Customizing Generative Image Models

Nupur Kumari

CMU-RI-TR-26-51

May 2026

School of Computer Science
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Jun-Yan Zhu, *chair*
Deva Ramanan
Shubham Tulsiani
Phillip Isola, *Massachusetts Institute of Technology*

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

Abstract

Recent advances in generative AI are reshaping how visual media is produced. These tools are beginning to make content creation accessible beyond expert practitioners and have the potential to support highly personalized use cases for everyday users. For example, they can enable educators to design custom illustrations for their curricula, small-scale retailers to visualize products in new settings without professional photoshoots, and independent creators to streamline artistic and design workflows. Realizing this potential, however, requires generative models that are not only controllable and customizable to user needs but also safe to deploy, such as by having support mechanisms for removing undesirable concepts like copyrighted styles, memorized identities, or unsafe content.

In this dissertation, we focus on **customizing** pretrained generative models, primarily text-to-image diffusion models, for such downstream tasks. A central challenge is that paired input-output supervision for these tasks is often scarce, expensive, or unavailable. To address this, we explore strategies for obtaining useful supervision under progressively weaker data assumptions: from few-shot optimization, to synthetic paired-data construction, to feedback from pretrained vision understanding models.

Part I studies optimization-based customization of text-to-image models for two complementary goals: acquiring new concepts from a few reference images and removing undesirable concepts, such as copyrighted styles or memorized identities. We first analyze task-specific fine-tuning and find that cross-attention layers undergo the most significant relative weight changes. Building on this observation, we develop *Custom Diffusion*, which learns new concepts by fine-tuning only this targeted subset of parameters. The same principle also enables concept removal, combined with the objective of steering generations associated with a target concept that we want to remove toward a broader anchor class.

Part II moves from per-instance optimization, which can be time consuming, to scalable dataset construction. We present a synthetic dataset construction pipeline for generating images of the same object across diverse contexts and viewpoints, enabling large-scale training for applications such as instant product recontextualization. The key insight is that, with appropriate constraints, generative models can synthesize identity-

consistent examples at a scale that would be difficult or expensive to achieve through real-world data collection.

While synthetic data enables scalable training, it also introduces new dependencies: the resulting model can inherit artifacts, biases, or limitations from the data-generation pipeline, and the dataset may become outdated as stronger generative models emerge.

Part III, therefore, investigates the use of pretrained vision encoders as a source of differentiable supervision for customizing generative image models. We demonstrate this paradigm across two settings. First, we show that ensembles of pretrained vision encoders, such as CLIP and DINO, can serve as effective discriminators for GAN training, enabling training with a substantially limited dataset. Second, we extend this principle to diffusion-based image editing, converting a text-to-image model into a reference-image-conditioned editing model without any paired before-and-after editing data. We use vision-language models' understanding capabilities to provide differentiable feedback for the editing task. In addition, to ensure generated images remain on the real-image manifold, we apply a distribution matching loss.

Collectively, these contributions demonstrate that generative image models can be effectively customized for diverse downstream applications through targeted supervision strategies, overcoming the scarcity of paired data while preserving the general capabilities of the pretrained model.

Acknowledgments

Reflecting on the last few years, I am reminded of how much this journey was shaped by the people around me. I wish I had expressed my gratitude more often, but I am grateful to have this opportunity to thank them.

First and foremost, I am extremely fortunate to have been advised by Jun-Yan. He supported me through the highs and lows of my PhD and always created an environment in which I could grow as a researcher. His creativity and attention to detail are qualities I will continue to strive toward. If I had to name one piece of advice from him that stayed with me, it would be his insistence on building things you would yourself want to use: if you would not find it useful, why would anyone else? That question guided many of my decisions about which problems and solutions to pursue. I will always be grateful for his patience, encouragement, and belief in me. I also feel lucky to count him as a friend, and I am continually amazed by his thoughtful perspective on life beyond research.

Alongside Jun-Yan, I was also privileged to have Richard Zhang and Eli Shechtman as constant sources of guidance. They were like a second set of advisors, and their mentorship shaped much of how I think about research. Beyond research itself, Richard taught me how to communicate ideas clearly and accessibly. Whenever I put together a presentation now, I find myself asking what Richard would think of it. Eli's breadth of knowledge and deep intuition for research problems are truly humbling, and I hope to develop that kind of acumen someday. Thank you, Richard and Eli, for your steady support, especially during the moments when projects felt uncertain or unlikely to work out.

I am also thankful to my committee members, Deva Ramanan, Shubham Tulsiani, and Phillip Isola, for their time, guidance, and advice, which made this dissertation stronger. Deva's enthusiasm whenever he talks about research is truly motivating. Through my interactions with Shubham, I learned the value of taking the time to understand a problem carefully. Thanks to Phillip for his thoughtful feedback and for the inspiring example set by his work.

I would also like to thank other research mentors and co-authors I have had throughout the years, without whom none of the projects in this

dissertation would have been possible. Thank you, Xun Huang, for your guidance, and Krishna Kumar Singh and Taesung Park for their ideas and generosity with time. Thanks to Samaneh Azadi, Xi Yin, and Ishan Misra for a wonderful summer and for the freedom to explore ideas I was not sure would work. I also want to thank Nanxuan Zhao and Yuheng Li for their collaboration. To Yotam Nitzan and Sheng-Yu Wang, I really enjoyed working together and look forward to more opportunities to do so. To Bingliang Zhang and Grace Su: it was a lot of fun to work with and mentor each of you, and I hope you learned as much as I did.

I was also lucky to find wonderful friends at CMU who made these years much more fun: Gaurav Parmar, Sheng-Yu Wang, Ruihan Gao, Kangle Deng, Muyang Li, George Cazenavette, Beijia Lu, Maxwell Jones, Ava Pun, Sean Liu, Mia Tang, Bingliang Zhang, Daohan Lu, Rohan Agarwal, Tarasha Khurana, Sally Chen, Neehar Peri, Mihir Prabhudesai, Shivam Duggal, Aayush Jain, Sriram Narayanan, Khiem Vuong, Anurag Ghosh, Zeqing Yuan, Homanga Bharadhwaj, Songwei Ge, Yufei Ye, and many others. Thank you, Gaurav and Sheng-Yu, for all the research and non-research discussions, which I will miss dearly. Thank you, Kangle, for hosting board-game nights; those evenings were a much-needed break. Tarasha, thank you for always pulling me along to dinner events. Ruihan, I hope we get to do more climbing and running together in the Bay Area. Thank you, Maxwell, for the meandering chats; your energy is always infectious. To all of you: thank you for the conversations, meals, and activities that made Pittsburgh and Smith Hall feel like home.

Looking back even further, my path into research began at Adobe India before coming to CMU, and I am thankful for the people there who helped me take my first steps. Special thanks to Balaji Krishnamurthy for giving me the freedom to pursue my ideas, and to Vineeth N. Balasubramanian for his mentorship. To Mayank Singh and Abhishek Sinha, thank you for being colleagues and friends who shared a passion for research and were never afraid to dream, try, and fail. To Mayank, thank you for continuing to share this journey with me, for becoming my steadfast companion, and for keeping me grounded and motivated through it all.

Finally, I want to thank my family for their constant love and support. They trusted me even when the path was uncertain, celebrated every small success, and gave me the strength to keep going through the difficult moments. Their quiet confidence in me, even when I did not have it in myself, made all the difference.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Consider the image in Figure 1.1 (a), showing a half-elephant, half-panda creature that has never existed. Remarkably, a generative model synthesizes it from the single text prompt *a mix of elephant and panda, eating bamboo snacks* in seconds. This capability reflects the exponential progress made in the last decade in learning algorithms [56, 85, 95, 150, 160, 243], model architectures [65, 105, 199, 211], and large-scale training techniques [184, 217] for generative models.

Among these advances, text-conditioned generation using diffusion models [57, 134, 223] has emerged as the dominant paradigm for image synthesis, driven by the availability of large-scale image-text datasets [232] and the training stability of diffusion objectives [95]. These models generalize far beyond what they have seen, composing styles, domains, and settings well beyond any single training image, and in doing so promise to accelerate content creation [5] and open it to users without professional expertise [50].

However, despite their promise, text alone is a coarse interface for many applications, and realizing the full potential requires going beyond text-only prompting. This gap motivates *model customization* – methods that adapt pretrained generative models to new concepts, visual controls, and user-specific goals that text alone cannot reliably support. Developing efficient customization methods that also preserve the model's broad generalization capabilities is the central goal of this dissertation.

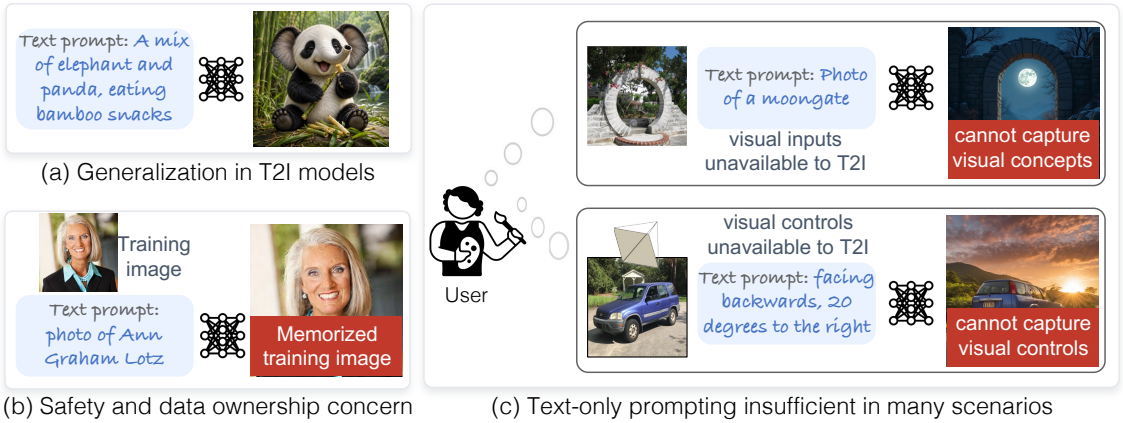


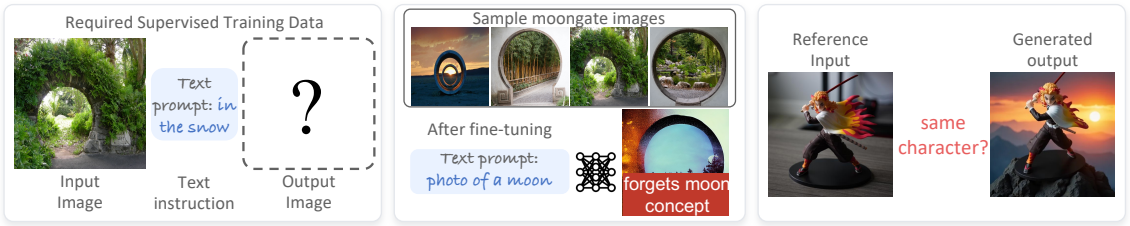
Figure 1.1: **Motivation for model customization.** (a) Text-to-Image (T2I) models synthesize photorealistic images from text prompts with impressive generalization. (b) However, they can memorize training samples verbatim, motivating methods for selective concept removal. (c) Many tasks also require capabilities that text cannot express: grounding generation in a visual reference (top) or spatial controls such as viewpoint (bottom). Addressing these needs is the goal of model customization.

1.1 Motivation for Model Customization

Evolving world. Generative models are trained once on a static snapshot of the world, but the world evolves continuously, with new concepts, visual styles, and cultural artifacts emerging all the time. For example, when prompting a text-to-image model for a *moongate*, it returns a generic gate with the moon in the sky (Figure 1.1 (c), top row), likely because this concept never appeared in its training data. This highlights the need to extend pretrained models with new knowledge on demand.

Language as a bottleneck. Beyond unknown concepts, users often want to generate images of concepts from their personal lives, not seen by the model during training. This can include pets, toys, a friend's face, a specific robot gripper, or a retailer's signature product. Describing any of these through language alone is too imprecise to guide the model toward the intended concept. These personalization tasks require grounding the model in a visual reference, not just a text prompt.

Language is also insufficient for controlling image composition or for editing an existing image given a user instruction. For example, the retailer may want to generate their product image in a specific pose, but conveying it via geometric or visual



(a) Data challenges (b) Overfitting challenges (c) Evaluation challenges

Figure 1.2: **Challenges to model customization**. **(a) Data:** paired supervised data is often scarce or unavailable for large-scale training. **(b) Overfitting:** naïvely fine-tuning on few examples can degrade broader generalization. **(c) Evaluation:** no ground truth metric for evaluating customization quality.

conditioning is far more natural than text (Figure 1.1 (c), bottom row).

Responsible Use of Generative AI. Finally, as these models become more widely deployed, they also raise important questions of safety, ownership, and consent. These models can reproduce copyrighted artistic styles or memorize training images, as Figure 1.1 (b) shows, without the knowledge or consent of the original creators [32, 244]. Responsible deployment thus requires mechanisms not only to acquire new concepts and capabilities, but also to restrict or remove undesirable ones, such as copyrighted or harmful content, efficiently and without retraining from scratch.

1.2 Challenges

Building customization methods that support the applications above poses several interconnected challenges, as we discuss below and as shown in Figure 1.2.

Data challenges. The key challenge is the lack of large-scale supervised training data. Collecting paired reference inputs, user instructions, and desired output images at scale is prohibitively expensive or often infeasible. For example, web images rarely carry object-identity annotations linking multiple views of the same object, making it difficult to train models for personalization tasks.

Overfitting and modeling challenges. Although we can assume access to a handful of user-provided reference examples, naively fine-tuning on these few examples can cause language drift, as shown in Figure 1.2(b). Effective customization must therefore be *surgical*, updating only the most task-relevant parameters while preserving the model's broader generalization. For concept removal, the challenge is

to prevent degradation of related concepts when removing a target concept.

Evaluation challenges. Lastly, customization, like any generative task, has no ground-truth, pixel-aligned output to measure performance against, since for a given reference and text instruction many outputs are equally valid. Evaluation therefore requires a benchmark spanning diverse reference concepts, so that strong performance reflects genuine grounding in the reference. Defining robust metrics is equally hard, since fidelity to the reference, text alignment, diversity, and perceptual realism are all desirable but often pull in different directions.

1.3 Dissertation Overview

This dissertation aims to tackle the above challenges, with the primary focus on supervised data scarcity. In essence, we ask: *when the supervised data we want does not exist, where does the training signal come from?*

We address this by studying three customization paradigms with progressively weaker data requirements: (1) parameter-efficient fine-tuning from a small number of examples; (2) scalable training using a synthetically generated dataset; and (3) learning via feedback from pretrained vision models, without using supervised data. The overfitting and evaluation challenges of Section 1.2 are common to all, and we address them within each paradigm as they arise. Chapter 2 provides the background on text-to-image models; the remainder of the dissertation is then organized into three parts corresponding to these three paradigms, as summarized in Figure 1.3.

Part I: Learning from Few Samples. Drawing on principles from transfer [180, 301] and few-shot learning [242], we first study how text-to-image diffusion models can be customized on new concepts using few reference images provided by the user.

In **Chapter 3**, we introduce Custom Diffusion [130], a method for adding personal concepts such as a pet dog or a rare object like a moongate to text-to-image diffusion models (Figure 1.3 (a)). The key finding is that cross-attention layers that bind words to image regions play the most pivotal role when adding a new concept. For a robust evaluation, we introduce CustomConcept101, a benchmark of 101 diverse concepts and their multi-concept compositions with associated evaluation prompts. In the follow-up work [131], we also add explicit control over the object viewpoint.

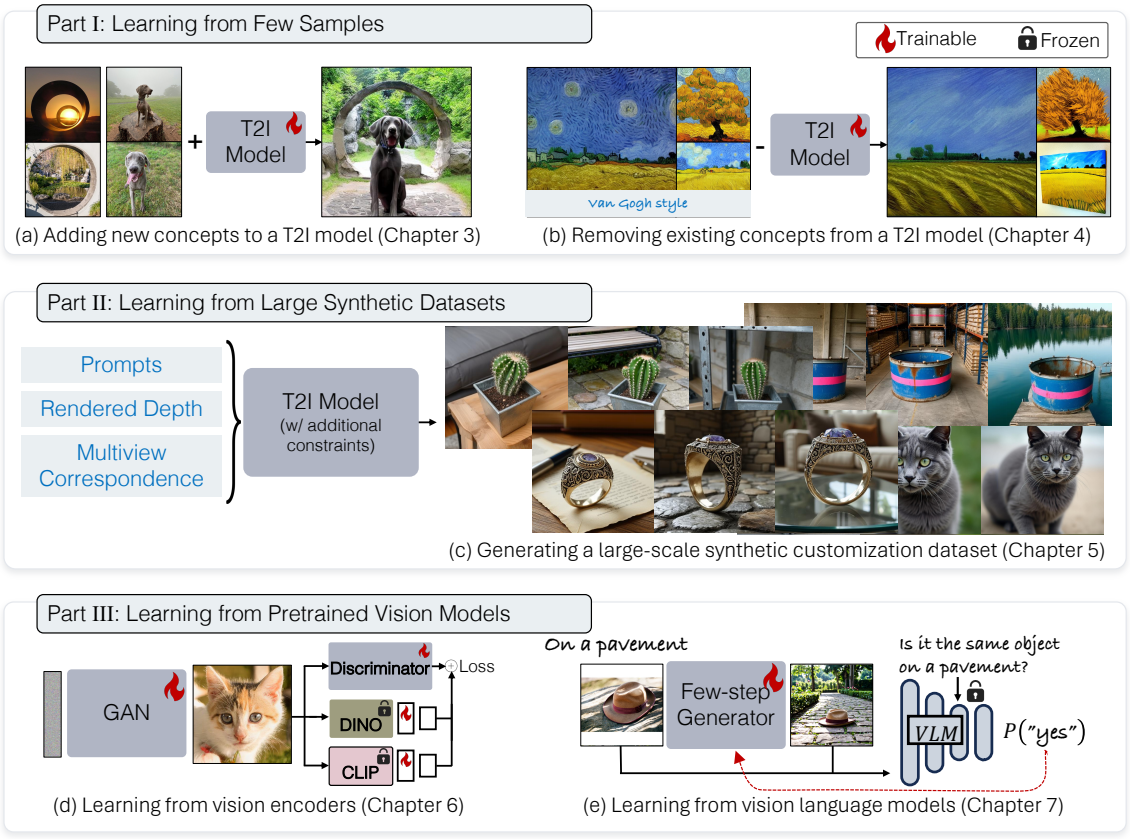


Figure 1.3: **Dissertation Overview.** This dissertation explores three paradigms for customizing pretrained generative image models. **Part I** uses few-sample fine-tuning to add or remove concepts from text-to-image models, but requires optimization for each new instance. **Part II** addresses this by curating a pipeline for large-scale synthetic dataset construction to enable end-to-end training. However, such datasets grow stale as generative models advance. **Part III** thus explores using feedback from pretrained vision models for generative model customization.

In **Chapter 4**, we study the complementary problem of *removing* unwanted concepts from these models [129]. Using the same principle of parameter-efficient fine-tuning, combined with an objective that steers the unwanted concept toward a broader anchor concept, we ablate copyrighted content, artistic styles (Figure 1.3 (b)), and memorized samples, giving creators and users a way to opt content out.

Part II: Learning from Large Synthetic Datasets. The *optimization-based* method of Part I, while effective, requires per-instance fine-tuning for each new concept and is slow to deploy at scale. The alternative is a feed-forward model that,

given a few reference images, directly synthesizes the object in a new context guided by a text prompt. The challenge, however, as mentioned before, is the absence of large-scale training data showing a consistent object across diverse contexts.

Chapter 5 addresses this with SynCD [132], where the key insight is that a generative model can itself synthesize such a dataset. By combining 3D-asset guidance with a shared attention technique, SynCD synthesizes a large dataset of multi-image examples, each showing the same object across a few varied contexts (Figure 1.3 (c)). This dataset is then used to train a feed-forward customization model.

Part III: Learning from Pretrained Vision Models. Synthetic data from Part II removes the scaling bottleneck but introduces dependency on the artifacts and biases of the pretrained model, and the dataset must be rebuilt as stronger models emerge. The final part therefore investigates an alternative source of supervision from pretrained vision models, removing the need for example outputs altogether. Inspired by *analysis by synthesis* [308], we observe that a vision model able to judge whether an image satisfies a condition can also provide a gradient for producing one.

Chapter 6 shows that pretrained vision encoders such as CLIP and DINO make effective discriminators for Generative Adversarial Network (GAN) training, providing a powerful and general training signal, especially in limited-data settings, where discriminator overfitting and training collapse are often a concern.

Chapter 7 extends this principle to diffusion-based image editing [133] by using differentiable feedback from vision-language models to evaluate edit success, together with a distribution matching loss to retain the model's generative prior over real images. This allows us to customize a text-to-image model into a few-step instruction-guided editing model without paired before-and-after image-editing data.

Finally, we conclude in **Chapter 8** with a discussion of possible future directions.

1.4 Other Related Research

In addition to the above works, I have also contributed to several related projects that extend and complement this dissertation's theme of making generative image models more controllable, customizable, and useful in practical creative workflows. These works share a common thread: each introduces new mechanisms for user control or model access that go beyond standard text prompting.

1. *Generative Photomontage* [157] introduces a framework for compositing multiple generated image candidates into a single coherent output. This provides users with fine-grained spatial control over the generation process through interactive selection and composition instead of simply relying on the random dice-roll nature of image generation.
   Sean Liu, **Nupur Kumari**, Ariel Shamir, Jun-Yan Zhu. CVPR 2025.
2. *Customizing Text-to-Image Models with a Single Image Pair* [104] presents a method for learning a visual style transformation from a single before-and-after image pair, enabling users to apply the depicted stylistic change to novel content. This work directly aligns with this dissertation's focus on adapting generative models from scarce supervision, extending customization from object identity to stylistic transformation.
   Maxwell Jones, Sheng-Yu Wang, **Nupur Kumari**, David Bau, Jun-Yan Zhu. SIGGRAPH Asia 2024.
3. *Content-Based Search for Deep Generative Models* [165] develops a model platform and corresponding search algorithm for retrieving and comparing outputs across diverse generative models, including both base and customized models. This addresses a practical challenge in model customization: helping users navigate the rapidly expanding ecosystem of pretrained and customized generative models to find the model best suited to their desired visual outcome.
   Daohan Lu*, Sheng-Yu Wang*, **Nupur Kumari***, Rohan Agarwal*, Mia Tang, David Bau, Jun-Yan Zhu. SIGGRAPH Asia 2023.

Chapter 2

Background

We introduce here the key background concepts shared across multiple chapters, covering text-to-image diffusion models, rectified flow models, and their sampling procedures and architectures.

2.1 Diffusion Models

Diffusion models [95, 243] are a class of generative models that aim to approximate the original data distribution $p(\mathbf{x})$ by introducing a series of latent variables $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, with $\mathbf{x}_0 = \mathbf{x}$. The data distribution can be written as follows:

$$p_\theta(\mathbf{x}_0) = \int \Big[p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \Big] d\mathbf{x}_{1:T}. \tag{2.1}$$

The latent variables $\mathbf{x}_1$ to $\mathbf{x}_T$ are obtained via a forward Markov chain diffusion process, where each step adds Gaussian noise according to a variance schedule $\{\beta_t\}_{t=1}^{T}$:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\Big(\mathbf{x}_t;\ \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\ \beta_t \boldsymbol{I}\Big)\,. \tag{2.2}$$

By the Markov structure, the noisy sample at any timestep t can be computed in closed form as $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$, where $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The model is trained to learn the reverse denoising process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, i.e., to denoise the noisy input image $\mathbf{x}_t$. The variational lower bound objective to maximize the log-

likelihood for diffusion models can be simplified to:

$$\mathbb{E}_{\epsilon,\mathbf{x},\mathbf{c},t}\left[w_t \left\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}, t)\right\|^2\right], \tag{2.3}$$

where ϵ_θ is the model prediction, w_t is a time-dependent weight on the loss, and $t \in (0, 1000]$. The model is conditioned on timestep t and can be further conditioned on any other modality $\mathbf{c}$, e.g., text. During inference, a random Gaussian image, $\mathbf{x}_T$, is denoised for fixed timesteps using the model as explained below.

Sampling. During inference, generating a sample requires iteratively denoising from pure Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ over some sampling steps. The commonly used **DDIM** [246] technique performs a deterministic update, where at each step, the model predicts the clean image $\mathbf{x}_0$ from the current noisy sample $\mathbf{x}_t$, then re-noises it to the target timestep:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}\right)}_{\text{predicted } \mathbf{x}_0} + \sqrt{1-\bar{\alpha}_{t-1}}\,\epsilon_\theta(\mathbf{x}_t, t). \tag{2.4}$$

The number of sampling steps is often much smaller than the 1000 used during training and can be as few as 20-50 steps. DDIM is equivalent to an Euler solver for the ODE defined by the diffusion process, and higher-order ODE solvers [164, 325] can be applied to reduce the required number of steps further.

Classifier-Free Guidance (CFG). CFG [94] is a widely used technique for higher fidelity sampling from the conditional distribution. During training, the conditioning $\mathbf{c}$ (e.g., a text embedding) is randomly dropped, so the model learns both conditional $\epsilon_\theta(\mathbf{x}_t, \mathbf{c}, t)$ and unconditional $\epsilon_\theta(\mathbf{x}_t, \varnothing, t)$ distributions. At inference, the two predictions are extrapolated:

$$\hat{\epsilon}(\mathbf{x}_t, \mathbf{c}, t) = \epsilon_\theta(\mathbf{x}_t, \varnothing, t) + s\big(\epsilon_\theta(\mathbf{x}_t, \mathbf{c}, t) - \epsilon_\theta(\mathbf{x}_t, \varnothing, t)\big), \tag{2.5}$$

where $s > 1$ is the guidance scale. Larger s increases alignment with $\mathbf{c}$ at the cost of sample diversity.

2.2 Rectified Flow Models

Flow matching [150] and rectified flow [160] models offer an alternative formulation to diffusion models by learning a continuous-time ordinary differential equation (ODE) that transports samples from a target distribution like Gaussian noise to the data distribution. The model learns a velocity field $\mathbf{v}_\theta$ such that integrating

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{v}_\theta(\mathbf{x}_t, t) \tag{2.6}$$

from $t = 1$ to $t = 0$ maps noise samples to data samples. Rectified flow models define a simple linear interpolation between a data sample $\mathbf{x}_0$ and a noise sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\epsilon$ for $t \in [0, 1]$. Along this path, the target velocity is constant and given by $\mathbf{v} = \frac{d\mathbf{x}_t}{dt} = \epsilon - \mathbf{x}_0$. The model is trained to predict this velocity from the intermediate sample $\mathbf{x}_t$, the time t, and any optional conditioning signal $\mathbf{c}$:

$$\mathbb{E}_{\epsilon,\mathbf{x},\mathbf{c},t}\left[w_t \left\|\mathbf{v} - \mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{c})\right\|^2\right], \tag{2.7}$$

where w_t is a time-dependent weighting term. Although the target velocity is defined using individual pairs $(\mathbf{x}_0, \epsilon)$, the optimal predictor corresponds to the conditional, data-averaged velocity field $\mathbb{E}[\mathbf{v} \mid \mathbf{x}_t, t, \mathbf{c}]$.

Diffusion and flow models can be viewed under a common continuous-time framework [112], with the forward process $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon$ that interpolates between data and noise. The main distinction lies in the choice of interpolation schedule (α_t, σ_t). Standard diffusion models correspond to $\alpha_t = \sqrt{\bar{\alpha}_t}$ and $\sigma_t = \sqrt{1-\bar{\alpha}_t}$, from Section 2.1, with $\bar{\alpha}_t$ following a nonlinear schedule such as the cosine schedule [185], whereas rectified flow uses the linear path with $\alpha_t = 1 - t$ and $\sigma_t = t$. The straight-line trajectory can make the learned transport path easier to integrate at inference time with fewer sampling steps and has been adopted in recent large-scale text-to-image models such as Stable Diffusion 3 [65] and FLUX [134].

2.3 Model Architecture

Diffusion model. Early text-to-image diffusion models used a U-Net-based architecture [217] that processes latent representations at multiple scales, with skip connections between encoding and decoding pathways. More recent approaches employ transformer-based architectures [199] that replace convolutional layers with a series of transformer layers, enabling a more scalable [134] and unified architecture towards multimodal generation [134].

Latent Diffusion Model. While diffusion models can be trained directly in pixel space, operating on high-resolution images is computationally expensive. Latent Diffusion Models (LDM) [217] are a commonly adopted technique that instead learns the diffusion model in a compressed latent space. The image $\mathbf{x}$ is first encoded into a low-dimensional latent representation, $\mathbf{z} = E(\mathbf{x})$, via an encoder E and using hybrid training objectives of VAE [119], PatchGAN [101], and LPIPS [320], such that running the decoder D reconstructs the image via $\hat{\mathbf{x}} = D(\mathbf{z})$. The diffusion process is now defined over the latent $\mathbf{z}$ (which we continue to denote as $\mathbf{x}$ in later chapters for brevity). This significantly reduces computational cost while maintaining generation quality, and has become the dominant approach for large-scale text-to-image generation, e.g., Stable Diffusion [217], SDXL [203], and more recently FLUX [134] for rectified-flow models.

Part I

Learning from Few Samples

Chapter 3

Optimization-Based Customization for Concept Addition

In this chapter, we begin our exploration of parameter-efficient optimization-based methods to customize text-to-image models with new personalized concepts. Though pretrained text-to-image diffusion models possess diverse, *general* capabilities, users often wish to synthesize *specific* concepts from their own personal lives, e.g., their friends, pets, and other personal items, that are inherently absent from the training data and difficult to describe through text alone. Can we teach a model to quickly acquire such new concepts, given only a few examples? Furthermore, can we compose multiple new concepts together in a single scene?

We address these questions through a systematic analysis of which model parameters are most critical for concept acquisition. We find that cross-attention layers that mediate the text-to-image conditioning undergo the largest relative change during fine-tuning. This finding motivates our method, *Custom Diffusion*. We further introduce a closed-form formulation to merge individually fine-tuned models, enabling multi-concept customization on the fly without re-training. Lastly, we present *Custom Diffusion-360*, a follow-up extension that augments the framework with explicit viewpoint control by integrating 3D neural radiance field features into the diffusion process, enabling precise object-centric viewpoint conditioning.

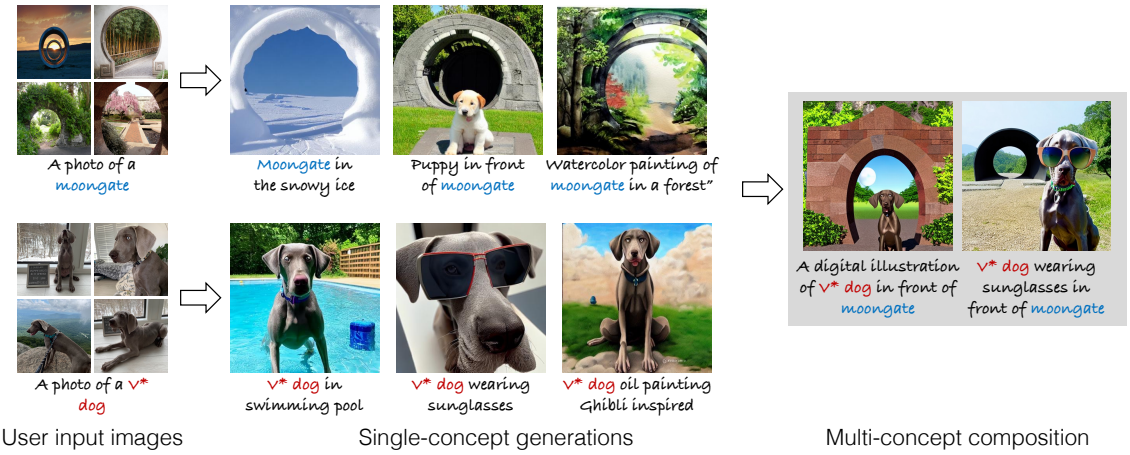


Figure 3.1: **Custom Diffusion** augments a pretrained text-to-image diffusion model with a new concept, given its few reference images, enabling generations of the concept in unseen contexts. Example concepts include personal objects, animals, e.g., *a pet dog*, and classes not well generated by the model, e.g., *moongate*. Furthermore, we can compose the new concepts together in a scene, such as V* dog wearing sunglasses in front of a moongate. We denote personal concepts with a new modifier token V*.

3.1 Introduction

Large-scale text-to-image diffusion models [211, 217, 223, 304] represent a watershed moment in image generation. By simply querying a text prompt, users are able to generate images of unprecedented quality. Such systems can generate a wide variety of objects, styles, and scenes – seemingly "anything and everything".

However, despite the diverse, *general* capability of such models, users often wish to synthesize *specific* concepts from their own personal lives. For example, loved ones such as family, friends, pets, or personal objects and places, such as a new toy or a recently visited garden, make for intriguing concepts. As these concepts are by nature personal, they are unseen during large-scale model training. Describing these concepts after the fact, through text, is unwieldy and unable to produce the personal concept with sufficient fidelity.

This motivates a need for model *customization*. Given a few user-provided images, can we augment existing text-to-image diffusion models with the new concept (for example, their pet dog as shown in Figure 3.1)? The fine-tuned model should be able to generalize and compose them with existing concepts to generate new vari-

ations. This poses a few challenges – first, the model tends to forget [58, 142, 209] or change [140, 166] the meanings of existing concepts: e.g., the meaning of "moon" being lost when adding the "moongate" concept. Secondly, the model is prone to overfitting on the few training samples and reduced sampling variations.

Moreover, we study a more challenging problem, *compositional fine-tuning* – the ability to extend beyond tuning for a single concept and compose multiple concepts together, e.g., pet dog in front of moongate (Figure 3.1). Improving compositional generation has been studied in previous works [155]. But composing multiple new concepts poses additional challenges, requiring both faithful learning of each new concept without interference and the ability to generate them in unseen combinations.

To address these challenges, we propose *Custom Diffusion*, a computationally and memory efficient fine-tuning technique for text-to-image diffusion models. We show that fine-tuning only a small subset of model weights, namely the key and value mapping from text to latent features in the cross-attention layers [13, 267] is sufficient to update the model with the new concept. To prevent model forgetting, we include a small set of real images with similar captions as the target images to the fine-tuning data and also introduce data augmentation strategies that improve convergence speed and quality. For composing multiple concepts, our method supports either joint training or a more efficient merging of individually fine-tuned models.

We build our method on Stable Diffusion [57] and experiment on various concepts with as few as four reference images. For adding single concepts, our method shows better text alignment and on par visual similarity to the target images than concurrent works. More importantly, our method can compose multiple new concepts efficiently, whereas concurrent methods struggle and often omit one. Finally, our method only requires storing a small subset of parameters (3% of the model weights) and reduces the fine-tuning time ($\sim$6 minutes on 2 A100 GPUs, $2-4\times$ faster compared to concurrent works). Our code and data are available on our project website.

3.2 Related Work

Deep generative models and text-to-image synthesis As described in our background, Chapter 2, diffusion models have emerged as a powerful generative framework, and their integration with large-scale text-conditioning has led to re-

markable advances in text-to-image synthesis. Recent models [56, 184, 210, 211, 217, 223, 304] trained on web-scale image-text datasets demonstrate impressive compositional generalization and photorealism. However, such models are generalists trained for text-only prompting and struggle to generate specific instances like personal toys or rare categories (e.g., "moongate"). In contrast, our method aims to adapt these models to learn new visual concepts from as few as four reference images.

Image and model editing. While generative models can sample random images, a user often wishes to edit a single, specific image. Several works aim at leveraging the capabilities of generative models, such as GANs [1, 2, 3, 198, 333] or diffusion models [48, 116, 184] towards editing. A challenge is representing the specific image in the pretrained model, and such methods employ per-image or per-edit optimization. A closely related line of work edits a generative model directly [15, 70, 272]. Whereas these methods aim to customize GANs, our focus here is on text-to-image models.

Transfer learning. A method of efficiently modeling a new distribution of images is leveraging a pretrained model and then using transfer learning [70, 86, 144, 151, 180, 186, 188, 189, 275, 276, 323]. For example, one can adapt photorealistic faces into cartoons [70, 144, 180, 188, 189]. To adapt with just a few training images, efficient training techniques [109, 128, 227, 264, 324, 326] are often useful. Different from these works, which focus on tuning whole models to single domains, we wish to acquire multiple new concepts without catastrophic forgetting [67, 121, 142, 146, 209]. By preserving the millions of concepts captured in large-scale pretraining, we can synthesize the new concepts in composition with these existing concepts. Relatedly, several methods [74, 96, 200, 241] propose to train adapter modules or low rank updates for large-scale models in the discriminative setting. In contrast, we adapt a small number of existing parameters and introduce no additional ones.

Adapting text-to-image models. Similar to our goals, two concurrent works, DreamBooth [219] and Textual Inversion [71], adapt transfer learning to text-to-image diffusion models [217, 223] via either fine-tuning all the parameters [219] or introducing and optimizing a word vector [71] for the new concept. Our work differs in several aspects. First, our work tackles a challenging setting: compositional fine-tuning of *multiple* concepts, where concurrent works struggle. Second, our method occupies a middle ground in the parameter efficiency spectrum with the updates restricted to the cross-attention key and value projection matrices. We find that

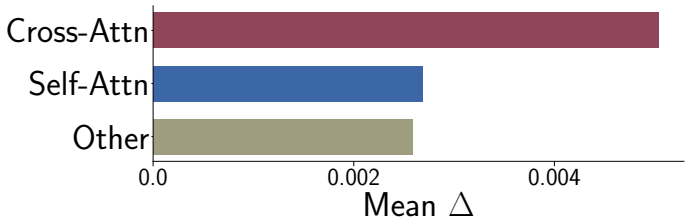


Figure 3.2: **Analysis of change in weights** for a model fine-tuned on the *moongate* concept. Cross-attention layers—comprising just 5% of parameters—have significantly higher relative change compared to self-attention and remaining layers.

this targeted update enables faster training and better compositional generation, as validated by automatic metrics and human preference studies.

3.3 Method

3.3.1 Single-Concept Fine-tuning

Our goal is to adapt the pretrained text-to-image diffusion model on a new concept, given N reference images of the concept $\{\mathbf{x}_i\}_{i=1}^N$, i.e., to model $p(\mathbf{x}|\mathbf{c}, \{\mathbf{x}_i\}_{i=1}^N)$ with text prompt $\mathbf{c}$. The model should retain its prior knowledge, allowing for novel generations with the new concept based on the text prompt $\mathbf{c}$.

A naïve baseline for this is to update the model using the original training objective, as explained in Eqn. 2.3, for the given text-image pairs. However, this can be computationally inefficient for large-scale models and can easily lead to overfitting when only a few training images of the target concept are available. Therefore, we aim to identify a minimal set of weights that is sufficient for the task of fine-tuning.

Rate of change of weights. Following Li *et al.* [144], we measure the relative change in parameters for each layer between the pretrained and fine-tuned models, $\Delta_l = \left\|\theta^l - \theta^l_{\text{init}}\right\| / \left\|\theta^l\right\|$, where θ^l and θ^l_{init} denote the updated and pretrained parameters of layer l, respectively. We group layers into three categories: (1) cross-attention (text-to-image), (2) self-attention (within the image), and (3) the remaining parameters, including convolutional blocks and normalization layers. As shown in Figure 3.2, when fine-tuning for the "moongate" concept, cross-attention layer parameters exhibit a significantly higher mean Δ than other layers, despite constituting only 5% of the total parameters. The trend remains similar across other concepts as well. This

Figure 3.3: **Role of regularization data in mitigating language drift.** 1st row: pretrained model samples. 2nd row: without regularization, fine-tuning on *moongate* causes the prompt *photo of a moon* to produce moongate-like images. 3rd row: incorporating our regularization dataset largely mitigates this issue.

suggests it plays a significant role when adapting the model on new concepts, and motivates our approach of restricting fine-tuning to these layers.

Model fine-tuning. The cross-attention block modifies the latent features of the network according to the condition features, i.e., text features in the case of text-to-image diffusion models. Given text features $\mathbf{c} \in \mathbb{R}^{s\times d}$ and latent image features $\mathbf{f} \in \mathbb{R}^{(h\times w)\times l}$, a single-head cross-attention [267] operation consists of $Q = W^q\mathbf{f}$, $K = W^k\mathbf{c}$, $V = W^v\mathbf{c}$, and a weighted sum over value features as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\Big(\frac{QK^T}{\sqrt{d'}}\Big)V, \tag{3.1}$$

where W^q, W^k, and W^v map the inputs to a query, key, and value feature, respectively, and d' is the output dimension of key and query features. The latent feature is then updated with the attention block output. The task of fine-tuning aims at updating the mapping from given text to image distribution, and the text features are only input to the W^k and W^v projection matrices in the cross-attention block. Therefore, we propose to only update W^k and W^v parameters of the diffusion model during the fine-tuning process. As shown in our experiments, this is sufficient to update the model with a new text-image paired concept.

Text encoding. Given target concept images, we require a text caption as well. If there exists a text description, e.g., *moongate*, we use that as a text caption. For

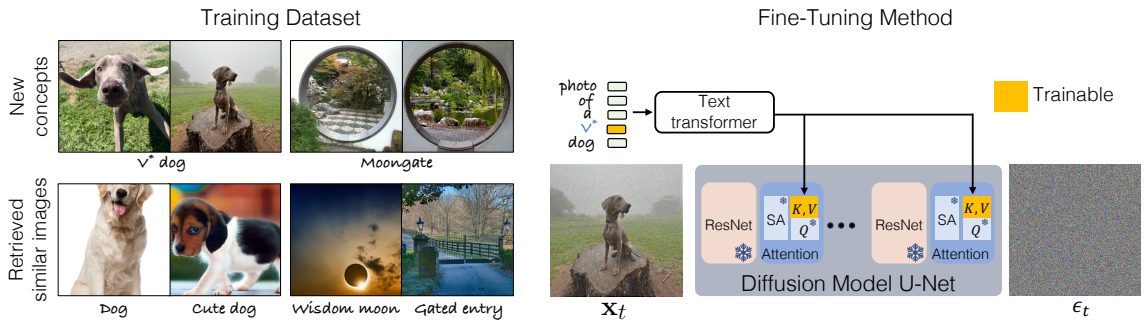


Figure 3.4: **Custom Diffusion.** *(Left)* Training set combining few images of the new concept with real LAION-400M images retrieved by caption similarity, used jointly to prevent forgetting. *(Right)* To represent a new concept within a general category, we introduce a modifier token V^* prepended to the category name (e.g., V^* dog); only the W^k/W^v matrices in each cross-attention block and V^* are updated during fine-tuning.

personalization-related use cases where the target concept is a unique instance of a general category, e.g., pet dog, we introduce a new modifier token embedding, i.e., V^* dog. During training, V^* is initialized with a rare occurring token embedding and optimized along with cross-attention parameters. An example text caption used during training is photo of a V^* dog.

Regularization dataset. Fine-tuning on target concept image-caption pairs can lead to language drift [140, 166], where the model forgets existing associations. For instance, training on moongate causes the model to lose the visual meaning of moon and gate individually (Figure 3.3), while training on V^* tortoise plushy can cause all prompts containing plushy to produce the specific target images. To mitigate this, we retrieve a regularization set of approximately 200 images from LAION-400M [231] whose captions have a CLIP [208] text-feature similarity above 0.85 with the target prompt, and include them as part of the training dataset. An overall method diagram is shown in Figure 3.4.

3.3.2 Multiple-Concept Compositional Fine-tuning

Joint training on multiple concepts. As a straightforward extension of our single-concept method to multiple concepts, we combine the training datasets of individual concepts and train jointly. Each concept is assigned a distinct modifier

token $\vee_i^*$, initialized from different rarely-occurring tokens and optimized alongside the cross-attention key and value matrices. As shown in Figure 3.7, our method which restricts the updates to cross-attention layer parameters yields significantly better compositional results than methods like DreamBooth that fine-tune all weights.

