

Unified Vision-Language Modeling

Alexander Joseph Swerdlow

CMU-RI-TR-25-23

May 2025

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee

Katerina Fragkiadaki	<i>Carnegie Mellon University, Chair</i>
Shubham Tulsiani	<i>Carnegie Mellon University</i>
Ayush Jain	<i>Carnegie Mellon University</i>

*Submitted in partial fulfillment of the requirements
for the degree of Master of Science.*

To my parents, Sharon and Jonathan.

Abstract

Recent developments in language modeling on large-scale datasets have found profound success in a variety of tasks. Given this, many researchers have sought to apply these same capabilities to other modalities, such as 2D or 3D vision. However, this effort has been met with a variety of challenges due to differences in 1) the underlying data representation, 2) the desired tasks for each modality, and 3) the availability of large, high-quality datasets.

In this thesis, we present two approaches for solving these challenges. First, we explore discrete diffusion models as a unified generative formulation in the joint text and image domain and demonstrate their advantages over autoregressive models including in improved control over quality versus diversity, joint multimodal inpainting, and greater controllability in generation through guidance. Second, we present a method for unifying the training of 2D and 3D vision-language models using a single, unified model, enabling the use of large-scale 2D datasets for 3D tasks.

The source code accompanying this thesis is publicly available at:

github.com/alexanderswerdlow/unidisc
github.com/facebookresearch/univlg

Acknowledgments

First and foremost, thank you to my thesis committee; I am grateful to my advisor, Katerina Fragkiadaki, for her support and guidance over these past two years, and to Shubham Tulsiani for his valuable feedback throughout my time at CMU. I am also very grateful to Deepak Pathak for his indispensable support.

Before coming to Carnegie Mellon, I had the privilege of working with many advisors who shaped me into the researcher I am today. In high school, I was very lucky to have Stephan Shapiro as a mentor for FIRST Robotics—he nurtured my passion for robotics and taught me how to be a good engineer. At UCLA, I first had the privilege of being advised by Paolo Tabuada and later by Dennis Hong. Finally, I was very fortunate to be advised by Bolei Zhou, who taught me so much about how to be a good researcher.

I am also immensely grateful to those whom I met during my time in industry, including Raquel Urtasun, Jeff Han, Karine Marabian, Lunjun Zhang, Bin Wang, Aditya Parikh, and many others. Moreover, throughout my academic journey, I have been lucky to receive guidance and learn from the following people: Carey Nachenberg, Stefano Soatto, Achuta Kadambi, Ankur Mehta, and Aviral Kumar.

At Carnegie Mellon, I was fortunate to be surrounded by a great group of people, including Chaitanya Chawla, Jeff Tan, Kashu Yamazaki, Neham Jain, Nikolaos Gkanatsios, Shantanu Jaiswal, Siddharth Gandhi, Sruthi Mukkamala, and Wen-Hsuan Chu.

Next, I want to express my gratitude to Ayush Jain. From my first visit to the lab, you have been a great friend and mentor—in more than just research. I aspire to be half the person you are.

I also want to specially thank Mihir Prabhudesai. You altered my path, and I am sincerely grateful for that. Who knew disk benchmarks could be so powerful? I have learned so much from you and will miss our many discussions.

I also want to thank the following highly incomplete list of people,¹ without whom I would not be here today: Abby Irby, Alex Wazzan, Austin Shalit, Ben Salk, Benjamin Lanir, Brian Raymond, Charlie Carter, Clay Weisberg, Daniel Schrager, David Rapia, Eli Yadidi, Ioana Ciuperca, Jason Weiner, Jim Olson, Jonah Rudner, Josh Levine, Kareena Kullar, Karielle Schnapp, Keaton Paquin, Laurel Thomas, Marcus Bernstein, Megan Kohanarieh, Natalya Krekorian, Nevin Liang, Nicole Sussman, Pablo Oger, Rita Gimelshein, Rishi Sankar, Samantha Renard, Shawn Fields-Meyer, Sierra Hagen, Talia Rubenstein, Tameez Latib, Villa Patt, Will Wloch, and Yoel Nozar. In ways large and small, you have shaped who I am, and I am sincerely grateful that you have been a part

¹In alphabetical order

of my life.

Finally, I want to thank my entire family for their unwavering support all these years—including my grandparents Abraham, Anita, Edith, and Harry, as well as Elaine, Jacob, Daniel, and Eliana Wasserman. To my brother Ari and my sister Leora—thank you for putting up with twenty-three years of me being me and for loving me all the same.

Any words I could write to describe my gratitude toward my parents, Sharon and Jonathan, would be insufficient, so I will keep this short. I am unreasonably lucky to have you as my parents. Thank you for the multitude of sacrifices you have made; thank you for your persistence and for your unconditional love.

Contents

1	Introduction	1
2	Unified Multimodal Discrete Diffusion	3
2.1	Abstract	3
2.2	Introduction	3
2.3	Related Work	5
2.3.1	Unified Multi-Modal Models	5
2.3.2	Discrete Diffusion Models	6
2.4	UniDisc: Unified Discrete Diffusion	6
2.4.1	Diffusion Models	6
2.4.2	Discrete Diffusion Models	7
2.4.3	Unified Training via UniDisc	8
2.4.4	Unified Sampling via UniDisc	8
2.5	Experiments	9
2.5.1	Evaluation of Multimodal Generation	9
2.5.2	Training Efficiency and Inference Speed	11
2.5.3	Multimodal Discriminative Performance	13
2.5.4	Scaling UniDisc	15
2.6	Conclusion	16
2.7	Acknowledgments	16
2.8	Appendix	17
2.8.1	Additional Experiment Details	19
2.8.2	Scaling Experiment Details	23
2.8.3	Training Details	24
2.8.4	Fine-tuning An Autoregressive model for Discrete Diffusion	24
2.8.5	Ablations	26
2.8.6	Large Scale Qualitative Results	27
2.8.7	Large Scale Quantitative Comparisons	34
2.8.8	Understanding the effect of Classifier Free Guidance (CFG)	35
2.8.9	Inference: Generation time vs. batch size	36
3	Unified 2D and 3D Vision-Language Understanding	37
3.1	Abstract	37
3.2	Introduction	37
3.3	Related Work	39

3.4	Method	41
3.4.1	Supervision Objective	43
3.5	Experiments	44
3.5.1	Evaluation on 3D Referential Grounding	44
3.5.2	Evaluation on Out-of-Domain 3D Referential Grounding	46
3.5.3	Evaluation on 3D Question Answering	46
3.5.4	Evaluation on 2D Referential Grounding	47
3.5.5	Ablations	48
3.6	Conclusion	49
3.7	Acknowledgements	49
3.8	Appendix	51
3.8.1	Evaluation on 3D Instance Segmentation	51
3.8.2	Additional Implementation details	51
3.8.3	More details on Out-of-domain LX3D datasets	52
3.8.4	Effect of Fine-tuning 2D backbones in UniVLG	53
3.8.5	Performance with different backbones	53
3.8.6	Additional Metrics on ScanQA Dataset	54
3.8.7	Visualizations of UniVLG on Referential Grounding Datasets	54
3.8.8	Visualization of common failure modes of UniVLG	54
3.8.9	Performance analysis with pose and depth noise	55
3.8.10	Miscellaneous Details	57
3.8.11	2D-3D Generalization Test	59
3.8.12	Discussion: Decoding Masks vs. Boxes	60
3.8.13	Evaluation with Mask based Intersection Over Union	61
3.8.14	Additional Related Work	61
4	Conclusions and Future Directions	63
	Bibliography	65