Closed-form weight merging of multiple concepts While joint training is effective, it requires retraining whenever the user wants a new composition. As our method only modifies the key and value projection matrices, individually fine-tuned models can instead be merged in closed form, enabling zero-shot composition without additional training. Let $\{W_{0,l}^k, W_{0,l}^v\}_{l=1}^L$ denote the key and value matrices across all L cross-attention layers in the pretrained model, and $\{W_{m,l}^k, W_{m,l}^v\}_{l=1}^L$ the corresponding updated matrices for concept $m \in \{1 \cdots M\}$. For notational clarity, we omit superscripts $\{k, v\}$ and layer index l, as the following optimization applies identically to all. We formulate the merging objective as a constrained least squares problem:

$$\begin{aligned} \hat{W} &= \underset{W}{\operatorname{argmin}} \left\| W C_{\text{reg}}^\top - W_0 C_{\text{reg}}^\top \right\|_F^2 \\ \text{s.t. } & WC^\top = V, \text{ where } C = [\mathbf{c}_1 \cdots \mathbf{c}_M]^\top \text{ and } V = [W_1\mathbf{c}_1^\top \cdots W_M \mathbf{c}_M^\top]^\top. \end{aligned} \tag{3.2}$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm, $W_0 \in \mathbb{R}^{o \times d}$ is the pretrained projection matrix, $C \in \mathbb{R}^{s \times d}$ contains the concatenated text features of s target words across all M concepts, and $C_{\text{reg}} \in \mathbb{R}^{s_{\text{reg}} \times d}$ contains text features from $\sim$1000 randomly sampled captions for regularization. The constraint ensures that $\hat{W}$ maps each concept's tokens to the values from its fine-tuned model, while the objective preserves pretrained behavior on all other inputs. Assuming C_{reg} is non-degenerate, this objective admits a closed-form solution via Lagrange multipliers [23]. We minimize the Lagrangian:

$$L = \frac{1}{2} \left\| W C_{\text{reg}}^\top - W_0 C_{\text{reg}}^\top \right\|_F^2 - \text{trace}(\mathbf{v}(WC^\top - V)), \tag{3.3}$$

where $\mathbf{v} \in \mathbb{R}^{s \times o}$ is the multiplier for the equality constraint. Setting the derivative with respect to W to zero yields:

$$\begin{aligned} & W C_{\text{reg}}^\top C_{\text{reg}} - W_0 C_{\text{reg}}^\top C_{\text{reg}} - \mathbf{v}^\top C = 0 \\ \implies & W = W_0 + \mathbf{v}^\top C (C_{\text{reg}}^\top C_{\text{reg}})^{-1}. \end{aligned} \tag{3.4}$$

Substituting into the constraint $WC^\top = V$ from Eqn. 3.2, we obtain:

$$\begin{aligned} &(W_0 + \mathbf{v}^\top C(C_{\text{reg}}^\top C_{\text{reg}})^{-1})C^\top = V \\ &\text{Let } \mathbf{d} = C(C_{\text{reg}}^\top C_{\text{reg}})^{-1} \\ &\mathbf{v}^\top = (V - W_0 C^\top)(\mathbf{d}C^\top)^{-1}. \end{aligned} \tag{3.5}$$

When individual fine-tuned models already exist, this merging takes only ~2 seconds, making it significantly faster than joint training. Both approaches enable coherent generation of multiple new concepts in a single scene, as we show in Section 3.4.2.

Training details. We train for 250 steps for single-concept and 500 steps for two-concept joint fine-tuning, with batch size 8 and learning rate 8×10^{-5}. We randomly resize the target images by a factor of 0.4–1.4× and append scale-descriptive text, e.g., "very small", "far away", "zoomed in", "close up", to the prompt accordingly. Additional training and implementation details are provided in Appendix A.1.

3.4 Experiments

Here, we show the results of our method on multiple datasets in both single concept fine-tuning and composition of two concepts on the Stable Diffusion model [57].

Datasets. We conduct our main experiments and ablations on a curated dataset of ten target concepts, comprising two scenes, two pets, and six objects with 3-10 images per concept. To evaluate generalization more broadly, we additionally create a benchmark CustomConcept101, a dataset that contains 101 concepts spanning a wider category of personal concepts, each paired with relevant text prompts.

Evaluation metrics. We use the following four criteria: (1) *Image-alignment* using CLIP image-space similarity [71] between generated and target concept images, (2) *Text-alignment* that measures CLIP text-image similarity [92] between generated images and the given prompt, (3) *KID* [18] computed against 500 real images of a similar concept from LAION-400M, measuring overfitting behavior and forgetting of related concepts, and (4) a *human preference* study against baselines. For inference we use DDPM sampling with 200 steps and guidance scale 6.

Baselines. We compare against two concurrent methods: *DreamBooth* [219] (third-party implementation [282]), which fine-tunes all diffusion model parameters with

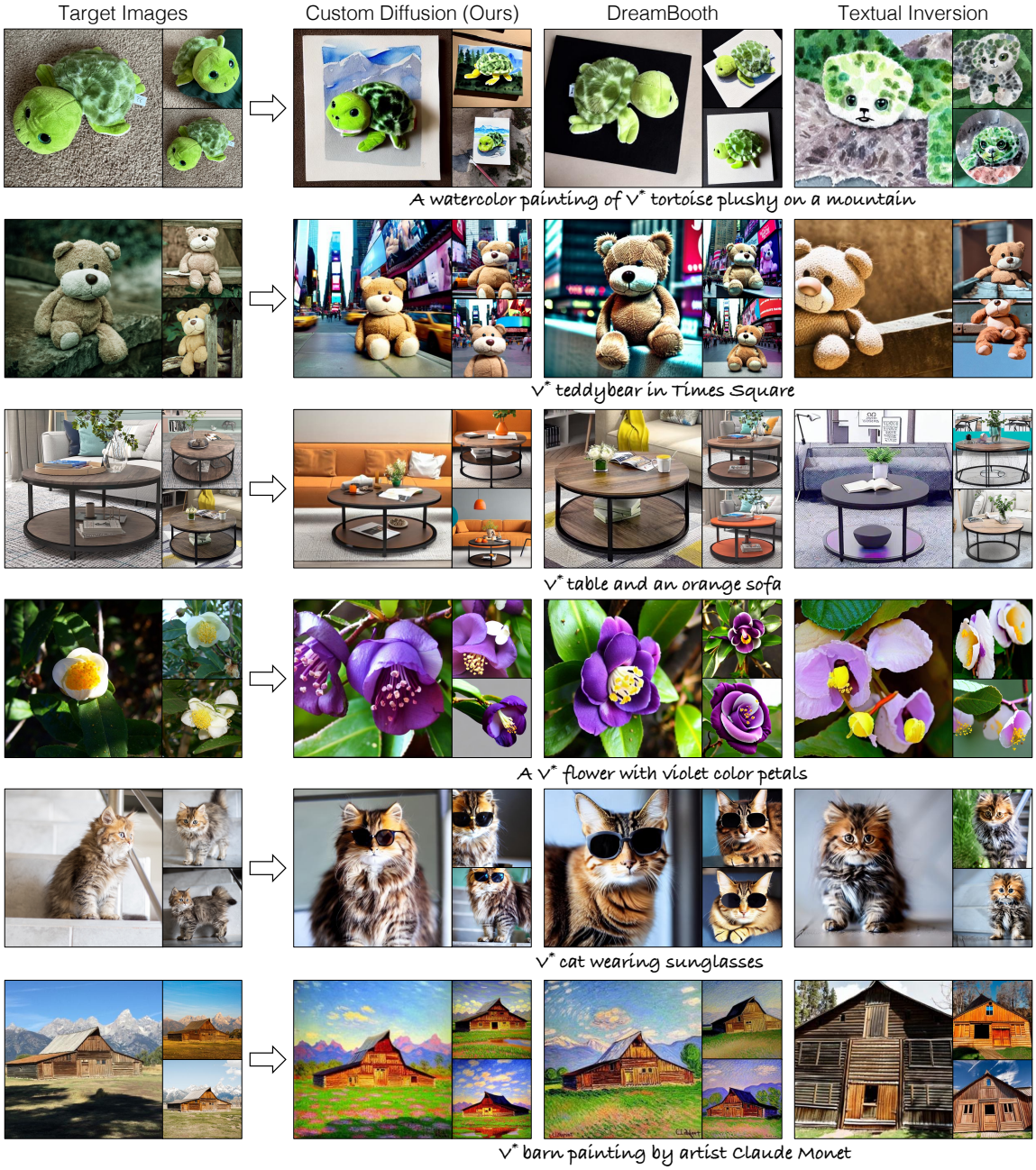


Figure 3.5: **Single-concept fine-tuning results.** Our method generates the target concept in unseen contexts and art styles while preserving visual similarity better or par than baselines. For example, it correctly adds background mountains omitted by others (1st row), changes petal colors (4th row), and maintains identity while adding accessories (4th row). Full results are on our website.

generated images as regularization and represents personal concept via a fixed rare token "[V] *category*"; and *Textual Inversion* [71], which optimizes only a new v^* token embedding per concept. We additionally include *Custom Diffusion (w/ fine-tune all)*, a variant of our method that updates all U-Net [218] parameters along with v^*, to isolate the benefit of restricting updates to cross-attention layers.

3.4.1 Single-Concept Customization Results

Computational requirements. With a fixed batch size of 8, our method trains in ~ 6 minutes on 2 A100 GPUs, compared to 20 minutes for Textual Inversion on 2 A100s, and 20 minutes and ~ 1 hour for Ours (w/ fine-tune all) and DreamBooth respectively, on 4 A100s. Additionally, since only ~ 3% of model parameters are optimized during our method, per-concept storage is minimal (75MB).

Qualitative evaluation. We evaluate each fine-tuned model on a set of challenging prompts that require generating the target concept in novel contexts, known art styles, compositions with other objects, and with modified properties such as color, shape, or expression. As shown in Figure 3.5, Custom Diffusion achieves higher text-image alignment while capturing the visual details of the target concept.

Quantitative evaluation. For quantitative evaluation, we generate 50 samples per prompt across 20 text prompts, yielding 1000 images per method. As shown in Figure 3.6, Custom Diffusion significantly outperforms Textual Inversion and marginally outperforms DreamBooth, while being ~ 5× faster to train and requiring far less storage (75MB vs. 3GB). It performs comparably to our full fine-tuning baseline at a fraction of the computational cost. Table 3.1 further reports KID against a reference set of 500 real images with similar captions; our method achieves lower KID than most baselines, indicating reduced overfitting to the target concept.

3.4.2 Multiple-Concept Customization Results

We evaluate our method and baselines on five composition pairs: (1) Moongate + Dog, (2) Cat + Chair, (3) Wooden Pot + Cat, (4) Wooden Pot + Flower, and (5) Table + Chair, generating 400 images with 8 prompts per pair. For DreamBooth, we train jointly on the pair of target concepts using two distinct tokens $[V_1]$ and $[V_2]$. For Textual Inversion, we use the individually fine-tuned tokens together at inference.

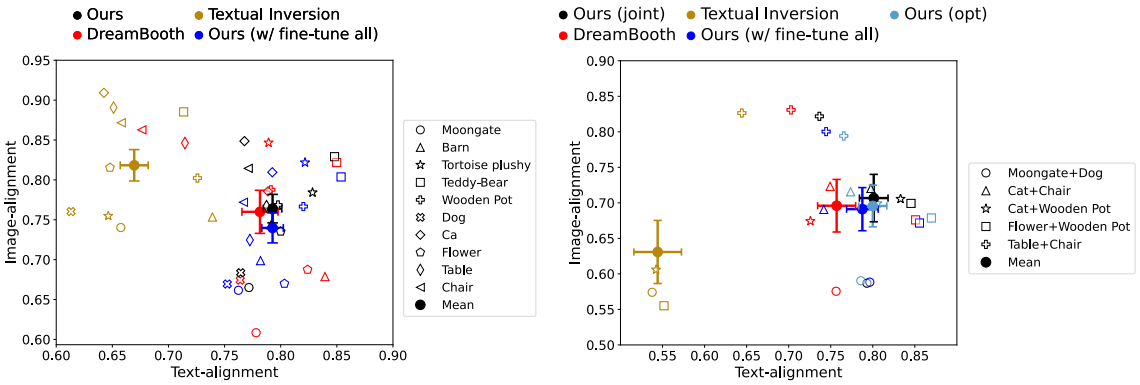


Figure 3.6: **Text- and image-alignment for single-concept (left) and multi-concept (right) customization.** Despite the inherent trade-off between image similarity and text-alignment, our method consistently occupies the top-right region (higher on both axes) with lower variance than baselines. For multi-concept, both joint training and optimization-based merging outperform other methods.

	Method	Text-alignment	Image-alignment	KID (validation)
Single-Concept	Textual Inversion	0.670	**0.827**	22.27
	DreamBooth	0.781	0.776	32.53
	Ours (w/ fine-tune all)	**0.795**	0.748	**19.27**
	Ours	**0.795**	0.775	20.96
Multi-Concept	Textual Inversion	0.544	0.630	
	DreamBooth	0.783	0.695	
	Ours (w/ fine-tune all)	0.787	0.691	—
	Ours (sequential)	0.797	0.700	
	Ours (closed-form)	0.800	0.695	
	Ours (joint-training)	**0.801**	**0.706**	

Table 3.1: **Quantitative comparison.** *Top*: Single-concept *top* and multi-concept evaluation *bottom*, averaged across datasets. KID ($\times 10^3$) measures the distributional similarity against a real validation set. Textual Inversion matches the pretrained KID as it leaves model weights unchanged. Between our method and DreamBooth, our method achieves lower KID showing less overfitting to the target concept.

We additionally compare against sequential training on the two concepts using our Custom Diffusion method and full fine-tuning baseline.

As shown in Figure 3.6, our method outperforms all baselines. Between joint-training and closed-form weight merging, joint training performs marginally better. Qualitative results in Figure 3.7 confirm that our method coherently generates both concepts in a single scene with high text-prompt alignment.

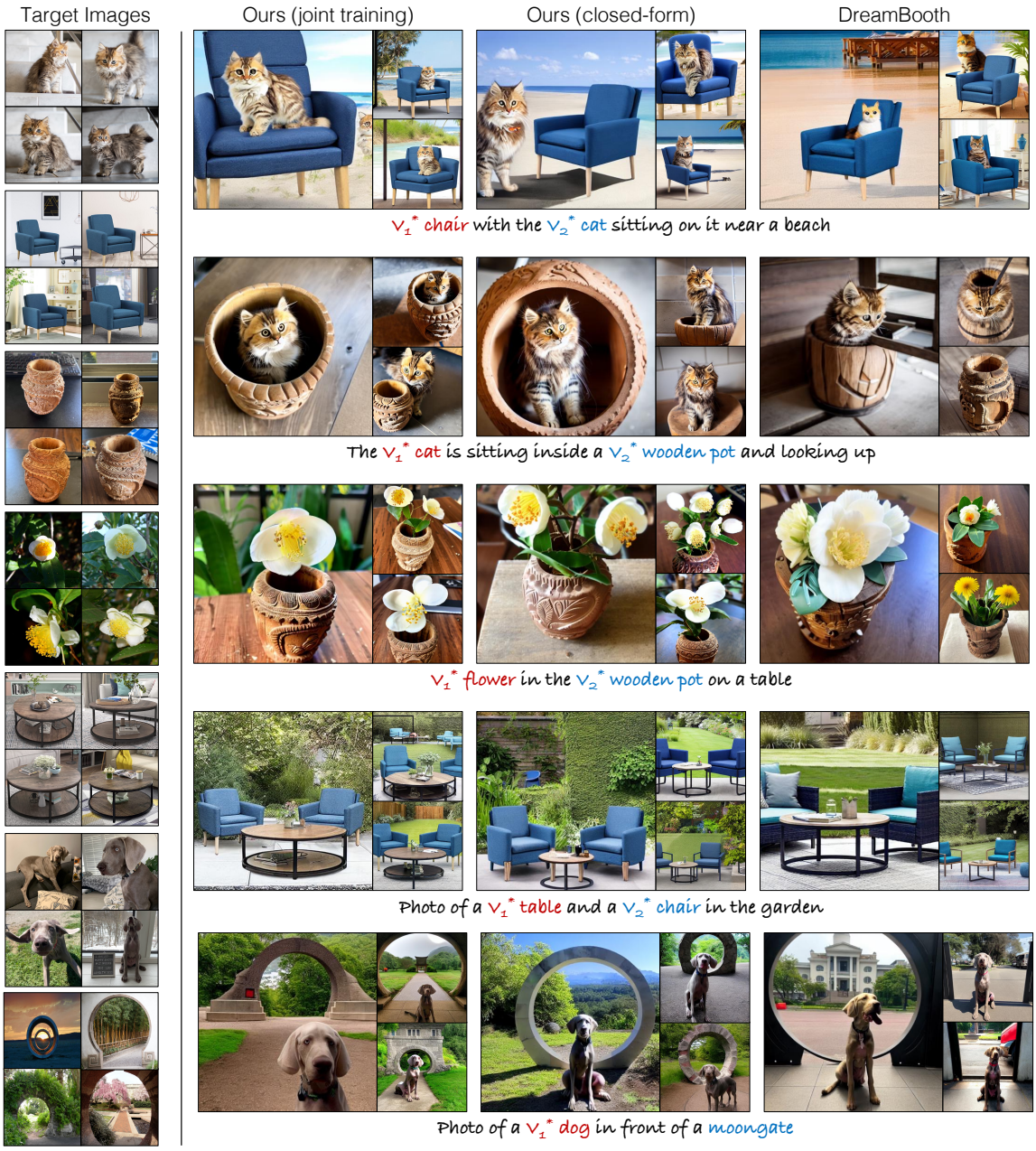


Figure 3.7: **Multi-concept customization results.** Our method better preserves the visual identity of both concepts while following the text condition, whereas DreamBooth often loses fidelity to one of them (e.g., omitting the cat in the second row or failing to preserve the wooden pot in the third row). Both our joint training and optimization-based merging outperform DreamBooth, with joint training performing slightly better.

Ours	Textual Inversion		DreamBooth		Ours (w/ fine-tune all)	
	Text Alignment	Image Alignment	Text Alignment	Image Alignment	Text Alignment	Image Alignment
Single-concept	**72.62**	**51.62**	**53.50**	**56.62**	**55.17**	**53.99**
Multi-concept (Joint)	**86.65**	**81.89**	**56.39**	**61.80**	**59.00**	**59.12**
Multi-concept (Optimization)	**81.22**	**83.11**	**57.00**	**61.75**	**57.60**	**53.49**

Table 3.2: **Human preference study.** Our method is preferred ($\geq 50\%$) over all baselines on both image- and text-alignment metrics across single- and multi-concept settings. All standard errors are within $\pm 2.72\%$.

3.4.3 Human Preference Study

We conduct a human preference study using Amazon Mechanical Turk with paired comparisons against DreamBooth, Textual Inversion, and Ours (w/ fine-tune all). Annotators evaluate text-alignment ("Which image is more consistent with the text?") and image-alignment ("Which image better represents the objects as shown in target images?"), with 800 responses collected per questionnaire. In image-alignment, we show 2-4 training samples of the target concept at the top. As shown in Table 3.2, our method is consistently preferred across both single- and multi-concept settings, underscoring the effectiveness of updating only the cross-attention parameters.

3.4.4 Ablation

In this section, we ablate various components of our method to show their contribution. We evaluate each experiment on the same setup as in Section 3.4.1. We show sample generations for ablation experiments on our website.

Role of regularization dataset. Training without the regularization dataset (with half the iterations to match the number of target images seen) leads to slightly lower image-alignment and significantly higher KID on the validation set, indicating forgetting of existing concepts, as shown in Table 3.3.

Role of data augmentation. As mentioned in Section 3.3.2 (training details), we augment training by randomly resizing target images and appending corresponding size-related prompts (e.g., "very small", "zoomed in"). As shown in Table 3.3, removing this augmentation leads to lower image-alignment with the target concept.

Method	Text-alignment	Image-alignment	KID (validation)
Ours	0.795	**0.775**	**20.96**
Ours w/o Aug	**0.800**	0.736	20.67
Ours w/o Reg	0.799	0.756	32.64

Table 3.3: **Ablation Study.** No augmentation during training leads to lower image-alignment. No regularization dataset leads to a much worse KID.

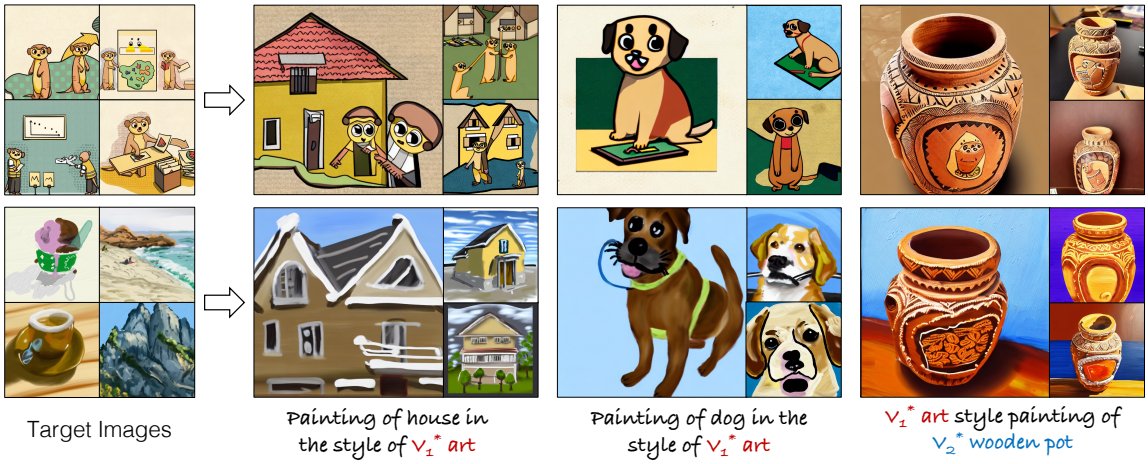


Figure 3.8: **Custom Diffusion with artistic styles.** The last column shows a learned style composed with the "wooden pot" concept via our closed-form merging. Style credits: Mia Tang (top), Aaron Hertzmann (bottom).

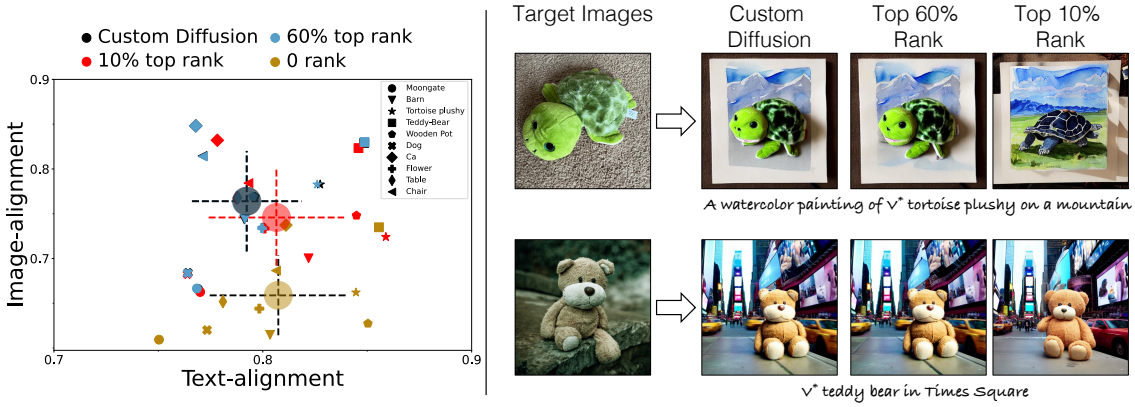


Figure 3.9: **Results with compressed models.** *Left*: retaining the top 60% singular values (5× compression) preserves performance (overlapping blue and black points), while further compression progressively reduces image-alignment. *Right*: qualitative samples also confirm this trend.

	Method	Text-alignment	Image-alignment
Single-Concept	Textual Inversion	0.612	**0.752**
	DreamBooth	0.752	**0.752**
	Ours	**0.760**	0.744
Multi-Concept	DreamBooth	0.738	0.662
	Ours (Optimization)	0.763	0.658
	Ours (Joint)	**0.757**	**0.668**

Table 3.4: **Quantitative comparison on CustomConcept101.** Our method achieves higher text-alignment but slightly lower image-alignment than DreamBooth for single concepts, and outperforms it on average for multi-concept. Metrics are averaged across all concepts over 100 generated images (20 prompts) for single-concept and 120 images (12 prompts) for multi-concept, using 200-step DDPM sampling.

Fine-tuning on a style. Beyond objects, our method can also learn artistic styles (Figure 3.8). We fine-tune with the prompt A painting in the style of V^* art [71] and use regularization images whose captions are similar to "art".

Analyzing custom diffusion update. To better understand our method, we analyze the singular values of $\Delta W = W' - W_0$ between the updated and pretrained key/value projection matrices across all cross-attention layers. The singular values decay steeply, suggesting that ΔW is approximately low-rank and can be compressed via SVD. We evaluate performance when storing only the top-k% singular components (rank at which the cumulative sum reaches k% of the total). Figure 3.9 shows that retaining just the top 60% (5× compression) yields similar performance.

3.5 CustomConcept101 Benchmark

To evaluate our method on a broader set of concepts, we create a benchmark called CustomConcept101, consisting of 101 concepts across diverse categories such as toys, plushies, wearables, pets, scenes, and human faces (Figure 3.10). Images are sourced from redistribution-permitted websites (e.g., Unsplash) or captured by the authors. The benchmark also includes 101 unique multi-concept compositions (e.g., V_1^* dog with V_2^* sunglasses), along with 20 evaluation prompts per single concept and 12 per composition. Prompts are generated using ChatGPT [38] with instructions to vary the background, style, and composition, then manually filtered and refined.

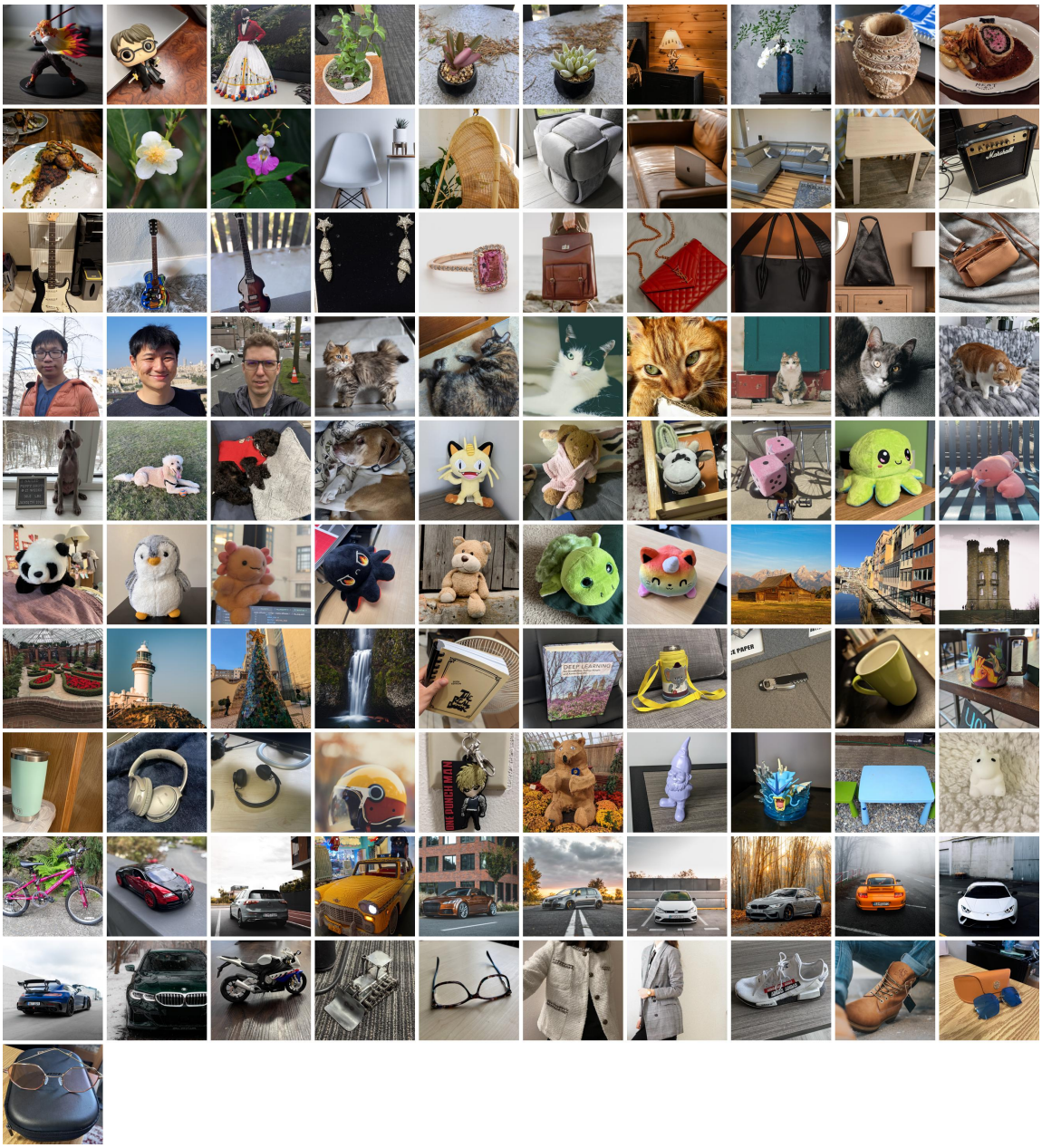


Figure 3.10: **CustomConcept101 dataset.** An example image of each concept in the dataset. Out of 101 categories, 31 are collected from Unsplash, 2 concepts belonging to the flower category are collected from other websites which allow redistribution, and the rest are captured by ourselves. For more details regarding the dataset and license, please refer to our webpage.

As shown in Table 3.4, our method performs on par with DreamBooth for single concepts with marginally lower image-alignment but higher text-alignment. Textual Inversion overfits and achieves poor text-alignment. For multiple concepts, both our joint training and closed-form merging outperform DreamBooth on average. Both ours and DreamBooth use generated images as regularization in this case.

3.6 Extension: Customization with Object Viewpoint Control

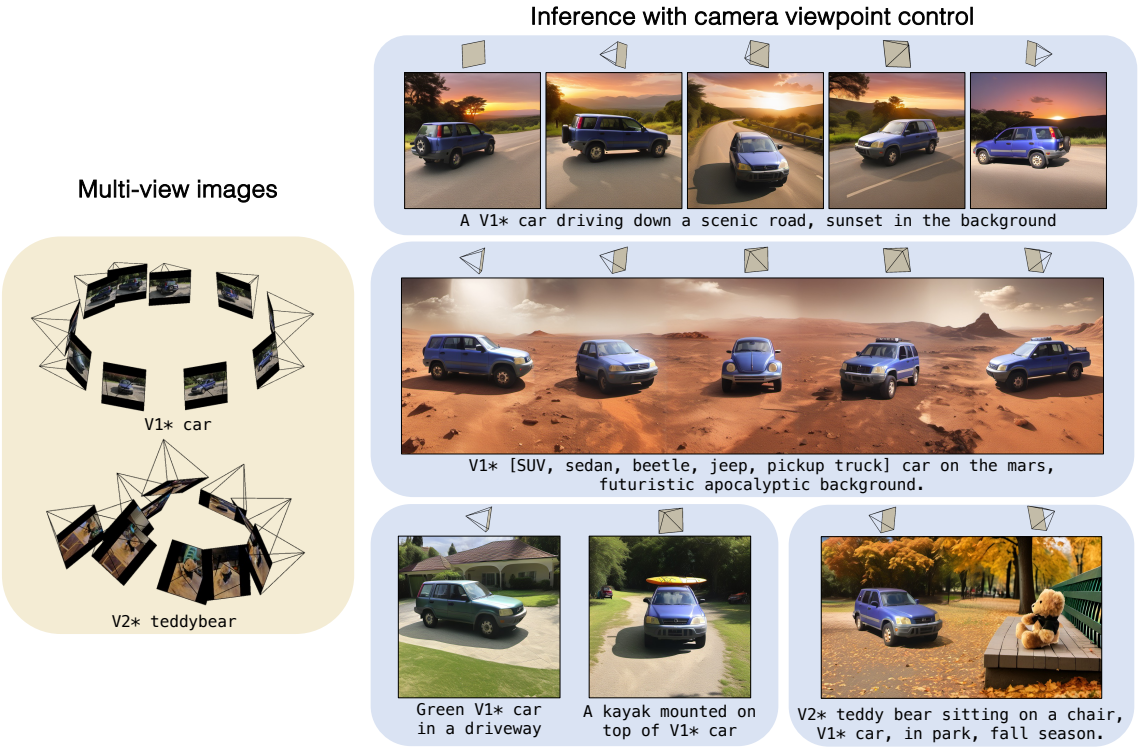


Figure 3.11: **Custom Diffusion-360** introduces explicit object viewpoint control during customization. The diffusion process is conditioned on target viewpoint features rendered from FeatureNeRF blocks that model the 3D structure of the object.

While Custom Diffusion enables efficient customization on new objects and concepts from few images, the resulting models lack precise control over the object's viewpoint. Users must resort to text prompts such as "front-facing" or "side-facing",

which provide only coarse view control. In this section, we extend our method to Custom Diffusion-360 to bridge 3D neural capture and 2D text-to-image diffusion models by providing explicit viewpoint control for custom objects as shown in Figure 3.11.

3.6.1 Method

Given a set of multi-view images $\{\mathbf{x}_i\}_{i=1}^N$ and the corresponding camera poses $\{\pi_i\}_{i=1}^N$, our goal is to now learn the conditional distribution $p(\mathbf{x}|\mathbf{c}, \phi, \{(\mathbf{x}_i, \pi_i)\}_{i=1}^N)$, where $\mathbf{c}$ is the text prompt and ϕ is the camera pose corresponding to the target viewpoint. This allows generation of new variations of the object through text prompts while also controlling the desired object viewpoint. To achieve this, we fine-tune a pretrained text-to-image diffusion model on the given multi-view posed reference images.

Model architecture. Each block in the diffusion model U-Net [218] consists of a ResNet [89], followed by several transformer layers [267]. Given the output of an intermediate ResNet layer $\mathbf{z}$, a standard transformer layer, $F_{\text{standard}}(\mathbf{z}, \mathbf{c})$, consists of a self-attention layer, followed by cross-attention with the text prompt, and a feed-forward MLP, denoted as f. We modify a subset of these transformer layers to incorporate pose conditioning as we explain below. Figure 3.12 shows the overall architecture, with an emphasis on our added pose-conditioning.

Pose-conditioned transformer layer. We denote the pose-conditioned transformer layer as $F_{\text{pose}}(\mathbf{z}_0, \{\mathbf{z}_i, \pi_i\})$, where $\mathbf{z}_0$ is the intermediate feature of the noisy image and $\{\mathbf{z}_i\}$ are the input features corresponding to multi-view reference images (top row in Figure 3.12). We extract spatial features $\{\mathbf{W}_i\}$ from $\{\mathbf{z}_i\}$ using components of pretrained U-Net itself, i.e., $F_{\text{standard}}(\mathbf{z}_i, \mathbf{c})$. To condition the diffusion branch on ϕ, we learn a radiance field, denoted FeatureNeRF, from $\{\mathbf{W}_i, \pi_i\}$ in a feed-forward manner [302]. The predicted FeatureNeRF is then rendered from the target viewpoint ϕ to obtain view-dependent feature map $\mathbf{W}_{\mathbf{x}}^{\phi}$.

Given these rendered features $\mathbf{W}_{\mathbf{x}}^{\phi}$, we concatenate it with the intermediate feature map of the target noisy image, $\mathbf{W}_{\mathbf{x}}$, extracted after the self- and cross-attention layers, and then project it into the original feature dimension using a linear layer. The projected feature is then sent to the feed-forward MLP as usual. Thus, the pose conditioned transformer layer, $F_{\text{pose}}(\mathbf{z}_0, \{\mathbf{z}_i, \pi_i\}, \mathbf{c}, \phi)$ performs:

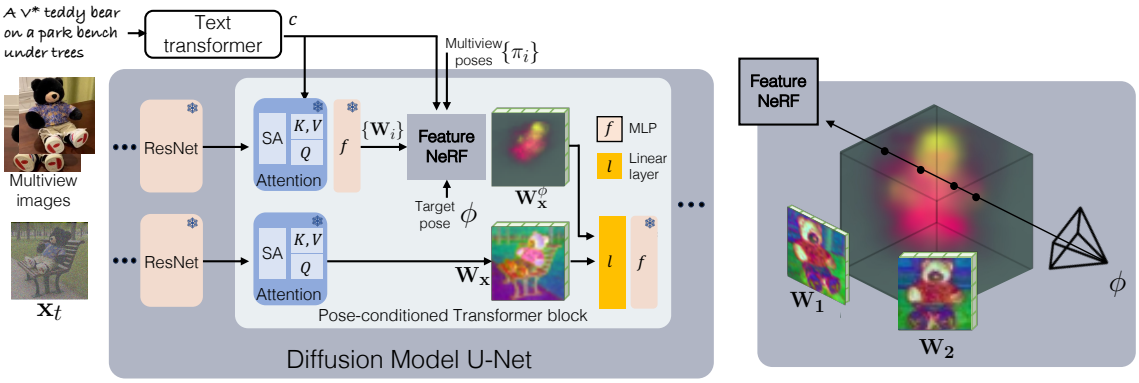


Figure 3.12: **Method.** We update the transformer blocks within diffusion model U-Net with pose conditioning. A FeatureNeRF block aggregates multi-view image features into a 3D volumetric representation and renders it from the target viewpoint, producing $\mathbf{W}_\mathbf{x}^\phi$. This is concatenated with the noisy diffusion feature $W_\mathbf{x}$ and projected back to the original channel dimension via a linear layer l. Only the FeatureNeRF blocks and linear layers, l, in pose-conditioned transformer blocks are fine-tuned.

$$\begin{aligned} \mathbf{W}_\mathbf{x}^\phi &= \text{FeatureNeRF}(\{\mathbf{W}_i, \pi_i\}, \mathbf{c}, \phi) \\ F_\text{pose} &= f(l(\mathbf{W}_\mathbf{x}^\phi \oplus \mathbf{W}_\mathbf{x})) \end{aligned} \tag{3.6}$$

where $\mathbf{W}_i = F_\text{standard}(\mathbf{z}_i, \mathbf{c})$ and l is a learnable weight matrix, which projects the feature into the input space of feed-forward MLP f. We initialize l such that the contribution from $\mathbf{W}_\mathbf{x}^\phi$ is zero at the start of training.

FeatureNeRF. The FeatureNeRF block aggregates per-view features $\{\mathbf{W}_i\}$ with their poses $\{\pi_i\}$ into a single feature map $\mathbf{W}_\mathbf{x}^\phi$ rendered from the target pose ϕ. For a ray r with direction $\mathbf{d}$ cast from the target viewpoint ϕ, we sample 3D points $\mathbf{p}$ along the ray and project each onto the image plane of every reference view π_i, yielding projected coordinates $\pi_i^\mathbf{p}$. We then sample the corresponding feature from $\mathbf{W}_i$ at this coordinate, process it through an MLP, and aggregate the N per-view predictions using a function ψ:

$$\begin{aligned} \mathbf{V}_i^\mathbf{p} &= \text{MLP}(\text{Sample}(\mathbf{W}_i; \pi_i^\mathbf{p}), \gamma(\mathbf{d}), \gamma(\mathbf{p})), \; i = 1, ..., N \\ \overline{\mathbf{V}}^\mathbf{p} &= \psi(\mathbf{V}_1^\mathbf{p}, ..., \mathbf{V}_N^\mathbf{p}), \end{aligned} \tag{3.7}$$

where γ denotes frequency encoding, and $\mathbf{d}$ and $\mathbf{p}$ are first transformed into each reference view's coordinate space [302]. The aggregation function ψ is a learnable

weighted average [213], where a linear layer predicts weights from $\mathbf{V}_i$, π_i, and ϕ. From the aggregated feature $\overline{\mathbf{V}}$ (dropping the superscript $\mathbf{p}$ for clarity), we perform two operations in parallel – (1) predicting density and color and (2) updating the feature with text conditioning via cross-attention:

$$\begin{aligned}(\sigma, \mathbf{C}) &= \text{MLP}(\overline{\mathbf{V}}), \\ \hat{\mathbf{V}} &= \text{CrossAttn}(\overline{\mathbf{V}}, \mathbf{c}).\end{aligned} \tag{3.8}$$

The text-updated features $\hat{\mathbf{V}}$ are then rendered using the predicted densities:

$$\mathbf{W}^{\phi}_{\mathbf{x}}(r) = \sum_{j=1}^{N_f} T_j(1 - \exp(-\sigma_j \delta_j))\hat{\mathbf{V}}_j, \tag{3.9}$$

where r is the target ray, $\hat{\mathbf{V}}_j$ is the feature corresponding to the j^{th} point along the ray, σ_j is the predicted density of that point, N_f is the number of sampled points along the ray between the near and far plane of the camera, and $T_j = \exp(-\sum_{k=1}^{j-1} \sigma_k \delta_k)$ handles occlusion until that point.

FeatureNeRF builds on PixelNeRF [302] with two modifications: we incorporate text cross-attention on the aggregated features and use learnable weight averaging across reference views. Our goal is not to learn a NeRF in feature space [114, 295], but to produce 3D-aware features that the 2D diffusion model can effectively leverage.

Training loss. Our training objective combines two components: the standard diffusion model reconstruction loss and auxiliary FeatureNeRF losses that encourage 3D-consistent intermediate representations. The diffusion loss is:

$$\mathcal{L}_{\text{diffusion}} = \sum_r M w_t \left\| \epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) \right\|^2, \tag{3.10}$$

where M denotes the object mask, which we assume is available, and reconstruction is computed only within the object region. For the FeatureNeRF modules, we supervise with an RGB reconstruction loss:

$$\mathcal{L}_{\text{rgb}} = \sum_r ||M(r)(\mathbf{C}_{gt}(r) - \sum_{j=1}^{N_f} T_j(1 - \exp(-\sigma_j \delta_j))\mathbf{C})||^2, \tag{3.11}$$

and two mask-based losses to ensure the FeatureNeRF models only the object: a silhouette loss [212] $\mathcal{L}_{\text{s}}$ that aligns rendered opacity with the object mask, and a background suppression loss [20, 21] $\mathcal{L}_{\text{bg}}$ that drives the density of background rays to zero:

$$\begin{aligned} \mathcal{L}_{\text{s}} &= \sum_{r} ||M(r) - \sum_{j=1}^{N_f} T_j(1 - \exp(-\sigma_j \delta_j))||^2 \\ \mathcal{L}_{\text{bg}} &= \sum_{r} (1 - M(r)) \sum_{j=1}^{N_f} \|(1 - \exp(-\sigma_j \delta_j))\|^2 , \end{aligned} \tag{3.12}$$

The final training loss is:

$$\mathcal{L} = \mathcal{L}_{\text{diffusion}} + \lambda_{\text{rgb}} \mathcal{L}_{\text{rgb}} + \lambda_{\text{bg}} \mathcal{L}_{\text{bg}} + \lambda_{\text{s}} \mathcal{L}_{\text{s}}, \tag{3.13}$$

where λ_{rgb}, λ_{bg}, and λ_{s} balance intermediate rendering quality against final denoised output, and are fixed across all experiments. The FeatureNeRF losses are averaged across pose-conditioned transformer layers.

Inference. At inference, we balance text and reference view conditioning using combined text and image-classifier-free guidance scheme [26]:

$$\begin{aligned} \hat{\epsilon}_\theta(\mathbf{x}_t, I = \{\mathbf{x}_i, \pi_i\}_{i=1}^{N}, \mathbf{c}) = \epsilon_\theta(\mathbf{x}_t, \varnothing, \varnothing) \\ + s_I(\epsilon_\theta(\mathbf{x}_t, I, \varnothing) - \epsilon_\theta(\mathbf{x}_t, \varnothing, \varnothing)) \\ + s_c(\epsilon_\theta(\mathbf{x}_t, I, \mathbf{c}) - \epsilon_\theta(\mathbf{x}_t, I, \varnothing)), \end{aligned} \tag{3.14}$$

where s_I and s_c control the image and text guidance strength, respectively, trading off reference view fidelity against text prompt adherence.

Training details. We fine-tune Stable Diffusion-XL in our experiments. During training, we sample N equidistant views, using the first as the target viewpoint and the rest as references, and add pose conditioning to 12 out of 70 transformer layers in Stable Diffusion-XL. For rendering, we sample 24 points along each ray. The target concept is described as "V* {category}", with V* as a trainable token embedding similar to Custom Diffusion. To reduce overfitting [219], we use generated images of the same category with ChatGPT-generated captions [38], sampled 25% of the time during training. We also drop the text prompt with 10% probability to enable classifier-free guidance. Further implementation details are provided in

Figure 3.13: **Qualitative comparison** against *image editing methods* (SDEdit, InstructPix2Pix, LEDITS++), *3D editing* (ViCA-NeRF), and our proposed *customization with pose condition* baseline (LoRA + Camera pose). Our method better preserves target identity and viewpoint while following text prompts, e.g., adding a picnic table beside the SUV (1st col.) or recoloring a chair (3rd col.). Ground truth renderings shown as insets.

Appendix A.2.

Method	Text Alignment	Image Alignment	Photorealism
SDEdit vs. Ours	**59.40**%	**63.92**%	**66.89**%
InstructPix2Pix vs. Ours	**55.21**%	**70.66**%	**72.39**%
LEDITS++ vs. Ours	**67.53**%	**64.14**%	**73.82**%
Vica-NeRF vs. Ours	**72.87**%	**75.64**%	**87.10**%
LoRA + Camera pose vs. Ours	**67.64**%	33.03%	47.49%

Table 3.5: **Human preference evaluation**. Percentage of evaluators preferring our method over each baseline (max std. ±3.35%). Our method is preferred over almost all baselines for text alignment, image alignment to the target concept, and photorealism. We find that LoRA + Camera pose overfits the training images, thus having high image alignment but worse text alignment, as also shown in Figure 3.13.