List of Figures

2.1	We show UniDisc’s ability to jointly inpaint image & text pairs. We do not explicitly optimize for this objective but it is intrinsic to UniDisc’s unified diffusion objective.	4
2.2	UniDisc is a unified multimodal discrete diffusion model that can jointly process and generate text and images. Each modality is converted into a sequence of discrete tokens and we jointly denoise, supervising with a weighted cross-entropy loss. At inference time we begin with a set of [MASK] tokens and iteratively unmask tokens.	7
2.3	Scaling Analysis for AR and UniDisc models: (Left) IsoFLOP curves for UniDisc, plotting varying model size for a fixed FLOP budget. (Right) Estimating optimal parameter size for each budget - minima of fitted parabola, we plot scaling laws for both AR and UniDisc. We find 13.2x more compute is required for UniDisc to achieve the same overall loss as AR.	10
2.4	Conditional generation results for both FID and CLIP metrics, across a range of CFG values. We find that AR is more sensitive to the CFG weighting, with a narrower optimal range.	10
2.5	Inference Comparisons for UniDisc and AR baseline: (a) Chameleon Perplexity (Text+Image) vs. Time - we perform similar to best AR method, (b) Chameleon Perplexity vs. Entropy - UniDisc has high diversity and low perplexity, while AR has significantly lower diversity, (c) Image FID vs. NFE, showing image generation saturates quickly with NFE (≈ 32), (d) GPT2 Generative Text Perplexity vs. NFE showing text generation benefits from more sampling steps (diminishing).	12
2.6	Uniform Concept Generation: We perform joint generation given only masked text input (left). We use a language conditioned segmentation model and find that UniDisc generates uniformly in <i>concept space</i> (right).	13
2.7	Image-Text Reasoning measured by QA and retrieval accuracy across datasets.	13
2.8	Zero-shot Image Editing: UniDisc can take corrupted and mismatched image/text pairs (left) and produce an aligned, high-quality pair (right), using the model’s own likelihood as a scoring function.	15
2.9	Conditional generation results for both FID and CLIP metrics, across a range of CFG values. We find that AR is more sensitive to the CFG weighting, with a narrower optimal range.	20

2.10	Unconditional multimodal generation results for UniDisc and AR baseline at 115M parameters - both models perform similarly.	21
2.11	Conditional generation results for UniDisc and AR baseline. Our model significantly outperforms the AR model when classifier free guidance is used.	21
2.12	We compare UniDisc with an AR model fine-tuned for joint inpainting and evaluate on a subset of DataComp1B.	22
2.13	Joint Retrieval Accuracy on DataComp1B. We outperform AR given the task of retrieving one correct image-text pair out of 16 possible pairs, implying better learnt representations.	23
2.14	Fine-tuning a pre-trained 270M parameter AR model on LM1B.	25
2.15	Training Loss Curve vs. Tokens on our 1.4B model.	25
2.16	UniDisc’s ability to generate an image, given unseen text as input.	27
2.17	UniDisc’s ability to generate text (captioning), given unseen image as input.	28
2.18	Zero-shot text-conditioned inpainting. UniDisc inpaints a masked region given a user-provided text prompt.	28
2.19	Zero-shot multimodal inpainting. UniDisc jointly inpaints in both image and text spaces.	29
2.20	Zero-shot multimodal editing. We provide a <i>clean</i> image and text pair and UniDisc automatically enhances both the image and text. In the final row, we fix the text and allow only the image to change.	30
2.21	We show how UniDisc jointly infills both image and text. $\arg\max p_{\theta}(x_0 x_t)$	32
2.22	We demonstrate that UniDisc uniformly generates all concepts at once.	33
2.23	We train UniDisc on 512x512 resolution images but demonstrate zero-shot inpainting at 1024x1024.	33
2.24	L2 distance between unconditional and conditional logits on currently masked tokens as sampling steps increase.	35
2.25	Comparing CLIP scores by applying CFG only on specific steps. This shows CFG has the most impact on the initial denoising steps (total steps = 24).	35
2.26	We show the effect of classifier-free guidance from left-to-right, starting with $w = 0$, and increasing linearly to $w = 8$ on the right, where output logits are $l_{\text{cfg}} = (1 + w)l_{\text{cond}} + w * l_{\text{uncond}}$. Caption: "crab meditating, surfboard, orange sun setting, rainbow clouds, zen beach"	36
2.27	Generative Perplexity vs. Time with various models and sampling strategies.	36
3.1	(A) UniVLG achieves state-of-the-art performance across a range of referential grounding, question answering, and instance segmentation benchmarks. (B) UniVLG is a unified model which accepts posed RGB-D sequences, or monocular 2D images which are then lifted to 3D pointmaps. (C) UniVLG significantly benefits from joint 2D-3D training, further boosted when all parameters are shared between modalities by using 2D-3D lifting. (D) Example task inputs/outputs for UniVLG.	38

3.2	UniVLG Architecture: A vision language transformer that accepts a language utterance and either (1) a sequence of posed RGB-D images or (2) a monocular RGB image, lifted to 3D (2D to 3D Projection). UniVLG fuses information across vision and language to predict 3D object segments or generate answers. It uses a ViT backbone followed by 3D relative attentions to produce a set of 3D feature tokens. The proposed decoder then iteratively updates a set of learnable queries as well as the 3D feature tokens through token - language - query attentions to decode object segments and match them to noun phrases in the input referential utterance. Masks are decoded through a dot-product between 3D feature tokens and learnable queries. A text decoder predicts answers for the input questions by conditioning on the set of updated object queries.	41
3.3	Visualizations of UniVLG on 3D Referential Grounding Datasets of ScanRefer, SR3D, and NR3D The red masks indicate UniVLG’s prediction for the target object, and pink masks indicate its predictions for the anchor object. Green masks and boxes indicate ground-truth target and anchor objects.	55
3.4	Visualizations of UniVLG on 2D Referential Grounding Datasets of RefCOCO, RefCOCO+, and RefCOCOg: The green masks indicate predictions of UniVLG.	56
3.5	Systematic failure modes of UniVLG: Green boxes and masks are ground-truth, red masks and boxes are UniVLG’s predictions.	56
3.6	We analyze the performance of UniVLG and BUTD-DETR on SR3D as the pose and depth error increases. We add gaussian noise to the pose and raw depth which affects the unprojected point cloud that both models observe.	58

List of Tables

2.1	We demonstrate how generative perplexity is an imperfect metric requiring calibration with entropy.	19
2.2	Model variants. The FFN hidden size is always 4x the overall d_model . .	24
2.3	Ablation w/115M parameter model of QK Norm, zero initialization of linear layers, RMSNorm, setting invalid tokens to $-\infty$ during training and generation, and Softmin SNR.	26
2.4	Ablation w/115M parameter model on different objective level decisions such as noising schedule, loss weighting and whether to use discrete time. .	26
2.5	We evaluate UniDisc on the GenEval [GHS23] benchmark.	34
2.6	We evaluate the 1.4B version of UniDisc on FID. We use evaluate on MS-COCO 30K [Che+15] and MJHQ-30K [Li+24a].	34
2.7	We compare our model to [Hu+23] on CUB200.	34
3.1	Results on 3D language grounding in 3D mesh and sensor point clouds (PC). We evaluate top-1 accuracy on the official validation set with assuming ground-truth (GT) or without assuming ground-truth proposals (Det).	45
3.2	Out-of-Domain 3D Referential Grounding Acc@25 in Det. From left-to-right, ScanNet++, HM3D, ARKitScenes, ScanNet (GT), ScanNet (SAM-Pro3D). See Section 3.8.3 for details.	46
3.3	Results on 3D Visual Question Answering on official validation sets. We evaluate top-1 exact match accuracy (EM@1).	47
3.4	Results on val sets of 2D Ref. grounding datasets	47
3.5	Analysis of Box Head vs Mask Head on ScanRefer Dataset with Acc@25 if not otherwise stated.	48
3.6	Analysis of 2D training strategies Acc@25 in DetSetup	48
3.7	Ablations Acc@25 in DetSetup	48
3.8	Evaluation on 3D Instance Segmentation Benchmarks. (S) and (M) denotes models trained on sensor and mesh point clouds respectively. . . .	51
3.9	Effect of Fine-tuning 2D backbones of UniVLG for Acc@25 in DetSetup. SR3D and NR3D are in-domain and ScanRefer is out-of-domain	53