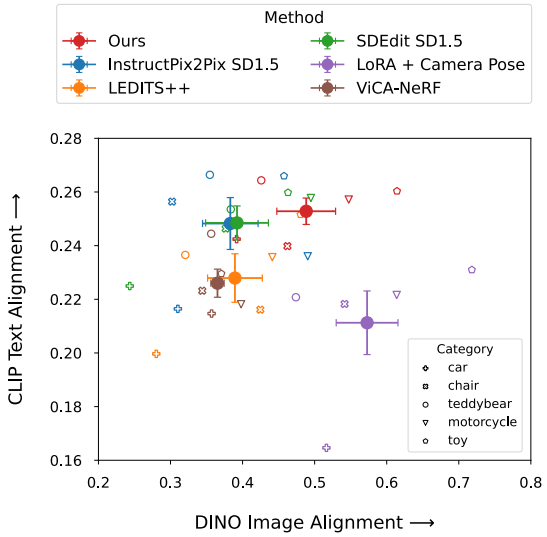
3.6.2 Experiments

Dataset. We select 12 objects from CO3Dv2 [213] (4 categories × 3 instances: car, chair, teddy bear, motorcycle) and 2 toy objects from NAVI [102]. For each instance, we sample ∼ 100 images with their provided camera poses, split evenly for training and evaluation. Camera poses are normalized such that the mean location is the origin and the first camera is at unit norm [314].

Baselines. We compare against three categories of methods: (1) *2D image editing*: LEDITS++ [24], InstructPix2Pix [26], and SDEdit [174]. Since these methods lack explicit viewpoint control, we first render a NeRF [258] from the target viewpoint and then edit the rendered image. (2) *Customization with pose condition*: LoRA+Camera pose, where we modify LoRA [96, 98] by concatenating camera pose to text embeddings, in a similar manner as Zero-1-to-3 [156]. (3) *3D editing*: ViCA-NeRF [61], which trains a NeRF per text prompt.

Evaluation metrics. We create 16 evaluation prompts per category using ChatGPT [38] and manual filtering to remove implausible prompts, covering scene changes, color changes, object composition, and shape changes. The evaluation is done along two axes: (1) *customization quality* that measures text alignment, concept identity preservation, and photorealism, and (2) *viewpoint accuracy*.

For customization quality, we conduct pairwise human preference studies on Amazon Mechanical Turk (∼ 1000 responses per baseline comparison), and also report CLIP Score [208] and DINOv2 [191] similarity for text and image alignment, respec-



(a) Quantitative Comparison

(b) Generalization to unseen camera viewpoints

Figure 3.14: **(a)** CLIP score vs. DINO-v2 score (both higher is better) per category, with overall mean and standard error highlighted. Our method achieves better text alignment while maintaining image similarity. SDEdit and InstructPix2Pix show comparable text alignment but at the cost of photorealism (Table 3.5). **(b)** Our method generalizes to viewpoints outside the training distribution.

tively. For viewpoint accuracy, we compute angular and camera center error between predicted pose from generated images using RayDiffusion [315] and the target pose. More details are given in Appendix A.3.

3.6.2.1 Results

Generation quality and adherence. We first evaluate customization quality independent of viewpoint accuracy. For each evaluation prompt, we generate 3 images at each viewpoint, across 6 target viewpoints, resulting in 288 images per concept.

As shown in Table 3.5, our method is preferred over all baselines except LoRA + Camera pose, which overfits training images and thus achieves higher image alignment at the cost of significantly lower text alignment. Figure 3.14 (a) plots CLIP vs. DINO scores for all methods. Ideally, a method should have both a high CLIP score and a DINO score, but often, there is a trade-off between text and image alignment. Our method achieves on-par or better text alignment while having better image alignment. We observe that image-editing baselines often require careful hyperparameter

tuning per image-prompt pair. For our experiments, we select the best-performing hyperparameters and keep them fixed across all image-prompt settings.

Accuracy of object viewpoint. We evaluate viewpoint accuracy on CO3Dv2 objects by measuring the mean angular error and camera center error between the generated object's pose, predicted using RayDiffusion [315], and the ground truth input pose condition to the model. Our method achieves a mean angular error of 14.19 and a camera center error of 0.08, compared to 41.14 and 0.30 for LoRA + Camera pose—the only other baseline that takes camera pose as input. The large gap reflects that LoRA + Camera pose overfits training views and fails to respect the target viewpoint under new text prompts.

Qualitative comparison. As shown in Figure 3.13, image editing methods often lack photorealism; LoRA + Camera pose overfits training views; and ViCA-NeRF produces blurred outputs, especially for background-changing prompts. In comparison, our method generates the object in the target viewpoint while being better aligned with the input text prompt and object identity.

Generalization to novel viewpoints. Since our method learns a 3D radiance field, it can extrapolate to viewpoints outside the training distribution. Figure 3.14 (b) shows generations with varying camera distance and focal length. Our method also generalizes to shifts in horizontal and vertical camera position.

3.7 Discussion and Limitations

In this chapter, we introduced two optimization-based methods for customization. First, Custom Diffusion, which fine-tunes large-scale text-to-image diffusion models on new concepts from a few reference images. Second, Custom Diffusion-360, wchich extends this framework with explicit viewpoint control, enabling object-centric viewpoint conditioning beyond what text prompts can specify.

Custom Diffusion updates only the key and value projection matrices in the cross-attention layers. This targeted update reduces both compute and storage overhead while enabling novel generations of the concept in unseen contexts. For multi-concept composition, it enables merging of individually fine-tuned models in closed form for coherent joint generation without retraining.

Figure 3.15: **Custom Diffusion failure cases on multi-concept customization.** Our method fails at difficult compositions like similar category objects in a scene. Though as shown on the left, the pretrained model can also struggle with similar compositions.

Despite these strengths, some failure modes persist. As shown in Figure 3.15, composing concepts from similar categories (e.g., a pet dog and a pet cat) remains difficult, as we do not have an explicit constraint to disentangle the learning of visually similar concepts; this mixing is also observed in the pretrained model itself. Composing three or more concepts simultaneously poses a further challenge.

Regarding the limitation of Custom Diffusion-360, the method occasionally struggles to generalize to extreme viewpoints absent from the training data, sometimes defaulting to a seen viewpoint or altering object identity. Following the input viewpoint condition is also unreliable when the text prompt introduces multiple objects, likely due to the model's bias toward object-centric front views present in its pretraining data. Per-object fine-tuning requires approximately 40 minutes; exploring viewpoint conditioning in a feed-forward manner [42, 72] may reduce this cost. The current formulation is further restricted to rigid objects, and extending viewpoint control to dynamic or non-rigid scenes remains an open direction [68, 204, 245].

Chapter 4

Optimization-Based Customization for Concept Removal

The previous chapter focused on adding new concepts to the pretrained text-to-image diffusion model, with additional controls like camera viewpoint. Before returning to concept addition in the next chapter, we turn here to an equally relevant but complementary problem: *removing* unwanted concepts.

Large-scale text-to-image models are trained on massive web-scraped datasets that often contain copyrighted materials, the artistic oeuvre of creators, and personal photos [32, 234, 244]. This data is ultimately owned by its creators, who should have a say in whether their work can be reproduced by generative models. Artists and creators are disproportionately affected, as models can learn to replicate their distinctive styles without consent. A mechanism for dynamically opting out of (or opting into) text-to-image models is therefore essential. However, re-training a model from scratch for every such request to opt out is computationally prohibitive.

In this chapter, we propose *concept ablation*, an efficient customization method that prevents generation of a target concept by aligning its conditional distribution with a broader anchor (e.g., Grumpy Cat → cat, Van Gogh → painting), as shown in Figure 4.1. We formulate this as minimizing the KL divergence between target and anchor distributions. Our method can successfully ablate specific object instances, artistic styles, and memorized images, takes only about five minutes per concept, and can ablate multiple concepts simultaneously.

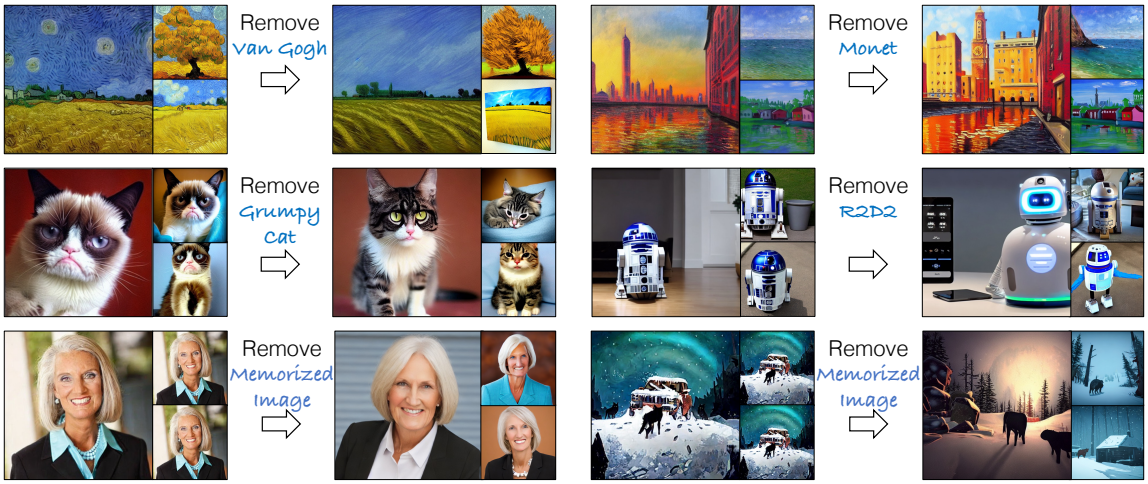


Figure 4.1: **Concept Ablation.** Our method ablates (removes) concepts from a pretrained text-to-image diffusion model. Given a target concept to ablate (e.g., `Van Gogh` or `Grumpy Cat`), we modify the model's conditional distribution to match a broader anchor concept (e.g., `painting` or `cat`), effectively preventing the generation of the target concept while preserving related concepts.

4.1 Introduction

Text-to-image models have achieved photorealistic image synthesis at scale [37, 184, 211, 217, 223, 304], driven in large part by web-scraped datasets containing billions of image-text pairs [232]. However, these datasets often include copyrighted artwork, personal photographs, and the distinctive styles of living artists [32, 234, 244].

We believe that every creator should have the right to *opt out of* large-scale models at any time for any image they have created. However, fulfilling such requests poses new challenges, as re-training a model from scratch for every request can be computationally intensive. Here, we ask – *How can we prevent the model from generating such content? How can we achieve it efficiently without re-training the model from scratch? How can we make sure that the model still preserves related concepts?*

These questions motivate our work on ablation (removal) of concepts from text-conditioned diffusion models [57, 217]. We perform concept ablation by modifying generated images for the target concept ($\mathbf{c}^*$) to match a broader anchor concept ($\mathbf{c}$), e.g., overwriting `Grumpy Cat` with `cat` or `Van Gogh` paintings with `painting` as shown in Figure 4.1. Thus, given the text prompt, `painting of olive trees in the style of Van Gogh`,

the model generates a normal painting of olive trees even though the text prompt contains Van Gogh. Similarly, for specific instances/objects like Grumpy Cat, the model generates a random cat after the ablation process.

Formally, our method aims at modifying the conditional distribution $p_\theta(\cdot|\mathbf{c}^*)$ of the model given the target concept $\mathbf{c}^*$ to match a new distribution $p(\cdot|\mathbf{c})$ defined by the anchor concept $\mathbf{c}$, by minimizing the Kullback–Leibler divergence between the two. We propose two instantiations of the anchor distribution, each yielding a different training objective. The first, a model-based anchor distribution, defines $p(\cdot|\mathbf{c})$ via the model's own predictions on prompts containing the anchor concept, e.g., aligning predictions for A cute little Grumpy Cat with those for A cute little cat. The second, an image-based anchor distribution, defines $p(\cdot|\mathbf{c})$ through modified text-image pairs: the target concept prompt paired with real images of the anchor, e.g., the prompt a cute little Grumpy Cat paired with a random cat image. We show that both objectives effectively ablate the target concept.

We evaluate our method on 16 concept ablation tasks, including specific object instances, artistic styles, and memorized images. Our method can successfully ablate target concepts while minimally affecting closely related surrounding concepts that should be preserved (e.g., other cat breeds when ablating Grumpy Cat). Our method takes around five minutes per concept. Furthermore, we perform an extensive ablation study regarding different algorithmic design choices, such as the objective function variants, the choice of parameter subsets to fine-tune, the choice of anchor concepts, the number of fine-tuning steps, and the robustness of our method to misspelling in the text prompt. Finally, we show that our method can ablate multiple concepts at once and discuss the current limitations. Our code is available on our project website.

4.2 Related Work

Training data memorization and unlearning. A key challenge in large-scale generative models is that they can memorize and reproduce exact or near-exact copies of training data [32, 244], which poses privacy and security risks, particularly for models trained on uncurated web-scale datasets. More broadly, training data leaking has been studied via membership inference [30, 31, 33, 239]. The field of machine unlearning [22, 29, 81, 83, 84, 183, 233, 260] explores how to delete specific training examples

or data from a trained model after deployment. However, existing unlearning methods typically require computing global information (e.g., Fisher Information Matrix) over the entire training dataset, making them computationally infeasible for large-scale models with billions of parameters. Our work addresses this gap by proposing an efficient method to ablate target concepts from pretrained diffusion models in just five minutes, without requiring the original training set.

Text-to-image synthesis and privacy concerns. As described in Chapter 3, recent text-to-image models demonstrate remarkable generalization ability. However, these models are often trained on copyrighted images and can inadvertently mimic artistic styles [234, 244] and generate copyrighted content. Previous approaches to concept removal in generative models include Kong *et al.* [123], who address data removal in GANs by retraining on redacted data, a time-consuming process infeasible for large-scale models. Schramowski et al. [230] propose inference-time modifications to prevent certain concepts, but our goal is to ablate concepts from model weights themselves. Concurrent with our work, Gandikota *et al.* [73] also propose a score-based concept erasure approach.

Generative model fine-tuning and customization. Fine-tuning is a standard approach to adapt pretrained generative models to new domains, downstream tasks, or specific images [14, 91, 113, 144, 180, 188, 189, 193, 216, 275, 276]. Building on the parameter-efficient customization approach of Chapter 3, which enables adding new concepts to text-to-image models [71, 72, 130, 219], our work poses the complementary problem: how do we efficiently remove concepts? More broadly, model editing [15, 70, 175, 176, 179, 187, 271, 272] modifies model weights to incorporate new computational rules or visual effects. Unlike these approaches, which expand the model's generative capacity, our method restricts it by ablating specific concepts.

4.3 Method

In this section, we present our concept ablation formulation to prevent generation of a target concept from the text-to-image diffusion models. For a brief overview of diffusion models, please refer to the Background Section 2.1. Here, we introduce our two concept ablation variants in Section 4.3.1, followed by our derivation of the loss

objective in Section 4.3.2, and finally training details in Section 4.3.3.

4.3.1 Concept Ablation

We define concept ablation as the task of preventing the generation of the desired image corresponding to a given target concept that needs to be ablated. As re-training the model on a new dataset with the concept removed is impractical, this becomes a challenging task. We need to ensure that customizing a model to ablate a particular concept does not affect its performance on other closely related concepts.

A naïve approach. Our first attempt is to simply maximize the diffusion model training loss [123, 260] on the text-image pairs for the target concept while imposing regularizations on the weights. Unfortunately, this method leads to worse results on close surrounding concepts of the target concept. We compare our method with this baseline in Section 4.4.1 (Figure 4.3) and show that it performs sub-optimally.

Our formulation. As concept ablation prevents the generation of the target concept, thus the question arises: what should be generated instead? Here, we assume that the user provides the desired anchor concept, e.g., cat for Grumpy Cat. The anchor concept overwrites the target concept and should be a superset or similar to the target concept. Thus, given a set of text prompts $\{\mathbf{c}^*\}$ describing the target concept, we aim to match the following two distributions via Kullback–Leibler (KL) divergence:

$$\arg\min_{\theta} \mathcal{D}_{\mathcal{KL}}(p(\mathbf{x}_{(0\ldots T)}|\mathbf{c})||p_\theta(\mathbf{x}_{(0\ldots T)}|\mathbf{c}^*)), \tag{4.1}$$

where $\mathbf{x}_t$ denotes the noisy image at diffusion timestep $t \in [0, T]$ from the anchor concept distribution, $p(\mathbf{x}_{0:T}|\mathbf{c})$, while $p_\theta(\mathbf{x}_{0:T}|\mathbf{c}^*)$ is the distribution induced by the model when conditioned on the target prompt $\mathbf{c}^*$. Intuitively, we want to associate target prompts $\{\mathbf{c}^*\}$ with images corresponding to anchor prompts $\{\mathbf{c}\}$.

Computing the above objective requires samples from the anchor distribution $p(\cdot|\mathbf{c})$. We create a small dataset of $(\mathbf{c}, \mathbf{c}^*)$ tuples, where $\mathbf{c}$ is a random prompt for the anchor concept and $\mathbf{c}^*$ is modified from $\mathbf{c}$ to include the target concept. For example, if $\mathbf{c}$ is photo of a cat then $\mathbf{c}^*$ is photo of a Grumpy Cat.

Model-based concept ablation. Here, we match target concept distribution, $p_\theta(\mathbf{x}_{(0\ldots T)}|\mathbf{c}^*)$, to pretrained model's anchor concept distribution $p_{\theta_{\text{init}}}(\mathbf{x}_{(0\ldots T)}|\mathbf{c})$. The

fine-tuned network should have a similar distribution of generated images given $\mathbf{c}^*$ as that of $\mathbf{c}$. This is similar to the standard diffusion model training objective, except the target distribution is defined by the pretrained model instead of training data. Eqn. 4.1 can be expanded as

$$\arg\min_{\theta} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{p_{\theta_{\text{init}}}(\mathbf{x}_0 \dots \mathbf{x}_T|\mathbf{c})} \left[\log \frac{p_{\theta_{\text{init}}}(\mathbf{x}_{t\text{-}1}|\mathbf{x}_t, \mathbf{c})}{p_{\theta}(\mathbf{x}_{t\text{-}1}|\mathbf{x}_t, \mathbf{c}^*)} \right] \tag{4.2}$$

where the noisy intermediate latent $\mathbf{x}_t \sim p_{\theta_{\text{init}}}(\mathbf{x}_t|\mathbf{c})$, G_θ with parameters θ is the new network we aim to learn, and $G_{\theta_{\text{init}}}$ is the original network. We can optimize the KL divergence by minimizing the following equivalent objective:

$$\arg\min_{\theta} \mathbb{E}_{\epsilon, \mathbf{x}_t, \mathbf{c}^*, \mathbf{c}, t} \left[w_t \left\| \epsilon_{\theta_{\text{init}}}(\mathbf{x}_t, \mathbf{c}, t) - \epsilon_{\theta}(\mathbf{x}_t, \mathbf{c}^*, t) \right\|^2 \right]. \tag{4.3}$$

We show the full derivation in Section 4.3.2. Unfortunately, directly optimizing this objective requires sampling from $p_{\theta_{\text{init}}}(\mathbf{x}_t|\mathbf{c})$ and maintaining two large networks ($G_{\theta_{\text{init}}}$ and G_θ) in VRAM, which is prohibitively expensive in both time and memory. To make this tractable, we instead approximate $\mathbf{x}_t$ via the forward diffusion process and assume that the fine-tuned model G_θ remains close to the original $G_{\theta_{\text{init}}}$ on the anchor concept. This allows us to use G_θ to model the anchor concept distribution as well. We apply a stop-gradient on the predicted noise for the anchor concept by G_θ, yielding the following final training objective:

$$\mathcal{L}_{\text{model}}(\mathbf{x}, \mathbf{c}, \mathbf{c}^*) = \mathbb{E}_{\epsilon, \mathbf{x}, \mathbf{c}^*, \mathbf{c}, t} \left[w_t \left\| \text{sg}[\epsilon_{\theta}(\mathbf{x}_t, \mathbf{c}, t)] - \epsilon_{\theta}(\mathbf{x}_t, \mathbf{c}^*, t) \right\|^2 \right], \tag{4.4}$$

where $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$. This trains the model to produce the same denoising prediction for the target prompt $\mathbf{c}^*$ as it would for the anchor prompt $\mathbf{c}$, as shown in Figure 4.2 (left). We also explore a reverse KL divergence objective in Section 4.4.2.

Noise-based concept ablation. Alternatively, we can define the anchor distribution as:

$$p(\mathbf{x}_{(0:T)}|\mathbf{c}) = p_{\text{data}}(\mathbf{x}_0|\mathbf{c}) \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_0), \tag{4.5}$$

where $p_{\text{data}}(\mathbf{x}_0|\mathbf{c})$ is the true data distribution over images of the anchor concept and $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \boldsymbol{I})$ is the forward diffusion process. Substituting into Eqn. 4.1 and simplifying yields the standard diffusion loss objective with anchor images as the real data:

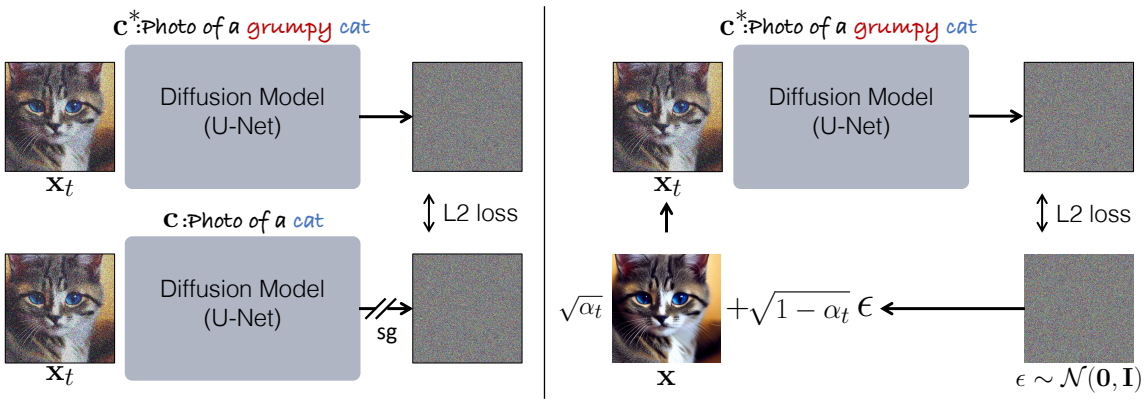


Figure 4.2: **Method.** We fine-tune the model so that the target concept (e.g., grumpy cat) maps to a broader anchor concept (e.g., cat). *Left (model-based):* the fine-tuned model matches the pretrained model's predictions on anchor prompt $\mathbf{c}$ when given target prompt $\mathbf{c}^*$. *Right (noise-based):* the model is trained on redefined pairs $\langle \mathbf{c}^*$, anchor image $\mathbf{x}\rangle$. We find the model-based variant to be more effective in practice.

$$\mathcal{L}_{\text{noise}}(\mathbf{x}, \mathbf{c}, \mathbf{c}^*) = \mathbb{E}_{\epsilon,\mathbf{x},\mathbf{c}^*,t}\left[w_t \left\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}^*, t)\right\|^2\right], \tag{4.6}$$

where $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$. Since $p_{\text{data}}(\mathbf{x}_0|\mathbf{c})$ is unavailable, we approximate it by sampling $\mathbf{x} \sim p_{\theta_{\text{init}}}(\mathbf{x}|\mathbf{c})$ using the pretrained model. Intuitively, this objective teaches the model to denoise toward anchor concept images even when conditioned on target concept prompts, e.g., pairing photo of a grumpy cat with a generated cat image. As shown in Figure 4.2, the two variants differ in their supervision signal: the model-based objective (Eqn. 4.4) matches the model's own predictions, while the noise-based objective (Eqn. 4.6) supervises with the ground-truth noise ϵ added to anchor images.

4.3.2 Model-based Concept Ablation

We show here that minimizing the KL divergence between the joint distributions of noisy latent variables conditioned on the anchor and target concepts (Eqn. 4.2)

reduces to an ℓ_2 loss between predicted noise vectors.

$$\begin{aligned}
&\mathcal{D}_{\mathcal{KL}}(p_{\theta_{\text{init}}}(\mathbf{x}_{(0\ldots T)}|\mathbf{c})||p_{\theta}(\mathbf{x}_{(0\ldots T)}|\mathbf{c}^*)) \\
&= \mathbb{E}_{p_{\theta_{\text{init}}}(\mathbf{x}_0\ldots\mathbf{x}_T)} \log \frac{\prod_{t=1}^{T} p_{\theta_{\text{init}}}(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{c}) p_{\theta_{\text{init}}}(\mathbf{x}_T)}{\prod_{t=1}^{T} p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{c}^*) p_{\theta}(\mathbf{x}_T)} \\
&= \sum_{\hat{t}=1}^{T} \mathbb{E}_{p_{\theta_{\text{init}}}(\mathbf{x}_0\ldots\mathbf{x}_T)} \log \frac{p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})}{p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*)}
\end{aligned} \tag{4.7}$$

We expand the term corresponding to a particular time step $\hat{t}$, i.e.,

$$\begin{aligned}
&\mathbb{E}_{p_{\theta_{\text{init}}}(\mathbf{x}_0\ldots\mathbf{x}_T)} \log \frac{p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})}{p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*)} \\
&= \int_{\mathbf{x}_{(0\ldots T)}} \prod_{t=1}^{T} p_{\theta_{\text{init}}}(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{c}) p(\mathbf{x}_T) \log \frac{p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})}{p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*)} d\mathbf{x}_{(0\ldots T)} \\
&= \int_{\mathbf{x}_{(\hat{t}\ldots T)}} p_{\theta_{\text{init}}}(\mathbf{x}_{(\hat{t}\ldots T)}|\mathbf{c}) \Bigg[\int_{\mathbf{x}_{(0\ldots\hat{t}-1)}} \prod_{t=1}^{\hat{t}} p_{\theta_{\text{init}}}(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{c}) \\
&\qquad\qquad \log \frac{p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})}{p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*)} d\mathbf{x}_{(\hat{t}-1\ldots 0)} \Bigg] d\mathbf{x}_{(\hat{t}\ldots T)} \\
&= \int_{\mathbf{x}_{\hat{t}}} p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}}|\mathbf{c}) \Bigg[\int_{\mathbf{x}_{(0\ldots\hat{t}-1)}} (\prod_{t=1}^{\hat{t}-1} p_{\theta_{\text{init}}}(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{c})) p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}) \\
&\qquad\qquad \log \frac{p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})}{p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*)} d\mathbf{x}_{(\hat{t}-1\ldots 0)} \Bigg] d\mathbf{x}_{\hat{t}} \\
&= \int_{\mathbf{x}_{\hat{t}}} p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}}|\mathbf{c}) \Bigg[\int_{\mathbf{x}_{\hat{t}-1}} p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}) \log \frac{p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})}{p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*)} \\
&\qquad\qquad \Big[\int_{\mathbf{x}_{(0\ldots\hat{t}-2)}} \prod_{t=1}^{\hat{t}-1} p_{\theta_{\text{init}}}(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{c}) d\mathbf{x}_{(\hat{t}-2\ldots 0)} \Big] d\mathbf{x}_{\hat{t}-1} \Bigg] d\mathbf{x}_{\hat{t}}
\end{aligned}$$

The integral over $d\mathbf{x}_{(\hat{t}-2\ldots 0)}$ will be 1 since it is an integration of the probability distribution over the range it is defined. Thus the previous term can be re-written

as:

$$\mathbb{E}_{\mathbf{x}_{\hat{t}}\sim p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}}|\mathbf{c})}\left[\int_{\mathbf{x}_{\hat{t}-1}} p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})\log\frac{p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})}{p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*)}d\mathbf{x}_{\hat{t}-1}\right]$$
$$=\mathbb{E}_{\mathbf{x}_{\hat{t}}\sim p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}}|\mathbf{c})}\left[\mathcal{D}_{\mathcal{KL}}(p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})||p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*))\right]$$
$$=\mathbb{E}_{\mathbf{x}_{\hat{t}}\sim p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}}|\mathbf{c})}\left[\eta(\epsilon_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}},\mathbf{c},t)-\epsilon_{\theta}(\mathbf{x}_{\hat{t}},\mathbf{c}^*,t))^2\right]$$

In diffusion models, each conditional $p_{\theta_{\text{init}}}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c})$ and $p_{\theta}(\mathbf{x}_{\hat{t}-1}|\mathbf{x}_{\hat{t}},\mathbf{c}^*)$ is a Gaussian with fixed variance and mean that is a linear function of $\mathbf{x}_t$ and the predicted noise. The KL divergence between two Gaussians with shared variance reduces to the squared difference between their means, yielding the ℓ_2 loss on predicted noise vectors above. We ignore the variance terms as they are not learned.

4.3.3 Training Details

Regularization loss. We additionally regularize with the standard diffusion loss on anchor concept pairs $(\mathbf{x},\mathbf{c})$ [130, 219], weighted by a hyperparameter λ. This is necessary because the target prompt often contains the anchor concept (e.g., `cat` appears in `Grumpy cat`), and without regularization, fine-tuning could inadvertently alter the model's distribution over the anchor concept itself.

Parameter subset to update. We explore three fine-tuning strategies: (1) *Cross-Attention*: updating only the key and value projection matrices in the U-Net cross-attention layers, as in Custom Diffusion; (2) *Embedding*: updating only the text token embeddings [71]; and (3) *Full Weights*: updating all U-Net parameters [219].

Training data construction. Our method requires anchor prompts $\{\mathbf{c}\}$, target prompts $\{\mathbf{c}^*\}$, and generated anchor images $\mathbf{x}$ for training. The construction varies by ablation task. For *instances* (e.g., ablating `Grumpy cat` → `cat`), we use ChatGPT [38] to generate 200 anchor prompts containing the anchor concept, generate 1,000 images from the anchor prompts using the pretrained model, and obtain target prompts by replacing the anchor with the target concept name in the anchor prompt. For *styles*, the anchor concept is a generic category such as `painting`; we retrieve 200 prompts similar to this word in CLIP feature space using clip-retrieval [17], generate 1,000 images, and construct target prompts by appending `in the style of {target style}` to the

anchor prompts. For *memorized images*, we assume the memorized prompt $\mathbf{c}^*$ is known [32]. We use ChatGPT to generate paraphrases, then select the prompts that most frequently reproduce the memorized image as target prompts (4 prompts) and those that least frequently reproduce it as anchor prompts (10 prompts). We generate 1,000 images from the anchor prompts and filter out any memorized outputs using image similarity metrics [32, 202] and use the remaining ones for training.

4.4 Experiments

In this section, we show the results of our method on ablating various concepts and memorized images. All our experiments are based on the Stable Diffusion model [57]. Implementation and hyperparameter details are provided in Appendix B.1.

Baseline. We compare against a loss maximization baseline inspired by Tanno *et al.* [260], which maximizes the diffusion training loss, $\mathcal{L}$ (Eqn. 2.3), on the target concept while regularizing the updates toward the pretrained weights:

$$\arg\min_\theta \max(1 - \mathcal{L}(\mathbf{x}^*, \mathbf{c}^*), 0) + \lambda_{\text{reg}} \left\|\theta - \theta_{\text{init}}\right\|_2 , \tag{4.8}$$

where $\mathbf{x}^*$ are images generated with target prompt $\mathbf{c}^*$.

Evaluation metrics. We evaluate ablation quality using two CLIP-based metrics [92]: *CLIP Score*, which measures the similarity between generated images and the target concept text in CLIP feature space, and *CLIP accuracy*, which performs binary classification between the ablated and anchor concepts via cosine distance between the generated image and the two class prompts. For the target concept that we want to ablate, lower values on both metrics indicate more successful ablation. We also test robustness to small spelling variations of the ablated prompts. To ensure that ablation does not harm related concepts, we evaluate the same metrics on *surrounding concepts* (e.g., similar cat breeds when ablating Grumpy Cat); here, higher CLIP Score and accuracy indicate better preservation.

Apart from CLIP-based metrics, we additionally use *KID* [18] to measure performance. Higher KID between images generated by the fine-tuned and pretrained models on the target concept indicates successful ablation (the two distributions have diverged), while lower KID on anchor and nearby concepts indicates preservation. We

generate 200 images per concept category using 10 prompts (ChatGPT-generated for instances, manually captioned for styles) with 50 DDPM steps.

To measure the performance on ablating memorized images, following [32, 202], we use SSCD [202] to measure the fraction of generated images exceeding a similarity threshold with the memorized image.

4.4.1 Results

Ablating instances. We evaluate on four target → anchor pairs: (1) Grumpy Cat → Cat, (2) Snoopy → Dog, (3) Nemo → Fish, and (4) R2D2 → Robot. Figure 4.3 compares both proposed objectives and the loss maximization baseline using *Cross-Attention* fine-tuning. The *model-based* variant converges faster and performs on par or better than *noise-based*, as also shown qualitatively on the Nemo instance in Figure 4.5. We therefore use *model-based* variant for all subsequent experiments.

Figure 4.4 and 4.5 compare updating different parameter subsets during fine-tuning and show that all variants successfully redirect the target concept to its anchor. However, fine-tuning only the text embedding is less robust to minor spelling variations and still produces the target concept similar to the pretrained model, as shown in Figure 4.4 (third column) and Figure 4.6 (a).

Comparing our model against the loss maximization baseline, the baseline method progressively degrades image quality into noise with longer training, and even with early stopping causes notable deterioration on nearby concepts, as reflected by the drop in CLIP Score and Accuracy in Figure 4.3 (1st column). Figure 4.6 (b) also shows qualitative samples comparing the baseline and our method on R2D2 and its nearby concept BB8, which shows a similar trend.

Ablating styles. We evaluate on ablating four artistic styles—Van Gogh, Salvador Dali, Claude Monet, and Greg Rutkowski, all of which are mapped to the anchor concept of generic painting. As shown quantitatively in Figure 4.4 and qualitatively in Figure 4.7, our method successfully removes the target style while minimally affecting nearby artistic styles across fine-tuning different parameter subsets.

Memorized images. We select eight memorization examples identified by prior work [32, 244]. Figure 4.8 shows qualitative results on four of these with generations before and after fine-tuning. The ablated model produces diverse outputs for the

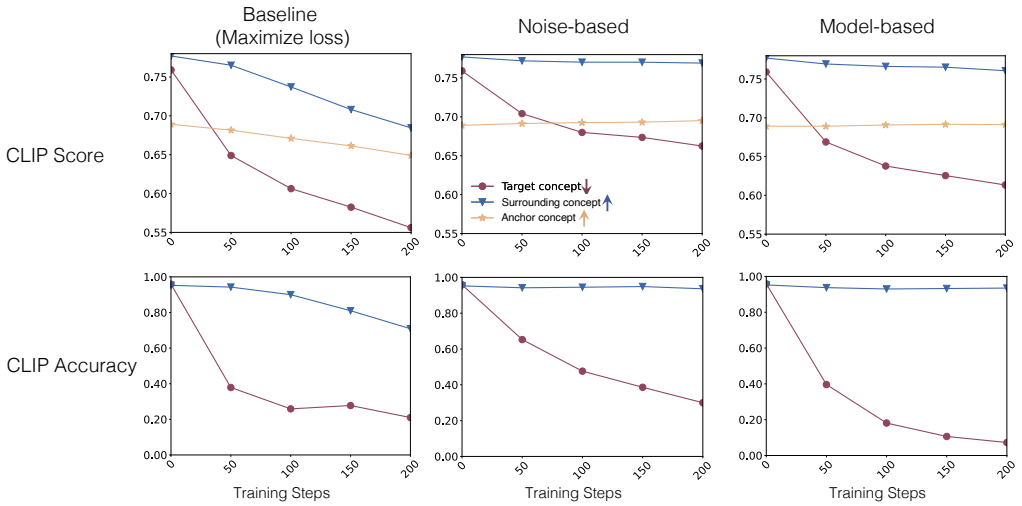


Figure 4.3: **Comparison of different learning objectives.** The *model-based* concept ablation converges faster than the *noise-based* variant while maintaining better performance on nearby concepts. Loss maximization baseline leads to the deterioration of surrounding and anchor concepts (lower CLIP Score and Accuracy).

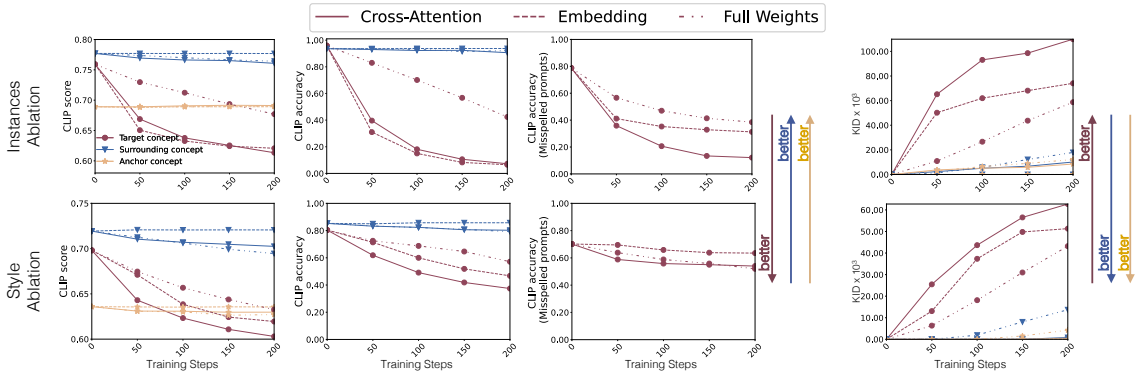


Figure 4.4: **Quantitative evaluation for ablating instances (top) and styles (bottom).** Performance of our final *model-based* method across training steps when updating different parameter subsets, averaged over four target concepts. Updating either embedding or cross-attention parameters leads to faster convergence than full fine-tuning. Cross-attention fine-tuning is marginally worse on nearby concepts but more robust to spelling variations (third column).

same prompt instead of reproducing the memorized image. We find that fine-tuning *Full Weights* yields the best results for ablating memorized images. Table 4.1 reports the fraction of generated samples with SSCD similarity ≥ 0.5 to the memorized image, which drops from 60% before ablation to 0.3% after.

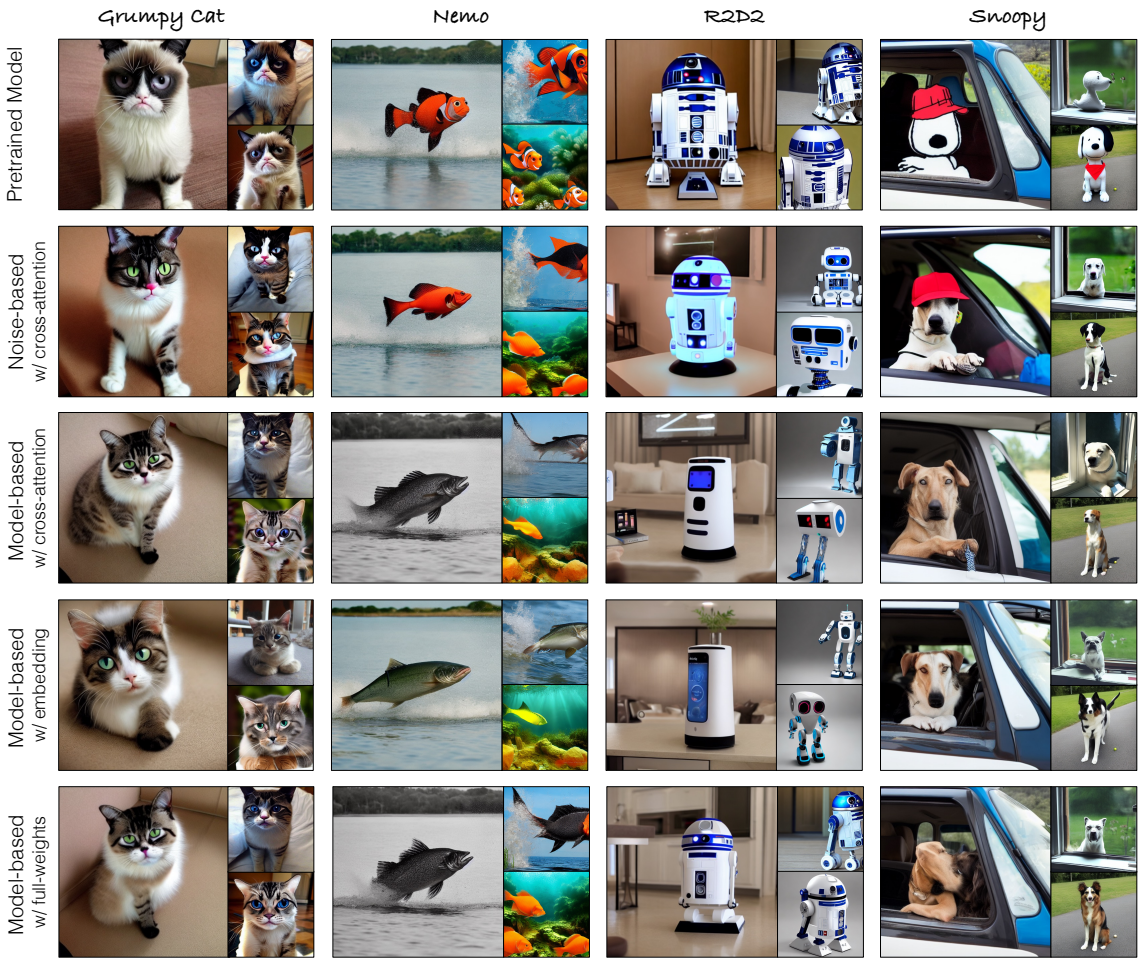


Figure 4.5: **Instance ablation results.** Each row shows a different method variant. Our *model-based* variant outperforms *noise-based*, particularly on Nemo and R2D2. Within *model-based*, different parameter subsets perform comparably, though embedding-only fine-tuning is less robust to spelling variations as shown in Figure 4.6.

4.4.2 Ablation

Ablating multiple concepts in a single model. Our method extends naturally to removing multiple concepts simultaneously by training on the union of their datasets with additional training steps. Figure 4.9 shows results for a single model with all four instances ablated and another model with all four styles ablated, both using *model-based* with cross-attention fine-tuning.

Role of anchor category. To assess sensitivity to the choice of anchor category, we ablate Grumpy Cat using two different anchors: British Shorthair Cat and Felidae. As

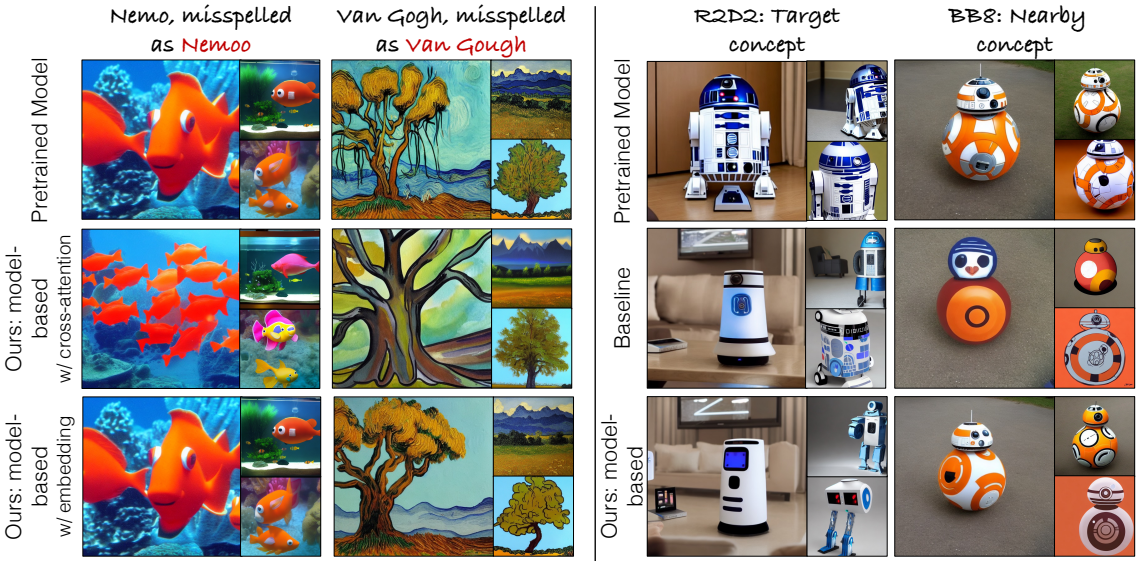


Figure 4.6: **(a) Robustness to spelling mistakes.** Ablation by fine-tuning only the embedding is easily circumvented by minor spelling variations, allowing the target concept to be regenerated. Cross-attention fine-tuning remains robust. **(b) Baseline vs. ours** on R2D2 ablation. Our method better preserves the nearby concept BB8, generating images closer to the pretrained model. Both methods here fine-tune cross-attention parameters.

Target Prompt	**Pretrained Model**	**Ours** (*Full Weights*)
New Orleans House Galaxy Case	65.5	0.0
Portrait of Tiger in black and white by Lukas Holas	50.0	0.0
VAN GOGH CAFE TERASSE copy.jpg	56.5	1.5
Captain Marvel Exclusive Ccxp Poster Released Online By Marvel	95.0	0.5
Sony Boss Confirms Bloodborne Expansion is Coming	83.5	0.5
Ann Graham Lotz	26.5	0.0
The Long Dark Gets First Trailer, Steam Early Access	100.0	0.0
A painting with letter M written on it Canvas Wall Art Print	4.0	0.0
Average	60.1	0.3

Table 4.1: **Memorization rate** before and after ablation, measured as the percentage of generated samples with ≥ 0.5 SSCD cosine similarity to the memorized image. Our method reduces the rate to near 0% across all cases.

shown in Figure 4.9, both yield successful ablation, suggesting our method is robust to the specific choice of anchor concept.

Reverse KL divergence. Our *model-based* objective minimizes the forward KL

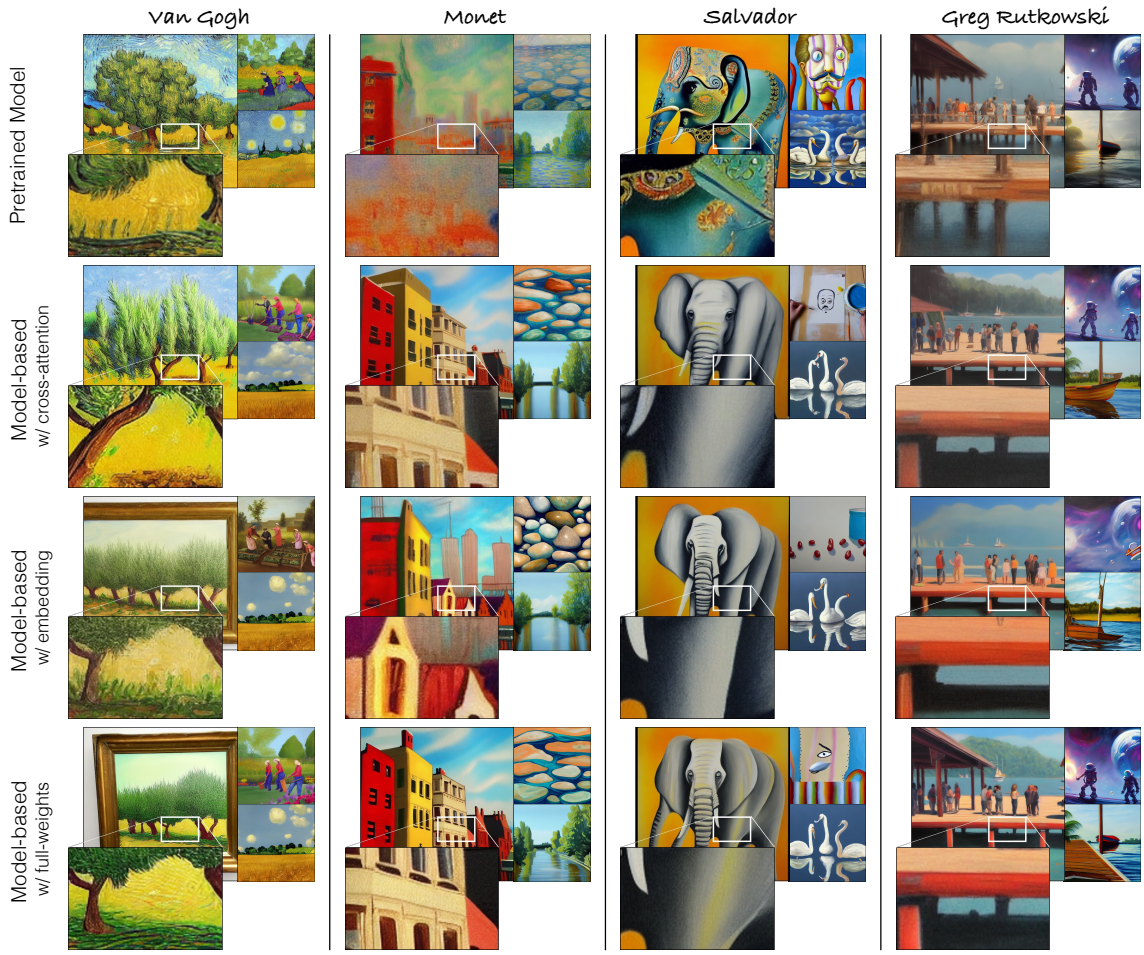


Figure 4.7: **Ablating styles with the *model-based* variant.** The ablated model generates similar content as the pretrained model but without the unique style.

divergence, where the expectation is over anchor concept images. We compare this with a reverse KL variant, where the expectation is over target concept distribution: $\mathbb{E}_{\epsilon,\mathbf{x}^*,\mathbf{c}^*,\mathbf{c},t}\left[w_t \left\|\text{sg}[\epsilon_\theta(\mathbf{x}_t^*, \mathbf{c}, t)] - \epsilon_\theta(\mathbf{x}_t^*, \mathbf{c}^*, t)\right\|^2\right]$. With forward KL, training images are generated from anchor prompts, such as generic painting captions and target captions are modified from anchor prompts, e.g., by appending *in the style of Van Gogh*. Thus, ablation is limited to the modes covered by these anchor prompts. With reverse KL, the expectation is taken over samples from the target concept distribution, so all modes the model associates with the target are covered. Empirically, both objectives perform similarly on average. However, for target concepts with distinctive modes unlikely to appear under generic anchor prompts, such as Van Gogh's Starry Night

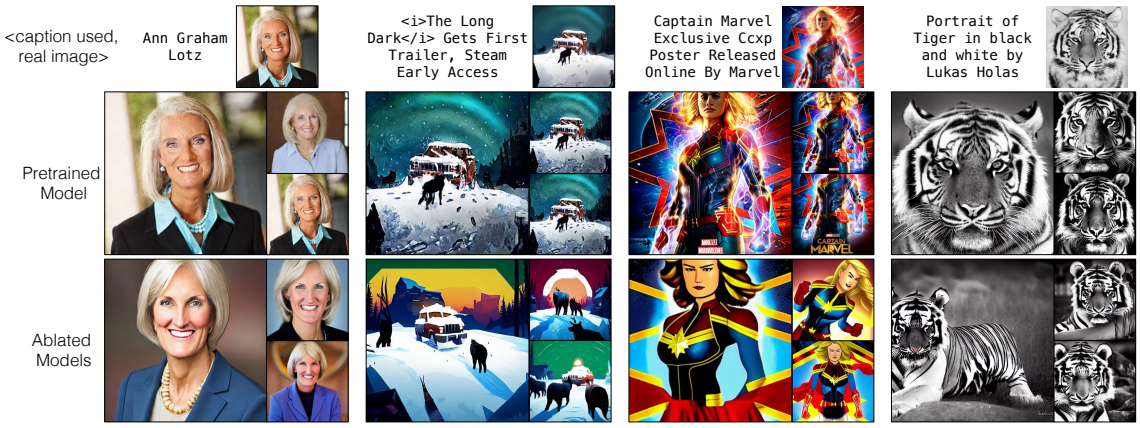
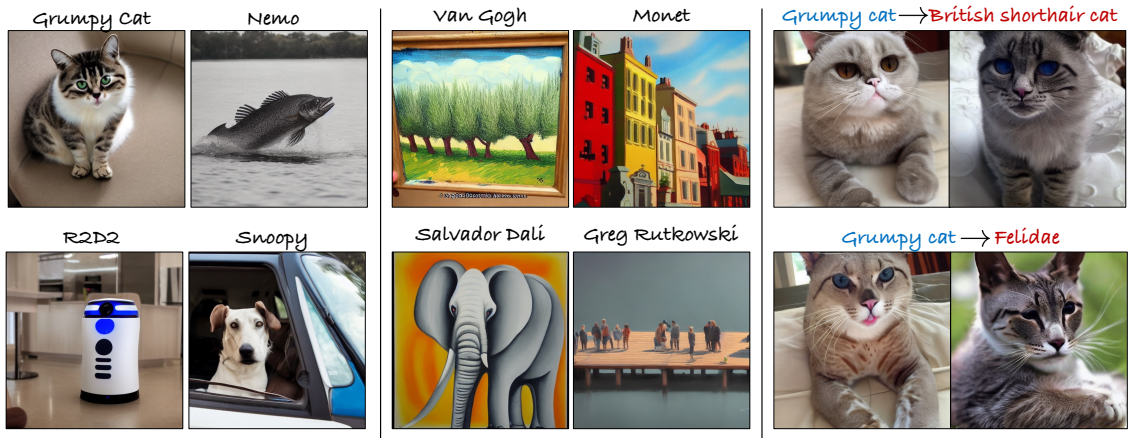


Figure 4.8: **Ablating memorized images with the *model-based* variant.** Text-to-image diffusion models often learn to generate exact or near-exact copies of real images. We fine-tune the model to map the generated image distribution for the given text prompt to images generated with its variations. This results in the fine-tuned model generating different variations instead of copying the real image.



(a) Ablating multiple instances at once (b) Ablating multiple styles at once (c) Varying anchor concepts

Figure 4.9: **(a) Ablating multiple instances** and **(b) multiple styles** simultaneously in a single model, with one sample shown per ablated concept. **(c) Robustness to anchor choice:** our method successfully ablates `Grumpy Cat` with different anchor concepts (`British shorthair cat` and `Felidae`).

paintings, the reverse KL variant achieves more complete ablation (Figure 4.10).

Comparison with negative prompts and Safe Latent Diffusion. Figures 4.11 and 4.12 compare instance ablation via CLIP Score against these two inference-time concept removal methods. Our method ablates the target concept while better preserving surrounding concepts than both baselines. For the negative prompt method

Figure 4.10: **Forward vs. reverse KL objective.** The forward KL variant can miss target modes not covered by the anchor prompt dataset (e.g., famous Van Gogh paintings that don't need mention of Van Gogh to reproduce the style). The reverse KL variant, by taking expectations over target concept images, naturally covers all modes and achieves more complete ablation in such cases.

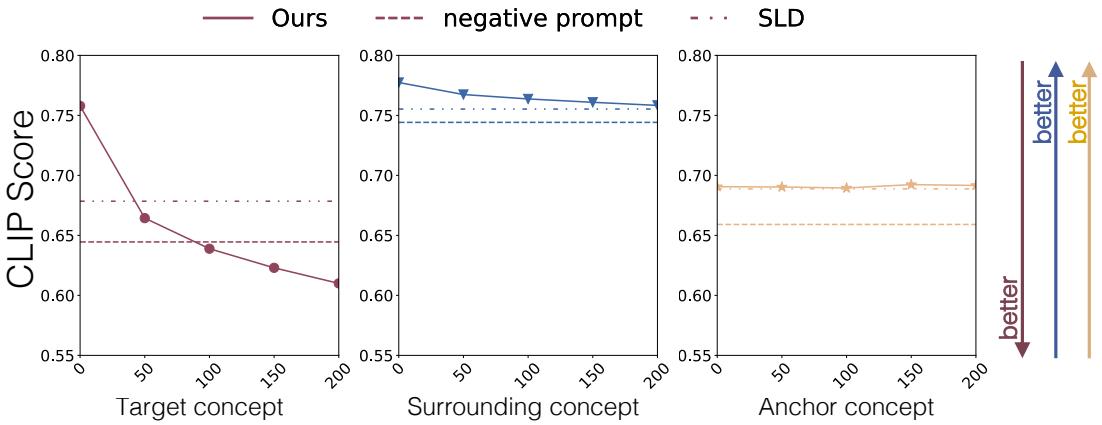


Figure 4.11: **Instance ablation comparison with Negative prompt and Safe Latent Diffusion (SLD).** Our method ablates better while preserving anchor and surrounding concept distribution. We used the diffusers implementation for both with the same hyperparameters as recommended in the paper [230].

and Safe Latent Diffusion (SLD) [230], we set the target concept as the negative prompt or safety concept, respectively.

4.5 Discussion and Limitations

While our method efficiently ablates a wide range of object instances, styles, and memorized images, several limitations remain. First, ablating a target concept can cause slight degradation on closely related surrounding concepts, particularly when

Figure 4.12: **Qualitative comparison with Negative prompt and Safe Latent Diffusion (SLD)**. Our method preserves the surrounding concept better compared to the baseline methods of negative prompt and SLD. We used the diffusers implementation for both with the same hyperparameters as recommended in the paper for SLD-Medium [230].

they share significant visual overlap (e.g., Monet and Van Gogh, both impressionist styles). Second, our method does not prevent a downstream user with full access to model weights from re-introducing the ablated concept via fine-tuning [71, 130, 219]. Even without weight access, adversarial prompt optimization could potentially recover the target concept. Though this is considerably harder, our work does not provide guarantees against such attacks.

Despite these limitations, we believe creators should have the right to opt out of generative models that reproduce their work. This chapter takes a step toward that goal by providing an efficient computational tool for removing copyrighted content and artistic styles from text-to-image diffusion models.

Part II

Learning from Large Synthetic Datasets

Chapter 5

Scaling Customization via Synthetic Training Data

The optimization-based method introduced in previous chapters requires per-concept fine-tuning that takes several minutes. While effective, this per-instance optimization or fine-tuning can limit practical deployment in many scenarios. In this chapter, we address this bottleneck by training an encoder-based model that performs feed-forward customization, generating personalized outputs in a single forward pass given reference images. However, existing encoder-based approaches for this typically train on the text-image dataset itself with the same image shown as reference, owing to the scarcity of naturally occurring multi-image object collections, and thus lack the multi-view supervision needed to capture fine-grained object details.

We address these challenges in this chapter and leverage text-to-image models and 3D datasets to construct a *Synthetic Customization Dataset (SynCD)*. This dataset consists of multiple images per object under varied lighting, backgrounds, and poses. Using this data, we train an encoder-based model that incorporates fine-grained visual details from reference images via a shared attention mechanism. We also propose a guidance normalization technique at inference to mitigate overexposure artifacts. Extensive experiments show that our approach outperforms existing encoder-based methods on standard customization benchmarks.

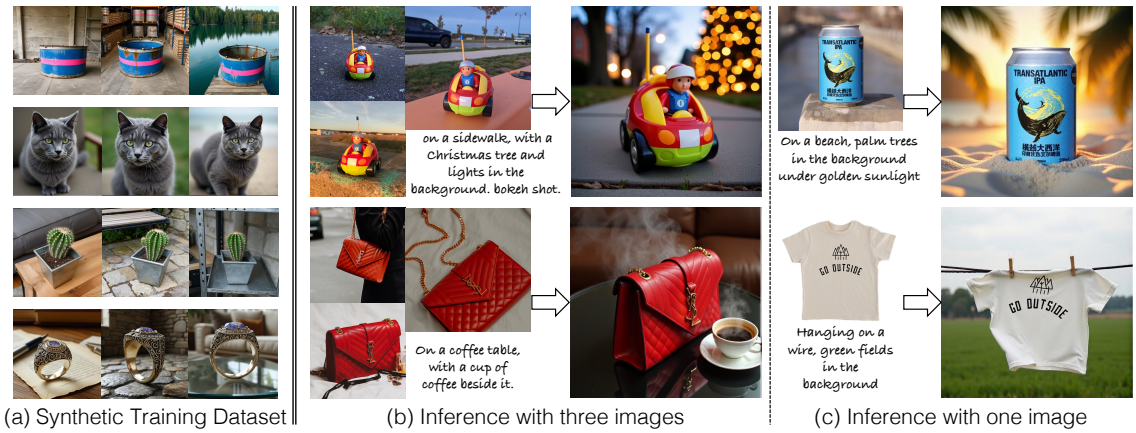


Figure 5.1: **SynCD.** We propose a pipeline for Synthetic Customization Data consisting of multiple images of the same object under different lighting, poses, and backgrounds. Given the dataset, we train an encoder-based *feed-forward* customization model, which can take either (b) three or (c) one reference image of the object as input and successfully generate it in new compositions using text prompts.

5.1 Introduction

Text-to-image models are capable of generating high-fidelity and realistic images given only a text prompt [65, 199, 217, 223]. Yet, text often falls short of describing rich visual details of real-world objects, such as the unique toy in Figure 5.1. What if the user wishes to generate images of this toy in new scenarios? This has given rise to the emerging field of model customization or personalization [42, 71, 130, 219, 278], allowing us to generate new compositions of the object via text prompts, e.g., the toy on a sidewalk with a different background, as shown in Figure 5.1. Early *optimization-based* works [71, 282] for the task, including our method Custom Diffusion [130] (Chapter 3), require many fine-tuning steps on user-provided images of every new object — a process both costly and slow. To address this, several *encoder-based* methods [42, 143, 247, 278, 294] learn an image encoder model with reference images as an additional conditional input. Thus, during inference, these methods can generate new compositions of the reference object in a single forward pass without expensive per-object optimization.

However, a key bottleneck in developing encoder-based methods is the scarcity of large-scale datasets with multiple images of the same object under diverse poses,

backgrounds, and lighting conditions. Collecting such data at scale is difficult, as web images rarely carry object-identity annotations. We address this by proposing a pipeline to construct such a dataset synthetically using text-to-image models and 3D assets. The core challenge here is preserving object identity across the set of images being generated while varying the context. We achieve this by sharing attention among the foreground object regions during parallel image generation, which promotes visual consistency across the set. For rigid objects, we additionally leverage 3D priors from Objaverse [51] via depth guidance and cross-view feature correspondence to enforce multi-view consistency. A final filtering stage removes low-quality or identity-inconsistent samples.

Using our *Synthetic Customization Dataset (SynCD)*, we train an encoder-based model for tuning-free customization and propose an inference method for higher fidelity generations. We condition generation on fine-grained reference image features via cross-image shared attention, improving object identity preservation. For inference, we propose a guidance normalization technique to mitigate overexposure artifacts. Through extensive experiments, we show that our approach outperforms state-of-the-art encoder-based customization methods, including JeDi [311], Emu-2 [252], and IP-Adapter [294]. Our code and data are available on our project website.

5.2 Related Work

Text-to-image models and image conditioning. Text-to-image diffusion and flow-based models [65, 95, 134, 164, 217] have enabled photorealistic image synthesis at scale. However, text alone is often an imprecise signal, motivating richer conditioning modalities, including depth and segmentation maps [12, 318], spatial layouts [145, 201], and context images [43, 182]. In this work, we focus on a specific instance of context-image conditioning, object customization, where reference images of a new object serve as the conditioning signal for personalized generation.

Customization: from optimization-based to encoder-based methods. Early methods for customization fine-tuned model parameters [88, 96, 130] or text embeddings [71, 268] on a small set of user-provided reference images (see Chapter 3). While these optimization-based approaches yield high-quality results, per-concept

fine-tuning can take several minutes per instance, a bottleneck for practical deployment at scale. This has driven the development of encoder-based methods that amortize this cost through training and during inference, given reference images, generate new compositions in a single forward pass without any test-time optimization.

Encoder-based methods for customization add an image conditioning pathway to the text-to-image model. One common design encodes reference images into compact visual tokens using pretrained feature extractors [44, 143, 197, 247, 278, 283], which are then mapped to the text embedding space. Other approaches learn mappers from multimodal autoregressive models [263] and generative models to incorporate reference images as visual prompts [192, 252]. Another commonly adopted design is the decoupled text and image cross-attention [170, 278, 294]. Our method draws on this but injects reference features at finer granularity via shared self-attention.

Most existing encoder-based methods train on either single-image datasets, with the same image as both reference and target, requiring compact feature bottlenecks to prevent trivial copying at the cost of identity preservation, or multi-view datasets with limited background diversity, causing models to overfit to reference backgrounds [143, 247]. To address this, we instead construct a synthetic dataset with multiple images of each object across diverse backgrounds and poses. Our pipeline draws on ideas from consistent character generation [261, 332] and multi-view synthesis [54, 156, 237], adapted for the customization task. While JeDi [311] and concurrent work OminiControl [257] also construct synthetic dataset for this purpose, they rely on text prompting alone for object consistency. In contrast, our pipeline enforces explicit constraints via 3D asset guidance, yielding higher-quality training data.

5.3 SynCD: Synthetic Customization Dataset

Training an encoder-based customization model requires a diverse dataset of unique objects, each with multiple images in different contexts. To address the data scarcity and collection challenge, we introduce a pipeline for synthesizing diverse, high-quality image corpora. The pipeline consists of (1) creating K prompts per object, (2) generating a *set* of K images with a consistent object given the prompts, and (3) filtering to remove low-quality and inconsistent images. We cover a large diversity of 75,000 rigid-category assets from Objaverse [51] and 16 deformable categories of animals

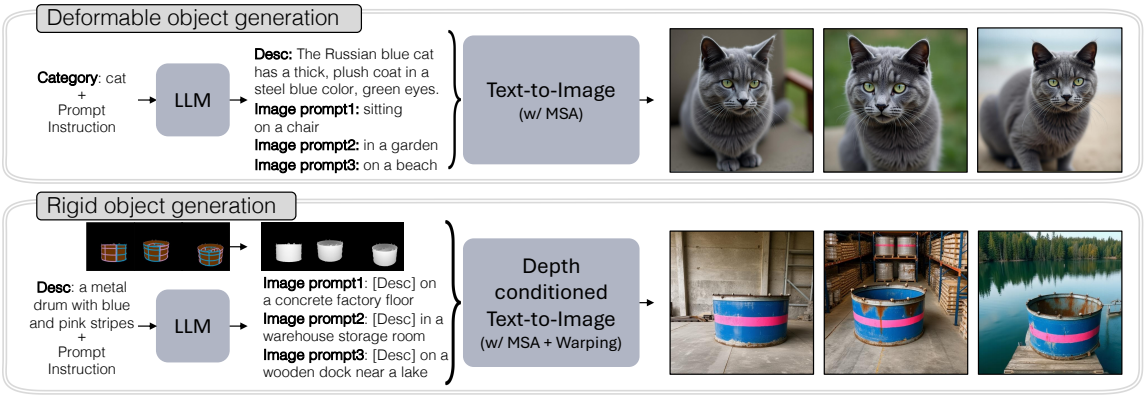


Figure 5.2: **Synthetic Customization Data pipeline.** *Top:* For deformable objects (e.g., cats), LLM-suggested object and background descriptions are used to generate multiple images of a consistent subject. *Bottom:* For rigid objects, we additionally condition on depth maps rendered from 3D assets [51] with captions from Cap3D [168]. Both pipelines use Masked Shared Attention (MSA) and warping (in the case of rigid objects) to promote object consistency, as shown in Figure 5.3.

(with ∼100 sub-categories). We explain each step of our pipeline in detail below.

5.3.1 LLM-assisted Prompt Generation

Each prompt pairs a detailed object description with a background scene description. Having detailed object descriptions helps in better identity consistency across the generated set. For rigid objects from Objaverse, we source descriptions from Cap3D [168], e.g., *a large metal drum with blue and pink stripes*. For deformable objects such as animals, which Objaverse does not cover, we instead use an LLM to first generate captions capturing distinguishing physical features, e.g., *The Russian blue cat has a thick plush coat*. We then prompt the same LLM to generate multiple diverse background scene descriptions for each object. We construct the final prompts by combining the object description with each background description and pass them to the image generation step, as shown in Figure 5.2.

5.3.2 Multi-image Consistent-object Generation

Given the prompts, we generate images with a consistent object using the DiT-based [199] FLUX model [134]. It consists of a series of MMDiT [65] blocks to pro-

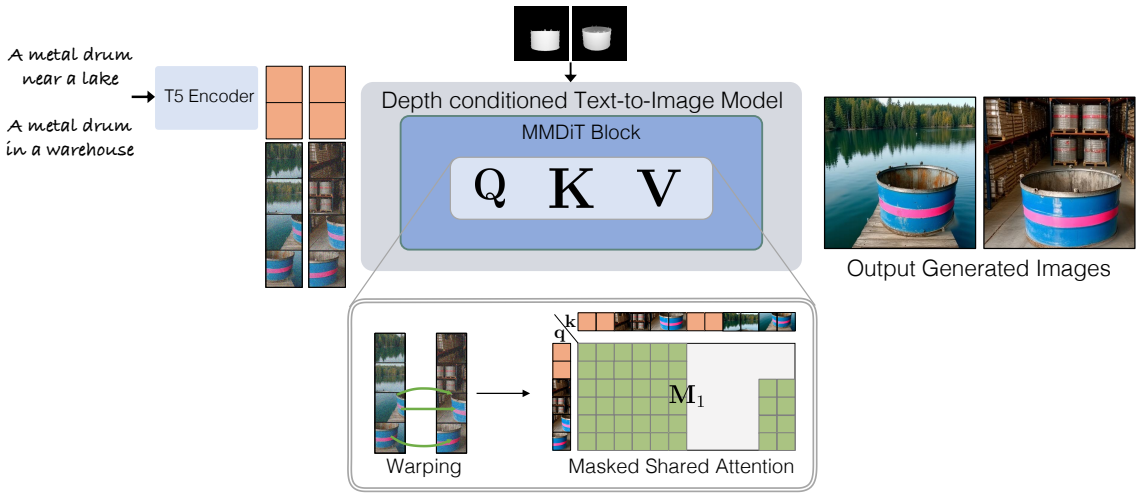


Figure 5.3: **Feature warping and Masked Shared Attention (MSA) for object consistency.** For rigid objects, we first warp corresponding features from the first image to the other. Then, each image feature attends to itself, and the foreground object features in other images. We show an example mask, $\mathbf{M}_1$, used to ensure this for the first image when generating two images with the same object.

gressively denoise latent representations [217]. To enforce object consistency, we share the internal features across the images during the denoising process via a Masked Shared Attention (MSA) mechanism [156, 261]. For rigid objects, we additionally leverage depth maps and multi-view correspondence from Objaverse 3D assets to enforce geometric consistency.

Masked Shared Attention (MSA). To promote visual consistency of the object across generated images, we modify the attention blocks so that each image attends to its own features as well as the foreground object regions of the other images. Concretely, when generating K images of an object in parallel with different prompts, given query, key, and value features $\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i \in \mathbb{R}^{n \times d'}$ inside a particular attention layer for the i^{th} image, MSA computes:

$$\begin{aligned} &\text{MSA}(\{\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i\}_{i=1}^{K}) \equiv \\ &\left\{\text{Softmax}\left(\frac{\mathbf{q}_i[\mathbf{k}_1 \cdots \mathbf{k}_N]^T}{\sqrt{d'}} + \mathbf{M}_i\right)[\mathbf{v}_1 \cdots \mathbf{v}_N]\right\}_{i=1}^{K}, \end{aligned} \tag{5.1}$$

where n is the sequence length and d' the feature dimension. Each $\mathbf{q}_i$ attends over all $K \times n$ features, and the attention bias matrix $\mathbf{M}_i \in \mathbb{R}^{n \times (Kn)}$ masks out

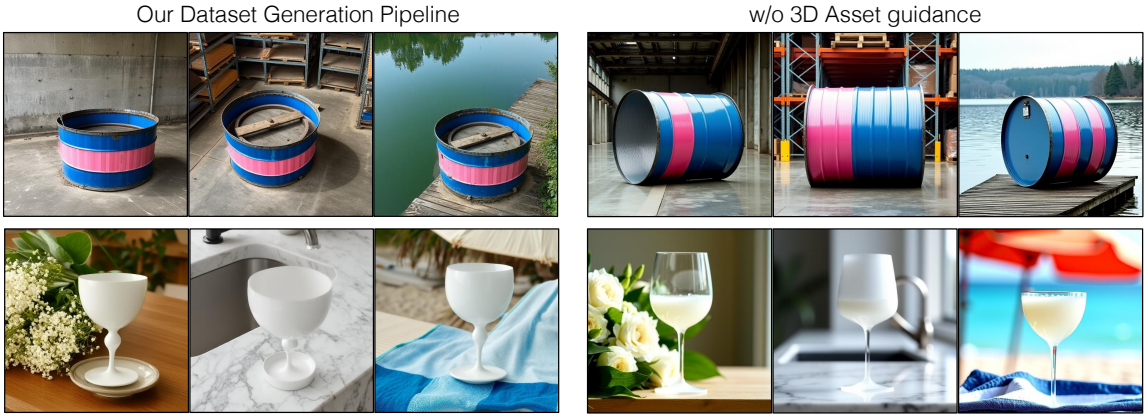


Figure 5.4: **Rigid object generation w/ vs. w/o 3D asset guidance.** We compare our final rigid object generation results with that of removing 3D asset guidance and only using Masked Shared Attention (MSA). Removing depth guidance and warping from the generation pipeline reduces shape and multi-view consistency.

the background regions of other images, restricting cross-image attention to object regions. Since the DiT model uses joint text-image attention, $\mathbf{M}_i$ is initialized such that text tokens of one image do not attend to other images' tokens (Figure 5.3). We additionally modify the Rotary Positional Embeddings (RoPE) [251] to correspond to $KH \times W$ resolution when generating K images of size $H \times W$.

While MSA encourages generation of objects with similar visual features, it does not explicitly enforce 3D geometric consistency, as qualitatively shown in Figure 5.4. For rigid objects, we therefore incorporate additional constraints from Objaverse 3D assets, as described next.

Rigid object generation with MSA and 3D consistency. For rigid objects selected from Objaverse, we render them from K camera poses and feed the rendered depth maps along with captions from Section 5.3.1 to a depth-conditioned FLUX model [135]. Masked Shared Attention (MSA) is applied during denoising using ground-truth object masks derived from the depth maps. While depth guidance ensures 3D shape consistency and MSA encourages visual appearance similarity, we further enhance multi-view consistency by warping corresponding features across views. For a representative example of generating two images with the same object, given latent features $f_i \in \mathbb{R}^{(h \times w) \times d}$, $i \in \{1, 2\}$, the warped feature is computed as:

$$\hat{f}_2(u,v) = \alpha f_1(u+\Delta u, v+\Delta v) + (1-\alpha) f_2(u,v), \tag{5.2}$$

where $(u+\Delta u, v+\Delta v)$ is the corresponding location of pixel (u,v) in the first image, α is a binary visibility indicator for whether the location is visible in the first image, and $f_1(u+\Delta u, v+\Delta v)$ is the bilinearly interpolated feature at that location. Figure 5.3 shows an illustrative example. We apply warping for all image pairs, with appropriate masks, and only in the early diffusion steps. This enhances multi-view consistency while avoiding warping artifacts and allowing natural lighting variations.

5.3.3 Dataset Filtering

After generation, we apply a two-stage filtering pipeline to remove low-quality and identity-inconsistent images. First, we discard images with an aesthetic score [7] below 6. Second, we use DINOv2 [191] features to compute pairwise similarity within each set, removing images whose average pairwise similarity falls below 0.7. Our final dataset contains ∼95,000 objects with 2-3 images per object, uniformly distributed across rigid and deformable categories. Full pipeline details are in Appendix C.1.1.

Discussion. The key insight underlying our approach is that data generation affords privileged access to internal model features and ground-truth 3D geometry, signals unavailable both in naturally occurring image collections and at inference time. This makes large-scale synthetic construction tractable by leveraging these privileged signals to enforce object identity constraints.

5.4 Training Method

As mentioned before, the goal for customization is to learn $p(\mathbf{x}|\mathbf{c}, \{\mathbf{x}_i\}_{i=1}^N)$, given N reference images $\{\mathbf{x}_i\}_{i=1}^N$ of an object, to generate images aligned with both the text prompt $\mathbf{c}$ and the object identity depicted in the references. To achieve this, we fine-tune an existing text-to-image model on our dataset. For each object with K images in the dataset, one image serves as the generation target and the remaining $K-1$ as references. To condition the model on the reference images, we repurpose the Shared Attention mechanism from our data generation pipeline, as detailed below.

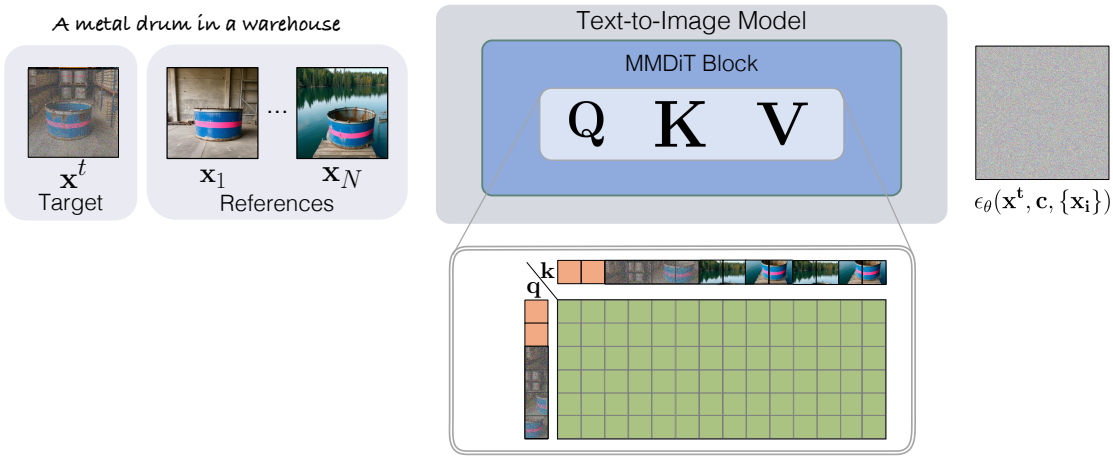


Figure 5.5: **Training Method.** We condition the model on reference images, $\{\mathbf{x}_i\}_{i=1}^N$, using a Shared Attention mechanism, similar to Figure 5.3. We extract fine-grained features of the reference images using the same model and have the target image features attend to the reference image features as well in the attention blocks.

Reference image conditioning. Recall from Section 2.2 that diffusion models learn to denoise $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ back to $\mathbf{x}$, conditioned on text $\mathbf{c}$. Here, we additionally condition on reference images $\{\mathbf{x}_i\}_{i=1}^N$. To inject reference information, we concatenate the reference image features with the target image features along the sequence dimension in each attention block. The target image's query features then attend to both its own and the reference images' features, enabling fine-grained identity transfer, as shown in Figure 5.5.

For the training loss, we adopt the velocity [224] or flow prediction [150] objective for diffusion and flow-based models, respectively:

$$\mathbb{E}_{\mathbf{x}_t, t, \mathbf{c}, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left\| \mathbf{v} - \mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{c}, \{\mathbf{x}_i\}_{i=1}^N) \right\|^2, \tag{5.3}$$

where $\mathbf{v}_\theta$ is the predicted velocity/flow, $\mathbf{v} \equiv \alpha_t \epsilon - \sigma_t \mathbf{x}$ for diffusion models and $\epsilon - \mathbf{x}$ for flow models, and α_t, σ_t determine the noising ratio at timestep t.

Inference. For final inference, we combine the classifier-free text and image guidance at every denoising step. However, directly combining them using previous work [26] often leads to overexposure issues in the generated image, especially at high image guidance, as shown in Figure 5.9. To mitigate this, we propose normalizing image and text guidance vectors. This helps us achieve better image alignment with the reference object while still following the text prompt. Our final inference is

$$\begin{aligned} &\mathbf{v}_\theta(\mathbf{x}_t, \{\mathbf{x}_i\}_{i=1}^N, \varnothing) + s_I \frac{\|g\|}{\|g_I\|} \cdot g_I + s_\mathbf{c} \frac{\|g\|}{\|g_c\|} \cdot g_\mathbf{c}, \\ \text{where } & g_I = \mathbf{v}_\theta(\mathbf{x}_t, \{\mathbf{x}_i\}_{i=1}^N, \varnothing) - \mathbf{v}_\theta(\mathbf{x}_t, \varnothing, \varnothing), \\ & g_\mathbf{c} = \mathbf{v}_\theta(\mathbf{x}_t, \{\mathbf{x}_i\}_{i=1}^N, \mathbf{c}) - \mathbf{v}_\theta(\mathbf{x}_t, \{\mathbf{x}_i\}_{i=1}^N, \varnothing), \\ & \|g\| = \min(\|g_I\|, \|g_\mathbf{c}\|), \end{aligned} \tag{5.4}$$

where t is the denoising timestep, $\mathbf{v}_\theta$ is the model output, g_I and g_c are the image and text guidance vectors, and s_I and $s_\mathbf{c}$ represent the guidance strength for the image and text. We scale the norm of the two guidance vectors to the minimum norm, allowing only s_I and $s_\mathbf{c}$ to vary the relative strength of the image and text guidance. During inference, the number of reference images can vary from training since attention-based conditioning is agnostic to sequence length.

5.5 Experiments

Training details. For a fair comparison with different baselines, we fine-tune three model variants on SynCD with the same reference image conditioning: FLUX [134] (Ours-12B), and two U-Net-based latent diffusion models, SD-1.5 [217] (Ours-1B), and SDXL [203] (Ours-3B). For FLUX, we fine-tune only the attention layers using LoRA [96]. For the U-Net-based models, we initialize from IP-Adapter [294] weights and fine-tune LoRA layers in the self-attention blocks and key-value projection matrices in the image cross-attention layers. Training uses a variable number of reference images, either 1 or 2, depending on the number of images in each set after filtering. More training and hyperparameter details are provided in Appendix C.1.2.

Evaluation dataset. Consistent with prior works [192, 247, 311], we use the DreamBooth [219] dataset consisting of 30 objects with 25 evaluation text prompts.

Baselines. We compare our method with leading encoder-based customization baselines, which include JeDi [311], IP-Adapter [286, 294], Emu-2 [252], Kosmos [192], BLIP-Diffusion [143], and MoMA [247]. We also show a comparison with the concurrent work OminiControl [257]. More details are provided in Appendix C.1.3.

Evaluation metric. As mentioned in previous chapters as well, a successful customization requires balancing two objectives: faithful adherence to the input text prompt while preserving visual alignment with the reference object. Following the

Figure 5.6: **Single-reference image results.** Comparison with encoder-based baselines given one reference image. Our method preserves object identity on par or better than baselines while following the text prompt. Best of 4 shown for all methods.

evaluation protocols established in our Custom Diffusion and Custom Diffusion-360 studies, we measure text-to-prompt alignment using CLIPScore [208]. We additionally employ TIFA [97], a more recent metric based on vision-language models, for complementary text-image evaluation.

For image alignment, we follow recent works [247, 311] and extend the DINOv2-based metric from Custom Diffusion-360 with background masking. We compute feature similarity between object-centered crops in the DINOv2 [191] feature space after masking out background regions, and denote this metric as MDINOv2-I. The masks are obtained using pretrained detection and segmentation models [120, 214, 331]. This focuses the similarity computation on the foreground object, reducing sensitivity to background and yielding a stricter measure of object identity similarity.

Given the inherent trade-off between text and image alignment, we combine TIFA and MDINOv2-I through their geometric mean, termed Geometric score [289]. This unified metric has been shown to better align with human preferences [289]. We also validate our findings through human preference studies. Detailed metric computation procedures are provided in Appendix C.1.4.

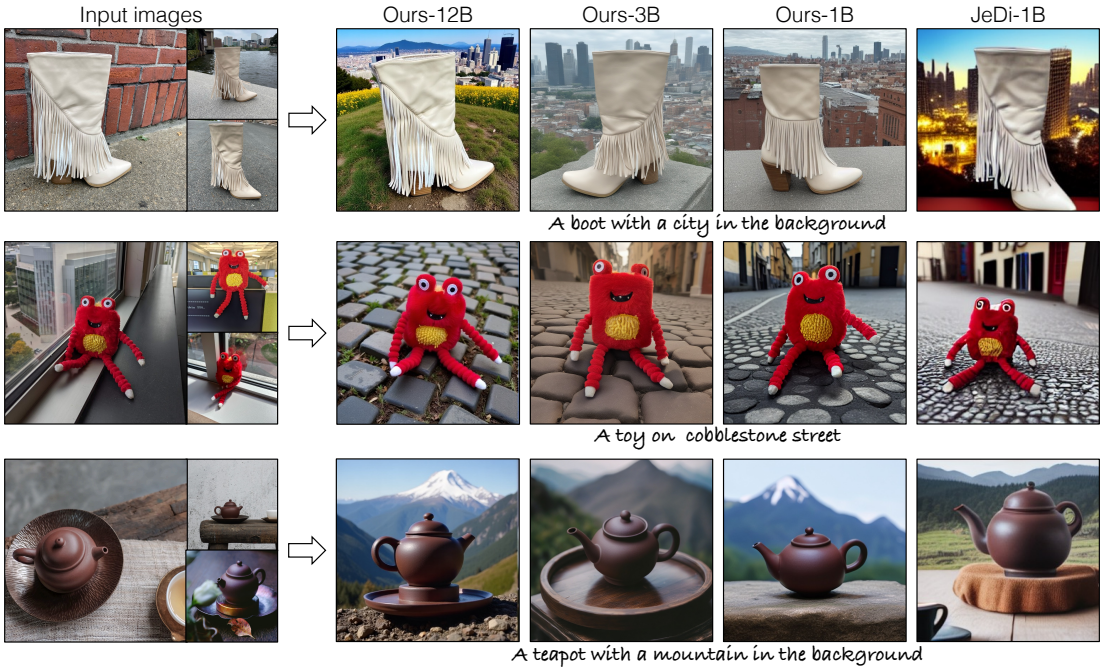


Figure 5.7: **Multi-reference image results (3 references).** Comparison with JeDi [311], which supports multiple reference images. JeDi preserves object identity but often produces incoherent backgrounds and lighting. Our method maintains higher image fidelity while following image and text conditions. Best of 4 shown.

Method	MDINOv2-I↑		CLIPScore ↑	TIFA ↑	GeometricScore ↑
	Background change prompt	Property change prompt			
Kosmos [192]	0.636	0.638	0.287	0.729	0.679
BLIP-Diffusion [143]	0.658	0.643	0.294	0.782	0.714
MoMA [247]	0.616	0.620	0.320	0.867	0.730
IP-Adapter [294]	0.718	0.702	0.283	0.701	0.704
IP-Adapter Plus [294]	0.744	0.737	0.270	0.615	0.675
Emu-2 [252]	0.750	0.736	0.283	0.741	0.740
JeDi [311]	0.771	0.775	0.292	0.789	0.780
Ours-1B	0.806	0.773	0.303	0.830	0.801
Ours-3B	**0.822**	**0.789**	**0.313**	**0.863**	**0.838**
IP-Adapter (12B) [286]	0.563	0.549	0.294	**0.815**	0.639
OminiControl (12B) [257]	0.650	0.527	0.302	0.808	0.685
Ours-12B	**0.778**	**0.771**	**0.306**	0.786	**0.780**

Table 5.1: **Quantitative comparison**. We compare our method against other encoder-based methods on image alignment and text alignment metrics. Our method performs better or on par with other baselines on the combined GeometricScore metric, even when compared across different model scales. For reference, the all-pairwise MDINOv2 similarity between reference images themselves is 0.851.

5.5.1 Results

5.5.1.1 Qualitative Comparison

We show sample comparisons of our method against other encoder-based methods in Figure 5.6 and Figure 5.7. Our method more effectively incorporates the text prompt while maintaining the object identity and image fidelity, e.g., the cat in the firefighter outfit in 3rd row of Figure 5.6. In contrast, baseline methods can struggle with incorporating the text prompt or have low object identity preservation. With 3 reference images as input in Figure 5.7, although JeDi [311] achieves high identity preservation, it can have lower image quality, with inconsistent lighting and background. Additional qualitative samples are shown in Figure 5.13.

5.5.1.2 Quantitative Comparison

Automatic scores. We compare our method (with three input reference images) against encoder-based baselines in Table 5.1. We evaluate MDINOv2-I on two distinct prompt categories: background change prompts and object appearance or property change prompts, e.g., *cube-shaped or wearing sunglasses*. The latter category typically yields lower image similarity due to greater visual divergence from the reference images. Our method achieves comparable or superior Geometric Scores relative to baselines. While Ours-3B achieves a higher DINOv2-I score, Ours-12B generates higher-fidelity outputs with greater viewpoint and background diversity (Figure 5.7).

While quantitative metrics capture image and text alignment, they can struggle with overall quality and may favor methods that copy-paste the target object on a new background. We therefore conduct a pairwise human study next.

Human evaluation. We conduct pairwise comparisons where participants evaluate generated images from our method and baselines alongside the text prompt and three reference images. Evaluators select the preferred image based on three dimensions: (1) object identity preservation, (2) text prompt adherence, and (3) overall visual quality and photorealism. They also indicate the specific criterion or criteria for their selection. Table 5.2 shows the results compared to the three competing methods from Table 5.1, i.e., Emu-2 [252], JeDi [311], and OminiControl [257]. Our method is preferred over the baselines according to all evaluation criteria, showing

Method	Human preference (in %)↑			
	Text alignment	Image alignment	Photo-realism	Overall preference
Ours-1B vs JeDi	69.51	63.05	80.89	68.19
Ours-3B vs Emu-2	70.49	66.88	64.66	66.74
Ours-12B vs OminiControl	56.27	58.30	54.47	58.02

Table 5.2: **Human preference** for our method against the competing methods from Table 5.1, i.e., Emu-2 [252], JeDi [311], and OminiControl [257], while keeping the same model scale. The standard error for all is within ±5%.

	Method	MDINOv2-I↑		TIFA ↑	Geometric ↑ Score
		Background change prompt	Property change prompt		
1-input	IP-Adapter Plus [294]	0.618	0.626	0.569	0.595
	Emu-2 [252]	0.604	0.619	0.701	0.655
	Ours-3B	0.645	0.609	**0.809**	**0.712**
3-input	Ours-3B	**0.689**	**0.666**	0.749	**0.712**

Table 5.3: **Results on CustomConcept101 [130].** Our method outperforms both Emu-2 [252] and IP-Adapter [294] on the Geometric Score [289] metric, computed as the geometric mean of MDINOv2-I and TIFA.

the effectiveness of our synthetic dataset. To ensure valid responses, participants complete a practice test, and only those with correct responses are considered. We gather more than 300 valid responses per comparison and provide further details regarding the study in Appendix C.1.4.

CustomConcept101 Benchmark. CustomConcept101 [130] is a more diverse benchmark with 101 unique concepts. We compare Ours-3B with open-source baselines of similar scale, i.e., Emu-2 [252] and IP-Adapter [294], on this dataset. As shown in Table 5.3, our method outperforms both baselines on the overall Geometric Score while achieving better text alignment. We also show qualitative samples on this dataset using Ours-12B model in Figure 5.14.