3.10	Ablation of visual backbones on 3D language grounding. We evaluate top-1 accuracy on the official validation set without assuming ground-truth proposals (Det).	54
3.11	Extra Metrics on ScanQA validation set	57
3.12	2D-3D Generalization Test	59
3.13	Mask mAP evaluation on 3D language grounding. We evaluate top-1 accuracy on the official validation set without assuming ground-truth proposals (Det).	61

Introduction

The success of large language models (LLMs) trained on vast text data has been remarkable. However, building models that integrate with visual understanding, across both 2D images and 3D scenes—as well as both understanding and generation—presents unique challenges. These challenges stem from differences in data representation and construction (e.g., text has a natural sequential structure whereas images do not), the tasks prevalent in each domain (e.g., generation vs. inpainting), and the significant disparities in the availability and scale of high-quality annotated data, with 3D data being notably scarcer than 2D images or text.

This thesis investigates methods for unified models that bridge these gaps. The core idea is to leverage shared representations, architectures, and training paradigms to allow knowledge transfer and joint learning across modalities.

The first part of this thesis (Chapter 2) focuses on unifying generative modeling across text and 2D image modalities. Current multimodal generative models are often dominated by autoregressive (AR) approaches, which generate data sequentially. While effective, AR models can be slow for high-dimensional data like images and offer limited controllability and editability. We introduce **Unified Multimodal Discrete Diffusion** (UniDisc), a model trained to reverse a discrete corruption process (masking) applied jointly to tokenized text and images. We demonstrate that this approach offers several advantages over AR models, including inherent support for joint multimodal inpainting (filling in missing text and image parts simultaneously), improved control over the quality-diversity trade-off during generation via techniques like classifier-free guidance, and enhanced controllability. We present a comprehensive comparison against AR counterparts, analyzing scaling properties, inference efficiency, and performance on various generative tasks, showing that UniDisc provides a compelling alternative for unified text-image generation.

The second part of this thesis (Chapter 3) addresses the challenge of unifying 2D and 3D vision-language understanding, motivated primarily by the relative scarcity of large-scale, richly annotated 3D datasets compared to their 2D counterparts. We introduce **Unified Vision-Language Grounding** (UniVLG), a single model architecture designed to process and learn from both 2D image-text data and 3D scene-text data (represented as posed RGB-D sequences). A key strategy is to leverage powerful, pre-trained 2D vision models

by initializing the majority of UniVLG’s weights from them. We further bridge the 2D-3D gap by incorporating 2D-to-3D lifting techniques, allowing the model to process 2D images through its 3D pathways, thus enabling effective knowledge transfer from abundant 2D data to improve 3D task performance. We introduce a novel language-conditioned mask decoder, shared across modalities, which proves more effective for precise 3D object grounding than traditional bounding box prediction. We demonstrate that UniVLG, especially when co-trained on 2D and 3D data, achieves state-of-the-art performance on multiple 3D vision-language benchmarks, particularly in more realistic settings that use sensor-derived point clouds and do not rely on ground-truth object proposals, all while maintaining strong performance on 2D tasks.

Unified Multimodal Discrete Diffusion

2.1 Abstract

Multimodal generative models that can understand and generate across multiple modalities are dominated by autoregressive (AR) approaches, which process tokens sequentially from left to right, or top to bottom. These models jointly handle images, text, video, and audio for various tasks such as image captioning, question answering, and image generation. In this work, we explore discrete diffusion models as a unified generative formulation in the joint text and image domain, building upon their recent success in text generation. Discrete diffusion models offer several advantages over AR models, including improved control over quality versus diversity of generated samples, the ability to perform joint multimodal inpainting (across both text and image domains), and greater controllability in generation through guidance. Leveraging these benefits, we present the first **Unified Multimodal Discrete Diffusion** (UniDisc) model which is capable of jointly understanding and generating text and images for a variety of downstream tasks. We compare UniDisc to multimodal AR models, performing a scaling analysis and demonstrating that UniDisc outperforms them in terms of both performance and inference-time compute, enhanced controllability, editability, inpainting, and flexible trade-off between inference time and generation quality. Code and additional visualizations are available at <https://unidisc.github.io>.

2.2 Introduction

Multimodal generative models—which can understand and generate a variety of modalities such as text, images, videos, and audio – can significantly improve the capabilities of an AI system, as these models can (1) leverage information from multiple sources to better understand the context (2) learn from any available data source, and (3) respond to a user’s request in a flexible manner, thus dynamically generating text, images, or audio as required. Although the choice of model architecture—transformers—is currently clear, the optimal generative objective remains unclear.



Figure 2.1: **We show UniDisc’s ability to jointly inpaint image & text pairs.** We do not explicitly optimize for this objective but it is intrinsic to UniDisc’s unified diffusion objective.

Current multimodal models are typically trained jointly using (an approximation to) a maximum likelihood objective over sequences consisting of images, text, and other modalities. Autoregressive (AR) models typically quantize data from continuous modalities and optimize the exact likelihood through a series of conditionals; during generation, they use a fixed token order, e.g., left-to-right, top-to-bottom (raster order) for images. They have demonstrated strong performance in both text and image generation, making them the current workhorse for multimodal models. However, generating image tokens autoregressively is slow and wasteful as nearby tokens are highly correlated, and this process results in many unnecessary forward passes through the network [Lu+22; Tea+23; Tea24]. Moreover, AR models are difficult to control [Li+22b], cannot inpaint or infill unless explicitly trained to, and cannot easily trade-off quality versus compute at inference time.

On the other hand, continuous diffusion models—which have been shown to work well for continuous modalities such as images—have fast inference, are highly controllable, and can easily trade-off quality vs. compute. These models corrupt data by adding Gaussian noise and are trained to denoise the data, maximizing a lower bound on the likelihood. However, these models have found to be significantly slower to train in text domain compared to AR models (by roughly 64 times) [GH24]. Text is inherently discrete, and adding continuous Gaussian noise to text token embeddings does not correspond to meaningful changes in the actual text. These trade-offs between different modeling strategies across modalities raises the question: what is the right unified generative formulation across text, image, and other modalities?

To address this, we present UniDisc, a unified multimodal model based on discrete

diffusion. While continuous Gaussian noise is inefficient with discrete data such as text and graphs, UniDisc corrupts data with discrete noise—specifically, randomly masking tokens—and learns to map mask tokens into multimodal tokens during inference. Discrete diffusion through masking has been explored separately for generating text [Aus+21; Sah+24] and images [Cha+22; Cha+23]. Such explorations have resulted in different noise schedules, transition kernels, and loss functions across the text and image domains. In this chapter, we explore a discrete diffusion formulation and its applicability in jointly modeling text and image modalities with a unified set of hyperparameters.