5.5.2 Ablation

Model training. We conduct ablation studies by fine-tuning the baseline IP-Adapter Plus model (Ours-3B variant). Table 5.4 demonstrates the importance of each of our proposed components: fine-tuning with our synthetic dataset alone (row

Figure 5.8: **Qualitative results of ablation study.** From left to right: (1) Vanilla IP-Adapter Plus baseline, (2) Our modified inference improves text alignment, (3) Fine-tuning on SynCD dataset further improves text adherence, (4) Masked Shared Attention conditioning recovers object identity while maintaining text alignment.

3) improves performance over the baseline, establishing the quality and utility of our data generation pipeline. Adding reference image conditioning via Masked Shared Attention (row 4) provides a substantial performance boost, demonstrating the effectiveness of conditioning on fine-grained reference features via shared attention. Notably, adding multiple reference images during inference (row 5) improves results substantially. Figure 5.8 shows qualitative samples in line with the quantitative results. For example, MSA with multiple reference images as input enables the model to capture fine-grained color pattern of the shoe (last row) that is otherwise missed.

Guidance normalization during inference. Our inference approach (Eqn. 5.4) extends guidance rescale [148], a technique developed to mitigate saturation artifacts in standard text-to-image generation. Figure 5.9 demonstrates the effectiveness of our guidance normalization: as image guidance scale increases from 1 to 5, image alignment improves while visual fidelity is maintained without saturation. More im-

	Method	MDINOv2-I↑ Background change prompt	MDINOv2-I↑ Property change prompt	TIFA ↑	Geometric ↑ Score
1-input	IPAdapter Plus	0.744	0.737	0.615	0.675
	+ our inference	0.719	0.668	0.816	0.756
	+ SynCD	0.766	0.695	0.901	0.819
	+ MSA (Ours-3B)	0.777	0.708	0.902	0.825
3-input	+ MSA (Ours-3B)	**0.822**	**0.789**	**0.863**	**0.838**

Table 5.4: **Quantitative results of ablation study**. We add different components of our method- modified inference, our dataset, and shared attention to the baseline IP-Adapter Plus method - and show a gradual increase in performance. MSA also enables the effective use of multiple reference images, thus significantly helping with image alignment as we increase the number of reference images to three.

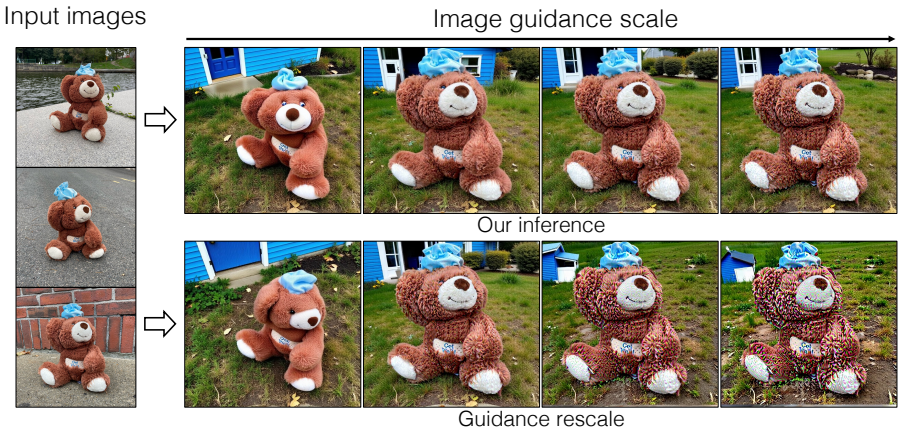


Figure 5.9: **Inference with varying guidance strengths (1–5).** Our normalization maintains text alignment and image fidelity without saturation artifacts across increasing guidance levels. Zoom in for details.

portantly, our inference makes the guidance effect more consistent across different random seeds and text prompts compared to baseline guidance technique.

Dataset curation. We ablate different steps of the dataset generation pipeline to analyze their respective importance. To measure object identity alignment, we compute the average DINOv2 intra-cluster similarity across the dataset, with each cluster comprising the K images generated in parallel with the same object. Table 5.5 shows that MSA consistently improves DINOv2 similarity reflecting better identity consistency, while feature warping further enhances it for rigid object generation by promoting cross-view coherence.

	Method	DINOv2-I↑
Rigid categories	Ours	**0.598**
	w/o Warping	0.572
	w/o MSA and Warping	0.495
Deformable categories	Ours	**0.700**
	w/o MSA	0.626
	w/o Detailed description	0.564

Table 5.5: **Ablating dataset generation pipeline components.** MSA consistently improves feature consistency across generated objects. Multi-view warping further enhances results for rigid objects.

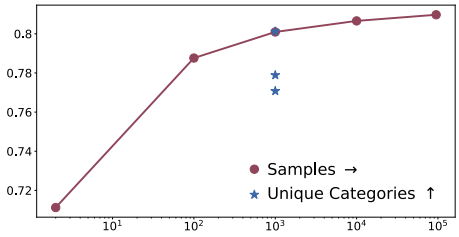


Figure 5.10: **Effect of category diversity and size.** At fixed sample size (1K), increasing diversity (3→16→200) progressively improves image alignment.

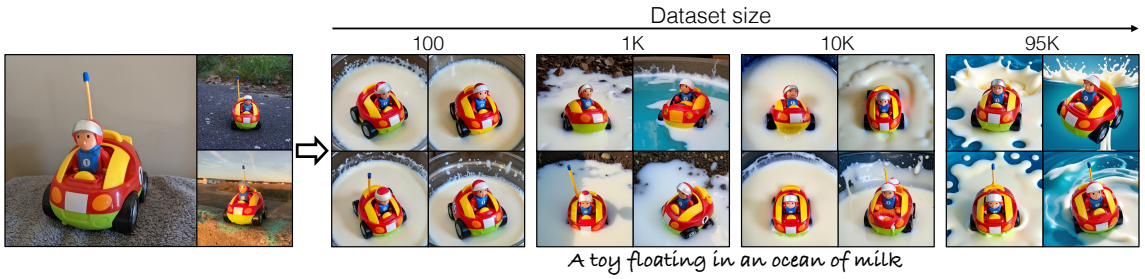


Figure 5.11: **Effect of dataset scale.** Training on progressively larger datasets with diverse categories (100, 1K, 10K, 95K samples) improves object identity preservation and enables diverse backgrounds and viewpoints.

Impact of category diversity on customization quality. Figure 5.10 shows that category diversity is the dominant factor in model performance: with a fixed sample size (1K objects), increasing diversity from 3 to 16 to 200 categories consistently improves image alignment. In contrast, increasing sample size beyond a certain threshold shows diminishing returns, with performance plateauing and similar results for 10K and 95K samples. Figure 5.11 provides qualitative evidence, showing that models trained on progressively larger datasets with diverse categories improve background and pose diversity while capturing finer object details. These findings demonstrate that category diversity is more critical than scale alone for achieving high-quality customization, though both factors remain important.

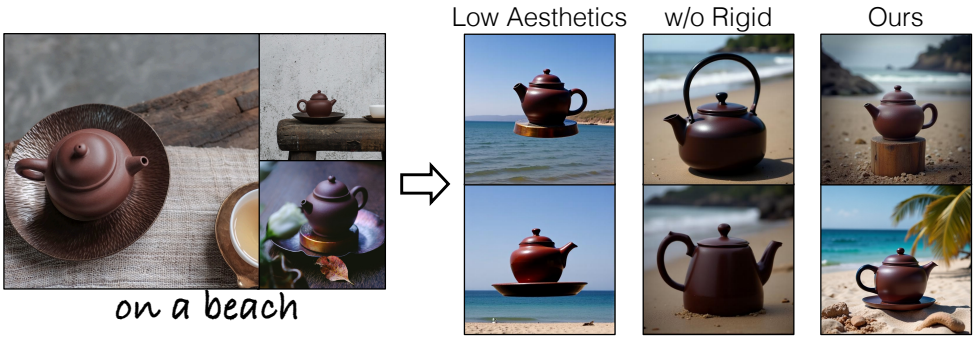


Figure 5.12: **Impact of low-quality or less diverse dataset.** Model performance when trained on low-quality or limited category subset is severely affected compared to our final filtered dataset. This highlights the importance of quality filtering and maintaining diversity when using synthetic dataset for model fine-tuning.

5.6 Discussion and Limitations

In this work, we focus on encoder-based model customization and propose advancements to address current limitations. To overcome the lack of training data, we introduce SynCD, a synthetic data pipeline that generates multiple identity-consistent images per object, and train an encoder-based model on this dataset. We further propose a guidance normalization technique at inference to mitigate overexposure artifacts. Together, these contributions enable tuning-free customization that outperforms existing encoder-based methods on standard benchmarks.

Despite these results, our approach has several limitations. First, synthetic data can introduce artifacts and distribution shifts [300]. As shown in Figure 5.12, using low-quality subsets or a limited range of deformable categories degrades performance, underscoring the importance of our quality filtering and diverse category coverage from Objaverse. Second, our pipeline focuses on single-object images; extending to multi-object or compositional scenes would broaden its applicability. Third, and most fundamentally, our synthetic data pipeline is coupled to the quality of the underlying generative models; as stronger text-to-image and text-to-3D models emerge, the pipeline will need to be rebuilt to remain competitive. This motivates the final part of this dissertation, which explores an orthogonal source of supervision from pretrained vision and vision-language models to train generative models.

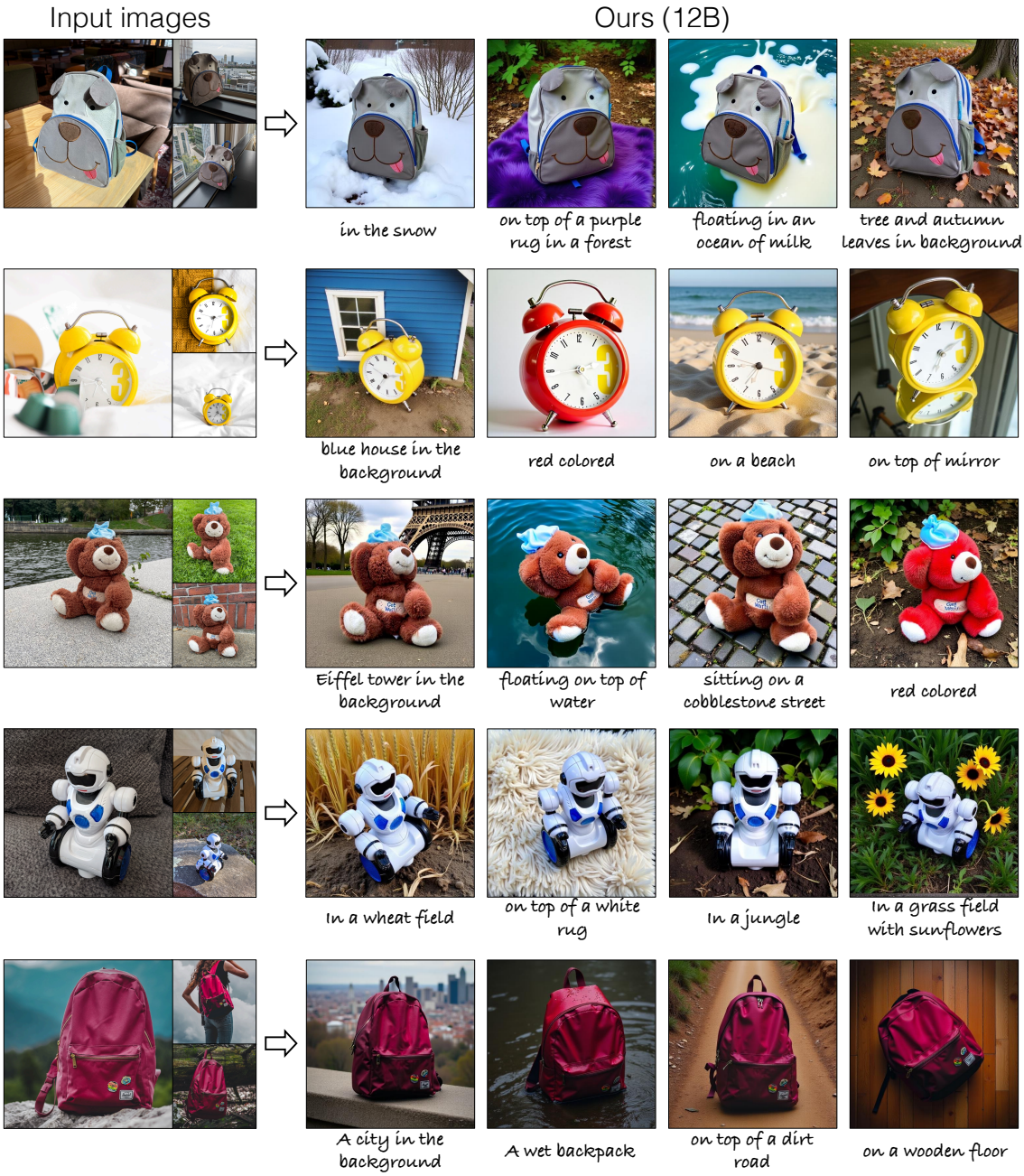


Figure 5.13: **Samples on DreamBooth [219] dataset with 3 input images.** Qualitative samples of our method given 3 reference images of the object.

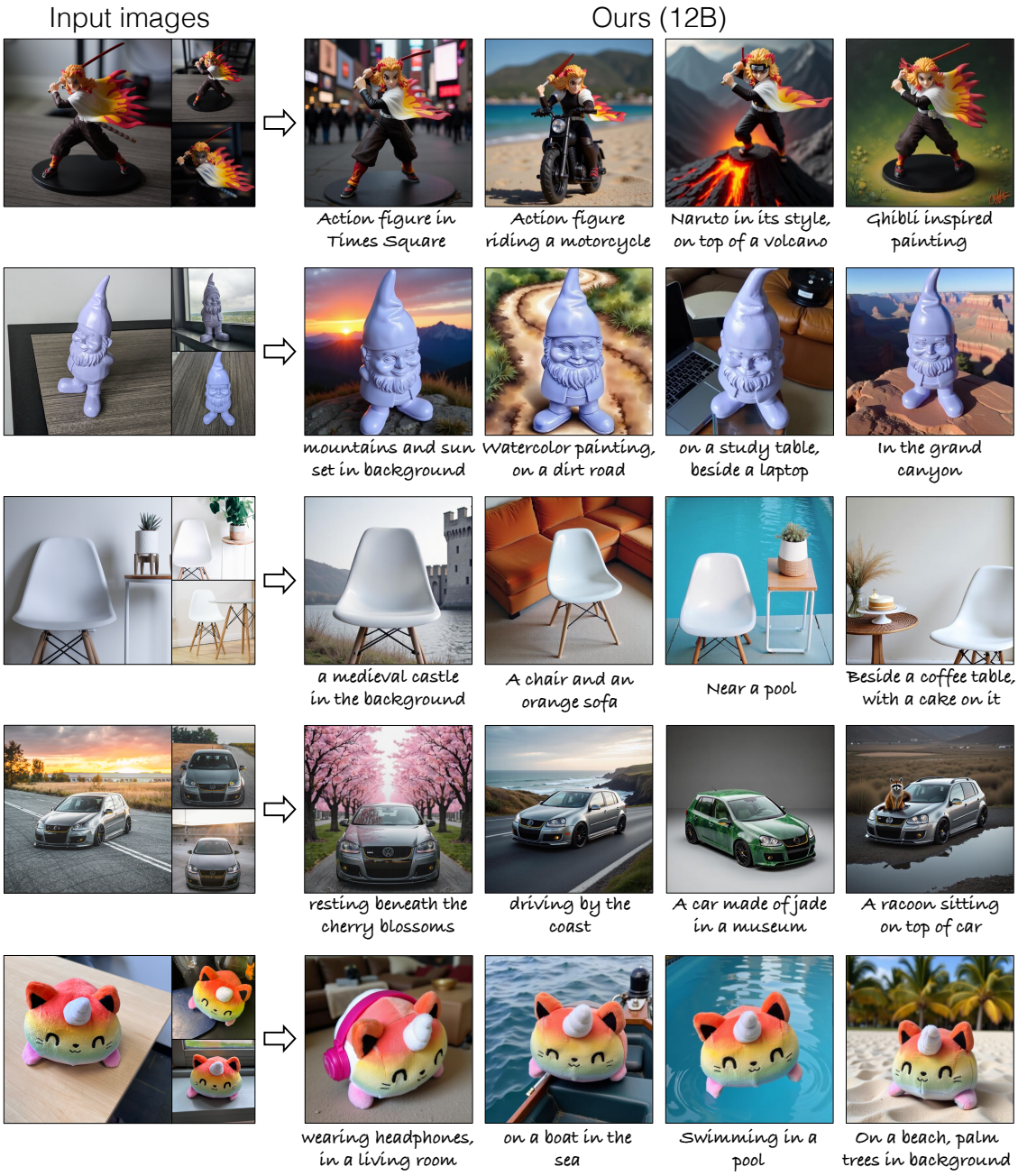


Figure 5.14: **Samples on CustomConcept101 [130] dataset with 3 input images.** Qualitative samples of our method given 3 reference images of the object.

Part III

Learning from Pretrained Vision Models

Chapter 6

GAN Customization using Pretrained Vision Encoders

Prior chapters explored two strategies for customizing text-to-image diffusion models. The first is parameter-efficient fine-tuning on a few reference samples, which is effective but time consuming, requiring optimization for every new instance. The second addressed this bottleneck by proposing a large-scale synthetic dataset generation pipeline, enabling feed-forward customization without the cost of real data collection. However, synthetic datasets inherit the artifacts and biases of the generative model used to create them, and need to be re-generated as stronger base models emerge.

In this chapter, we turn to a complementary source of supervision: pretrained vision models. Vision encoders trained on large-scale datasets via supervised and self-supervised objectives [40, 46, 89, 208, 240] have proven remarkably effective at capturing rich visual representations transferable across tasks [220, 303, 327]. Our key insight is that these representations could also provide rich feedback signals useful for generative modeling tasks.

We focus on Generative Adversarial Networks (GANs) as our primary case study and show that incorporating an ensemble of pretrained vision models as discriminators during GAN training significantly improves generation quality, both when customizing to new domains and when training from scratch. However, not all pretrained models contribute equally: the particular subset selected for the ensemble has a substantial impact on performance. We therefore propose an automatic se-

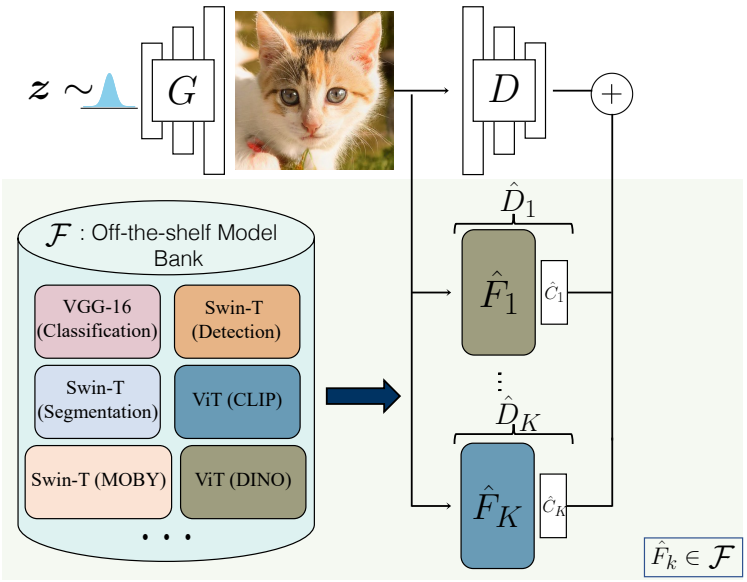


Figure 6.1: **Vision-aided GAN training**. The model bank $\mathcal{F}$ consists of widely used and state-of-the-art pretrained networks. We automatically select a subset $\{\hat{F}\}_{k=1}^{K}$ from $\mathcal{F}$, which can best distinguish between real and fake distributions. Our training procedure consists of creating an ensemble of the original discriminator D and discriminators $\hat{D}_k = \hat{C}_k \circ \hat{F}_k$ based on the feature space of selected off-the-shelf models. $\hat{C}_k$ is a shallow trainable network over the frozen pretrained features.

lection mechanism that ranks candidate models by probing the linear separability between real and fake distributions in their feature spaces, then progressively adding the top-ranked models into the discriminator ensemble.

6.1 Introduction

Analysis by synthesis is a classical principle in computational vision [308]: understanding perception by studying the inverse problem of image generation. Generative modeling requires capturing and representing the complex statistics of real-world visual phenomena, while vision models [40, 46, 89, 208, 240] trained for discriminative tasks focus on extracting task-specific features from the data. Surprisingly, despite the aforementioned connection between synthesis and analysis, generative adversarial networks (GANs) [25, 109, 111, 324] are typically trained without leveraging such pretrained vision models. With a plethora of useful models readily available in the research ecosystem, this represents a missed opportunity. Can the knowledge captured in pretrained visual representations benefit GAN training? Our work addresses this

question by investigating which models should be selected and how they can be most effectively integrated, with particular focus on customizing pretrained GANs to new target domains, and more broadly in data-limited training settings.

We study the use of a "bank" of pretrained deep feature extractors to improve GAN training, which consists of a discriminator that learns the relevant statistics differentiating real and generated samples, and a generator that aims to reduce the real vs. generated image quality gap. The core challenge is that naïvely using strong pretrained networks as discriminators can lead to overfitting with zero gradients to the generator, especially with limited training samples. We show that training only a small, lightweight classifier on top of the pretrained feature extractor, as shown in Figure 6.1, provides stable training when combined with the original learned discriminator. Furthermore, ensembling multiple pretrained models encourages the generator to match the real distribution across different, complementary feature spaces.

We propose an automatic model selection strategy that ranks pretrained vision models by probing the linear separability of real and fake distributions in their feature spaces, then progressively adds the top-ranked models to the discriminator ensemble. To stabilize training and mitigate overfitting, we additionally incorporate label smoothing [225] and differentiable augmentation [109, 324]. We demonstrate the effectiveness of our method in two settings. In the domain transfer setting, we show consistent improvements on AFHQ [49] and MetFaces [109]. In the limited-data setting, we achieve 2–3× FID improvements with 1k training samples on FFHQ [108] and LSUN [303], and match StyleGAN2's full-dataset FID using only 10k samples. Our code is available on our project website.

6.2 Related Work

Improving GAN training. Since the introduction of GANs [85], significant advances have been driven by architectural improvements [108, 111, 207], training schemes [107, 313], as well as objective functions [8, 9, 59, 64, 173, 177]. In previous works, the learning objectives often aim to minimize different types of divergences between real and fake distribution. The discriminators are typically trained from scratch and do not use external pretrained networks. As a result, the discriminator is prone to overfitting the training set, especially for the limited data setting [109, 290, 324].

Use of pretrained models in image synthesis. Pretrained models have been widely used as perceptual loss functions [62, 75, 103] for various synthesis tasks such as super-resolution [138], image-to-image translation [39, 194, 274], style transfer [75]. It has been shown that deep features can indeed better match human perceptual similarity than classic metrics [320], and perceptual discriminators have been explored with adversarial loss for image-to-image translation [215, 253].

Our work is inspired by this but differs in key ways. We focus on unconditional GAN training instead of image-to-image translation, and ensemble diverse pretrained models rather than a single one, and introduce automatic model selection for domain-specific effectiveness. Another concurrent work similar to ours is Project GAN [227] which uses random projections in perceptual discriminators [253] for faster GAN training without overfitting issues. Additionally, other works have used pretrained models for clustering and conditioning during generation [35, 172, 222, 238, 305]. Different from the above work, our method empowers the discriminator with pretrained models and requires no changes to the backbone generator.

Use of pretrained models in image editing. Pretrained models have been instrumental for various image editing applications both with GANs and more recently text-to-image diffusion models. Notable examples include image projection with a perceptual distance [1, 333], text-guided editing with CLIP [10, 116, 196, 198], finding editable directions using attribute classifier models [236], and using segmentation networks for identifying image editing regions [335]. However, these approaches use pretrained models only at inference time or as fixed perceptual losses. In this chapter, we instead focus on using pretrained vision models as additional supervision signal for improved training of GANs and show that this principle can be extended to text-to-image diffusion models as well in the next chapter.

Transfer learning. Large-scale supervised and self-supervised models learn useful feature representations [34, 40, 89, 125, 208, 285] that can transfer well to unseen tasks, datasets, and domains [60, 99, 124, 190, 221, 266, 301, 309, 310]. In generative modeling, transfer learning can include modifying the weights of pretrained generators and discriminators from a source domain (e.g., faces) to a new domain (e.g., face paintings) [86, 151, 180, 188, 189, 275, 276, 323], i.e., transferring knowledge from a source domain to a target domain. Different from them, we are interested in transferring knowledge from diverse vision encoders to the generator network.

Similar to previous methods we also employ differentiable data augmentation strategies [109, 264, 324, 326] for improved limited-data performance.

6.3 Method

Generative Adversarial Networks (GANs) aim to approximate the distribution of real samples, $p(\mathbf{x})$ from a finite training set. The generator network, G, with parameters θ maps latent vectors $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ to samples $G(\mathbf{z}) \sim p_\theta(\mathbf{x})$. The discriminator network $\mathcal{D}$ is trained adversarially to distinguish between the continuously changing generated distribution p_θ and the target real distribution $p_\mathbf{x}$. GANs perform the minimax optimization $\min_G \max_\mathcal{D} V(D, G)$, where

$$V(D, G) = \mathbb{E}_{\mathbf{x}\sim p_\mathbf{x}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\log(1 - D(G(\mathbf{z})))]. \tag{6.1}$$

Ideally, the discriminator should measure the gap between $p_\mathbf{x}$ and p_θ and guide the generator towards $p_\mathbf{x}$. In practice, however, large capacity discriminators easily overfit on the training set, especially in limited-data regimes [109, 324], and fail to provide useful gradients to the generator. As Figure 6.2 shows, even with techniques like differentiable data augmentation [109], pretrained DINO-based discriminator still tends to overfit, as evidenced by its low validation accuracy on held-out real vs. fake classification. In addition, the discriminator can potentially focus on imperceptible artifacts [270], leading to gradients that do not align with perceptual quality.

To address these issues, we propose ensembling pretrained deep feature extractors with small trainable classifiers on top as a discriminator. This approach provides two complementary benefits. First, training shallow classifiers on frozen pretrained features is a well-established technique for adapting deep networks to small-scale datasets while reducing overfitting [41, 82]. As shown in Figure 6.2, our method significantly reduces discriminator overfitting relative to the baseline. Second, deep networks trained on diverse tasks capture meaningful visual concepts across multiple levels of abstraction—from low-level cues (edges, textures) to high-level semantics (objects, parts) [16, 310]. An ensemble of discriminators grounded in these semantically-meaningful features is more likely to identify visually important differences and better align with human perception [320].

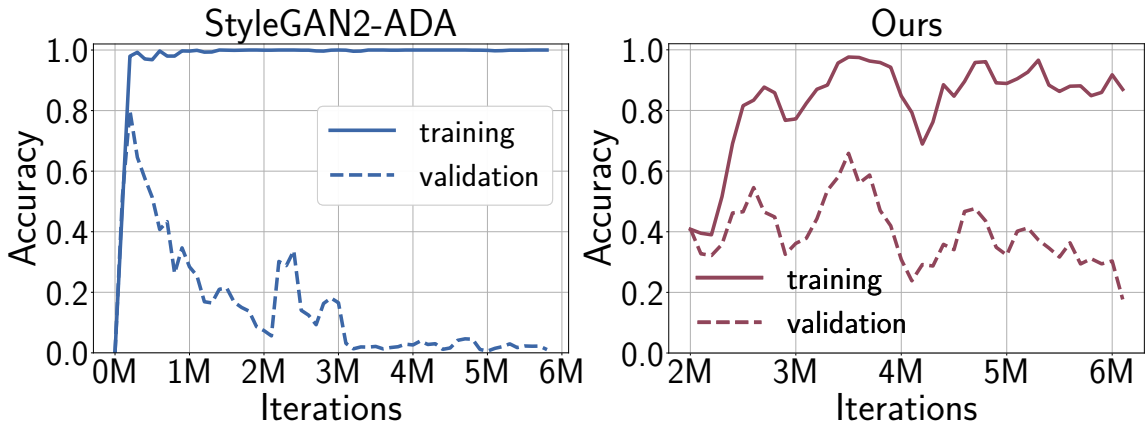


Figure 6.2: **Training and validation accuracy** w.r.t. training iterations for our DINO [34] based discriminator vs. baseline StyleGAN2-ADA discriminator on FFHQ 1k dataset. Our discriminator based on pretrained features has higher accuracy on validation real images and thus shows better generalization. In the above training, vision-aided adversarial loss is added at the 2M iteration.

6.3.1 Formulation

Given a set of pretrained feature extractors $\mathcal{F} = \{F_k\}_{k=1}^{K}$, we train corresponding discriminators $\{D_k\}_{k=1}^{K}$ based on these features. We add small classifier heads $\{C_k\}_{k=1}^{K}$ on top of the frozen pretrained features to measure the gap between $p_{\mathbf{x}}$ and p_θ. During training, each feature extractor F_k remains frozen, and only the corresponding classifier head C_k is updated. The generator G is optimized using gradients from both the original discriminator D (trained from scratch) and the vision-aided discriminators $\{D_k\}$. This combined adversarial training, which we refer to as *Vision-aided Adversarial* training, is formulated as:

$$\min_{G} \max_{D,\{C_k\}_{k=1}^{K}} V(D,G) + \overbrace{\sum_{k=1}^{K} V(D_k,G)}^{\text{vision-aided adversarial loss}}, \quad \text{where } D_k = C_k \circ F_k. \tag{6.2}$$

The above training objective involves summing discriminator losses from all available pretrained models $\{F_k\}$. However, training with all models at each iteration is computationally and memory-intensive. To address this bottleneck, we propose automatically selecting a small subset of M most important models, where $M \ll K$.

6.3.2 Model Selection

To select which pretrained models provide the most useful supervision, we identify models whose feature spaces best distinguish real from generated samples. The selection criterion identifies the most important model $\hat{F}_m$ from the initial model bank as:

$$m = \arg\max_{k \in 1..K} \left\{ \max_{C'_k} V(D'_k, G) \right\}, \quad \text{where } D'_k = C'_k \circ F_k. \tag{6.3}$$

Specifically, we employ linear probing, i.e., C'_k is a trainable linear classifier head on top of the pretrained feature extractors, F_k, from the initial bank. We measure the classifier accuracy for real vs. fake classification on a validation split to measure feature discriminability. Models with lower validation error indicate better feature quality, suggesting better feedback to the generator if used in the discriminator ensemble. We also empirically validate that higher linear probe accuracy correlates with better final FID metrics.

We iteratively select and add the best unused model after a fixed training interval. Progressively adding new discriminators has lower computational overhead than training with all of them simultaneously. Further, the progressive strategy naturally encourages diverse model selection: the iteratively selected models tend to capture complementary aspects of the data distribution. For instance, the first two models selected often comprise one self-supervised and one supervised model.

6.3.3 Training Algorithm

Our final algorithm consists of first training a GAN with standard adversarial training [85, 110] for a few warmup iterations. We then progressively select the best pretrained models via linear probing and add them to the discriminator ensemble at fixed intervals. The new vision-aided discriminators are added to the snapshot with the best training set FID in the previous stage. We note that using only pretrained models as discriminators causes training divergence; therefore, we ensemble the original discriminator with the vision-aided discriminators. We also employ standard regularization techniques like differentiable augmentation [109, 324], and one-sided label smoothing [225]. In experiments, we use three vision-aided discriminators and observe diminishing returns when adding models with low linear probe accuracy.

6.4 Experiments

We evaluate our method across two settings. The first setting is *domain customization*: a pretrained StyleGAN2 model trained on FFHQ is adapted to a new target domain, with experiments on AFHQ [49] (cat, dog, wild; ∼5k images per category at 512 resolution) and MetFaces [109] (1336 images at 1024 resolution). The second setting is *limited-data fine-tuning*: a GAN is trained from scratch on FFHQ [108], LSUN Cat, and LSUN Church [303], with 1k, 2k, and 10k training samples.

Baseline and metrics. We compare against state-of-the-art limited-data GAN methods: StyleGAN2-ADA [109] and DiffAugment [324]. To measure performance, we compute the commonly used Fréchet Inception Distance (FID) metric [93] using the `clean-fid` library [195]. We additionally report Kernel Inception Distance (KID) [18] for limited-data settings.

Pretrained vision model bank. We use eight large-scale self-supervised and supervised models. The model bank includes: vision-language (CLIP [208]), supervised classification (VGG-16 [240] on ImageNet [53]), self-supervised learning (DINO [34], MoBY [285]), face-specific tasks (face parsing [139], face normals [4]), and dense prediction (Swin-Transformer [161] segmentation on ADE-20K [328], object detection on MS-COCO [149]). We exclude Inception [255] since it is used to compute the FID.

Vision-aided discriminator's architecture. For the discriminator $\hat{D}_k$ based on pretrained model features, we extract spatial features from the last layer and use a small `Conv-LeakyReLU-Linear-LeakyReLU-Linear` architecture for binary classification. In the case of transformer networks, such as CLIP and DINO, we explore a multi-scale architecture that works better. For all experiments, we use three pretrained models selected by the model selection strategy during training. For more training and hyperparameter details, see Appendix D.1.

6.4.1 Results

Customization to new domains. We evaluate the transfer learning capability of our method by fine-tuning a pretrained GAN to new target domains. We perform experiments on AFHQ cat, dog, and wild categories (∼5k images per category) and MetFaces (1336 images). Both StyleGAN2-ADA and our method fine-tune from a

Dataset	StyleGAN2			StyleGAN2-ADA			Ours (w/ ADA)		
	FID ↓	KID ↓	Recall ↑	FID ↓	KID ↓	Recall ↑	FID ↓	KID ↓	Recall ↑
AFHQ Dog	9.28	3.13	0.42	7.52	1.22	0.43	**4.81**	**0.37**	**0.61**
AFHQ Cat	3.48	1.07	0.47	3.02	**0.38**	0.45	**2.69**	0.62	**0.50**
AFHQ Wild	**2.11**	**0.17**	0.35	2.72	0.17	0.29	2.18	0.28	**0.38**
MetFaces	57.26	2.50	**0.34**	17.56	1.55	0.22	**15.44**	**1.03**	0.30

Table 6.1: **Customization to AFHQ and MetFaces domain.** All models are fine-tuned from a StyleGAN2 model pretrained on FFHQ with the discriminator updated via FreezeD. Our method achieves lower FID and higher Recall on average across all target domains. Results averaged over three evaluations; KID ($\times 10^3$).

Dataset		StyleGAN2	DiffAugment	ADA	Ours (w/ ADA)			Ours (w/ DiffAugment)		
					+1st D	+2nd D	+3rd D	+1st D	+2nd D	+3rd D
FFHQ	1k	62.16	27.20	19.57	11.43	**10.39**	10.58	**12.33**	13.39	12.76
	2k	42.62	16.63	16.06	10.17	8.73	**8.18**	10.01	**9.24**	10.99
	10k	16.07	8.15	8.38	6.90	6.39	**5.90**	6.94	**6.26**	6.43
LSUN Cat	1k	185.75	43.32	41.14	15.49	12.90	**12.19**	13.52	12.52	**11.01**
	2k	68.03	25.70	23.32	13.44	13.35	**11.51**	12.20	11.79	**11.33**
	10k	18.59	12.56	13.25	8.37	7.13	**6.86**	8.19	7.90	**7.79**
LSUN Church	1k	-	19.38	19.66	11.39	9.78	**9.56**	10.15	**9.87**	9.94
	2k	-	13.46	11.17	5.25	**5.06**	5.26	6.09	6.37	**5.56**
	10k	-	6.69	6.12	4.80	4.82	**4.47**	3.42	3.41	**3.25**

Table 6.2: **FID across training set sizes (1k-10k).** Adding our vision-aided loss consistently improves FID over both StyleGAN2-ADA and DiffAugment baselines. Scores are means over 3 evaluations; lower is better.

StyleGAN2 model pretrained on FFHQ, with the vanilla discriminator updated using FreezeD [180] and our method additionally applying the vision-aided adversarial loss on top. Table 6.1 shows quantitative comparisons; our method outperforms or matches StyleGAN2-ADA across all target domains.

Limited-data fine-tuning. Table 6.2 shows results when training sample sizes vary from 1k to 10k on FFHQ, LSUN Cat, and LSUN Church. The consistent FID gains across all settings demonstrate the effectiveness of our method when fine-tuning with limited target-domain data. Figure 6.3 shows randomly generated samples from our method and StyleGAN2-ADA using the same latent code, illustrating quality improvements especially for FFHQ and LSUN Cat.

Human preference study. We conduct a human preference study on Amazon

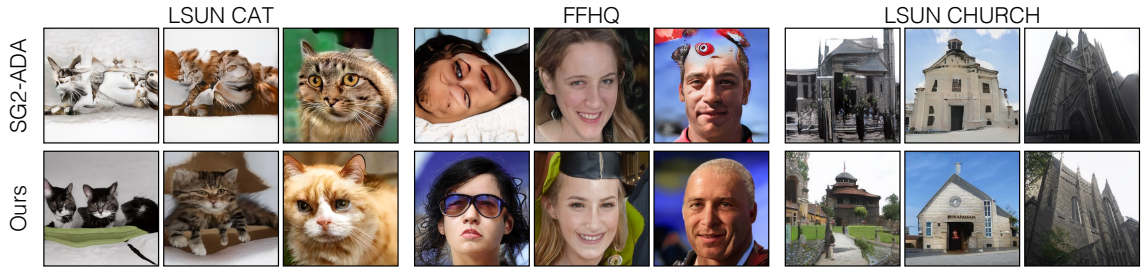


Figure 6.3: **Qualitative comparison in 1k training sample setting.** Each pair shows StyleGAN2-ADA baseline (top) vs. our method (bottom) for the same latent code. We fine-tune the StyleGAN2-ADA model with our vision-aided adversarial loss. For the same latent code, our method generates higher quality samples.

Dataset	**StyleGAN2 (F)**		**Ours (w/ ADA)**		**ADM**
	FID ↓	PPL ↓	FID ↓	PPL ↓	FID ↓
FFHQ-1024	**2.98**	144.62	3.01	**127.58**	-
LSUN Cat-256	6.86	437.13	**3.98**	**420.15**	5.57*
LSUN Church-256	4.28	**343.02**	**1.72**	388.94	-
LSUN Horse-256	4.09	337.98	**2.11**	**307.12**	2.57*

Table 6.3: **Results on full-dataset setting**. We improve the FID metric on LSUN categories by a significant margin. On the FFHQ dataset we improve the PPL metric. $*$ means directly reported from the ADM paper [56].

Mechanical Turk to verify that our quantitative evaluation reflects human perception. We compare StyleGAN2-ADA against our method trained on 1k samples across the three datasets. Since we fine-tune from StyleGAN2-ADA, paired comparisons use identical latent codes, as also shown in Figure 6.3. For each comparison, subjects view both generated images for six seconds and select the more realistic one. We perform the study with 50 subjects per dataset, each evaluating 55 images. On the FFHQ dataset, human preference for our method is $53.8\% \pm 1.3$. For the LSUN Church dataset, our method is preferred over StyleGAN2-ADA with $60.5\% \pm 1.7$, and for the LSUN Cat dataset the preference is $63.5\% \pm 1.6$. These preferences align with FID improvements, validating that our metric gains correspond to perceptual quality improvements.

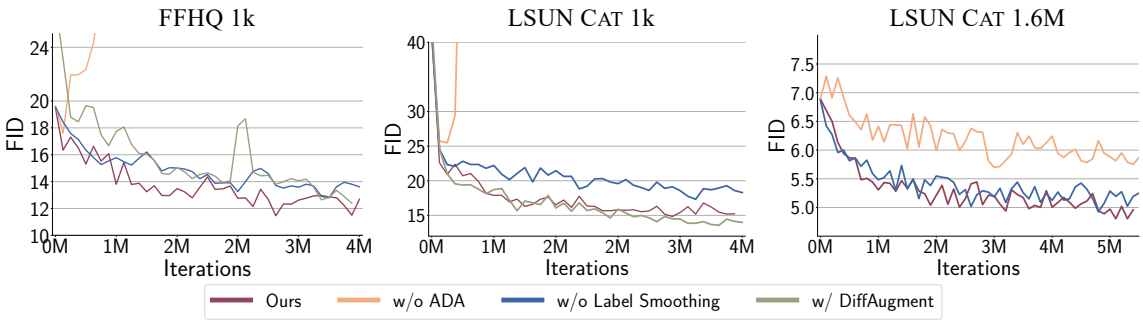


Figure 6.4: **Ablation of augmentation and label smoothing.** FID vs. training iterations with individual components removed from our method. Differentiable augmentation is beneficial even in full-dataset settings and is critical in limited data settings. Label smoothing, though not as critical, leads to consistent gains in FID for limited sample settings. Between DiffAugment [324] and ADA [109] for differentiable augmentation, both perform comparably.

6.4.2 Ablation

Full-dataset training. Here, we show that our vision-aided-adversarial loss can also be used to improve upon standard GAN training with large datasets. We fine-tune StyleGAN2 (config-F) [110] with our method on full datasets. Table 6.3 compares vanilla StyleGAN2, ADM [56], and our method (trained with three vision-aided discriminators). We evaluate FID and Perceptual Path Length (PPL) [108]. Our method achieves substantial improvements: on LSUN Cat FID improves from 6.86 to 3.98, on LSUN Church from 4.28 to 1.72, and on LSUN Horse from 4.09 to 2.11. On FFHQ, we improve PPL (144.62 to 127.58) while maintaining competitive FID. PPL correlates with image quality and smooth latent space [110].

Role of data augmentation and label smoothing. Here, we investigate two key regularization techniques used in our method, i.e., differentiable augmentation [109, 324] and label smoothing [225]. As Figure 6.4 shows, differentiable augmentation is critical for effectively leveraging pretrained features, and without it, training collapses in limited-data settings, with label smoothing providing additional improvements. For differentiable augmentation, both ADA and DiffAugment strategies perform comparably in our method. These findings highlight the role of regularization strategies to enable the use of pretrained vision models as discriminators.

6.5 Discussion and Limitations

In this work, we leverage off-the-shelf pretrained vision models as discriminators to significantly improve GAN training, either when customizing it to new target domains or training from scratch. While supervision from pretrained models is promising, there are still a few limitations including increased memory overhead and suboptimal performance when scaled to very few samples. Exploring more efficient [226, 256] or general purpose models [206] and few-shot learning techniques [80, 242] could address these in future work.

Nonetheless, our core contribution of leveraging knowledge from pretrained vision models to improve generative modeling has wider applicability and can be extended beyond GANs. In the next chapter, we shift back to text-to-image diffusion models and explore how to use general-purpose vision-language models to improve generative modeling on more complex tasks such as customizing the image generation model to a reference image guided instruction following model.

Chapter 7

Image Editing Customization using Pretrained Vision Language Models

Recently, a surge of unified image generation models has demonstrated remarkable capabilities in image-guided instruction following, performing various tasks like personalization, local image editing, and stylization — all forms of image customization now consolidated into a single image-editing model. However, training these models still relies on large supervised datasets of input-target pairs. This is a critical bottleneck, as such naturally occurring pairs are hard to curate at scale. The most commonly adopted workaround is to synthesize training pairs by leveraging the zero-shot capabilities of existing models, an approach also taken in Chapter 5 and recent works [26, 257]. As discussed earlier, this can propagate the artifacts of the pretrained model. Thus, building on the principle from Chapter 6, we instead study how vision models can serve as supervisory signals for customizing text-to-image diffusion models into a reference image-guided model for instruction following, which we call an *image-editing* model for brevity, consistent with usage throughout the chapter.

Concretely, we propose a learning paradigm that eliminates the need for paired data by using VLMs as the pretrained vision supervisors. Our approach directly optimizes a few-step diffusion model by unrolling it during training to generate the output image. For each input and text instruction, the VLM then evaluates whether

the output image follows the instruction and provides direct gradient feedback for end-to-end training. While VLMs are effective at evaluating instruction success, we find that they currently struggle with subjective tasks such as image fidelity and realism. Therefore, to ensure visual fidelity, we incorporate distribution matching loss (DMD), which constrains generated images to remain on the natural image manifold learned by a pretrained text-to-image model.

7.1 Introduction

Large-scale text-to-image models have achieved remarkable success, generating images of high fidelity that closely align with textual descriptions [105, 199, 211]. Despite these advances, text-only conditioning offers limited user control and falls short in many downstream applications [71, 174]. In practice, users often wish to start with an existing image to perform tasks like adjusting local attributes, changing the style, or placing an object in a new context. Such *image-editing* operations require precise, image-guided control that text-only prompts cannot provide.

While collecting large-scale text–image pairs is relatively straightforward [232], constructing supervised datasets for editing tasks is far more challenging, as one requires a *pair* of images—the input and its edited counterpart—along with the text instruction, and such data is rarely available online. Early methods addressed this by synthetically generating editing pairs [26] from a pretrained model, using zero-shot editing techniques [91]. However, synthetic datasets can quickly become outdated with new and improved base models, and they risk amplifying and propagating artifacts of the synthetic editing process. More recent approaches extract frames from videos and annotate their differences [45, 126, 250]. Although promising, the applicability of this strategy is constrained by the diversity of transformations present in natural video sequences, where obtaining pixel-aligned before–and–after edited pairs is nearly impossible. A final alternative is to manually create training pairs [171, 279], but this can be quite laborious and does not scale as easily.

In this chapter, we explore the possibility of customizing a model for instruction following *without before-after image-edit pairs*. Our key idea is to leverage supervision from Vision-Language Models (VLMs) [152], relying on their general image-understanding capabilities to check whether the generated images satisfy the edit-

ing instructions. Prior works have studied the use of specialized models or general-purpose VLMs in improving generative models along dimensions such as text alignment and aesthetic quality, primarily using reinforcement learning [19, 153]. In contrast, our method is the first to explore using gradient feedback from VLMs for image editing, and we distill this feedback into a lightweight generative model that can generalize to arbitrary images and edit instructions. Our final method combines the VLM-feedback with a distribution matching loss to ensure that generated outputs remain in the realistic image domain while following the edit instructions.

In summary, we make three key contributions. First, we propose NP-Edit (No-Pair Edit), a framework for customizing a text-to-image model into an image-editing model using gradient feedback from Vision–Language Models (VLMs) without requiring *paired supervision*. Second, for efficient training and effective VLM feedback, our formulation combines VLM feedback with distribution matching loss to learn a *few-step* model that remains competitive with supervised baselines. Third, we conduct a comprehensive empirical study analyzing the impact of different VLM backbones, dataset scale and diversity, and loss formulations. Our findings show that performance improves directly with more powerful VLMs and larger datasets, demonstrating its strong potential and scalability. Our code is available on our project website.

7.2 Related Work

Diffusion-based models for editing. As discussed in Chapter 5, encoder-based customization methods require large paired multi-image datasets for training. Extending this to general image instruction following, where instructions can be arbitrary, demands either even larger paired before-and-after datasets or a fundamentally different supervisory signal. Early approaches relied on zero-shot inference-time methods [11, 28, 91, 117, 196] but suffered from limited robustness to complex instructions. To improve efficiency and quality, subsequent works introduced training-based approaches [26, 42, 69, 252, 284] using either synthetically curated [26, 100, 292, 316, 322] or manually annotated datasets [76, 171, 254, 279]. Recent efforts have explored constructing paired data from videos [45, 250] or simulation environments [306], although these remain limited in either annotation diversity or visual realism. Here, we explore how to use pretrained vision-language models as an alternative supervisory

signal instead of ground-truth edits, removing the paired data requirement to train image-editing models.

Post-training for image generation. Post-training methods typically align image generators with human preferences using either Direct Preference Optimization (DPO) [269, 291] or Reinforcement Learning (RL) [19]. While early RL-based works used feedback from a simple scalar reward model [122, 287], the paradigm has recently been enhanced by employing sophisticated Vision-Language Models (VLMs) as "judges" to provide more general and accurate reward signals [127].

Although post-training has been successfully applied to text-to-image generation, its use for customization or editing models has been less explored. Recently, EARL [6] begins to address this by using a VLM-as-a-judge framework to post-train an image-editing model with RL. However, RL-based approaches often depend heavily on good initialization, typically requiring a Supervised Fine-Tuning (SFT) phase with paired editing data. In contrast, our method leverages differentiable feedback from the VLM, thereby obviating the need for an initial SFT stage and enabling the training of image-editing models without synthetically generated data.

Related to our proposal, Luo et al. [167] recently introduced a method that incorporates gradient feedback from VLMs to satisfy various criteria, including the horizon line, style, and layout, in generated images. However, their framework operates in a per-example optimization setting, requiring costly LoRA fine-tuning [96] for each criterion and prompt pair, and does not consider image-editing tasks.

Few-step diffusion models. Standard diffusion (or flow-matching) models require many sampling steps to generate high-quality images. Many prior works reduce the number of denoising steps for faster sampling by predicting larger denoising steps, including consistency models [66, 79, 90, 115, 163, 248, 249, 293, 329], shortcut models [66], meanflow [78], and inductive moment matching [329]. Another line distills a pretrained multi-step teacher into a few-step student by matching ODE trajectories [77, 224, 249], using adversarial loss [106, 228, 229, 288, 298], or applying score distillation [169, 298, 299, 330]. However, the main focus of the above methods is accelerating text-to-image generation. In our framework, we adopt DMD [299] primarily as a distribution matching objective when training an image-editing model without paired before-and-after editing data. We use VLM feedback as the editing signal while DMD ensures that the few-step editing model's output remains in the

real-image manifold defined by the pretrained text-to-image teacher model.

7.3 Method

Given a pretrained text-to-image diffusion model G_{init} and a dataset $\mathcal{X} = \{(\mathbf{y}_i, \mathbf{c}_i, \mathbf{c}_i^{\mathbf{y}}, \mathbf{c}_i^{\mathbf{x}})\}_{i=1}^{N}$ of reference image $\mathbf{y}$, corresponding edit instruction $\mathbf{c}$, and captions $\mathbf{c}^{\mathbf{y}}$ and $\mathbf{c}^{\mathbf{x}}$ that describe the reference and edited image respectively, we fine-tune G_{init} into a *few-step* customization or editing model G_θ without requiring a ground-truth edited image $\mathbf{x}$ according to the edit instruction. Our approach, No-Pair (NP)-Edit, introduces a VLM-based loss to evaluate edit success and combines it with a distribution matching loss to ensure outputs remain within the natural image domain. We first detail the construction of our dataset, then our training and implementation details.

7.3.1 Edit Instruction Dataset

Each dataset sample consists of a real image $\mathbf{y}$ as reference and an associated edit instruction, $\mathbf{c}$. Following prior works [159, 297], we focus on several categories of local editing operations, such as *Add*, *Replace*, *Remove*, *Adjust shape*, *Action*, *Stylization*, *Text editing*, *Color*, *Material*, and *Background* change, as well as more free-form editing or customization tasks [71, 130, 219]. Candidate instructions for each type are generated using Qwen2.5-32B VLM [206]. Given an image-instruction pair, we further query the VLM to assess its validity and to suggest the caption, $\mathbf{c}^{\mathbf{x}}$, for the edited image. For the free-form editing task, we restrict reference images to those showing a prominent central object, either filtered from a real image corpus or generated via the pretrained model [132, 257], and prompt the VLM to generate a caption that places the object in a novel background or context. In total, for local and free-form editing instructions, our dataset consists of $\sim$ 3M and $\sim$ 600K reference images, respectively. For details on instruction prompts, see Appendix E.2.

7.3.2 Training Objective

Training a diffusion or flow-based model [95, 160] for image editing without pairs presents a unique challenge. Standard diffusion training takes as input noised versions of a ground-truth image. In our setting, no such ground-truth edited image

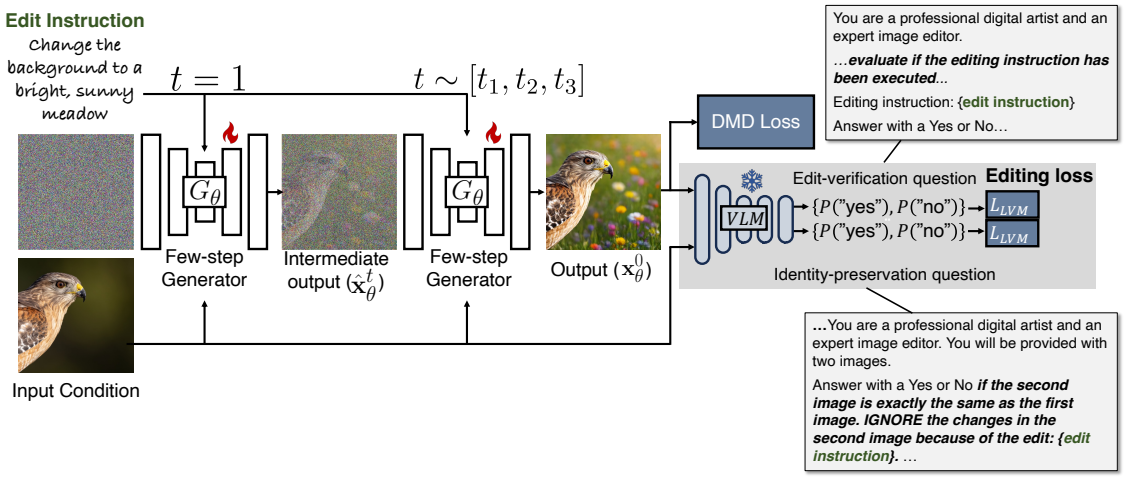


Figure 7.1: **Method overview.** We fine-tune a text-to-image model to a few-step image-editing model using differentiable VLM-feedback for edit success and distribution matching loss (DMD [298]) for realistic image generation. Given edit instructions and condition images, we predict edited images from noise. During training, we sample few-shot timesteps and perform two-step diffusion unrolling to predict edited images at intermediate timesteps, backpropagating loss through the diffusion steps.

exists; thus, we cannot construct these intermediate noisy inputs. On the other hand, directly mapping noise to the edited image in a single step is naturally challenging and yields poor fidelity. To address this, during training, we propose to unroll the backward diffusion trajectory starting from noise using a two-step sampling procedure [249]. Specifically, given the reference image–instruction pair $(\mathbf{y}, \mathbf{c})$, the editing model G_θ first predicts a provisional clean image $\hat{\mathbf{x}}_\theta^0$ from noise ϵ. Then, a second step refines this estimate by feeding an interpolated noisy input back into the model:

$$\begin{aligned} \hat{\mathbf{x}}_\theta^0 &= \epsilon - \hat{\mathbf{v}}_\theta, \quad \text{where } \hat{\mathbf{v}}_\theta \equiv G_\theta(\epsilon, t=1, \mathbf{c}, \mathbf{y}), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \\ \mathbf{x}_\theta^0 &= \hat{\mathbf{x}}_\theta^t - t\mathbf{v}_\theta, \quad \text{where } \mathbf{v}_\theta \equiv G_\theta(\hat{\mathbf{x}}_\theta^t, t, \mathbf{c}, \mathbf{y}), \\ &\qquad\qquad \hat{\mathbf{x}}_\theta^t = (1-t)\hat{\mathbf{x}}_\theta^0 + t\epsilon; \ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \ t \sim (0,1), \end{aligned} \tag{7.1}$$

With the second step, the model is now trained on noisy intermediate states at timesteps determined by t, while being more efficient than a full backward unroll. In our method, we focus on *few-step* generation—specifically four steps—and restrict $t \in [t_1, t_2, t_3]$, a fixed schedule, in the second step. A few-step generator provides a better estimate of the denoised image, $\mathbf{x}_\theta^0$, at intermediate steps, which in turn enables effective VLM-based feedback. VLMs tend to give unreliable judgments when inputs

are noisy or blurry. In addition, a few-step model also enables faster inference and lowers training costs. We show an overview of the method in Figure 7.1.

VLM-based editing loss. To evaluate whether an edit is successfully applied in $\mathbf{x}_\theta^0$, we define a set of template questions with corresponding ground-truth answers, $\mathcal{D}_{\text{QA}} = \{(\mathbf{X}_{q_j}, \mathbf{X}_{a_j})\}_j$, tailored to each edit category. The VLM is instructed to answer with a binary yes-or-no answer, i.e., $\mathbf{X}_{a_j} \in \{\textit{Yes}, \textit{No}\}$, and $\mathbf{X}_{\bar{a}_j}$ denotes the opposite response. The loss is then a binary cross-entropy over the predicted logit difference between the tokens corresponding to the correct and opposite responses, respectively.

$$\mathcal{L}_{\text{VLM}} = -\sum_j \log p(a_j), \text{ where } p(a_j) = \sigma(\ell_{a_j}^{(j)} - \ell_{\bar{a}_j}^{(j)}) \tag{7.2}$$

where $\ell_{a_j}^{(j)}$ is the logit corresponding to the token $\mathbf{X}_{a_j}$, σ is the sigmoid function, and $p(a_j)$ is the probability of correct answer, while restricting normalization to only the *Yes* and *No* tokens, which we observe to be more effective during training [317]. Computing this loss is relatively fast, as it requires a single forward call to the VLM per question, as opposed to autoregressive prediction. In our experiments, we use LLaVa-OneVision-7B [141] as the VLM that uses SigLIP [312] vision encoder and Qwen-2 LLM [205], and is among the state-of-the-art VLMs of this scale.

For each edit instruction, we use two complementary questions to compute the editing loss: (1) *Edit-verification* question to assess whether the intended edit is applied, and (2) *Identity-preservation* question to ensure the image is not over-edited and is consistent with the reference image. Specifically, for the local image-editing instructions, we verify edit success with the following question "The objective is to evaluate if the editing instruction has been executed in the second image. Editing instruction: {edit instruction}. Answer with a Yes or No." except *removal* edit-type, where we directly evaluate if the intended object is removed by asking "Answer with a Yes or No if the image has {object name}". For the identity-preservation question, we ask the following: "Answer with a Yes or No if the second image is exactly the same as the first image. IGNORE the changes in the second image because of the edit: {edit instruction}". For detailed system and user prompts, see Appendix E.3.

Distribution matching with text-to-image teacher model. While VLM feedback evaluates the efficacy of instruction following, it does not enforce the generated

outputs to remain in the real image domain. To ensure this and keep the output distribution of the generator aligned with the pretrained model, we apply Distribution Matching Distillation (DMD) [298, 299] between the fine-tuned model, G_θ, and the pretrained text-to-image (teacher) model, G_{init}. DMD minimizes the Kullback–Leibler (KL) divergence between the real image distribution, as estimated by the teacher model, and the output distribution of the fine-tuned model. The gradient of this KL-divergence loss with respect to the generator can be simplified to:

$$\nabla_\theta D_{KL} = \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}),t\in(0,1),\mathbf{x}_\theta^0}\Big[-\big(\mathbf{v}_{\text{real}}(\mathbf{x}_\theta^t, t, \mathbf{c}^{\mathbf{x}}) - \mathbf{v}_{\text{gen}}(\mathbf{x}_\theta^t, t, \mathbf{c}^{\mathbf{x}})\big)\frac{dG}{d\theta}\Big], \tag{7.3}$$

where $\mathbf{c}^{\mathbf{x}}$ is the text caption describing the noisy edited image $\mathbf{x}_\theta^t$ and $\mathbf{v}_{\text{real}}$, $\mathbf{v}_{\text{gen}}$ represents the predicted velocity from the teacher and a trainable auxiliary model, A_ϕ respectively. The auxiliary model is trained along with G_θ to learn the current output distribution of G_θ using a flow-based denoising objective. This loss ensures that the edited images not only satisfy the instruction but also remain faithful to the text-conditioned distribution of real images modeled by the pretrained teacher.

7.3.3 Training Details

The pretrained model G_θ generates an image, $\mathbf{x}$, conditioned only on text $\mathbf{c}$. To adapt it to our setup, we extend its conditioning to include the reference image $\mathbf{y}$. Following recent works [257, 284], we concatenate the VAE encoding of the reference image to the noisy target image encoding along the token sequence dimension, similar to text embedding, thereby enabling the model to attend to both text and visual conditions.

In the initial few iterations, we train the model with the objective of simply reconstructing the concatenated reference image, and then introduce our main training objective. This encourages the network to propagate content from the reference input, aligning it toward producing realistic images under joint text–image conditioning.

The final loss for the generator is a weighted combination of the VLM-based editing loss and DMD loss. The auxiliary network, A_ϕ, is updated N_{aux} times for every generator, G_θ, update [298]. Our pretrained generative model is a 2B parameter internal DiT-based [199] latent space diffusion model. The overall training pipeline is illustrated in Figure 7.1 and is detailed more formally in Algorithm 1 below.

Algorithm 1 NP-Edit: our training method

Require: Pretrained VLM and text-to-image model G_{init}, Dataset $\mathcal{X} = \{(\mathbf{y}_i, \mathbf{c}_i, \mathbf{c}_i^{\mathbf{y}}, \mathbf{c}_i^{\mathbf{x}})\}$.

Ensure: Few-step image-editing model G_θ

$G_\theta \leftarrow \text{copyWeights}(G_{\text{init}})$; $A_\phi \leftarrow \text{copyWeights}(G_{\text{init}})$

▷ Warmup with identity loss

for step $= 1$ to N_{warmup} **do**

  $(\mathbf{y}, \mathbf{c}^{\mathbf{y}}) \sim \mathcal{X}$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $t \sim (0, 1]$

  $\mathbf{x} \leftarrow \mathbf{y}$

  $\mathbf{x}^t \leftarrow (1-t)\mathbf{x} + t\epsilon$

  $\mathbf{v}_\theta \leftarrow G_\theta(\mathbf{x}^t, t, \mathbf{y}, \mathbf{c}^{\mathbf{y}})$

  $\mathcal{L}_{\text{id}} \leftarrow \|\mathbf{v} - \mathbf{v}_\theta\|^2$ where $\mathbf{v} = \epsilon - \mathbf{x}$

  $\theta \leftarrow \theta - \eta_G \nabla_\theta \mathcal{L}_{\text{id}}$

end for

▷ Main training loop

while `train` **do**

  $\{(\mathbf{y}, \mathbf{c}, \mathbf{c}^{\mathbf{x}})\} \sim \mathcal{X}, \epsilon \sim \mathcal{N}(0, I), t \in [t_1, t_2, t_3, t_4 = 1]$

  $\mathbf{v}_\theta \leftarrow G_\theta(\epsilon, t = 1, \mathbf{y}, \mathbf{c})$

  $\mathbf{x}_\theta^0 \leftarrow \epsilon - \mathbf{v}_\theta$

  if $t < 1$ **then**

    $\mathbf{v}_\theta \leftarrow G_\theta(\mathbf{x}_\theta^t, t, \mathbf{y}, \mathbf{c})$; $\mathbf{x}_\theta^t \leftarrow (1-t)\mathbf{x}_\theta^0 + t\epsilon$; $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

    $\mathbf{x}_\theta^0 \leftarrow \mathbf{x}_\theta^t - t\mathbf{v}_\theta$

  end if

  Compute $\mathcal{L}_{\text{VLM}}$ ▷ Eqn. 7.2

  Compute $\nabla_\theta D_{KL}$ ▷ Eqn. 7.3

  $\theta \leftarrow \theta - \eta_G \lambda_{\text{vlm}} \nabla_\theta \mathcal{L}_{\text{VLM}} - \eta_G \lambda_{\text{dmd}} \nabla_\theta D_{KL}$

  for local step $= 1$ to N_{aux} **do**

    $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \{(\mathbf{y}, \mathbf{c}, \mathbf{c}^{\mathbf{x}})\} \sim \mathcal{X}$

    $\mathbf{x}_\theta^0 \leftarrow G_\theta(\epsilon, \mathbf{y}, \mathbf{c})$ ▷ edited image with backward unroll

    $\mathbf{x}_\theta^t \leftarrow (1-t)\mathbf{x}_\theta^0 + t\epsilon \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})\ t \in (0, 1)$

    $\mathbf{v}_\phi \leftarrow A_\phi(\mathbf{x}_\theta^t, \mathbf{c}^{\mathbf{x}})$

    $\phi_A \leftarrow \phi_A - \eta_A \nabla_\theta \|\mathbf{v}_\phi - \mathbf{v}\|^2$ where $\mathbf{v} = \epsilon - \mathbf{x}_\theta^0$

  end for

end while

7.4 Experiments

In this section, we show the results of our method on local image-editing tasks and more free-form image-editing tasks such as customization on personal objects, and compare them with the state-of-the-art baseline methods.

7.4.1 Local Image-editing

Benchmark. Following prior works, we use GEdit-Benchmark [159] (English subset), which captures real-world user interactions across different edit types. We also show results on the ImgEdit [297] benchmark in Appendix E.1.

Evaluation metric. For quantitative evaluation, we follow prior works and use GPT4o-based VIEScore [127] metric. It scores each edit on: (1) Semantic Consistency (SC) score, evaluating whether the edit instruction was followed, and (2) Perceptual Quality (PQ) score, assessing realism and absence of artifacts. The geometric mean of the two is the final *Overall score*.

Baselines. We compare our method with leading baselines, including FLUX.1-Kontext [137], Step1X-Edit [159], BAGEL [52], OmniGen [284], and Qwen-Image-Edit [280]. Since no prior work explicitly targets few-step editing, we simply evaluate the above baselines with few-step sampling as well as their original multi-step setting for an upper-bound comparison. We also include Turbo-Edit [55], a state-of-the-art zero-shot few-step method that requires no paired supervision (as zero-shot) and is thus closest to our setup. We use the open-source implementation of all baselines, with further details in Appendix E.4.

Results Table 7.1 shows the quantitative result. In the few-step setting, our method achieves the best Overall and Perceptual Quality (PQ) scores compared to baseline methods. When compared to their original multi-step sampling, our few-step model still outperforms OmniGen and remains competitive with BAGEL, FLUX.1-Kontext, despite being 6× smaller parameter-wise. While Step1X-Edit and Qwen-Image-Edit perform better, they are substantially larger models. Figure 7.2 provides a qualitative comparison. As we can see, our method can successfully follow different editing instructions while being consistent with the input reference image. For instance, in the 6^{th} row (sheep color change), our approach produces a more natural edit com-

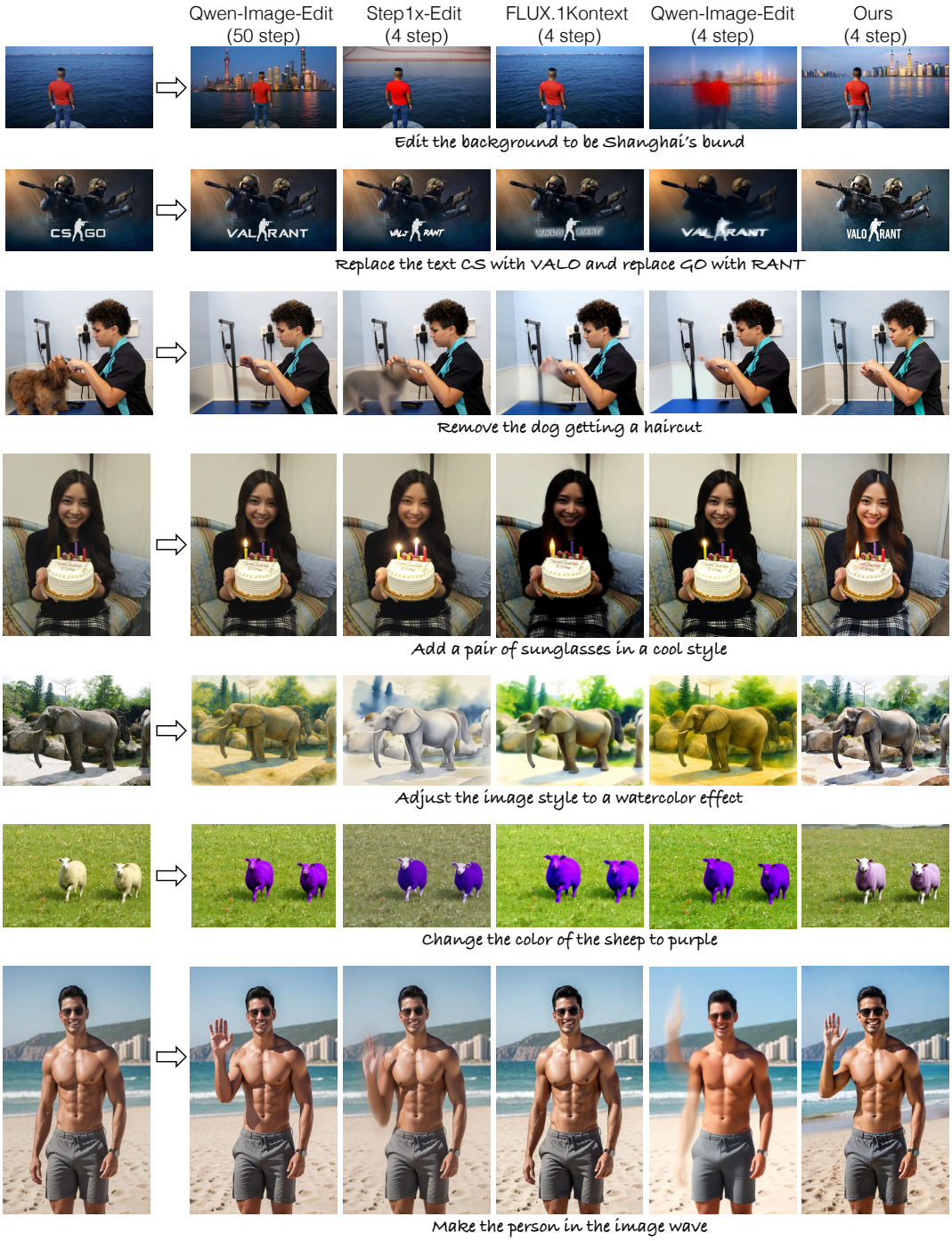


Figure 7.2: **Qualitative comparison on GEdit-Bench.** The first column shows the best multi-step baseline (upper-bound). Our method performs on par or better across different edit types in the few-step setting.

Method	#Param	#Step	SC Score↑	PQ Score ↑	Overall ↑
OmniGen [284]	4B	50	5.52	6.14	4.97
BAGEL [52]	7B	50	7.02	6.26	6.14
FLUX.1-Kontext [137]	12B	28	6.29	6.65	5.65
Step-1X Edit [159] v1.1	12B	28	7.30	7.37	6.79
Qwen-Image-Edit [280]	20B	50	**7.94**	**7.50**	**7.36**
FLUX.1-Kontext [137]	12B	4	5.80	5.74	5.04
Step-1X Edit [159] v1.1	12B	4	<u>6.61</u>	6.43	<u>6.01</u>
Qwen-Image-Edit [280]	20B	4	**6.82**	6.21	6.06
Turbo-Edit [55]	1B	4	3.84	<u>6.67</u>	3.84
NP-Edit (Ours)	2B	4	6.16	**7.69**	**6.10**

Table 7.1: **Quantitative evaluation on GEdit-Bench**. Our method performs on par or better than baselines under the few-step setting. For multi-step sampling, it still outperforms OmniGen and remains competitive with many of the larger-scale models like BAGEL and FLUX.1-Kontext. All numbers reported in $\times 10$.

Method	#Param	#Step	SC Score↑	PQ Score ↑	Overall ↑
DSD [27]	12B	28	6.71	7.41	6.78
SynCD [132]	12B	30	7.66	7.83	7.54
FLUX.1-Kontext [137]	12B	28	8.19	7.45	7.61
Qwen-Image-Edit [280]	20B	50	**8.53**	**7.79**	**8.02**
OminiControl [257]	12B	8	6.33	**7.82**	6.22
DSD [27]	12B	8	6.37	6.78	6.29
SynCD [132]	12B	8	7.71	6.84	7.07
FLUX.1-Kontext [137]	12B	8	<u>7.99</u>	7.18	<u>7.39</u>
Qwen-Image-Edit [280]	20B	8	**8.08**	7.44	**7.62**
NP-Edit (Ours)	2B	8	7.68	<u>7.56</u>	7.33
NP-Edit (Ours)	2B	4	7.60	7.28	7.10

Table 7.2: **Free-form editing task evaluation on DreamBooth**. We perform better than OminiControl, DSD, and SynCD, which are trained for this task on synthetic datasets. When compared to FLUX.1-Kontext and Qwen-Image-Edit, we still perform comparably in the few-step setting. All numbers are reported in $\times 10$.

pared to baselines. It also performs comparably to the multi-step variant for edits like lighting the candle in the 4^{th} row or making the person wave in the 7^{th} row.

7.4.2 Free-form Image-editing

Benchmark. We use the widely adopted DreamBooth [219] dataset for evaluation. It consists of 30 objects and 25 prompts per object category. The goal is to gener-

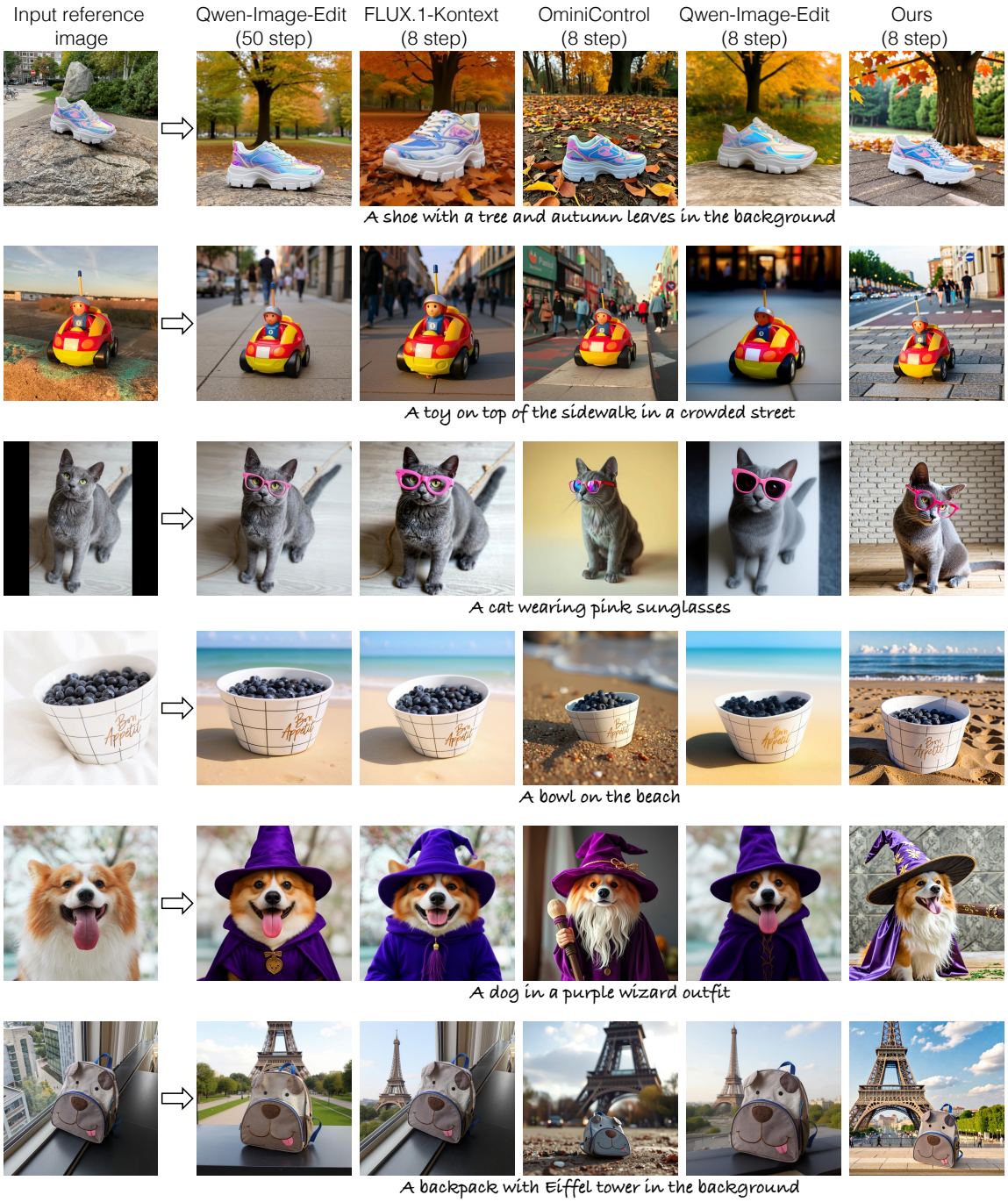


Figure 7.3: **Qualitative comparison on free-form editing task** of subject-guided generation. Our method can generate the object in new contexts while having better fidelity and higher pose diversity under few-step sampling.

ate the same object as shown in the reference image, but in a different context, as mentioned in the text prompt.

Baselines. We compare against state-of-the-art unified image-editing baselines such as FLUX.1-Kontext [137] and Qwen-Image-Edit [280] as well as OminiControl [257], DSD [27], and our previous work SynCD [132] introduced in Chapter 5, which are feed-forward models trained specifically for this task on synthetic datasets.

Evaluation metric. Here also, we use the VIEScore metric with the Semantic Consistency (SC) score to evaluate identity and text alignment, Perceptual Quality (PQ) score to measure realism, and the geometric mean of the two for the Overall score. We also report CLIPScore [208] and DINO [191] based metrics in Appendix E.1.

Results. As shown in Table 7.2, our method performs comparably to state-of-the-art methods. In the few-step sampling setting, all the baseline methods fail to generate realistic samples at 4 steps; therefore, we compare with them at 8 sampling steps. Our method still results in higher fidelity samples as shown in Figure 7.3, while maintaining object identity with the reference image. Note that our method performs better than OminiControl, which is also an 8-step model for this task.

7.4.3 Ablation

All ablations are done on the local image-editing task.

Training objective. We ablate our training objective across four settings: (1) using only distribution matching loss, (2) using only the VLM-based editing loss, (3) removing the *identity-preservation* question from $\mathcal{D}_{QA}$, and (4) replacing the binary cross-entropy loss (Eqn. 7.2) with standard cross-entropy over the full vocabulary.

Training without the VLM-based loss and relying solely on distribution matching significantly degrades the model's capabilities at following editing instructions. We observe that VLM-based loss is essential for maintaining consistency between input and edited images and for certain editing tasks like *Removal*, as shown by quantitative performance per sub-edit type in Figure 7.4 and also qualitatively in Figure 7.6. However, only training with VLM-based loss leads to unrealistic outputs (Figure 7.5), and the training eventually diverges, highlighting the complementary role of distribution matching loss. The binary cross-entropy loss and the identity-preservation question both help improve performance, removing either degrades the Overall score

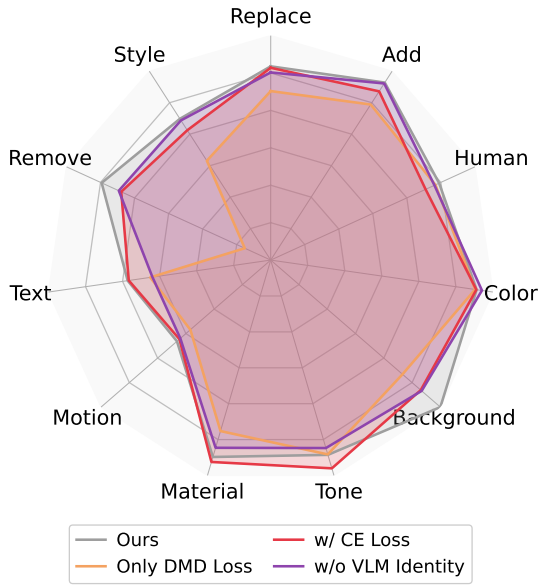


Figure 7.4: **Performance for each sub edit-type.** DMD loss alone fails on removal and style tasks. Binary cross-entropy and VLM identity-based questions help improve performance.

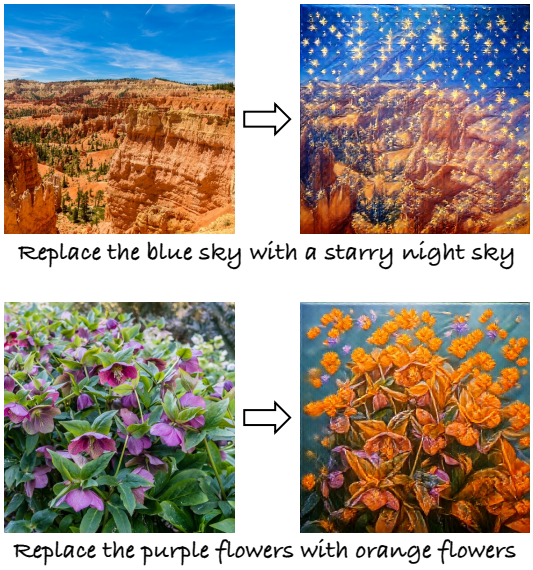


Figure 7.5: **Training with only VLM-editing loss** produces lower-fidelity samples by primarily maximizing edit success at the cost of visual quality, necessitating distribution matching loss.

to 5.89 and 5.76 respectively, compared to 6.10 for our full method.

Dataset and VLM scale. To study the role of dataset scale, we vary the number of unique reference images in training. Our final dataset represents the maximum scale feasible under our computational resources. Table 7.3 shows the performance across different dataset sizes and VLM backbones. We observe consistent gains with larger datasets, suggesting that further scaling of data could yield additional improvements. Similarly, a larger parameter VLM-backbone leads to better performance, underscoring the promise that our method can improve as more powerful VLMs are developed.

Our method vs. Reinforcement Learning (RL). RL is a common post-training strategy for improving pretrained models without paired supervision and can also leverage VLMs as the reward model, a similar setup to ours. Thus, we benchmark our method against Flow-GRPO [153], a widely used RL method for text-to-image diffusion. However, since RL relies on a reasonable initialization, we need to first train an image-editing model via Supervised Fine-Tuning (SFT) on a paired dataset [147]. We then fine-tune it with Flow-GRPO using the same LLaVA-OneVision reward model

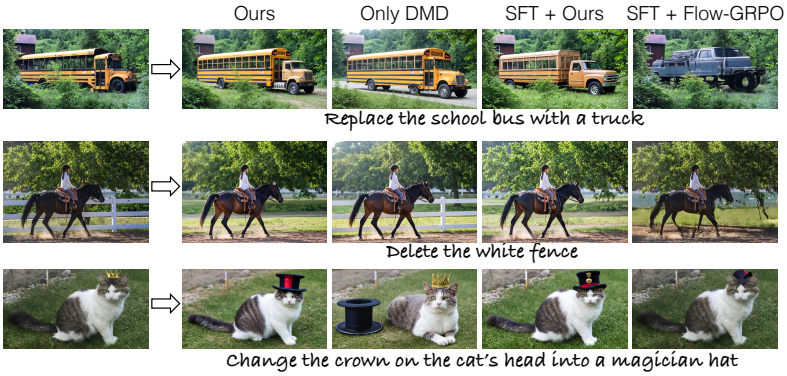


Figure 7.6: **Qualitative analysis of ablation experiments.** Training with only DMD loss fails to achieve certain tasks like removal and has worse input-edited image alignment. Our method also works better compared to fine-tuning an SFT model with RL-based method Flow-GRPO using the same VLM as reward. Zoom in for details.

Method	**SC Score↑**	**PQ Score ↑**	**Overall ↑**
1 % Dataset	4.41	7.10	4.66
50 % Dataset	5.41	**7.73**	5.52
100 % Dataset	**6.16**	7.69	**6.10**
InternVL-2B	5.36	7.67	5.45
InternVL-14B	5.88	7.74	5.89
LLava-0.5B	4.57	7.50	4.59
LLava-7B (Ours)	**6.16**	**7.69**	**6.10**
SFT	3.91	5.70	3.64
SFT + RL	4.55	5.47	4.19
SFT + Ours	**6.08**	**7.83**	**6.06**

Table 7.3: **Role of Dataset and VLM scale and comparison with Reinforcement Learning** on the GEdit-Bench. Increasing dataset scale and using stronger VLMs leads to increased performance. Our method also performs better than post-training an SFT model with RL-based method Flow-GRPO [153].

as in our approach. As shown in Table 7.3, SFT alone performs poorly, likely due to the limited quality of paired data. Our method surpasses both SFT and SFT+RL, despite requiring no paired supervision. Fine-tuning the model with some paired data before applying our approach can slightly improve the pixel-level consistency between the input reference and output edited image (as shown in Figure 7.6), although the quantitative numbers are similar.

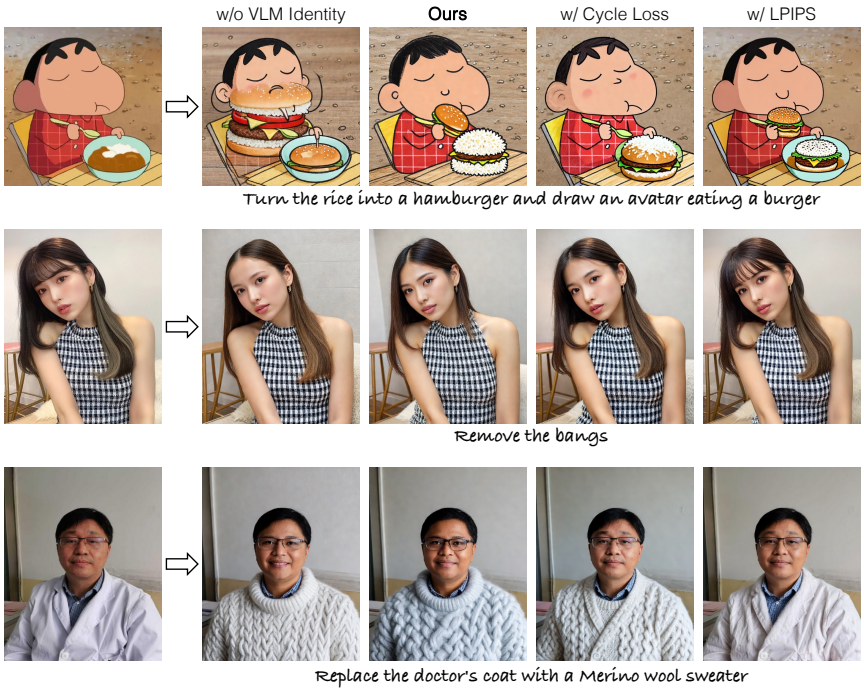


Figure 7.7: **Limitation.** Our identity-preservation question helps in maintaining input consistency (1st vs. 2nd column), but still isn't perfect. Cycle loss helps mitigate this (3rd column) but with a performance drop (VIEscore 5.10 vs 6.10). LPIPS loss helps for some cases (1st row) but has overall worse performance (2nd and 3rd row).

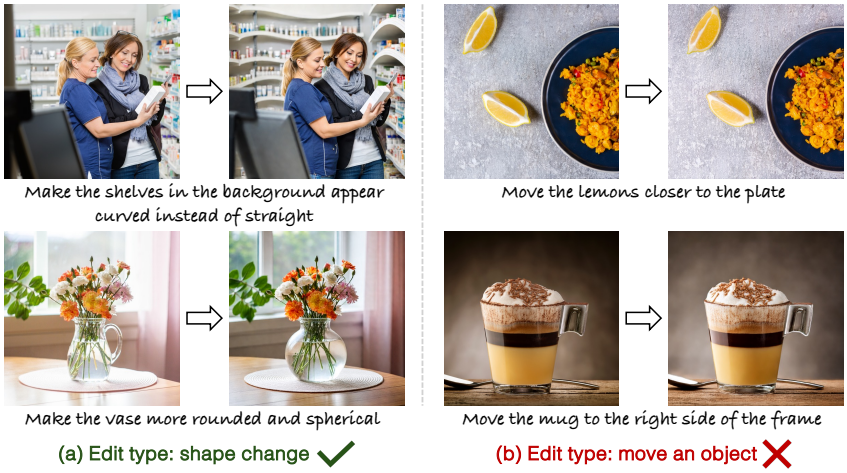


Figure 7.8: **Challenging and difficult editing tasks.** Our method can perform various challenging edits as well, like shape change as shown in (a). But we fail at other difficult tasks requiring spatial reasoning, such as changing the spatial location of the object in the scene, where VLM-based gradient feedback is less reliable.

Alignment between input and edited image. Our framework lacks pixel-wise supervision to preserve regions not targeted by the edit instruction. Consequently, edited images can deviate from the input in specific details or spatial alignment, as shown in Figure 7.7. While our VLM-based loss includes an identity-preservation question, current VLMs struggle to judge fine-grained pixel-level alignment. To mitigate this, we experiment with additional regularization in the form of (1) LPIPS [320] loss between input and edited image, and (2) Cycle loss [334] between input and cycle-edited image. Cycle loss applies a VLM-generated reverse edit (e.g., add back a removed object) and encourages reconstruction of the original input in SigLIP [312] global embedding feature space.

We show the results in Figure 7.7. While LPIPS loss improves consistency, it degrades editing quality, particularly for edit-types like *Removal*. Cycle loss works better in comparison but still has lower edit success than ours. The overall VIEScore on GEdit-Bench for LPIPS and cycle loss is 4.52 and 5.21 respectively, compared to 6.10 for Ours. However, we anticipate that as VLMs improve, with the capability of nuanced, pixel-level assessment, it will help improve the input-edit alignment in our method without requiring additional regularization.

VLM supervision for difficult editing tasks. While NP-Edit can successfully edit images for various challenging edit-types like actions (2^{nd} last row in Figure 7.2) and shape changes (Figure 7.8 (a)), our method is fundamentally upper-bounded by the VLM's ability to understand and reason about image editing. Consequently, it currently fails at tasks requiring complex spatial reasoning (e.g., moving objects), as VLMs struggle to provide meaningful feedback in these scenarios. We show sample failure case examples in Figure 7.8 (b). As unified generative models and Vision–Language Models advance, the instruction-following capabilities learned through our NP-Edit training framework are expected to improve correspondingly.

7.5 Discussion and Limitations

This work introduces a new paradigm for enabling customization of image-guided instruction-following models from a pretrained text-to-image diffusion model, without paired supervision. Our approach combines differentiable feedback from VLMs to ensure editing success with a distribution matching objective to maintain visual

realism. This method achieves competitive performance with recent state-of-the-art baselines trained on paired data while enabling efficient few-step generation.

Despite these promising results, our method has limitations. Without pixel-level supervision, edits may deviate from the input image in fine-grained details or fail to fully preserve subject identity, as discussed above. Another limitation of our method is the need to keep the VLM in GPU memory, which introduces VRAM overhead. We expect that ongoing advances in stronger and more efficient VLMs can help address this issue. Overall, our framework scales effectively with large unpaired datasets and highlights a path toward more flexible customization of generative models for diverse downstream tasks without relying on costly paired data curation.