We propose a unified architecture that jointly tokenizes text and images, and uses full self-attention to learn to map a masked token sequence to a clean token sequence by sampling from a joint vocabulary of text and image tokens. We evaluate UniDisc across multimodal conditional and unconditional generation on multiple image-text datasets and compare to its AR counterpart. First, we find UniDisc achieves a higher FID and CLIP score than AR (Figure 2.4), which we attribute to the effect of classifier-free guidance. We show that UniDisc exhibits strong joint image-text inpainting abilities that are not possible with prior unified generative models as seen in Figure 2.1. Second, we find that UniDisc consistently outperforms its AR counterpart in inference efficiency: at a given inference compute budget, our model achieves generations of higher quality and diversity (Figure 2.5). Third, we show UniDisc showcases stronger discriminative ability than AR on retrieval tasks due to its variable number of sampling steps (Figure 2.7). Lastly, we scale UniDisc to a 1.4B parameter model, trained on web-scale image-text datasets.

Our code, model weights, and dataset are publicly available. More qualitative visualizations are available at <https://unidisc.github.io>.

2.3 Related Work

2.3.1 Unified Multi-Modal Models

In recent years, unified models for processing multiple modalities have advanced significantly. Models like Flamingo [Ala+22] and PaLM-E [Dri+23] demonstrate strong few-shot learning capabilities across tasks. LLaVA [Liu+23] enhances LLaMa [Tou+23] with multimodal fine-tuning, but still uses separate encoders, limiting true unification and image generation. Recent efforts, like Perceiver IO [Jac+21] and Unified-IO [Lu+22; Lu+23b], attempt modality unification but at a smaller scale. The Chameleon project [Tea24] scales this up with a 34B parameter model trained on image-text data. However these approaches largely focus on autoregressive generation which is inefficient for high-dimensional data. Moreover, 4M [Miz+23; Bac+24] proposes an encoder-decoder approach with image-like modalities using a discrete-diffusion (referred to as *masked modeling*) approach and text generated autoregressively.

Relevant to our work, UniD3 [Hu+23] considered discrete diffusion on image and text but made several design decisions that separated each modality, using both absorbing and uniform masking, decoupling the modalities inside the model with separate operations on each. Further we couldn’t compare against their model—no training code is available and

were unable to reproduce their reported results using their publicly available code.

2.3.2 Discrete Diffusion Models

Discrete diffusion models have emerged as a promising alternative to continuous diffusion for discrete data types. [Soh+] introduced the first discrete diffusion model over binary variables, [Hoo+] extended the noising process to categorical variables, demonstrating its effectiveness on image generation tasks. D3PM [Aus+21] later extended discrete diffusion to a more general set of noising processes, allowing for more flexible noise schedules. Recent work by SEDD [LME] introduced score entropy, a novel loss function for discrete diffusion models that bridges the gap between continuous and discrete spaces, and more recently, [Sah+24; Shi+24] showed text perplexity competitive with GPT-2. Most recently, [Nie+24] looked at the scaling properties of discrete diffusion on text. While this approach shows promise for improving discrete diffusion models, these methods were primarily focused on language modeling tasks. Our work extends the application of discrete diffusion to multiple modalities and demonstrates its effectiveness in a unified architecture.

2.4 UniDisc: Unified Discrete Diffusion

2.4.1 Diffusion Models

Diffusion models [HJA20; Soh+; Son+20] are a class of generative models that learn to construct a data distribution by gradually reversing a process that introduces noise into clean data samples. This approach models the transformation of a data sample x_0 from a clean state through increasingly noisy states until it reaches a pure noise distribution.

The forward diffusion process is described by a series of transitions, where each latent variable x_t at time step t is sampled from a Gaussian distribution as follows:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I)$$

Here, $\bar{\alpha}_t = \prod_{s=0}^t \alpha_s$ represents the cumulative product of noise levels, making x_t increasingly distant from x_0 as t increases. The variable x_t represents the noisy version of x_0 at time t , modeled to progressively approximate Gaussian noise as t approaches the final time step.

The reverse diffusion process then aims to reconstruct the original data by progressively denoising these samples. This involves learning the reverse transitions, with the goal to train the model $p_\theta(x_{t-1}|x_t)$ to approximate the true reverse process and effectively recover the original data point x_0 from the noisy samples.

Given T timesteps of diffusion, the loss using the Evidence Lower Bound (ELBO) for the diffusion process equals¹:

¹We skip the prior matching term from the loss as it is zero.

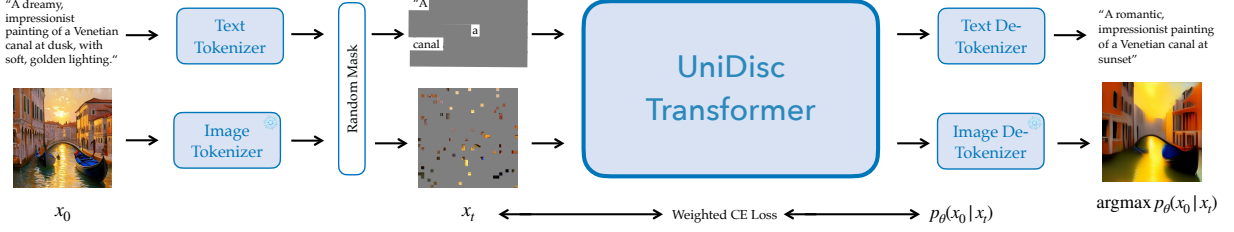


Figure 2.2: UniDisc is a unified multimodal discrete diffusion model that can jointly process and generate text and images. Each modality is converted into a sequence of discrete tokens and we jointly denoise, supervising with a weighted cross-entropy loss. At inference time we begin with a set of [MASK] tokens and iteratively unmask tokens.

$$\mathcal{L}_{\text{diff}} = \underbrace{-\mathbb{E}_{q(x_1|x_0)} [\log p_\theta(x_0|x_1)]}_{\text{reconstruction term}} + \underbrace{\sum_{t=2}^T \mathbb{E}_{q(x_t|x_0)} [D_{KL}(q(x_{t-1}|x_t, x_0) || p_\theta(x_{t-1}|x_t))]}_{\text{denoising matching term}} \quad (2.1)$$

2.4.2 Discrete Diffusion Models

Building on the foundations of continuous diffusion models, discrete diffusion models adapt these concepts to structures that are inherently discrete. Unlike their continuous counterparts that model transitions of x_t given x_{t-1} with Gaussian distributions, discrete models define transitions using categorical distributions. The forward process for discrete models is thus characterized as:

$$q(x_t|x_0) = \text{Cat}(x_t; x_0 \cdot \bar{Q}_t) \quad (2.2)$$

Here, $\bar{Q}_t = \prod_{i=0}^{t-1} Q_i$ is a $N \times N$ matrix where N is the size of the vocabulary. $\bar{Q}_t$ represents the cumulative transition matrix at each discrete time step t , and Q_t is a transition matrix $[Q_t]_{ij} = q(x_t = j | x_{t-1} = i)$ dictating the probabilities of moving from one discrete state (a token in the vocabulary) x_{t-1} to another x_t discrete state (a token in the vocabulary), x_0 is a one-hot vector of the input data sample. D3PM [Aus+21] generalizes this framework over various transition matrices (Q_t), the popular ones mainly include uniform and absorbing transition matrix. In UniDisc, we use the absorbing transition matrix as empirically it has been found to work the best across text and images [Aus+21; LME]. Absorbing transition matrix requires having an absorbing state namely the [MASK] token. The matrix is represented as $Q_t = \alpha_t I + (1 - \alpha_t) \mathbb{1} e_m^T$, where $\mathbb{1}$ is a column vector of ones and e_m is a one-hot vector with one on the mask state m . This ends up being a matrix with all zeros except $i = j \neq m$ is α and $j = m, i \neq m$ is $1 - \alpha$ and $i = j = m$ is 1.