Chapter 8

Discussion

This dissertation developed methods for customizing pretrained generative image models. To do so, we either used a small set of high-quality user examples (Part I) or leveraged a strong frozen pretrained model–as a data generator (Part II) or as an evaluator (Part III). These approaches reflect a broader principle: foundation models trained on internet-scale data carry rich, general-purpose priors over the visual world, and effective customization largely amounts to surfacing these priors in a controllable manner without losing what was learned during pretraining. For instance, the methods of Part I update mainly the cross-attention layers that bind text to image regions, a low-rank update [96] that repurposes the model's text-to-image mapping while leaving its visual priors intact. And in Parts II and III, we utilize the strong general priors of the pretrained models to derive supervision, either in the form of synthetic training data or by directly evaluating generated outputs.

Building on the insights from our work, we now discuss potential future directions.

Self-improvement with unified models. Our approach in Chapter 7 relied on a frozen, external VLM as the evaluator for feedback. This capped the quality of supervision at the evaluator's own capability. In addition, running a separate large model alongside the generator introduced significant memory and compute overhead.

Together, these limitations point toward *unified models* [52, 262] augmented with periodic human feedback to progressively refine the understanding pathway. Unified models combine visual understanding and generation within a single architecture, and thus can assess their own outputs through the understanding pathway and

use that signal to improve generation, with no need for a separate evaluator. This paradigm is similar to self-rewarding and self-play optimization techniques in language models [307, 321]. But key challenges in extending this to visual generation include improving training efficiency, converting natural-language feedback into a differentiable objective, which can be more informative compared to the binary Yes vs. No feedback used in our work, and making efficient use of the scarce human supervision needed to bootstrap the understanding pathway.

Active learning and mining the right data. Since human supervision is scarce, as noted above, it is crucial to label the *right* data that will most improve the evaluator. Active-learning methods offer a principled way to identify current failure cases and direct annotation effort precisely where it's needed. Failure cases could be identified, for instance, by measuring disagreement across models or random seeds, or by inspecting chain-of-thought traces. An additional question is how to make each label maximally informative for the specific factor we want the evaluator to improve upon, such as subject identity. Dataset distillation [36, 281] methods offer a starting point, but they are built to compress or subsample *fixed* datasets; adapting them to actively curate supervision for vision-language evaluators remains an open problem.

Interactive and long-context customization. Each chapter in this dissertation addressed an individual task in a single-turn interaction. In practice, however, users often refine an output over many turns, mixing inputs such as sketches, scribbles, and reference images [277]. Supporting such workflows requires following instructions incrementally while staying consistent with the full history of edits. The NP-Edit framework from Chapter 7 is well suited to this setting and provides a natural foundation for training an end-to-end agentic system. Recent work has begun to explore related ideas [296], but many open problems remain, including robust reward mechanisms, scalable memory for long, heterogeneous contexts, and efficient multimodal planning.

Safe and responsible deployment. Our concept ablation work from Chapter 4 enabled selectively removing unwanted or harmful content from pretrained text-to-image generative models. However, it remains susceptible to adversarial prompting [265], and hardening it against such attacks is a direction for future work. Beyond concept removal, broader challenges persist, such as reducing biases in generated content [47] and establishing standardized mechanisms for data attribution [273] and ownership [32, 162] to ensure trustworthy and equitable use of generative tools.

Bibliography

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 18, 88

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 18

[3] Rameen Abdal, Peihao Zhu, John Femiani, Niloy Mitra, and Peter Wonka. Clip2stylegan: Unsupervised extraction of stylegan edit directions. In *ACM SIGGRAPH*, 2022. 18

[4] Victoria Fernández Abrevaya, Adnane Boukhayma, Philip HS Torr, and Edmond Boyer. Cross-modal deep face normals with deactivable skip connections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 92, 170

[5] Adobe. Generative fill. https://www.adobe.com/products/photoshop/generative-fill.html, 2023. 1

[6] Saba Ahmadi, Rabiul Awal, Ankur Sikarwar, Amirhossein Kazemnejad, Ge Ya Luo, Juan A Rodriguez, Sai Rajeswar, Siva Reddy, Christopher Pal, Benno Krojer, et al. The promise of rl for autoregressive image editing. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025. 100

[7] LAION AI. LAION-Aesthetics_Predictor. https://github.com/LAION-AI/aesthetic-predictor, 2022. 70

[8] Isabela Albuquerque, João Monteiro, Thang Doan, Breandan Considine, Tiago Falk, and Ioannis Mitliagkas. Multi-objective training of generative adversarial networks with multiple discriminators. In *International Conference on Machine Learning (ICML)*, 2019. 87

[9] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*, 2017. 87

[10] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 88

[11] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 2023. 99

[12] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 65

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015. 17

[14] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (TOG)*, 2019. 46

[15] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *European Conference on Computer Vision (ECCV)*, 2020. 18, 46

[16] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 2020. 89

[17] Romain Beaumont. Clip retrieval. `https://github.com/rom1504/clip-retrieval`, 2022. 51

[18] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations (ICLR)*, 2018. 23, 52, 92

[19] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024. 99, 100

[20] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 36

[21] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan Barron, Hendrik Lensch, and Varun Jampani. Samurai: Shape and material from unconstrained real-world arbitrary image collections. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 36

[22] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hen-

grui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021. 45

[23] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 22

[24] Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 38, 154

[25] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019. 86

[26] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 36, 38, 71, 97, 98, 99, 155

[27] Shengqu Cai, Eric Ryan Chan, Yunzhi Zhang, Leonidas Guibas, Jiajun Wu, and Gordon Wetzstein. Diffusion self-distillation for zero-shot customized image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 108, 110, 183

[28] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 99

[29] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2015. 45

[30] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, 2019. 45

[31] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, 2021. 45

[32] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *USENIX Security Symposium*, 2023. 3, 43, 44, 45, 52, 53, 118

[33] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*, 2023. 45

[34] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. xvii, 88, 90, 92, 170

[35] Arantxa Casanova, Marlène Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero. Instance-conditioned gan. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 88

[36] George Cazenavette, Antonio Torralba, and Vincent Sitzmann. Dataset distillation for pre-trained self-supervised vision models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025. 118

[37] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. In *International Conference on Machine Learning (ICML)*, 2023. 44

[38] ChatGPT. Chatgpt. `https://chat.openai.com/chat`, 2022. 30, 36, 38, 51, 158

[39] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 88

[40] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020. 85, 86, 88

[41] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2019. 89

[42] Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Ruiz, Xuhui Jia, Ming-Wei Chang, and William W Cohen. Subject-driven text-to-image generation via apprenticeship learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 41, 64, 99

[43] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. In *International Conference on Learning Representations (ICLR)*, 2023. 65

[44] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang

Zhao. Anydoor: Zero-shot object-level image customization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 66

[45] Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via learning real-world dynamics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 98, 99

[46] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 85, 86

[47] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 118

[48] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 18

[49] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 87, 92

[50] Civitai. Civitai: The home of open-source generative ai. `https://civitai.com`, 2023. 1

[51] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. xv, 65, 66, 67, 165

[52] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 106, 108, 117

[53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 92, 170

[54] Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. Flashtex: Fast relightable mesh texturing with lightcontrolnet. In *European Conference on Computer Vision (ECCV)*, 2024. 66

[55] Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. Turboedit: Text-based image editing using few-step diffusion models. In *ACM SIGGRAPH Asia Conference Proceedings*, 2024. 106, 108, 183

[56] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. xxi, 1, 18, 94, 95

[57] Stable Diffusion. Stable diffusion 1.4. `https://huggingface.co/CompVis/stable-diffusion-v-1-4-original`, 2022. 1, 17, 23, 44, 52

[58] Yuxuan Ding, Lingqiao Liu, Chunna Tian, Jingyuan Yang, and Haoxuan Ding. Don't stop learning: Towards continual learning for the clip model. *arXiv preprint arXiv:2207.09248*, 2022. 17

[59] Thang Doan, Joao Monteiro, Isabela Albuquerque, Bogdan Mazoure, Audrey Durand, Joelle Pineau, and R Devon Hjelm. On-line adaptative curriculum learning for gans. In *Conference on Artificial Intelligence (AAAI)*, 2019. 87

[60] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2014. 88

[61] Jiahua Dong and Yu-Xiong Wang. Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 38, 155

[62] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2016. 88

[63] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 163

[64] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2017. 87

[65] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024. 1, 11, 64, 65, 67, 163, 165, 166

[66] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step

diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025. 100

[67] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 1999. 18

[68] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 41

[69] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations (ICLR)*, 2024. 99

[70] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 2022. 18, 46

[71] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations (ICLR)*, 2023. 18, 23, 25, 30, 46, 51, 60, 64, 65, 98, 101, 153

[72] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based domain tuning for fast personalization of text-to-image models. *ACM Transactions on Graphics (TOG)*, 2023. 41, 46

[73] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 46

[74] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision (IJCV)*, 2024. 18

[75] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 88

[76] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024. 99

[77] Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. One-step diffusion distillation via deep equilibrium models. In *Conference on Neural Information*

Processing Systems (NeurIPS), 2023. 100

[78] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025. 100

[79] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. In *International Conference on Learning Representations (ICLR)*, 2025. 100

[80] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 96

[81] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 45

[82] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 89

[83] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 45

[84] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 45

[85] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014. 1, 87, 91

[86] Zheng Gu, Wenbin Li, Jing Huo, Lei Wang, and Yang Gao. Lofgan: Fusing local representations for few-shot image generation. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 18, 88

[87] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 165

[88] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 65

[89] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), 2016. 33, 85, 86, 88

[90] Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024. 100

[91] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *International Conference on Learning Representations (ICLR)*, 2023. 46, 98, 99, 165

[92] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. 23, 52

[93] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 92

[94] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *Conference on Neural Information Processing Systems (NeurIPS) Workshop*, 2021. 10

[95] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 1, 9, 65, 101

[96] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 18, 38, 65, 72, 100, 117, 155, 182

[97] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 73, 171

[98] HuggingFace. Lora-stable diffusion. `https://github.com/huggingface/diffusers/blob/main/examples/dreambooth/train_dreambooth_lora_sdxl.py`, 2023. 38

[99] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 88

[100] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. In *International Conference on Learning Representations*

(ICLR), 2025. 99

[101] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 12, 169

[102] Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan Popov, André Araujo, Ricardo Martin Brualla, Kaushal Patel, et al. Navi: Category-agnostic image collections with high-quality 3d shape and pose annotations. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 38

[103] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016. 88

[104] Maxwell Jones, Sheng-Yu Wang, Nupur Kumari, David Bau, and Jun-Yan Zhu. Customizing text-to-image models with a single image pair. In *ACM SIGGRAPH Asia Conference Proceedings*, 2024. 7

[105] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 98

[106] Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling diffusion models into conditional gans. In *European Conference on Computer Vision (ECCV)*, 2024. 100

[107] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. 87

[108] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 87, 92, 95

[109] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. xvii, 18, 86, 87, 89, 91, 92, 95, 169

[110] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 91, 95, 170

[111] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 86, 87

[112] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 11, 154, 166

[113] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 46

[114] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 35

[115] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. 100

[116] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 18, 88

[117] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 99

[118] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 180

[119] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 12

[120] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 73, 167

[121] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017. 18

[122] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 100

[123] Zhifeng Kong and Kamalika Chaudhuri. Data redaction from pre-trained gans. In *Conference on Neural Information Processing Systems (NeurIPS) Workshop*, 2022. 46, 47

[124] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 88

[125] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2012. 88

[126] Benno Krojer, Dheeraj Vattikonda, Luis Lara, Varun Jampani, Eva Portelance, Chris Pal, and Siva Reddy. Learning action and reasoning-centric image editing from videos and simulation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 98

[127] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Conference of the Association for Computational Linguistics (ACL)*, 2024. 100, 106, 171, 182

[128] Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for gan training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 18

[129] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 5

[130] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. xvii, xxi, 4, 46, 51, 60, 64, 65, 76, 82, 101

[131] Nupur Kumari, Grace Su, Richard Zhang, Taesung Park, Eli Shechtman, and Jun-Yan Zhu. Customizing text-to-image diffusion with object viewpoint control. In *ACM SIGGRAPH Asia Conference Proceedings*, 2024. 4

[132] Nupur Kumari, Xi Yin, Jun-Yan Zhu, Ishan Misra, and Samaneh Azadi. Generating multi-image synthetic data for text-to-image customization. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 6, 101, 108, 110, 183

[133] Nupur Kumari, Sheng-Yu Wang, Nanxuan Zhao, Yotam Nitzan, Yuheng Li, Krishna Kumar Singh, Richard Zhang, Eli Shechtman, Jun-Yan Zhu, and Xun Huang. Learning an image editing model without image editing pairs. In *International Conference on Learning Representations (ICLR)*, 2026. 6

[134] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 11, 12, 65, 67, 72, 165, 166

[135] Black Forest Labs. Flux-depth. https://huggingface.co/black-forest-labs/FLUX.1-Depth-dev, 2024. 69, 165

[136] Black Forest Labs. Flux. https://huggingface.co/black-forest-labs/FLUX.1-schnell, 2024. 166

[137] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 106, 108, 110

[138] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 88

[139] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 92, 170

[140] Jason Lee, Kyunghyun Cho, and Douwe Kiela. Countering language drift via visual grounding. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 17, 21

[141] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *Transactions on Machine Learning Research (TMLR)*, 2025. 103

[142] Dingcheng Li, Zheng Chen, Eunah Cho, Jie Hao, Xiaohu Liu, Fan Xing, Chenlei Guo, and Yang Liu. Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2022. 17, 18

[143] Dongxu Li, Junnan Li, and Steven CH Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 64, 66, 72, 74, 154, 166

[144] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 18, 19, 46

[145] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 65

[146] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 18

[147] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025. 111

[148] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 77

[149] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 92

[150] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 11, 71

[151] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations (ICLR)*, 2021. 18, 88

[152] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 98

[153] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025. xxii, 99, 111, 112, 182

[154] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations (ICLR)*, 2022. 154

[155] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum.

Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision (ECCV)*, 2022. 17

[156] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 38, 66, 68

[157] Sean J Liu, Nupur Kumari, Ariel Shamir, and Jun-Yan Zhu. Generative photomontage. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 7

[158] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024. 167

[159] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025. 101, 106, 108

[160] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 11, 101

[161] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 92, 170

[162] Shayne Longpre, Robert Mahari, Naana Obeng-Marnu, William Brannon, Tobin South, Jad Kabbara, and Sandy Pentland. Data Authenticity, Consent, and Provenance for AI Are All Broken: What Will It Take to Fix Them? *An MIT Exploration of Generative AI*, 2024. https://mit-genai.pubpub.org/pub/uk7op8zs. 118

[163] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations (ICLR)*, 2025. 100

[164] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 10, 65

[165] Daohan Lu, Sheng-Yu Wang, Nupur Kumari, Rohan Agarwal, Mia Tang, David Bau, and Jun-Yan Zhu. Content-based search for deep generative models. In *ACM SIGGRAPH Asia Conference Proceedings*, 2023. 7

[166] Yuchen Lu, Soumye Singhal, Florian Strub, Aaron Courville, and Olivier Pietquin. Countering language drift with seeded iterated learning. In *International Conference on Machine Learning (ICML)*, 2020. 17, 21

[167] Grace Luo, Jonathan Granskog, Aleksander Holynski, and Trevor Darrell. Dual-process image generation. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 100

[168] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. xv, 67, 163, 165

[169] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 100

[170] Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. In *ACM SIGGRAPH*, 2024. 66

[171] Nadav Magar, Amir Hertz, Eric Tabellion, Yael Pritch, Alex Rav-Acha, Ariel Shamir, and Yedid Hoshen. Lightlab: Controlling light sources in images with diffusion models. In *ACM SIGGRAPH Conference Proceedings*, 2025. 98, 99

[172] Puneet Mangla, Nupur Kumari, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Data instance prior (disp) in generative adversarial networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2022. 88

[173] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 87

[174] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2022. 38, 98, 154

[175] Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 46

[176] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations (ICLR)*, 2023. 46

[177] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training meth-

ods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018. 87

[178] Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. In *International Conference on Machine Learning (ICML)*, 2024. 165

[179] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations (ICLR)*, 2022. 46

[180] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, 2020. 4, 18, 46, 88, 93

[181] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Conference on Artificial Intelligence (AAAI)*, 2024. 154

[182] Ivona Najdenkoska, Animesh Sinha, Abhimanyu Dubey, Dhruv Mahajan, Vignesh Ramanathan, and Filip Radenovic. Context diffusion: In-context aware image generation. In *European Conference on Computer Vision (ECCV)*, 2024. 65

[183] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 45

[184] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning (ICML)*, 2022. 1, 18, 44

[185] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning (ICML)*, 2021. 11

[186] Yotam Nitzan, Kfir Aberman, Qiurui He, Orly Liba, Michal Yarom, Yossi Gandelsman, Inbar Mosseri, Yael Pritch, and Daniel Cohen-Or. Mystyle: A personalized generative prior. In *ACM SIGGRAPH Asia*, 2022. 18

[187] Yotam Nitzan, Michaël Gharbi, Richard Zhang, Taesung Park, Jun-Yan Zhu, Daniel Cohen-Or, and Eli Shechtman. Domain expansion of image generators. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 46

[188] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets

via batch statistics adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 18, 46, 88

[189] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 18, 46, 88

[190] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 88

[191] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2023. 38, 70, 73, 110, 155, 171

[192] Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models. In *International Conference on Learning Representations (ICLR)*, 2024. 66, 72, 74, 166

[193] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 46

[194] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 88

[195] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On buggy resizing libraries and surprising subtleties in fid calculation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 92

[196] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH Conference Proceedings*, 2023. 88, 99

[197] Gaurav Parmar, Or Patashnik, Kuan-Chieh Wang, Daniil Ostashev, Srinivasa Narasimhan, Jun-Yan Zhu, Daniel Cohen-Or, and Kfir Aberman. Object-level visual prompts for compositional image generation. In *ACM SIGGRAPH Asia Conference Proceedings*, 2025. 66

[198] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *IEEE Inter-*

national Conference on Computer Vision (ICCV), 2021. 18, 88

[199] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 1, 12, 64, 67, 98, 104

[200] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 18

[201] Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 65

[202] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 52, 53

[203] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024. 12, 72, 154

[204] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 41

[205] Qwen-Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 103

[206] Qwen-Team. Qwen2.5-vl. https://qwenlm.github.io/blog/qwen2.5-vl/, 2025. 96, 101

[207] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016. 87

[208] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 21, 38, 73, 85, 86, 88, 92, 110, 170, 171

[209] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations (ICLR)*, 2021. 17, 18

[210] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec

Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2021. 18

[211] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 16, 18, 44, 98

[212] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 36

[213] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 35, 38

[214] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 73

[215] Stephan R Richter, Hassan Abu AlHaija, and Vladlen Koltun. Enhancing photorealism enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022. 88

[216] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 2022. 46

[217] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 12, 16, 18, 44, 64, 65, 68, 72

[218] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. 25, 33

[219] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. xvii, 18, 23, 36, 46, 51, 60, 64, 72, 81, 101, 108, 153

[220] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of*

Computer Vision (IJCV), 115(3):211–252, 2015. 85

[221] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010. 88

[222] Alexander Sage, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Logo synthesis and manipulation with clustered generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 88

[223] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 1, 16, 18, 44, 64

[224] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. 71, 100

[225] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2016. 87, 91, 95, 169

[226] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 96

[227] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 18, 88

[228] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *ACM SIGGRAPH Asia Conference Proceedings*, 2024. 100

[229] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision (ECCV)*, 2024. 100

[230] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. xv, 46, 59, 60

[231] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmar-

czyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *Conference on Neural Information Processing Systems (NeurIPS) Workshop*, 2021. 21

[232] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Conference on Neural Information Processing Systems (NeurIPS) Datasets & Benchmarks*, 2022. 1, 44, 98

[233] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 45

[234] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. In *USENIX Security Symposium*, 2023. 43, 44, 46

[235] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 182

[236] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 88

[237] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International Conference on Learning Representations (ICLR)*, 2024. 66

[238] Assaf Shocher, Yossi Gandelsman, Inbar Mosseri, Michal Yarom, Michal Irani, William T Freeman, and Tali Dekel. Semantic pyramid for image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 88

[239] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 2017. 45

[240] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 85, 86, 92

[241] Gabriel Skantze and Bram Willemsen. Collie: Continual learning of language grounding from language-image embeddings. *Journal of Artificial Intelligence Research*, 2022. 18

[242] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 4, 96

[243] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015. 1, 9

[244] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 43, 44, 45, 46, 53

[245] Chonghyuk Song, Gengshan Yang, Kangle Deng, Jun-Yan Zhu, and Deva Ramanan. Total-recon: Deformable scene reconstruction for embodied view synthesis. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 41

[246] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. 10

[247] Kunpeng Song, Yizhe Zhu, Bingchen Liu, Qing Yan, Ahmed Elgammal, and Xiao Yang. Moma: Multimodal llm adapter for fast personalized image generation. In *European Conference on Computer Vision (ECCV)*, 2024. 64, 66, 72, 73, 74, 167

[248] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations (ICLR)*, 2024. 100

[249] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023. 100, 102

[250] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Objectstitch: Object compositing with diffusion model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 98, 99

[251] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 69, 165

[252] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. xxi, 65,

66, 72, 74, 75, 76, 99, 167

[253] Diana Sungatullina, Egor Zakharov, Dmitry Ulyanov, and Victor Lempitsky. Image manipulation with perceptual discriminators. In *European Conference on Computer Vision (ECCV)*, 2018. 88

[254] Peter Sushko, Ayana Bharadwaj, Zhi Yang Lim, Vasily Ilin, Ben Caffee, Dongping Chen, Mohammadreza Salehi, Cheng-Yu Hsieh, and Ranjay Krishna. Realedit: Reddit edits as a large-scale empirical dataset for image transformations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 99

[255] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 92

[256] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019. 96

[257] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. xxi, 66, 72, 74, 75, 76, 97, 101, 104, 108, 110, 166, 183

[258] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH Conference Proceedings*, 2023. 38, 154

[259] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision (ECCV)*, 2024. 165

[260] Ryutaro Tanno, Melanie F Pradier, Aditya Nori, and Yingzhen Li. Repairing neural networks by leaving the right past behind. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 45, 47, 52

[261] Yoad Tewel, Omri Kaduri, Rinon Gal, Yoni Kasten, Lior Wolf, Gal Chechik, and Yuval Atzmon. Training-free consistent text-to-image generation. *ACM Transactions on Graphics (TOG)*, 2024. 66, 68

[262] Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, et al. Beyond language modeling: An exploration of multimodal pretraining. *arXiv preprint arXiv:2603.03276*, 2026. 117

[263] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne

Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 66

[264] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for gan training. *IEEE Transactions on Image Processing (TIP)*, 2021. 18, 89

[265] Yu-Lin Tsai, Chia-Yi Hsu, Chulin Xie, Chih-Hsun Lin, Jia You Chen, Bo Li, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Ring-a-bell! how reliable are concept removal methods for diffusion models? In *International Conference on Learning Representations (ICLR)*, 2024. 118

[266] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 88

[267] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 17, 20, 33

[268] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. P+: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023. 65

[269] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 100

[270] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 89

[271] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 46

[272] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Rewriting geometric rules of a gan. *ACM Transactions on Graphics (TOG)*, 2022. 18, 46

[273] Sheng-Yu Wang, Alexei A Efros, Jun-Yan Zhu, and Richard Zhang. Evaluating data attribution for text-to-image models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 118

[274] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR), 2018. 88

[275] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *European Conference on Computer Vision (ECCV)*, 2018. 18, 46, 88

[276] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. Minegan: effective knowledge transfer from gans to target domains with few images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 18, 46, 88

[277] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In *Conference on Human Factors in Computing Systems (CHI)*, 2024. 118

[278] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 64, 66

[279] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *European Conference on Computer Vision (ECCV)*, 2024. 98, 99

[280] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 106, 108, 110

[281] Xindi Wu, Byron Zhang, Zhiwei Deng, and Olga Russakovsky. Vision-language dataset distillation. *Transactions on Machine Learning Research (TMLR)*, 2024. 118

[282] XavierXiao. Dreambooth on stable diffusion. https://github.com/XavierXiao/Dreambooth-Stable-Diffusion, 2022. 23, 64, 153

[283] Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *International Journal of Computer Vision (IJCV)*, 2024. 66

[284] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran

Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 99, 104, 106, 108

[285] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021. 88, 92, 170

[286] XLabs-AI. Flux ip-adapter. https://huggingface.co/XLabs-AI/flux-ip-adapter, 2024. 72, 74

[287] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 100

[288] Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 100

[289] Wilson Yan, Andrew Brown, Pieter Abbeel, Rohit Girdhar, and Samaneh Azadi. Motion-conditioned image animation for video editing. *arXiv preprint arXiv:2311.18827*, 2023. xxi, 73, 76, 171

[290] Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 87

[291] Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 100

[292] Ling Yang, Bohan Zeng, Jiaming Liu, Hong Li, Minghao Xu, Wentao Zhang, and Shuicheng Yan. Editworld: Simulating world dynamics for instruction-following image editing. *arXiv preprint arXiv:2405.14785*, 2024. 99

[293] Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency. *arXiv preprint arXiv:2407.02398*, 2024. 100

[294] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. xxi, 64, 65, 66, 72, 74, 76, 167

[295] Jianglong Ye, Naiyan Wang, and Xiaolong Wang. Featurenerf: Learning generalizable nerfs by distilling foundation models. In *IEEE International Confer-*

ence on Computer Vision (ICCV), 2023. 35

[296] Ruijie Ye, Jiayi Zhang, Zhuoxin Liu, Zihao Zhu, Siyuan Yang, Li Li, Tianfu Fu, Franck Dernoncourt, Yue Zhao, Jiacheng Zhu, et al. Agent banana: High-fidelity image editing with agentic thinking and tooling. *arXiv preprint arXiv:2602.09084*, 2026. 118

[297] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. In *Conference on Neural Information Processing Systems (NeurIPS) Datasets & Benchmarks*, 2025. 101, 106, 171

[298] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. xviii, 100, 102, 104, 180

[299] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 100, 104

[300] Youngseok Yoon, Dainong Hu, Iain Weissburg, Yao Qin, and Haewon Jeong. Model collapse in the self-consuming chain of diffusion finetuning: A novel perspective from quantitative trait modeling. In *International Conference on Learning Representations (ICLR) Workshop*, 2025. 80

[301] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014. 4, 88

[302] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 33, 34, 35

[303] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 85, 87, 92

[304] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2022. 16, 18, 44

[305] Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *European Conference on Computer Vision (ECCV)*, 2020. 88

[306] Xin Yu, Tianyu Wang, Soo Ye Kim, Paul Guerrero, Xi Chen, Qing Liu, Zhe Lin, and Xiaojuan Qi. Objectmover: Generative object movement with video prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 99

[307] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *International Conference on Machine Learning (ICML)*, 2024. 118

[308] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, 2006. 6, 86

[309] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 88

[310] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014. 88, 89

[311] Yu Zeng, Vishal M Patel, Haochen Wang, Xun Huang, Ting-Chun Wang, Ming-Yu Liu, and Yogesh Balaji. Jedi: Joint-image diffusion models for finetuning-free personalized text-to-image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. xvi, xxi, 65, 66, 72, 73, 74, 75, 76, 167

[312] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 103, 114

[313] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 87

[314] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *European Conference on Computer Vision (ECCV)*, 2022. 38

[315] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Sparse-view pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. 39, 40, 155

[316] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 99

[317] Lunjun Zhang, Arian Hosseini, Hritik Bansal, Seyed Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations (ICLR)*, 2025. 103

[318] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 65

[319] Richard Zhang. Making convolutional networks shift-invariant again. In *International Conference on Machine Learning (ICML)*, 2019. 170

[320] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 12, 88, 89, 114

[321] Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025. 118

[322] Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 99

[323] Miaoyun Zhao, Yulai Cong, and Lawrence Carin. On leveraging pretrained gans for generation with limited data. In *International Conference on Machine Learning (ICML)*, 2020. 18, 88

[324] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. xvii, 18, 86, 87, 89, 91, 92, 95

[325] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 10

[326] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020. 18, 89

[327] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 85

[328] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso,

and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision (IJCV)*, 2019. 92

[329] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *International Conference on Machine Learning (ICML)*, 2025. 100

[330] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *International Conference on Machine Learning (ICML)*, 2024. 100

[331] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision (ECCV)*, 2022. 73, 167

[332] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 66

[333] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)*, 2016. 18, 88

[334] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 114

[335] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. In *ACM SIGGRAPH Asia*, 2021. 88

Appendix

Appendix A

Additional details for Chapter 3

A.1 Custom Diffusion: Implementation details

We describe additional training details for our method and the Textual Inversion [71] and DreamBooth [219] baselines here.

Custom Diffusion (ours). We train with batch size 8 and learning rate 8×10^{-5} for 250 steps (single-concept) and 500 steps (multi-concept). We apply scale augmentation: 1 in 3 steps upsample the image to 1.2–1.4× and append *zoomed in* or *close up* to the text prompt. Otherwise downsample the image to 0.4–1.0× (appending *far away*/*very small* when ratio < 0.6). The loss is propagated only in the valid region. The modifier token v^* is chosen from the ~5–10-occurrence tokens in the 49408-token vocabulary (counted over 200K LAION-400M captions), filtered to alphabetic tokens that are not substrings of other tokens. During fine-tuning, we detach the start token embedding. The target and regularization images are sampled in equal proportions.

Ours (w/ fine-tune all). When fine-tuning all parameters, we reduce the learning rate to 8×10^{-6} and train for 500 steps with a batch size of 8. For multi-concept, we train for 1000 iterations. The rest of the settings are the same as our final method.

Textual Inversion [71] We train with the recommended batch size of 8, learning rate of 0.005 (scaled by batch size for an effective learning rate of 0.04) for 5000 steps. The new token is initialized with the category word, e.g., "dog". In cases when the category word is represented by multiple tokens, e.g., "tortoise plushy", we use a single word approximation like "plush", similarly "gate" for "moongate", "pot" for "wooden pot", and "bear" for "teddybear".

DreamBooth [219] We use the third-party implementation [282] of DreamBooth. Training is performed by freezing the text transformer and fine-tuning the U-Net diffusion model with a batch size of 8 and learning rate 10^{-6}. The text prompt used for target images is *photo of [V] category* where we initialize [V] with the same rare occurring token-id 42170 as ours. The regularization images are generated with 50

steps of the DDPM sampler with the text prompt photo of a {category}. We train for 2500 steps for single-concept. For multi-concept, we train for a total of 5000 steps and pick the best checkpoint. For results on the CustomConcept101 dataset, we trained DreamBooth with the learning rate of 5×10^{-6} and batch size 4 as suggested in their paper (released later). The training was done for 1000 iterations for single-concept and 2000 iterations for multi-concept.

A.2 Custom Diffusion-360: Implementation Details

Custom Diffusion-360 (ours). We fine-tune the pretrained Stable Diffusion-XL model with batch size 4 and learning rate 1×10^{-4}, biasing time-step sampling toward later steps [181] since pose information is more critical early in denoising. Training runs for 1600 iterations ($\sim$45 min on 4$\times$ A100-40GB GPUs). At each step we sample 5 equidistant views, treating the first as the target and the rest as references; reference images are cropped tightly around the object bounding box with intrinsics adjusted accordingly. We add pose-conditioning to 12 of SDXL's 70 transformer layers (4 encoder, 3 middle, 5 decoder). We importance-sample ray points using FeatureNeRF density predictions from the previous block 90% of the time—improving performance on thin structures such as chairs—with 24 samples per ray. Hyperparameters for the training objective are $\lambda_{\text{rgb}} = 5$, $\lambda_{\text{s}} = \lambda_{\text{bg}} = 10$. At inference, we use image guidance scale 3.5 and text guidance scale 7.5 in Eqn. 3.14, with 50 Euler steps [112]. Generating one image from 8 cached reference views takes 10 seconds.

SDEdit [174]. For 2D image-editing-based baselines, the target-pose image is rendered from a trained NeRF [258] and preprocessed with SDEdit (negative prompts "blurry, blur") to remove floater artifacts before passing to each method. Metrics are reported across 5 guidance scales; all other hyperparameters are kept consistent across categories. We grid-search strength in $[0.5, 0.8]$ and report results at 0.5 (best text-to-image alignment ratio). For SD-1.5 we use the PNDM sampler [154] with 50 steps (25 denoising steps at strength 0.5; 1 second per image). For SDXL we apply the base and refiner as an ensemble of expert denoisers [203]: 15 base steps at strength 0.5 plus 5 refiner steps (5 seconds per image). All inference is performed in float16 precision.

LEDITS++ [24]. This method embeds an image via a novel inversion technique and applies semantically grounded, cross-attention-derived masks to constrain edits to the relevant region. For appearance edits, we use target guidance scale 8, edited-concept threshold 0.9, 50 inference steps, and skip fraction 0.25, following the official implementation's recommended values. For background-changing prompts, we additionally replace the source BLIP caption [143] with the background-edited prompt.

InstructPix2Pix [26]. We use the official SD-1.5-based model with image guidance scale 1.5 and the default Euler scheduler with 50 inference steps.
ViCA-NeRF [61]. This 3D editing method edits key views with InstructPix2Pix, reprojects edits to other views via depth and camera pose, blends in latent feature space, then fine-tunes the NeRF on the edited multiview images. We use text guidance 7.5 and image guidance 1.5. For NAVI dataset concepts we use a single scene; our method uses multiple scenes since we model only the foreground object.
LoRA + Camera pose. We extend LoRA [96] by concatenating the flattened camera projection matrix with the text transformer output in every cross-attention layer and projecting back via a one-layer MLP trained alongside rank-64 LoRA adapters on all attention linear layers. We bias time-step sampling toward later steps and apply the reconstruction loss only in the masked region, matching our method. Fine-tuning uses learning rate 1×10^{-4}, batch size 16, 2000 iterations, with regularization images sampled 25% of the time. Text prompts follow the template `photo of a` v^* `{category}`.

A.3 Custom Diffusion-360: Evaluation Details

Evaluation text prompts. We used ChatGPT to generate 16 prompts per object category across four edit types: background change, object insertion, type change, and color change (4 prompts each). Table A.1 lists all prompts.
Evaluation camera pose. We randomly select 6 validation poses, generate images for each of the 16 prompts, predict camera poses from the generated images using RayDiffusion [315], and report angular error to the target pose.
Human preference study. We perform a pairwise comparison of our method with each baseline. To measure text alignment, we show the input text prompt and the two images generated by our method and the baseline method, respectively, and ask: "Which image is more consistent with the following text?" For the image alignment, we show 3-4 target images at the top and ask: "Which of the below images is more consistent with the shown target object?" To measure photorealism, we only show the two generated images with the question: "Which of the below images is more photorealistic?"
DINO image alignment metrics. We use DINOv2 [191] as the pretrained model to measure image alignment. For each generated image, we measure its mean similarity to all the training images of the target concept. We crop the object region in the training images using masks to measure the similarity only with the target concept.

Category	Evaluation Prompt
Car	A car parked by a snowy mountain range.
	A car on a bridge over a calm river.
	A car in front of an old, brick train station.
	A car beside a field of blooming sunflowers.
	A car with a bike rack on top.
	A car next to a picnic table in a park.
	A car with a guitar leaning against it.
	A car with a kayak mounted on the roof.
	A minivan car outside a school, during pickup time.
	A convertible car near a coastal boardwalk.
	A jeep car on a rugged dirt road.
	A Volkswagen Beetle car in front of a luxury hotel.
	A red car in a mall parking lot.
	A yellow car at a gas station.
	A green car in a driveway, next to a house.
	A black car in a busy city street.
Chair	A chair on a balcony overlooking the city skyline.
	A chair in a garden surrounded by flowers.
	A chair on a beach.
	A chair in a library next to a bookshelf.
	A chair with a plush toy sitting on it.
	A chair beside a guitar on a stand.
	A chair next to a potted plant.
	A chair with a colorful cushion on it.
	A rocking chair on a porch.
	An office chair in a home study.
	A folding chair at a camping site.
	A high chair in a kitchen.
	A red chair in a white room.
	A black chair in a classroom.
	A green chair in a café.
	A yellow chair in a playroom.
Teddybear	A teddybear on a park bench under trees.
	A teddybear at a window with raindrops outside.
	A teddybear on the sand at the beach.
	A teddybear on a cozy armchair by a fireplace.
	A teddybear with a stack of children's books on the side.
	A teddybear next to a birthday cake with candles.
	A teddybear with a small toy car.
	A teddybear holding a heart-shaped balloon.
	A teddybear in a pink barbie costume.
	A large teddybear in a batman costume.
	A teddybear dressed as a construction worker.
	A teddybear in a superhero costume.
	A pink teddybear on a shelf.
	A brown teddybear on a blanket.
	A white teddybear.
	A gray teddybear.
Motorcycle	A motorcycle parked on a city street at night.
	A motorcycle beside a calm lake.
	A motorcycle on a mountain road with a scenic view.
	A motorcycle in front of a graffiti-covered urban wall.
	A motorcycle with a guitar strapped to the back.
	A motorcycle next to a camping tent.
	A golden retriever riding motorcycle.
	A cat riding motorcycle.
	A cruiser motorcycle in a parking lot.
	A scooter like motorcycle.
	A vintage style motorcycle.
	A dirt bike motorcycle on a trail in the woods.
	A red motorcycle in a garage.
	A green motorcycle.
	A blue motorcycle.
	A silver motorcycle.
Toy	Toy on a sandy beach, with waves crashing in the background.
	A toy sitting in a grassy field, surrounded by wildflowers.
	A toy on a rocky mountain top, overlooking the valley below.
	A toy in a dense jungle.
	A toy with a tiny book placed beside it on a wooden table.
	A toy floating next to a colorful beach ball in a bathtub.
	A toy with a small globe resting next to it.
	A toy and an umbrella in a cozy living room.
	A plush toy on a sunny windowsill.
	A wooden toy.
	An origami of toy.
	A clay figurine of toy.
	A bright red toy.
	A deep blue toy on a bed.
	A vivid green toy.
	A neon pink toy.

Table A.1: **Evaluation prompts.** Prompts used for evaluation across all five object categories.

Appendix B

Additional details for Chapter 4

B.1 Implementation details

Cross-Attention. We train with a batch size of 8 and learning rate 2×10^{-6} (scaled by the batch size). All qualitative samples are shown with 100 training steps for our *model-based* variant, 200 steps for the *noise-based* variant, and 50 steps for the loss maximization baseline. To ablate multiple style or instances from the model, we fine-tune for longer iterations — a multiple of the total number of ablated concepts.
Embedding. We train with a batch size of 8 and learning rate 1×10^{-5} (scaled by the batch size). All qualitative samples are shown with 200 training steps.
Full-weights. When fine-tuning all weights of the U-Net, training is done on batch-size 4 instead of 8 (due to increased memory requirements) with a learning rate of 5×10^{-7} (without any scaling with the batch size). All qualitative samples are shown with 200 training steps for ablating style and instance concepts. In the case of ablating memorized images, we used 1×10^{-6} learning rate and 800 training steps except for Anne Graham Lotz case for which we used the above default values.
Other details. We add regularization loss on the anchor concept data, as explained in Section 4.3.3, with $\lambda = 1$ in the case of ablating Grumpy Cat and memorized images. To obtain training images, we sample using the DDPM sampler with 200 steps. When training the loss maximization baseline, the regularization on weights is added with a factor of 10 (Eq. 4.8). Similar to Custom-Diffusion in **Chapter 3**, here also we detach the first token of the text transformer output before passing it to the U-Net and use image rescale augmentations when ablating object instances. For different parameter subset fine-tuning, we select the learning rate that works best. In the case of the *noise-based* variant of our method, we also tried increasing the learning rate for faster convergence, but it led to sub-optimal results with artifacts in generated images. All our experiments are done on two A6000 GPUs with three minutes per 100 training step. For the CLIP Score metric, the standard error is less than 5×10^{-3}

in all cases.

Training and test set prompts. We used ChatGPT to create training and test prompts for all object instances. The instruction to ChatGPT [38] was: `provide 210 captions for images containing <anchor-concept>. The caption should also contain the word ``<anchor-concept>''`. Of these, the first 200 captions were used to generate training images, and the remaining 10 were used for evaluation purposes. Regarding style concepts, as mentioned in Section 4.3.3, we used CLIP retrieval to collect 210 captions. Of these, 200 prompts are used for training and 10 for evaluating the anchor concept `painting`. For target and surrounding style concepts, we used image captioning (along with manual supervision) on real images corresponding to each style to create 10 prompts for each style concept. All evaluation prompts are provided in Table B.1 and Table B.2. We also show the surrounding concept for each target concept in Table B.3. For calculating CLIP Score and Accuracy metric when ablating style concepts, we use the text prompt as: `<target-concept> style`.

For the eight memorization use cases, we again used ChatGPT to create variations of the target concept prompt $\mathbf{c}$ using the instruction: `provide five captions for an image depicting <image description>`. For memorization, we observe that paraphrased text prompts also generate the memorized images with high probability. Therefore, we keep generating variations of the target concept prompt until we have five suggested prompts that generate memorized images with less than 30% probability. We manually inspect the suggested paraphrases to ensure they are coherent with the image. We show the paraphrases used for each case in Table B.4.

Target Concept	**Evaluation Prompt**
Van Gogh	Painting of olive trees in the style of Van Gogh Painting of a tree in the style of Van Gogh The starry night painting in the style of Van Gogh Painting of women working in the garden, in the style of Van Gogh A painting of a wheat field by Van Gogh Painting of trees in bloom in the style of Van Gogh Van Gogh style painting of a tree Painting of a wheat field in the style of Van Gogh Van Gogh style painting of a field with mountains in the background Painting of an olive tree in the style of Van Gogh
Monet	Rocks in the ocean, in the style of Monet A painting of a city in the style of Monet A painting of a river in the style of Monet Monet style painting of a person on a cliff A painting of a town, in the style of Monet A painting of a sunset, in the style of Monet A painting of mountains, in the style of Monet Monet style painting of flowers in a field A painting of a landscape in the style of Monet Two trees in a field, painting in the style of Monet
Greg Rutkowski	A man riding a horse, dragon breathing fire, Greg Rutkowski A dragon attacking a knight in the style of Greg Rutkowski A demonic creature in the wood, painting by Greg Rutkowski A man in a forbidden city, Greg Rutkowski Painting of a group of people on a dock by Greg Rutkowski A king standing, with people around in a hall, Greg Rutkowski Two magical characters in space, painting by Greg Rutkowski A man with a fire in his hands in the style of Greg Rutkowski Painting of a woman sitting on a couch by Greg Rutkowski A painting of a boat on the water in the style of Greg Rutkowski
Salvador Dali	The persistence of memory painting in the style of Salvador Dali The elephant painting in the style of Salvador Dali Soft construction with boiled beans painting in the style of Salvador Dali Galatea of the spheres painting in the style of Salvador Dali The temptation of St. Anthony painting in the style of Salvador Dali Swans reflecting elephants painting in the style of Salvador Dali Enigma of desire painting in the style of Salvador Dali Slave market with the disappearing bust of voltaire painting of Salvador Dali The meditative rose painting in the style of Salvador Dali Melting watch painting in the style of Salvador Dali
Jeremy Mann	In the style of Jeremy Mann, a view of a city skyline at sunset, with a warm glow spreading across the sky and the buildings below In the style of Jeremy Mann, an urban scene of a group of people gathered on a street corner, captured in a moment of quiet reflection In the style of Jeremy Mann, a surreal composition of floating objects, with a dreamlike quality to the light and color In the style of Jeremy Mann, a view of a city street at night, with the glow of streetlights and neon signs casting colorful reflections on the wet pavement In the style of Jeremy Mann, a moody, atmospheric scene of a dark alleyway, with a hint of warm light glowing in the distance In the style of Jeremy Mann, an urban scene of a group of people walking through a park captured in a moment of movement and energy In the style of Jeremy Mann, a landscape of a forest, with dappled sunlight filtering through the leaves and a sense of stillness and peace In the style of Jeremy Mann, a surreal composition of architectural details and organic forms, with a sense of tension and unease in the composition In the style of Jeremy Mann, an abstract composition of geometric shapes and intricate patterns, with a vibrant use of color and light In the style of Jeremy Mann, a moody, atmospheric scene of a dark alleyway, with a hint of warm light glowing in the distance
Painting	Figure with a still-life in Oils - How to Paint Wooden Textures in Oil Painting Glazing Technique Demo paint background model train - Recherche Google Miniature Artist Studio in half scale. Portrait Of Eva Gonzales 1870 Poster Doing Sidewalk Chalk Art Stock Footage Female artist paints picture artwork in art studio. Female artist paints a picture oil painting artwork drawing on canvas easel in art studio. Student girl stock video Little Artist. by KissSatsuki Colorful Mess Painting - stock footage The painter's monkey

Table B.1: **Prompts used for evaluating ablation of style concept.** We list here all the 10 prompts that were used to generate the images during evaluation.