Intuitively this means that during the forward transition, the probability of an input token x_0 to stay the same is α , the probability of it being masked is $1 - \alpha$, and the probability of a masked token to be unmasked is 0.

Given the forward diffusion in Equation (2.2), [Soh+] uses the same objective function as Equation (2.1) to optimize their model, where $q(x_{t-1}|x_t)$ ends up being a Bernoulli distribution instead of a Gaussian distribution. MDLM [Sah+24] simplifies this objective function by considering continuous time-diffusion and applying loss only on the masked tokens. The final loss simply ends up being a re-weighted masked generative modeling loss:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t \sim \mathcal{U}(0,1), q(x_t|x)} \left[\frac{\alpha'_t}{1 - \alpha_t} \log p_\theta(x_0 | x_t) \right] \quad (2.3)$$

where $\alpha'_t = \alpha_t - \alpha_{t-1}$, and α_t is the probability of the token not being masked. MaskGIT [Cha+22] and Muse, state-of-the-art masked image generative model use the same loss as Equation (2.3), except there is no reweighting term and the time is discrete time instead of continuous time. The noising schedule α_t is also different, while language discrete diffusion models such as [Aus+21; Sah+24] use a linear-time schedule, MaskGIT and Muse [Cha+22; Cha+23] use a cosine schedule. We ablate these different design choices in Section 2.8.5.

2.4.3 Unified Training via UniDisc

We train a bidirectional decoder-only transformer [Vas+17] using 2D RoPE embeddings [Liu+24] for all image tokens, 1D RoPE [Su+23] embeddings for text tokens, and add learned modality-specific embeddings to each token. This allows our model both flexibility in resolution at inference, and the ability to use compute effectively by performing the majority of training at a lower resolution. We use the same objective function as Equation (2.3), except for us x_0 is $[x_0^{img}, x_0^{txt}]$

Classifier-Free guidance (CFG) [HS22] has been used in continuous diffusion models to trade-off between quality and diversity of generation. We apply this idea to discrete diffusion, with a probability of 0.1 we set all the tokens of a random modality to be mask tokens, this allows UniDisc to learn unconditional likelihood for image and text modality. During inference we use CFG for conditional generation (image-to-text or text-to-image) to trade-off between quality and diversity of generation as shown in Figure 2.5.

To improve training stability, we use Query-Key Normalization [Wor+23] and use RMSNorm [ZS19] for all other norms. We use Sandwich Normalization—normalization before and after each FFN, as we found this helps control activations in deeper layers as previously reported in [Din+21; Zhu+24c].

To further improve the convergence speed of discrete diffusion we analyze the noising schedule and find that linear schedule in [Sah+24; Aus+21] results in excessively high weighting for early timesteps, impairing the convergence speed. Following Min-SNR trick in continuous diffusion [Han+23], we limit the minimum weighting to 5. An architecture diagram is provided in Figure 2.2 and pseudo-code for training procedure is provided in Section 2.8.

2.4.4 Unified Sampling via UniDisc

Sampling in masked discrete diffusion, involves mapping a set of masked tokens m to a set of visible tokens x_0 using T timesteps of denoising. A variety of sampling strate-

gies have been previously proposed [Soh+; Aus+21; Zhe+24b; Cha+22; Sah+24; LME] for masked discrete diffusion. MaskGIT [Cha+22] proposes a confidence-based sampling, where they decode the most confident tokens at each step of denoising. D3PM [Aus+21] and MLDM [Sah+24] uses a sampling mechanism similar to [HJA20] except applied to a bernoulli distribution, which we refer to as DDPM sampling. This results in a random set of tokens being decoded, instead of the most confident ones as in MaskGIT. We ablate these sampling strategies in Figure 2.5 and find the confidence-based sampling proposed in MaskGIT to work the best for unified modeling.

2.5 Experiments

We compare UniDisc against an autoregressive (AR) baseline across various tasks, metrics and datasets. We use the same architecture and hyper-parameters, and data, only differing in the attention mask and respective loss functions. For our autoregressive baseline we use a standard language model architecture from Chameleon [Tea24]—that is a decoder-only transformer with causal attention and rotary positional embeddings. To enable classifier-free guidance, we dropout modalities with 10% probability during training. For UniDisc, we dropout both modalities and for the AR baseline we dropout only the first modality in the input sequence as in [Liu+24].

Our experiments aim to answer the following questions:

1. How does UniDisc compare against AR models in unconditional and conditional multi-modal generation of image/text pairs?
2. How effective is classifier-free guidance in conditional generation for AR models and for UniDisc?
3. How does UniDisc compare against AR models in terms of training efficiency with varying the ratio of image-text tokens?
4. How do various sampling strategies for UniDisc affect its generation results and inference speed?
5. How does UniDisc compare against AR models across image-language reasoning tasks?
6. How do various design choices of UniDisc contribute to its performance?

Lastly, we show that we can successfully scale UniDisc, to a 1.4B parameter model, trained on 500B tokens. We qualitatively evaluate this model, to demonstrate its capabilities.

Datasets: In Section 2.5.1, 2.5.2, and 2.5.3, we conduct experiments with different train and validation sets. Our training set includes DataComp1B [Gad+24], CC12M [Cha+21], CLEVR-math [LA22], and CLEVR-Ref [Liu+19]. Our evaluation datasets include a held-out validation set of DataComp1B and CC12M, along with Flickr [Plu+16], MS-COCO30k [Che+15], and Winoground [Thr+22].

2.5.1 Evaluation of Multimodal Generation

We evaluate UniDisc and AR models in unconditional and conditional generation tasks.

Evaluation metrics: We consider the following three evaluation metrics, most commonly

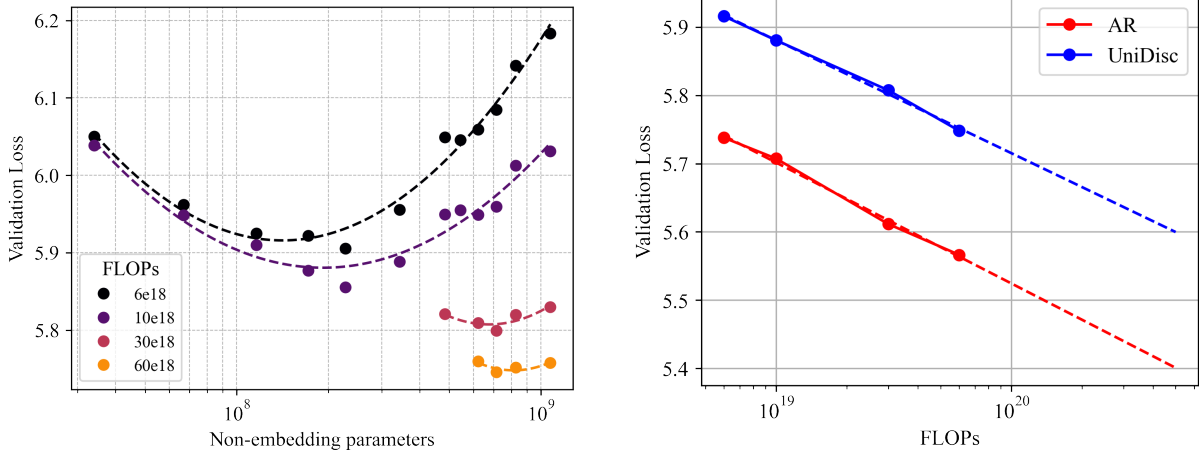


Figure 2.3: **Scaling Analysis for AR and UniDisc models:** (Left) IsoFLOP curves for UniDisc, plotting varying model size for a fixed FLOP budget. (Right) Estimating optimal parameter size for each budget - minima of fitted parabola, we plot scaling laws for both AR and UniDisc. We find 13.2x more compute is required for UniDisc to achieve the same overall loss as AR.