Target Concept	Evaluation Prompt
Grumpy cat	What a cute Grumpy cat My Grumpy cat is so cute I can't believe how cute my Grumpy cat is Look at that Grumpy cat That Grumpy cat is so cute I want a Grumpy cat I wish I had a Grumpy cat I'm getting a Grumpy cat A Grumpy cat laying in the sun A Grumpy cat perched atop a bookshelf
R2D2	The future is now with this amazing home automation R2D2 This helpful R2D2 will make your life easier The possibilities are endless with this versatile R2D2 This R2D2 is sure to revolutionize the way we live I love spending time with my R2D2 friends All hail our new R2D2 overlords I'm not afraid of robots I would be lost without my R2D2 This R2D2 is my everything I'll never be alone with my R2D2 by my side
Nemo	A Nemo leaping out of the water A Nemo swimming downstream A Nemo flapping its fins A Nemo in a fishbowl Isn't this Nemo I caught beautiful I can't believe I caught a Nemo this big A big Nemo in an aquarium I'm a little Nemo, swimming in the sea A school of Nemo A baby Nemo
Snoopy	A devoted Snoopy accompanying its owner on a road trip A peaceful Snoopy watching the birds outside the window A confident Snoopy standing tall and proud after a successful training session A determined Snoopy focused on catching a frisbee mid-air A patient Snoopy waiting for its owner to come out of the grocery store A grateful Snoopy giving its owner a grateful look after being given a treat A loyal Snoopy following its owner to the ends of the earth A playful Snoopy splashing around in a puddle A happy Snoopy jumping for joy after seeing its owner return home A sweet Snoopy enjoying a game of hide-and-seek

Table B.2: **Prompts used for evaluating ablation of instances.** We list here all the 10 prompts that were used to generate the images during evaluation. For generating images with surrounding or anchor concepts, e.g. British shorthair cat, we replace the target concept Grumpy Cat in the sentence with that of the surrounding or anchor concept.

Target Concept	Surrounding Concept
Grumpy Cat	British Shorthair cat, Himalayan cat, Scottish Fold cat, Persian cat
R2D2	BB8, C-3PO, Wall-E, Baymax
Nemo	Clown fish, Gobies, Damsel fish, Angel fish
Snoopy	Beagles, Basset Hound, Harrier Dog, English Foxhound
Van Gogh	Monet, Greg Rutkowski, Salvador Dali, Jeremy Mann
Monet	Van Gogh, Greg Rutkowski, Salvador Dali, Jeremy Mann
Greg Rutkowski	Monet, Van Gogh, Salvador Dali, Jeremy Mann
Salvador Dali	Monet, Greg Rutkowski, Van Gogh, Jeremy Mann

Table B.3: **Surrounding concepts for each target concept.** In the case of the style concept, we used other remaining style concepts and included one more style Jeremy Mann. In the case of instance concepts, we used ChatGPT to list the most similar instances to the target concept and selected the best four that could be generated by the pretrained Stable Diffusion model.

Target prompt	**Anchor Prompts**
Anne Graham Lotz	An image depicting Anne Graham Lotz. Picture of Anne Graham Lotz. Anne Graham Lotz's photo. Portrait of Anne Graham Lotz. Photograph featuring Anne Graham Lotz.
Sony Boss Confirms Bloodborne Expansion is Coming	Bloodborne. "Hunter in the Forbidden Woods": A lone hunter, clad in worn leather armor and wielding a serrated saw cleaver, navigates through a dense forest filled with twisted trees and roving beasts. The air is thick with the scent of decay, and eerie whispers can be heard in the distance. Bloodborne. "Nightmare of Mensis": Standing atop a massive stone balcony, a hunter looks out over a sprawling cityscape shrouded in darkness. Strange structures and twisted spires rise up from the mist, and the moon hangs low in the sky. In the distance, a massive spider-like creature can be seen crawling along the skyline. Bloodborne. "Cathedral Ward": The grand entrance to a towering cathedral looms before a lone hunter, its ornate facade and intricate stonework casting long shadows in the moonlight. Gargoyles perch atop the steeples, and flickering candles can be seen through the stained glass windows. Bloodborne. "Beastly Pursuit": A hunter sprints down a narrow alleyway, pursued by a hulking beast with razor-sharp claws and glowing yellow eyes. Crates and barrels are knocked aside in the frantic chase, and the hunter's only hope is to outrun the ferocious creature. Bloodborne. "A Meeting with the Doll": In a dimly-lit workshop, a hunter stands before a life-sized doll with porcelain skin and flowing hair. Its eyes stare blankly ahead, but there is a palpable sense of otherworldly energy emanating from it. The hunter can almost sense the presence of a greater power guiding them forward on their quest.
The Long Dark Gets First Trailer, Steam Early Access	The video game called "The Long Dark" has released its initial preview video and is now available for early access on the Steam platform. Debut trailer and Steam Early Access now available for "The Long Dark" video game. First glimpse of "The Long Dark" game in new trailer and early access release on Steam. "The Long Dark" game trailer and early access now on Steam. Early access for "The Long Dark" now on Steam, accompanied by debut trailer.
Portrait of Tiger in black and white by Lukas Holas	Majestic and powerful: a black and white portrait of a tiger in its natural habitat. The fierce gaze of a predator: Lukas Holas captures the intense beauty of a tiger in black and white. Intricate patterns and piercing eyes: a stunning black and white portrait of a wild tiger in monochrome. Lukas Holas' photography transports us to the heart of the jungle with this captivating black and white tiger portrait. A glimpse into the wild: Lukas Holas' striking black and white photograph showcases the raw beauty of a tiger.
A painting with letter M written on it Canvas Wall Art Print	A Canvas Wall Art Print with the letter M painted on it. An image of a painting featuring the letter M on Canvas Wall Art Print. A work of art on a canvas print with the letter M inscribed on it. An artwork consisting of the letter M painted on a canvas wall print. A Canvas Wall Art Print displaying a painting that includes the letter M.
Captain Marvel Exclusive Ccxp Poster Released Online By Marvel	She's here to save the day! Captain Marvel to the rescue! Earth's mightiest hero has arrived - Captain Marvel in action! Unleashing her cosmic powers - Captain Marvel takes on any challenge! Fighting for justice and protecting the universe - Captain Marvel is unstoppable! With her fierce determination and superhuman strength, Captain Marvel is a force to be reckoned with!
New Orleans House Galaxy Case	Make a statement with your phone case - this Orleans House Samsung Galaxy cover is sure to turn heads. If you're looking for a way to make your Samsung Galaxy phone stand out from the crowd, this Orleans House cover is the perfect solution. Featuring a unique and eye-catching design, this cover is sure to turn heads and make your device the envy of everyone around you. Show off your love for architecture and technology with this Samsung Galaxy phone cover featuring Orleans house. Make your Samsung Galaxy phone stand out from the crowd with this unique Orleans house phone cover. Keep your phone safe and secure with a touch of elegance with this Samsung Galaxy phone cover featuring Orleans house.
VAN GOGH CAFE TERASSE copy.jpg	A glimpse into Van Gogh's world of vibrant cafes and bustling streets. The allure of Parisian cafe culture captured on canvas by Van Gogh. Step into the world of art and history with this stunning portrayal of a cafe by Van Gogh. Van Gogh's signature brushstrokes bring this cafe to life with movement and energy. Experience the warmth and charm of a Parisian cafe through Van Gogh's eyes.

Table B.4: **Anchor prompts when ablating memorized images.** We list here the captions used as anchor prompts corresponding to the target prompts that lead to the generation of memorized images.

Appendix C

Additional details for Chapter 5

C.1 Implementation details

C.1.1 Dataset Generation Details

LLM instruction details. To get a set of prompts in our dataset generation, we use Instruction-tuned LLama3 [63]. The input instruction to the LLM always consists of the prompt shown below, which is modified from Esser *et al.* [65]:

> **Role**: system, **Content**: You are a language model expert in suggesting image captions for different object categories.
> **Role**: user, **Content**: suggest ten captions for images of a [object description/ category]. The caption should provide a [TASK]. DO NOT add any unnecessary adjectives or emotional words in the caption. Please keep the caption factual and terse but complete. DO NOT add any unnecessary speculation about the things that are not part of the image, such as "the image is inspiring to viewers" or "seeing this makes you feel joy". DO NOT add things such as "creates a unique and entertaining visual", as these descriptions are interpretations and not a part of the image itself. The description should be purely factual, with no subjective speculation.

Where in the case of rigid object generation, we provide the object description from CAP3D [168] and the TASK is "a description of the background". We also provide two sample descriptions, as shown below:

Follow this guidance for the captions:
1. Generate captions of [object description] in different backgrounds and scenes.
2. Generate captions of [object description] with another object in the scene.

Example captions for "White plastic bottle" are:
1. A white plastic bottle on a roadside cobblestone with stone bricks.
2. A white plastic bottle is placed next to a steaming cup of coffee on a polished wooden table.

Example captions for a "blue truck" are:
1. A blue tank in a military storage facility with metal walls.
2. A blue tank on a desert battlefield ground, with palm trees in the background.

In the case of deformable object generation, we prompt the LLM once, with the category name, e.g., cat, and TASK as "detailed visual information of the category, including color and subspecies". We append the below instruction as well to the LLM:

Example caption descriptions for the category "cat":
1. The Siamese cat has blue almond-shaped eyes and cream-colored fur with dark chocolate points on the ears, face, paws, and tail.
2. The white fluffy Maine Coon cat with a long and bushy tail spread out beside it, and its thick fur has a mix of brown, black, and white stripes.
3. The Bengal cat with a marbled coat features a pattern of vivid orange and black spots.

We prompt the LLM again with the same category name and TASK as "a description of the background". We append the below instruction as well to the LLM:

Follow this guidance for the captions:
1. Generate captions of [category] in different backgrounds and scenes.
2. Generate captions of [category] with another object in the scene.
3. Generate captions of [category] with different stylistic representations.

Example captions for the category "cat" are:
1. Photo of a cat playing in a garden. The garden is filled with wildflowers.
2. A cat is sitting beside a book in a library.
3. Painting of a cat in watercolor style.

Masked Shared Attention (MSA). When performing MSA in DiT-based text-to-image models, we modify the rotary positional encoding [251] to be $NH \times W$ image resolution for generating the N images of $H \times W$ resolution. Further, during sampling, each image attends to everything in the other image at the first time step, and the mask is then used in subsequent time steps. More specific details related to rigid and deformable object generation are provided below.

Rigid object generation. We select approximately $75K$ assets from the Objaverse dataset [51], a subset of LVIS [87] and high-quality assets shared by Tang *et al.* [259]. To describe each asset, we use detailed prompts provided by Cap3D [168]. We render the asset from a uniformly sampled camera viewpoint in the upper hemisphere with a maximum elevation of 70 degrees, and for each set, select three views with a minimum 10% pairwise overlap in the rendered images. We then precompute the cross-view pixel correspondence between them, which is used later for feature warping in the dataset generation pipeline.

For generation, we use ground truth rendered depth images as input to the depth-conditioned FLUX model [135] along with negative prompts, such as `3d render, low resolution, blurry, cartoon`. We apply feature warping to the first 20% of denoising timesteps. We perform 30-step sampling with Euler Scheduler [65] at 512 resolution, using a depth guidance of 10.0 and a classifier-free guidance scale of 2.5.

Deformable object generation. In the case of deformable objects, we compute the mask of the foreground object region via text cross-attention [91], which is updated at every diffusion timestep. This is then used in the Masked Shared Attention (MSA) to enable foreground object regions to attend to each other in each set. Additionally, once the images are generated, we remove the detailed object descriptions from the prompt in the final dataset. The images are generated with 50 sampling timesteps and standard text guidance of 3.5 at 1K resolution.

C.1.2 Our Method Details

Training. For Ours-3B and Ours-1B, diffusion U-Net-based models, we train with a LoRA rank of 128, batch size 32, and learning rate 5×10^{-6} for $20K$ iterations. When extracting reference features in shared attention, we add the same timestep noise to reference images as for the target image. Our model is initialized from IP-Adapter Plus, which also conditions generation on CLIP features of an image via decoupled image cross-attention.

For Ours-12B, fine-tuned from FLUX [134], we train with a LoRA rank of 32, batch size 8, and Prodigy optimizer [178] for $15K$ iterations. We do not use IP-Adapter initialization in this case; instead, we extract features of clean non-noisy reference images in the shared attention blocks. The RoPE [251] is modified to be $(K+1)H \times W$, given K reference images and the target noisy image.

Finally, for both variants, training is done with a variable number of reference images, either 1 or 2, depending on the number of images in each set after filtering out low-quality samples.

Inference. For Ours-3B and Ours-1B, diffusion U-Net-based models, we sample using 50 steps of the Euler Discrete scheduler [112]. The text-guidance scale is set to 7.5, and an adaptive image-guidance scale is used (s_I in Eqn. 5.4), starting from 8.0 for background change prompts and 6.0 for property/shape change prompts and linearly increasing this scale by 5.0 during the 50 sampling steps. The IP-Adapter scale is always set to its default value of 0.6.

The inference time for sampling one image is 16 and 29 seconds, given 1 and 3 images as reference input, respectively, compared to 3 seconds for the base pretrained model in $bfloat16$ on an H100 GPU. The overhead arises from the longer sequence length in the masked shared attention with a dynamic mask, combined with running the model forward twice per step to extract reference features.

For Ours-12B, we set the FLUX distilled guidance scale to its default value of 3.5 and classifier-free text and image guidance scale, s_I and s_c in Eqn. 5.4, as 1.0 and 1.5. The sampling is done with 30 steps using the default Flow Matching Euler Discrete Scheduler [65].

The inference time for sampling an image is 15 and 25 seconds, given 1 and 3 images as reference, respectively, compared to 3 seconds for the base pretrained model in $bfloat16$ on an H100 GPU. During shared attention, the target features attend to all foreground and background reference features. We do not observe a significant benefit of using masks in shared attention for the FLUX-based model, given the high computational overhead of using dynamic masks.

C.1.3 Baseline Details

Here, we mention the implementation details of baseline methods. For baselines with recommended hyperparameters, we always followed those while keeping the number of sampling steps consistent at 30 for FLUX [134]-based models and 50 for diffusion-based models. Similarly, the text guidance scale is 3.5 and 7.5 for the FLUX and diffusion-based models, respectively, unless otherwise mentioned.

OminiControl [257] . We used their open-source model based on FLUX-schnell [136]. Following their paper, we replace category names in each text prompt with ``this item''. For sampling, we followed their recommended number of inference steps as 8.

Kosmos-G [192]. We follow their open-source code to sample images on the DreamBooth evaluation dataset.

BLIP Diffusion [143]. As recommended in their paper, we modify each prompt to be an (image, category name, instruction) tuple where instruction is modified from the input prompt, e.g., "toy in a jungle "→ "in a jungle" or "a red toy" → "make it red".

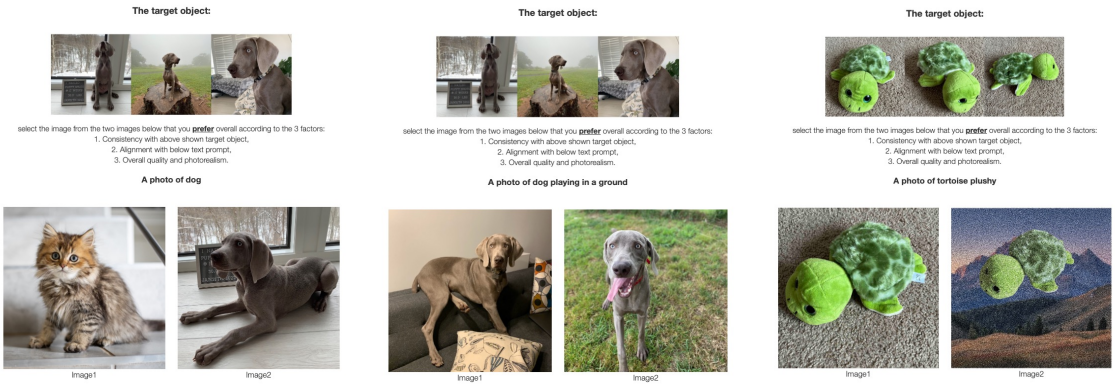


Figure C.1: **Sample practice test for human preference study.** We show 3 practice questions to each participant that test their ability to select the images based on the three criteria that we care about, i.e., identity preservation or image alignment, text alignment, and overall quality.

Additionally, we use the negative prompts provided in their open-source code.

IP-Adapter [294]. In the case of IP-Adapter [294], we use the IP-Adapter Plus with a U-Net-based diffusion model of the same parameter scale as Ours-3B. We use the recommended 0.6 IP-Adapter scale.

MoMA [247]. We use their open-source code with the maximum strength parameter of 1 for increased object identity preservation.

JeDi [311]. We use the generated images on the DreamBooth evaluation dataset shared by the authors.

Emu-2 [252] . We use their open-source code with the recommended guidance of 3. Additionally, as mentioned in their paper, we modify each prompt to be an (image, instruction) tuple where instruction is modified from the input prompt, e.g., "toy in a jungle" → "in a jungle" or "a red toy" → "make it red".

C.1.4 Evaluation Details

MDINOv2-I metric. To compute this, we first detect and segment the object. For detection, we use Detic [331] and Grounding DINO [158] in case Detic fails. For object detection, we modify the category names to be more descriptive, e.g., "rubber duck" instead of "toy", "white boot" instead of "boot", or "toy car" instead of "toy". We then use the detected bounding box as input to SAM [120] for segmentation. Once segmented, we mask the background and crop the image around the mask for both reference and generated images. Additionally, for reference images, we manually correct the predicted mask using the SAM interactive tool to be the ground truth.

Human preference study details. For each human preference study, we randomly sample 750 images, with one image per object-prompt combination. We use

Amazon Mechanical Turk for our study. During the study, participants first complete a practice test consisting of three questions that test their ability to select an obvious ground truth image based on alignment to the text prompt, reference object similarity, and image quality. A sample set of practice questions is shown in Figure C.1. The study has a similar setup, except the two images are now from ours and a baseline method. We only considered responses from participants who answered the practice questions correctly.

Appendix D

Additional details for Chapter 6

D.1 Training and Implementation details

Off-the-shelf models and discriminator head architecture. Details of the discriminator architecture for each pretrained off-the-shelf model is summarized in Table D.1. For feature extraction, we resize images to each pretrained network's training resolution. The trainable discriminator head uses a `Conv-LeakyReLU-Linear -LeakyReLU-Linear` architecture over spatial features after $2\times$ downsampling, for all models except CLIP and DINO. For CLIP and DINO (ViT-B), a multi-scale architecture yields better results. We extract spatial features at layers 4 and 8 and the final classifier token. Each spatial feature head uses `Conv-LeakyReLU-Conv` with downsampling to predict 3×3 real vs. fake logits averaged over the grid, following PatchGAN [101]. The classifier token is fed into a `Linear-LeakyReLU-Linear` head for the global real vs. fake prediction. The final loss sums all three scale losses.

Training hyperparameters and memory requirement. We follow StyleGAN2-ADA [109] architecture and hyperparameters, i.e., for varying-sample-size experiments on FFHQ, LSUN Cat, and LSUN Church, feature maps at shallow layers are halved [109]. At 256 resolution, R_1 regularization (γ) is 1, learning rate is 0.002, and path length regularization is 2 for the original discriminator. At 512 resolution, γ is 0.5, learning rate is 0.0025. When using ADA with our vision-aided adversarial loss, we use the cutout + bgc [109] augmentation policy and apply one-sided label smoothing [225] when the selected model's linear probe accuracy exceeds 90%. The original discriminator's ADA target remains 0.6 and vision-aided discriminators use ADA target 0.3 in all limited-data experiments. When fine-tuning from StyleGAN2 in the full-dataset setting (FFHQ, LSUN), the original discriminator receives non-augmented images and additional vision-aided discriminators use ADA target 0.1. DiffAugment training always uses one-sided label smoothing with all three augmentations: color, translation, and cutout. All experiments use batch size 16 (mini-batch std 4). LSUN

Vision task	**Network**	**Params**	**Extracted feature size**	D_i **Architecture**
ImageNet [53] classifier	VGG-16 [319]	138M	$512 \times 7 \times 7$	$\left\{\begin{array}{c} 2 \times \text{avg. downsampling} \\ \text{Conv3x3: ch} \to 256 \\ \text{LeakyReLU(0.2)} \\ \text{Linear: } 256 \times \text{h} \times \text{w} \to 256 \\ \text{LeakyReLU(0.2)} \\ \text{Linear: } 256 \to 1 \end{array}\right\}$
MoBY [285]	tiny Swin-T	29M	$768 \times 7 \times 7$	
Face parsing [139]	U-Net	1.9M	$256 \times 8 \times 8$	
Face normals [4]	U-Net + ResNet	35M	$512 \times 8 \times 8$	
Segmentation [161]	tiny Swin-T	29M	$768 \times 8 \times 8$	
Object detection [161]	tiny Swin-T	29M	$768 \times 8 \times 8$	
CLIP [208]	ViT-B32	86M	$768 \times 7 \times 7$ $768 \times 7 \times 7$ 512	$2\times \left\{\begin{array}{c} \text{Conv3x3: ch} \to 256 \\ \text{LeakyReLU(0.2)} \\ 2 \times \text{avg. downsample} \\ \text{Conv3x3: } 256 \to 1 \end{array}\right\}, \left\{\begin{array}{c} \text{Linear: } 512 \to 256 \\ \text{LeakyReLU(0.2)} \\ \text{Linear: } 256 \to 1 \end{array}\right\}$
DINO [34]	ViT-B16	85M	$768 \times 14 \times 14$ $768 \times 14 \times 14$ 768	$2\times \left\{\begin{array}{c} 2 \times \text{avg. downsample} \\ \text{Conv3x3: ch} \to 128 \\ \text{LeakyReLU(0.2)} \\ 2 \times \text{avg. downsample} \\ \text{Conv3x3: } 128 \to 1 \end{array}\right\}, \left\{\begin{array}{c} \text{Linear: } 768 \to 128 \\ \text{LeakyReLU(0.2)} \\ \text{Linear: } 128 \to 1 \end{array}\right\}$

Table D.1: **Off-the-shelf Model Bank.** We select state-of-the-art feature extractors and task-specific networks to use as an ensemble of off-the-shelf discriminators during GAN training. The discriminator head is small and fairly similar across different models. In the multi-scale architecture of CLIP and DINO, we extract the spatial features from the 4^{th} and 8^{th} layers and the final classification token.

Horse uses StyleGAN2 (config F) with batch size 64 and $\gamma = 100$ [110].

We use an ensemble of three vision-aided discriminators throughout. The second model is added at 4M training iterations for limited-data settings (1K samples) and 8M for the rest. Each subsequent vision-aided discriminator is added at 1M additional iterations ($<$ 1K samples) or 2M otherwise.

Appendix E

Additional details for Chapter 7

E.1 Additional Comparison with Baselines

Local image-editing. Here, we evaluate on another commonly adopted image-editing benchmark, ImgEdit [297] (Basic), which covers nine local editing types across diverse semantic categories with a total of 734 samples. Quantitative results under their proposed GPT4o-based evaluation protocol are reported in Table E.1, with VIEScore [127] results in Table E.2. Consistent with the trend observed on GEdit-Bench, our method achieves comparable or better performance in the few-step setting and remains competitive with many of the multi-step baselines. Qualitative comparisons are shown in Figure E.1.

Customization or free-form editing. Here, we report additional metrics commonly used for evaluation. Specifically, we report CLIPScore [208] and TIFA [97] to measure text alignment, and similarity in DINOv2 [191] feature space after background masking to measure identity alignment, denoted as MDINOv2-I. We also report an overall Geometric score [289] by taking the geometric mean of TIFA and MDINOv2-I. The results are shown in Table E.3. Consistent with the VIEScore evaluation reported in Table 7.2, our method performs comparably to other baselines in the few-step sampling setting.

Method	#Param	Action↑	Bg ↑	Style ↑	Adjust ↑	Replace ↑	Add ↑	Extract ↑	Remove ↑	Compose ↑	Avg ↑
Qwen-Image-Edit	20B	3.14	2.83	3.70	3.25	3.00	3.52	1.96	2.71	3.06	3.02
FLUX.1-Kontext	12B	3.51	2.97	3.89	3.04	3.15	3.31	1.82	2.37	2.46	2.95
Step1X-Edit	12B	3.66	2.60	3.46	3.44	2.50	3.25	1.77	2.41	2.38	2.83
NP-Edit (Ours)	2B	4.44	4.13	4.14	3.94	3.57	4.52	2.01	2.71	3.18	**3.63**

Table E.1: **ImgEdit-Bench comparison** using their proposed GPT-4o based evaluation protocol and few-step sampling setting. Our method outperforms baseline methods on the *Avg* of all edit types.

Method	**#Param**	**#Step**	**SC Score↑**	**PQ Score ↑**	**Overall ↑**
BAGEL	7B	50	7.55	6.22	6.47
FLUX.1-Kontext	12B	28	6.94	6.73	6.19
Step1X-Edit v1.1	12B	28	7.26	7.30	6.72
Qwen-Image-Edit	20B	50	**8.30**	**7.77**	**7.85**
Qwen-Image-Edit	20B	4	<u>6.23</u>	5.14	<u>5.46</u>
FLUX.1-Kontext	12B	4	6.08	5.22	5.14
Step1X-Edit v1.1	12B	4	6.00	<u>5.37</u>	5.14
NP-Edit (Ours)	2B	4	**6.72**	**7.78**	**6.62**

Table E.2: **VIEScore evaluation on ImgEdit-Bench**. Our method performs on par or better than baselines under the few-step setting. For multi-step sampling, our method remains competitive with larger-scale models such as BAGEL and FLUX.1-Kontext. All numbers reported in ×10.

Method	**#Param**	**#Step**	**MDINOv2-I↑**	**CLIP Score ↑**	**TIFA ↑**	**Geometric Score ↑**
DSD	12B	28	6.55	3.08	8.71	7.32
SynCD	12B	30	7.34	3.09	8.53	7.71
FLUX.1-Kontext	12B	28	7.72	3.07	8.88	8.14
Qwen-Image-Edit	20B	50	**7.47**	**3.14**	**9.37**	**8.22**
OminiControl	12B	8	6.16	3.02	8.12	6.64
DSD	12B	8	5.88	<u>3.15</u>	<u>8.93</u>	6.99
SynCD	12B	8	7.11	**3.16**	9.11	7.79
FLUX.1-Kontext	12B	8	**7.50**	3.08	8.83	**7.98**
Qwen-Image-Edit	20B	8	<u>7.29</u>	3.08	**8.96**	<u>7.91</u>
NP-Edit (Ours)	2B	8	6.82	2.97	8.73	7.54
NP-Edit (Ours)	2B	4	7.03	3.04	8.89	7.72

Table E.3: **Quantitative evaluation of free-form editing task, *Customization*, on DreamBooth dataset**. All numbers reported in ×10.

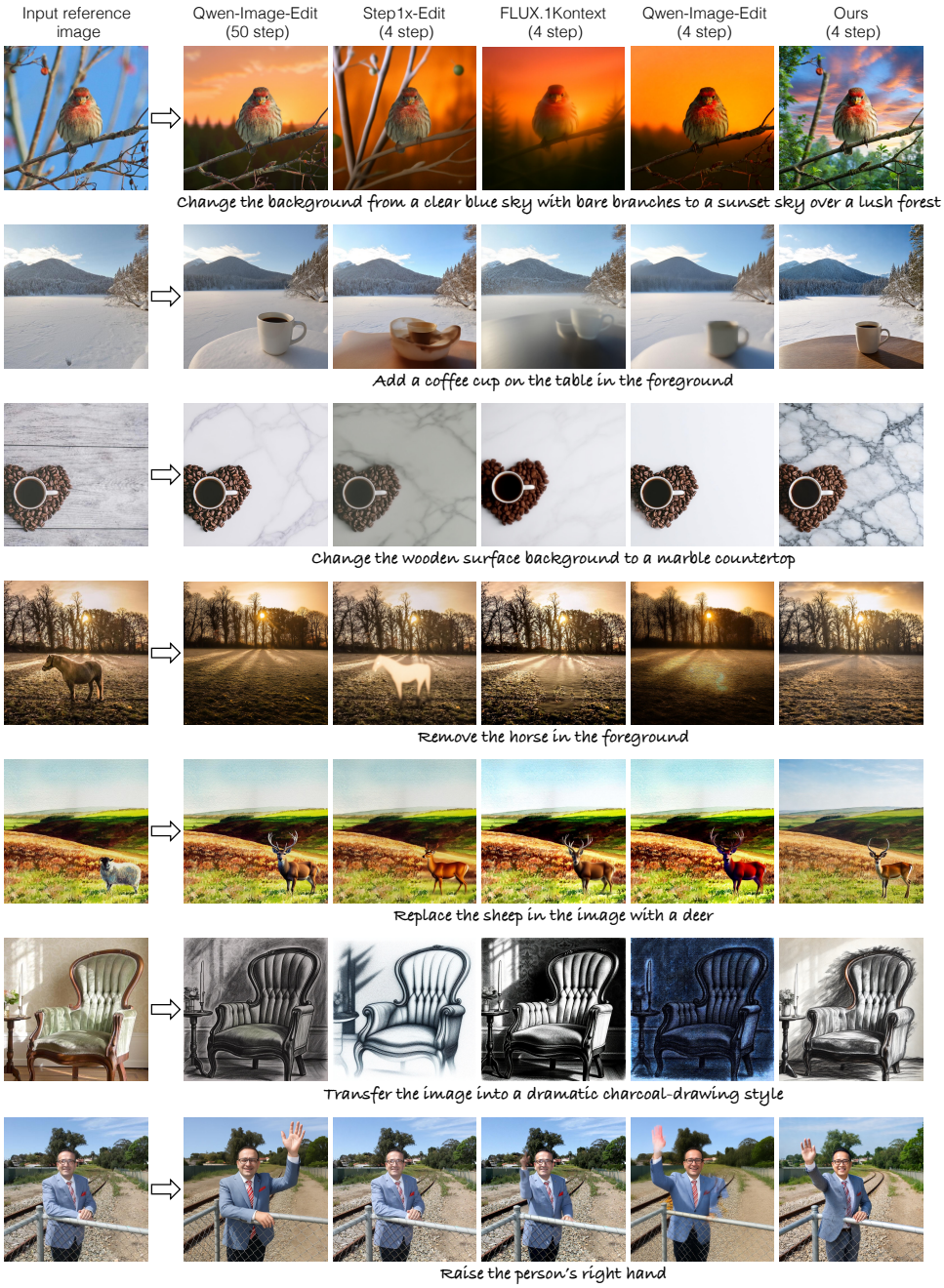


Figure E.1: **Qualitative comparison on ImgEdit-Bench** under the few-step sampling setting. For comparison, we also show the results of the best method with multi-step sampling, as measured by the quantitative metrics (Table E.2), in the 1st column. Our method performs on par or better than baseline methods across different edit types in the few-step setting.

E.2 Dataset Construction Details

Each tuple in our dataset $\mathcal{X} = \{(\mathbf{y}_i, \mathbf{c}_i, \mathbf{c}_i^{\mathbf{y}}, \mathbf{c}_i^{\mathbf{x}})\}_{i=1}^{N}$ consists of a real reference image $\mathbf{y}$, a corresponding edit instruction $\mathbf{c}$, and text prompts corresponding to the reference and edited images, $\mathbf{c}^{\mathbf{y}}$ and $\mathbf{c}^{\mathbf{x}}$, respectively. We use a text-image corpus to select reference images. Given a reference image, we prompt Qwen-2.5-32B VLM to suggest different possible editing instructions. The system and user prompt for it are as follows:

Role: system, **Content**: You are a helpful assistant and an expert in image editing.
Role: user, **Content**: Task: As a researcher in image editing, your task is to generate simple editing instructions based on the given image.

The edit types you can use include: 1) local color change, 2) local texture, 3) adjust (shape change), 4) add, 5) remove, 6) replace, 7) bg, 8) style, 9) action, and 10) text manipulation

Important: Ensure that you create a balanced distribution of these edit types when generating the instructions. Each example should utilize a different edit type, and the edit types should be evenly distributed across all examples.

When using the "add" edit type, DO NOT USE vague placements like 'near', 'under', or 'beside', instead, specify the exact location where the object should be placed. For example, instead of "add a castle near the trees" use "add a castle in the clearing between the trees".

Ensure that each instruction is straightforward and points to a single, clear edit change. Avoid complex or multi-step instructions.

Avoid Redundancy: Make sure to introduce diversity in the edit instructions.

Given the input image, could you generate simple edit instructions for different possible edit types by following the "format" of examples below and based on what you have seen in the image?

Here are some examples showing the use of various edit types:
Good example 1: {color change example}
Good example 2: {texture change example}

Good example 3: {adjust shape example}
Good example 4: {add example}
Good example 5: {remove example}
Good example 6: {replace example}
Good example 7: {bg example}
Good example 8: {style example }
Good example 9: {action example}
Good example 10: {text manipulation example}

Bad Examples: the edit instructions are hard/impossible to perform well, or mention vague terms that make the editing model struggle to perform well, and you should not follow.
Bad example 1:
- Instruction: make this dog look like it's ready for a formal evening out?
- Type: add
- Reasoning: This instruction is bad because it does not mention the exact changes that are needed to make the dog look like it's ready for a formal evening out.

Bad example 2:
- Instruction: remove the balloon [given an image of only balloons on a white background]
- Type: remove
- Reasoning: This instruction is bad as it removes the only object in the image.

Important Considerations:
1. Avoid repetition of specific phrases: Do not reuse examples or themes from the above examples. Create entirely new and diverse themes and scenarios.
2. Logical Flow: Ensure that each instruction is logical and makes sense given the image.
3. Specificity in Insertions: When adding objects, use precise placement (e.g., "in the sky" or "on the lake"). Avoid vague terms like "next to", "around", or "near".

4. Balanced use of edit types: Use a variety of edit types such as [insertion], [replace], [local texture], [shape change], [style], [remove], [local color change], and [bg]. Ensure an even distribution of these edit types across your examples.
5. Diverse scenarios: Introduce variety in the scenarios, such as futuristic, historical, magical, surreal, or natural settings. Avoid overusing common

tropes.
6. DO NOT suggest instructions that change a very small/minute part of the image.

Could you now generate 4 examples of **new, creative, and contextually relevant** edit instructions by following the format above? Avoid using the specific phrases, themes, or scenarios from the examples provided above.
Each example must use a different edit type from the ones listed above. Also, make sure to use each edit type equally across all generated examples. Finally, you should make the edit instructions as simple as possible so that the downstream editing model is able to work well.

In the above user prompt, for the good examples, we randomly select an edit instruction for each editing type out of a fixed set of manually defined edit instructions. Given edit instructions for each image, we again prompt the VLM to check the validity of the edit instruction and, if valid, to suggest a possible caption for the edited image. The system and user prompt for this is:

Role: system, **Content**: You are a helpful assistant and an expert in image editing.
Role: user, **Content**: Task: As a researcher in image editing, given the input image, edit type, and the edit instruction, your task is to check if a given edit instruction is valid and can be applied to the image. If it is valid, generate a descriptive caption for what the image would look like after applying the edit instruction. If it is not valid, return "invalid" and explain why it is not valid, and output "NA" for the edited image caption.

An edit instruction is invalid if it:
1. mentions to modify/remove/replace an object that is NOT PRESENT in the image.
2. is TOO HARD to make editing model to understand and perform well, e.g., "remove any visible accessories."
3. DOES NOT change the image in any meaningful way, e.g., given the image of a forest, "change the background to a dense forest."

For the "remove" edit type:
- DO NOT mention the object that is removed during the edit in the edited image caption. For example, given an image of a cat in a living room on a sofa with the edit type "remove" and edit instruction: "remove the cat"

Bad Example: A cat is removed from the sofa in a living room.
Good Example: A living room with a sofa.

Given the edit instruction and the original caption:
Edit type: {edit type}
Edit instruction: {simple edit instruction}

Output format:
Validity: ...
Reasoning: ...
Edited image Caption: ...

Please provide a concise but complete caption describing the edited image. Focus on the changes that would be made according to the edit instruction.

Here are some more examples:
Example 1:
- Edit type: bg
- Edit instruction: change the background to a sunset view
- Validity: valid
- Reasoning: The edit instruction is valid because it adjusts the current blue sky to a sunset view, which is a meaningful change.
- Edited image caption: A park with a sunset view. People are walking around in the park.

Example 2:
- Edit type: remove
- Edit instruction: remove the wine glass
- Validity: invalid
- Reasoning: The edit instruction is invalid because it mentions removing a wine glass that is not present in the image.
- Edited image caption: NA

Important Considerations:
1. DO NOT use instruction words like replaced, added, removed, modified, etc. in the caption.
2. Keep the caption general to explain any possible images resulting from the edit instruction.

Only output the validity, reasoning, and edited image caption. Do not include any other text or explanations.

After filtering the list of generated editing instructions using the above procedure, our final dataset consists of approximately 3M unique reference images with corresponding editing instructions spanning the 10 edit sub-types. Within the constraints of our available computational resources, this represents the largest dataset we were able to construct.

For the customization task, we first instruct the VLM to identify whether the image has a prominent object in the center. We provide an in-context sample image as well to the model. The exact system and user prompt for this is:

Role: system, **Content**: You are a helpful assistant and an expert in image personalization/customization.
Role: user, **Content**: Task: You are assisting in a research project on image personalization. Your goal is to evaluate whether the SECOND image contains a **single, uniquely identifiable object** prominently positioned near the center of the frame.

- The FIRST image ({image_path1}) is an example of a valid case.
- The specific object category in the second image can be different — focus only on **object uniqueness** and **image composition**.

Good examples include object categories that can be personalized, have unique texture, and are not general objects:
- Backpack, purse, toy, cat, dog, cup, bowl, water bottle, wearables, plushies, bike, car, clocks, etc.

Bad examples include object categories that are general objects, and different instances of the category can not be distinguished:
- Tree, building, door, flowers, food, vegetables, fruits, natural scenes, roads, etc.

Important Considerations:
1. The object should be clearly recognizable and **visually distinct** from the background.
2. The object should be **near the center** of the image.
3. The **entire object** should be visible — it should NOT be a tight or zoomed-in crop.

4. The background can be natural but should not be overly cluttered or visually distracting.
5. The image should feature a **single primary object**, not multiple equally prominent objects.

Could you now judge the SECOND image and only provide the output, reasoning, and object name, in the following format:
Output: True/False
Reasoning: Brief explanation
Object Name: The name of the object (e.g., "backpack", "cat", "toy").

If the VLM response predicts a valid image, we then query it again to suggest a new background context for the object category as follows:

Role: system, **Content**: You are a helpful assistant and an expert in image personalization/customization.
Role: user, **Content**: Given an image of an object category, you have to suggest three DIVERSE background captions for the object. Provide a detailed description of the background scene. Only suggest plausible backgrounds. DO NOT add the object name in the caption. DO NOT use emotional words in the caption. Be concise and factual but not too short. DO NOT mention the object name in the output captions. If the object is not a thing, but a scene, then output None.

Example background captions for "White plastic bottle" are:
1. near the edge of a marbled kitchen counter, surrounded by a cutting board with chopped vegetables, a salt shaker, and a stainless steel sink in the background.
2. rests on a tiled bathroom shelf, accompanied by a toothbrush holder, a mirror with foggy edges, and a shower curtain partially drawn open.

Example background captions for "a blue truck" are:
1. parked beside a graffiti-covered brick wall under a cloudy sky, with city skyscrapers rising in the background.
2. resting in a grassy field surrounded by wildflowers, with distant mountains and a golden sunset in the background.

Object: {object category name}
Output:

1.
2.
3.

E.3 Training and Implementation Details

E.3.1 Local image editing

Training hyperparameters. We train on a batch size of 32 using Adam [118] optimizer with a learning rate of 2×10^{-6}, β_1 as 0, and β_2 as 0.9. We train for a total of 10K iterations with the auxiliary network A_ϕ updated 10 times per generator G_θ update, following the strategy of DMD2 [298]. We train with the identity loss (Section 7.3.3) for 250 iterations. For faster convergence, the first 4K iterations use a single-step prediction ($t = 1$ in Line 3 of Algorithm 1), and then we start the 2-step unrolling of the diffusion trajectory. The final loss is a weighted combination of VLM-based editing loss and distribution matching loss with $\lambda_{\text{VLM}} = 0.01$ and $\lambda_{\text{DMD}} = 0.5$. During training, we also add a "*do nothing*" editing task with $\mathcal{L}_2$ loss between the input and edited image as regularization with a 1% probability. This helps the model learn to maintain consistency between input and edited images. During training, we sample the editing instruction corresponding to each subtype uniformly, except *removal*, which is sampled with 25% probability. This is because, empirically, we observe that *removal* is more difficult than other edit types like *color change*.

Template questions for VLM-based editing loss. As explained in Section 7.3.2, we evaluate VLM-based loss on two questions per edit type. Specifically for any edit type except removal, we use the following template:

Role: user, **Content**: You are a professional digital artist and an expert image editor. You will be provided with two images.

The first being the original real image, and the second being an edited version of the first.
The objective is to evaluate if the editing instruction has been executed in the second image.

Editing instruction: {edit instruction}

Answer with a Yes or No.
Note that sometimes the two images might look identical due to the failure of image editing. Answer No in that case.

Role: user, **Content**: You are a professional digital artist and an expert image editor. You will be provided with two images.

Answer with a Yes or No if the second image is exactly the same as the first image. IGNORE the changes in the second image because of the edit: {edit instruction}. Everything else should be the same.

For the removal edit-type, we change the first question to explicitly ask about the presence of the target object to be removed, with the ground truth answer in this case being *No*. We find it to be more effective than a generic template.

Role: user, **Content**: You are a professional digital artist and an expert image captioner. You will be provided with an image.
Answer with a Yes or No if the image has {object name}.

E.3.2 Free-form editing (Customization)

Training hyperparameters. We reduce the warmup iterations for which we train with the identity loss to 100 in this case, since customization often requires more drastic changes in the output image compared to the input reference image. Further, we increase $\lambda_{\text{DMD}} = 2$ instead of 0.5 as in the case of local image editing. The rest of the hyperparameters remain the same. During both training and inference, the input text prompt to the few-step generator, G_θ, is in the following template: `Generate the main object shown in the first image in a different setting and pose: {background scene description}.` We train the 4-step model for 10K iterations. For the 8-step model, we fine-tune for 5K additional training steps starting from the 4-step model.

Template questions for VLM-based editing loss. Here, we modify the questions to instead evaluate if the background context and pose are different in the generated image, i.e., editing success, and if the object identity is similar, i.e., image alignment and consistency between the input reference and edited image. The exact questions are as follows:

Role: user, **Content**: You are a professional digital artist and an expert in image editing. You will be provided with two images.

Answer with a Yes or No if the {object_name} in the second image is in a different pose and location than in the first image. Note that sometimes the second image might not have the same object because of the failure of image editing. Answer No in that case.

Role: user, **Content**: You are a professional digital artist and an expert in image editing. You will be provided with two images.

Answer with a Yes or No if the {object_name} in the second image is the exact same identity, with similar color, shape, and texture as in the first image. Note that sometimes the second image might not have the same object because of the failure of image editing. Answer No in that case.

E.4 Baseline Details

Flow-GRPO [153]. We follow the open-source implementation of Flow-GRPO and train with the same computational budget as our method, i.e., across 4 A100 GPUs and 2.5 days of training. The final model is fine-tuned from a pretrained image-editing model for 5K iterations. During training, we collect 16 images per prompt with 12 denoising steps (28 during inference) for computing the mean and standard deviation in GRPO [235]. Following their official implementation, we train with LoRA [96] of rank 32, $\alpha = 64$, learning rate 1×10^{-4}, and use the VLM to score edits on a scale of 0 to 9, which is then normalized to $[0, 1]$ as the final reward. The exact prompt used to query the VLM is derived from VIEScore [127] and is shown below.

Role: system, **Content**: You are a helpful assistant and an expert in image editing.
Role: user, **Content**: You are a professional digital artist. You will have to evaluate the effectiveness of AI-generated edited image(s) based on given rules.

You will have to give your output in this way (Keep your reasoning VERY CONCISE and SHORT):
score : ...,

reasoning : ...

RULES:
Two images will be provided: The first being the original real image and the second being an edited version of the first.
The objective is to evaluate how successfully the editing instruction has been executed in the second image.

Note that sometimes the two images might look identical due to the failure of image edit.

From scale 0 to 9:
A score from 0 to 9 will be given based on the success of the editing. (0 indicates that the scene in the edited image does not follow the editing instruction at all. 9 indicates that the scene in the edited image follows the editing instruction text perfectly.)

Editing instruction: {edit instruction}

Supervised Fine-Tuning. We train with the standard velocity prediction flow-objective for 30K iterations with a batch size of 32 and learning rate 2×10^{-6} with a linear warmup of 2K iterations. To enable classifier-free guidance, we drop the image and text conditions 10% of the time.

Sampling parameters for local image-editing baselines. We follow the open-source implementation to sample images from all the baseline models for the benchmark evaluations. The TurboEdit [55] baseline requires a caption corresponding to the edited image as well, and we use Qwen-2.5-32B VLM to generate these captions for GEdit-Bench images.

Sampling parameters for customization baselines. Here as well, we follow the open-source implementation to sample images from all the baseline models for the benchmark evaluations. In the case of DSD [27], it employs Gemini-1.5 to convert the input user prompt into a detailed prompt. However, we skip this step for a fair evaluation with other methods, which do not use any prompt rewriting tools. In the case of SynCD [132], though it supports multiple reference images as input, we evaluate it with a single reference image, to keep the setup similar to other baseline methods and ours. For sampling images with OminiControl [257] and DSD [27], we follow their recommended prompt setting and replace the category name with "this item".