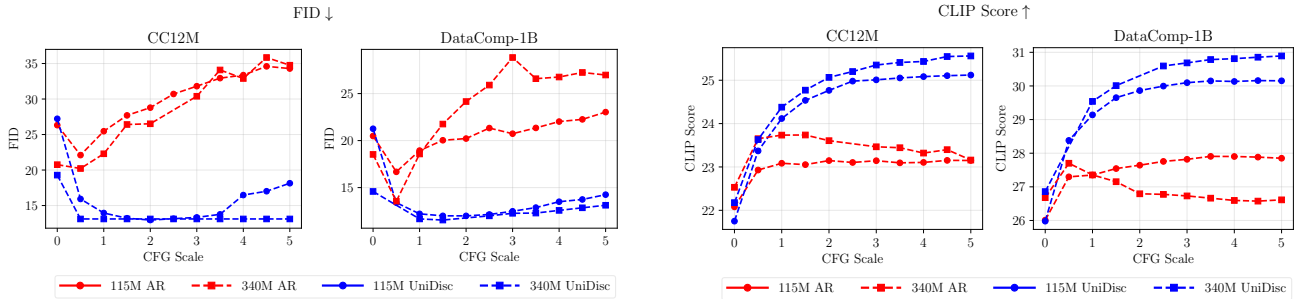


Figure 2.4: **Conditional generation results for both FID and CLIP metrics, across a range of CFG values.** We find that AR is more sensitive to the CFG weighting, with a narrower optimal range.

used in previous works: **i) Joint perplexity** indicates a model’s ability to fit to different validation sets. Note that this metric is jointly calculated across image-text tokens. The perplexity values from the autoregressive Chameleon baseline are exact likelihoods, the values from UniDisc are upper bounds. While perplexity is a good metric for assessing the fitting ability of a model, it cannot be used to evaluate its generation ability. **ii) Fréchet inception distance (FID)** [Heu+17] is a popular metric in image-generation to quantify the quality and diversity of image generation. **iii) CLIP score** is used for calculating image-text coherence. While we could not find an equivalent to the FID metric for text, we use CLIP score to evaluate generated image-text coherence, conditioning our model on an input image.

Experimental details:

We show conditional image-text generation results in Figure 2.4. We condition on an image to generate the corresponding language description, and vice versa, condition on

the language description to generate the corresponding image. For unconditional results please refer to [Figure 2.10](#) in the Appendix. We use a dataset comprising 30M image-text pairs from DataComp1B [\[Gad+24\]](#) and CC12M [\[Cha+21\]](#), please refer to [Section 2.8.1](#) for further details.

UniDisc significantly outperforms AR in conditional generation while performing equally well in unconditional generation). We attribute this performance gap to classifier-free guidance (CFG). As can be seen in [Figure 2.4](#), the results without CFG (Scale = 0) are similar between AR and UniDisc, but increasing CFG disproportionately benefits UniDisc.

The iterative generation process of diffusion makes it easy to blend conditional and unconditional predictions to guide the output. Autoregressive models, on the other hand, generate data sequentially in a fixed order, without any iterative refinement, which makes it difficult to mix in unconditional predictions to guide generation. We study this in detail in [Section 2.8.8](#) in Appendix.

Joint Image-Text Inpainting In [Figure 2.1](#), we show that UniDisc can inpaint in a joint text and image space—without any fine-tuning. Currently, none of the popular generative models have this capability, because most multimodal generative models are either autoregressive [\[Tea24\]](#) or use mixed modeling [\[Zho+24\]](#), which prevents them from easily inpainting jointly over image and text. In [Section 2.8.1](#) we explicitly fine-tune an AR model for joint inpainting, UniDisc zero-shot still shows far better performance. For more qualitative results, please refer to [Figure 2.18](#), and [Figure 2.19](#) in the Appendix.

2.5.2 Training Efficiency and Inference Speed

With the ever growing scale of recent generative models, an important aspect of their performance is their compute efficiency. Prior works [\[Kap+20; Hof+22\]](#) have extensively measured the training scaling laws of AR models, finding a power-law relationship between compute cost and distribution fitting, measured by negative log likelihood (NLL). In contrast, there has been little work that has measured the training efficiency of discrete diffusion models: the closest work is that of [\[GH24\]](#), which finds that the training efficiency of continuous diffusion models is approximately 64x worse than AR models on text. Recently, concurrent work [\[Zhe+24b\]](#) studied discrete diffusion models, again only on text, and found a scaling factor of 16x compared to AR models.

Although discrete diffusion is thought to be comparatively more efficient on other modalities such as images, we are not aware of prior work that has empirically measured this. We perform an ISO Flop analysis [\[Hof+22\]](#) of UniDisc and our AR baseline, changing only the attention mask and loss function. As in prior work, we select a set of compute budgets C_i and, within each budget, vary the non-embedding parameters (incl. LM head) N , and total tokens during training D , keeping the total compute, measured in FLOPs, fixed using an approximation of $C \approx 6ND$.

We compare the training efficiency of UniDisc and our AR baseline in [Figure 2.3](#) (right) and find that the rough training-inefficiency factor for discrete diffusion to that of AR models for unified training is about 13.2—i.e. one needs to train UniDisc 13.2x longer to achieve the same loss. Additional experimental details are available in [Section 2.8.2](#).

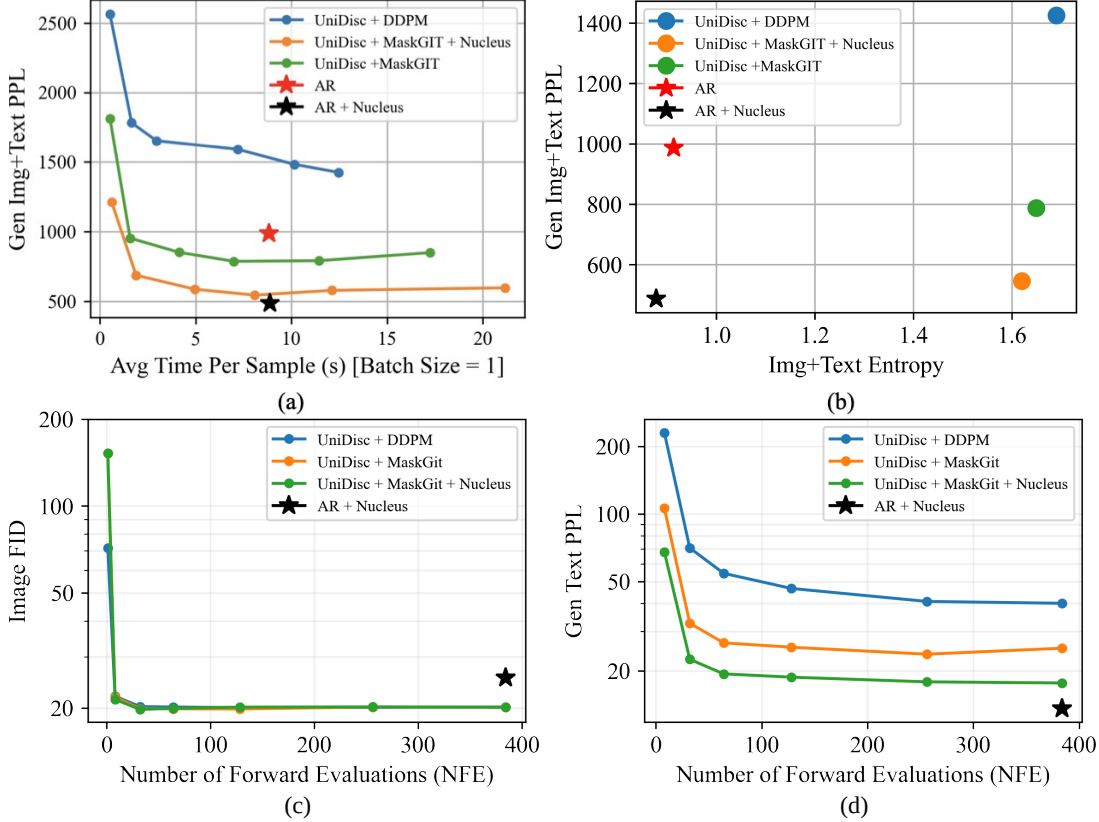


Figure 2.5: **Inference Comparisons for UniDisc and AR baseline:** (a) Chameleon Perplexity (Text+Image) vs. Time - we perform similar to best AR method, (b) Chameleon Perplexity vs. Entropy - UniDisc has high diversity and low perplexity, while AR has significantly lower diversity, (c) Image FID vs. NFE, showing image generation saturates quickly with NFE (≈ 32), (d) GPT2 Generative Text Perplexity vs. NFE showing text generation benefits from more sampling steps (diminishing).

While training efficiency is important, inference efficiency is equally—if not more—important as we deploy these models at wide scale. Thus, we compare the inference efficiency of UniDisc and our AR baseline in Figure 2.5 (a), (c) and (d). In (a), we measure the joint generative perplexity using Chameleon, In (c) we measure the Image FID and in (d) we measure the Text Perplexity. While it might appear from (a) and (d) that AR does better than UniDisc, in Figure 2.5 (b), we find that UniDisc has far higher entropy at a given perplexity.

We note that solely looking at the generative perplexity is not sufficient, as it has been previously found [Zhe+24b] that very low perplexity can be achieved by repeating the same tokens, which we find often happens with AR w/nucleus sampling and low temperature. We demonstrate such degenerate cases in Section 2.8.1. Therefore, Generative Perplexity + Entropy should be considered jointly for evaluating the quality of generation results.

2.5.3 Multimodal Discriminative Performance

Generative models can act as strong discriminative models as shown in several recent works [Li+23a; JCG24; Pra+23]. Moreover, [RM23] show the discriminative ability of a generative model can be a good metric for its generation performance. In this section, we compare the discriminative capabilities of AR models and UniDisc on cross-model retrieval tasks (image/text/joint).

We evaluate on Winoground [Thr+22] and a held-out DataComp1B validation set [Gad+24], using 18M text/image pairs from DataComp1B as our training set. To enable text retrieval during inference for the AR model, we train with flipping the order of modalities, putting the image first 20% of the time, following [Zho+24]. We find that this improves the retrieval for the AR model. All other hyperparameters follow those in Section 2.5.1. Details of evaluations on CLEVR-VQA and CLEVR-Ref are available in Section 2.8.1

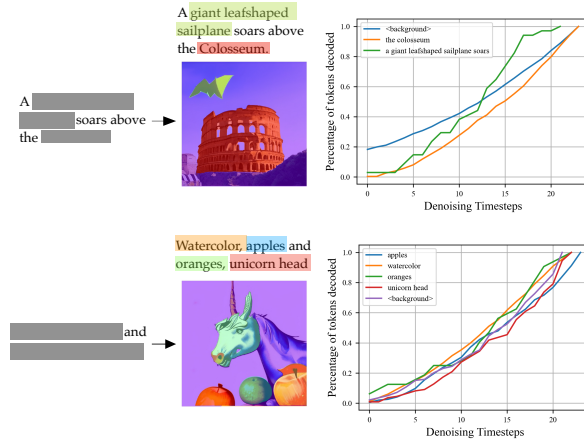


Figure 2.6: **Uniform Concept Generation:** We perform joint generation given only masked text input (left). We use a language conditioned segmentation model and find that UniDisc generates uniformly in *concept space* (right).

	Clevr-VQA	Clevr-Ret	Datacomp	Winoground
Text Retrieval				
AR	0.60	0.81	0.85	0.24
UniDisc	0.63	0.94	0.85	0.31
Image Retrieval				
AR	N/A	0.06	0.96	0.25
UniDisc	N/A	0.25	0.95	0.27
Joint Retrieval				
AR	N/A	0.06	0.17	0.06
UniDisc	N/A	0.5	0.64	0.20

Figure 2.7: **Image-Text Reasoning measured by QA and retrieval accuracy across datasets.**

In Figure 2.7, we report the image retrieval, text retrieval and joint retrieval accuracy for AR and UniDisc. For image retrieval, the model is given a text caption paired with 16 images, out of which only one image is correctly paired and the rest are random. The goal is to accurately classify the correct image. To evaluate the model’s retrieval accuracy we check if the correct image has the highest $p(x^{img}|x^{txt})$ among all other images. We do the same for text retrieval, where we check $p(x^{txt}|x^{img})$. For joint retrieval, only a single pair has the correct mapping, and every other pair has a random image and text. We check if the correct pair has the highest joint probability $p(x_{img}, x_{txt})$.

We find that UniDisc significantly outperforms the AR model on all retrieval tasks. To further investigate this, we measure the joint retrieval accuracy across denoising steps & CFG values in Figure 2.13 in Appendix. We find CFG and the number of denoising steps to play a large role in UniDisc’s retrieval accuracy. While the number of denoising steps in

an AR model is fixed to the sequence length, the denoising steps for UniDisc can be much higher.

2.5.4 Scaling UniDisc

We show that UniDisc scales well across parameters and dataset size. We train a 1.4B parameter model with web-scale data. Our model is trained in two stages, with a low-resolution pre-training stage and a second high-resolution fine-tuning stage. Our first-stage consists of 250M image/caption pairs at 256×256 resolution. We curate our dataset from several sources, with 200M open-web images from [Gad+24], which were re-captioned by a VLM to create higher-quality descriptions by [Li+24b]. We also add a set of smaller datasets consisting of PixelProse [Sin+24], JourneyDB [Sun+23], and Cambrian-10M [Ton+24]. In addition, we construct a high-quality, custom dataset of 18M synthetic images, following findings by [Zhu+24c; Seh+24] on the importance of image/caption alignment for image generation. We construct our dataset by prompting an LLM to augment a set of 250K human prompts and use [Ess+24] for generations. In both stages, we account for dataset imbalance and sample more from higher-quality sources. Finally, we fine-tune our model in a second stage, interpolating the RoPE 2D embeddings to train at 512×512 on 30M image/caption pairs.

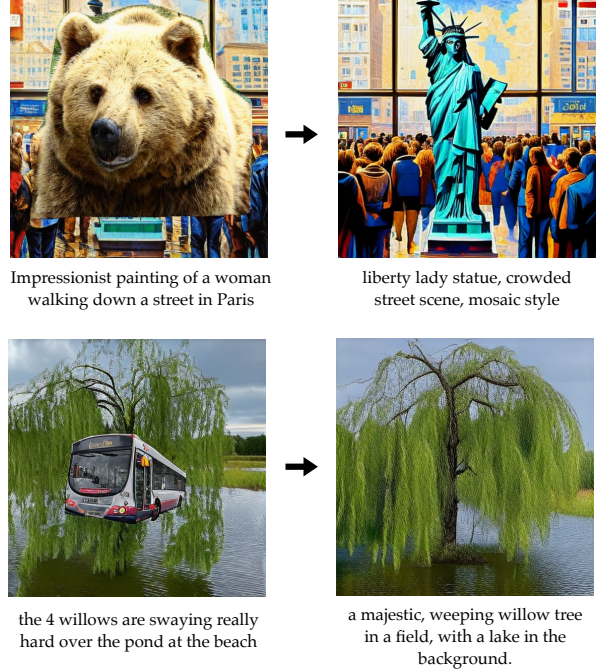


Figure 2.8: **Zero-shot Image Editing:** UniDisc can take corrupted and mismatched image/text pairs (left) and produce an aligned, high-quality pair (right), using the model’s own likelihood as a scoring function.

Further due to lack of space, we ablate several architecture and objective design choices on a smaller model in Section 2.8.5 and we show the training curve of our 1.4B model is available in Figure 2.15. We also compare UniDisc to recent multimodal models on standard image generation benchmarks in Section 2.8.7. Qualitative results from the model are available in Section 2.8.6, demonstrating zero-shot text-conditioned image inpainting (Figure 2.18), standard text-to-image (Figure 2.16), and image-to-text (Figure 2.17) generation. Moreover, we demonstrate a form of image editing in Figure 2.8 and Section 2.8.6, showing that UniDisc can, without any specialized fine-tuning, *automatically* improve a text & image pair by noising and denoising, using the model’s likelihood as a judge. Additionally, we analyze the joint generation of UniDisc in Figure 2.6 and Section 2.8.6, finding that the model generates images roughly *uniformly in concepts instead of in area*.

2.6 Conclusion

In this chapter, we introduced UniDisc, the first large-scale unified multimodal discrete diffusion model capable of generating, inpainting and editing both images and text. By leveraging discrete diffusion processes, we showed that UniDisc surpasses autoregressive models in both inference efficiency and quality. Our model unifies various design choices in discrete diffusion space, across modalities, through extensive ablations and analysis. We hope that our work inspires future research in this direction.

2.7 Acknowledgments

This chapter was authored jointly with Mihir Prabhudesai and contributed to by: Sidharth Gandhi, Deepak Pathak, Katerina Fragkiadaki.

This work was funded by LambdaLabs, ONR award N00014-23-1-2415, AFOSR FA9550-23-1-0257, AFOSR FA9550-23-1-0747, ONR MURI N00014-24-1-2748, ONR MURI N00014-22-1-2773 and DARPA No. HR00112490375 from the U.S. DARPA Friction for Accountability in Conversational Transactions (FACT) program. We thank CMU FLAME Cluster, LambdaLabs and Google TPU Research cloud for providing us with compute for this work. We also thank Alex Li, Unnat Jain and Lili Chen for helpful discussion in regards to the project.

2.8 Appendix

UniDisc Training

We describe the detailed algorithm for unified discrete diffusion training on image and text below in Algorithm 1.

Algorithm 1 UniDisc Training

```
1: Require: Training data  $x$ 
2: Require: Noising Schedule  $\alpha_t$  i.e., Linear or Cosine
3: Require: Unconditional probability  $p_{uncond}$ 
4: Initialize: Model parameters  $\theta$ 
5: repeat
6:    $[x_0^{img}, x_0^{txt}] = x_0 \sim p(x, c)$  ▷ Sample image and text data
7:    $t \sim \mathcal{U}(0, 1)$  ▷ Sample random timestep
8:
9:    $x_t^v \sim q(x_t^v | x_0^v) = \alpha_t x_0 + (1 - \alpha_t) e_m$  for  $v \in \{\text{img}, \text{txt}\}$  ▷ mask all tokens
10:
11:   With probability  $p_{uncond}$  ▷ For Classifier-Free Guidance:
12:     If  $\text{rand}() < 0.5$ : ▷ Randomly set one of the modalities to mask tokens
13:        $x_t^{img} \leftarrow m$ 
14:     Else:
15:        $x_t^{txt} \leftarrow m$ 
16:    $x_0^{pred} = p_\theta([x_t^{img}, x_t^{txt}])$  ▷ Estimate model prediction from masked sequence
17:   Compute loss as:  $\mathcal{L}_{\text{diff}} = \frac{\alpha'_t}{1 - \alpha_t} \log \langle x_0^{pred}, x_0 \rangle$  ▷ Loss function over the logits of inputs
18:   Perform gradient step on  $\mathcal{L}$  to update  $\theta$ 
19: until converged
```

Sampling Algorithms

Here we describe the implementations of UniDisc’s sampling algorithm and MaskGIT [Cha+22].

Algorithm 2 MaskGIT Sampling

1: **Initialize:** $x_T \leftarrow [m, m, \dots, m]$ ▷ All tokens are masked
2: **Require:** Sampling steps T
3: **Require:** Num Tokens to Unmask: $f(t)$. We set $f(t)$ as $\frac{1-\alpha_t}{\sum_{t=1}^T 1-\alpha_t}$
4: **for** $t = T$ **down to** 1 **do**
5: $p_{x_0} \leftarrow p_\theta(x_0 \mid x_t)$ ▷ Model prediction
6: $p_{x_0}^{(p)} \leftarrow \text{Top}_p(p_{x_0})$ ▷ Top- p (Nucleus) sampling on logits
7: $p_{x_0}^{(k)} \leftarrow \frac{p_{x_0}^{(k)}}{\tau(t)}$ ▷ Apply temperature annealing
8: Sample $x_{\text{new}} \sim \text{Categorical}(p_{x_0}^{(k)})$ ▷ Sample new tokens
9: $M \leftarrow \lfloor f(t) \times N \rfloor$ ▷ Determine number of tokens to unmask
10: Select M most confident tokens based on $p_{x_0}^{(k)}$
11: Update $x_{t-1}[i] \leftarrow x_{\text{new}}[i] \quad \forall i \in \text{selected positions}$
12: **Keep** previously unmasked tokens unchanged
13: **end for**

2.8.1 Additional Experiment Details

Conditional and Unconditional Experiment Details

For unconditional and conditional results in Figure 2.10 and 2.11 we use a dataset of 11B tokens comprising 30M images from DataComp1B [Gad+24] and CC12M [Cha+21] as our training set, with a fraction of 20% text tokens and 80% image tokens after excluding pad tokens. For faster convergence, we train only on DataComp1B for results in Figure 2.5 and Figure 2.13. We tokenize the image and text tokens using separate tokenizers. We use lookup-free quantization (LFQ) from [Yu+23; Luo+24] for as our image tokenizer, and use the tokenizer from [Tou+23] as our text tokenizer. We use an image resolution of 256×256 , and a downsampling ratio of 16, resulting in a sequence length of 384 with 256 with image tokens and 128 text tokens. Note that we use the same tokenizers for all the baselines, ensuring fair comparisons. We train UniDisc for 300 L40S GPU hours and train the autoregressive model for a proportionate amount of time such that it achieves the same validation loss. Our model comprises 115M/340M non-embedding parameters and we use a batch size of 512, a learning rate of $3e-4$, and weight decay of 0.05, following [Sun+24].

Conditional and Unconditional Evaluations

We extend Figure 2.4, adding results on Flickr-30K and MS-COCO below in Figure 2.9. We show unconditional results in Figure 2.10 and conditional results (taking the optimal CFG weight for both UniDisc and the AR model) in Figure 2.11.

Generative Perplexity — Qualitative

Text	Chameleon Perplexity	GPT2 Perplexity
"ICLR is globally renowned for presenting..." (Continued)	32.836	35.780
"This is simple. This is simple." (Repeated)	8.423	3.930
"Words Words Words Words" (Repeated)	2.226	3.583
"AAAAAAAAAAAAA" (Repeated)	2.732	1.904
"(Spaces Repeated)"	80.240	1.095

Table 2.1: We demonstrate how generative perplexity is an imperfect metric requiring calibration with entropy.

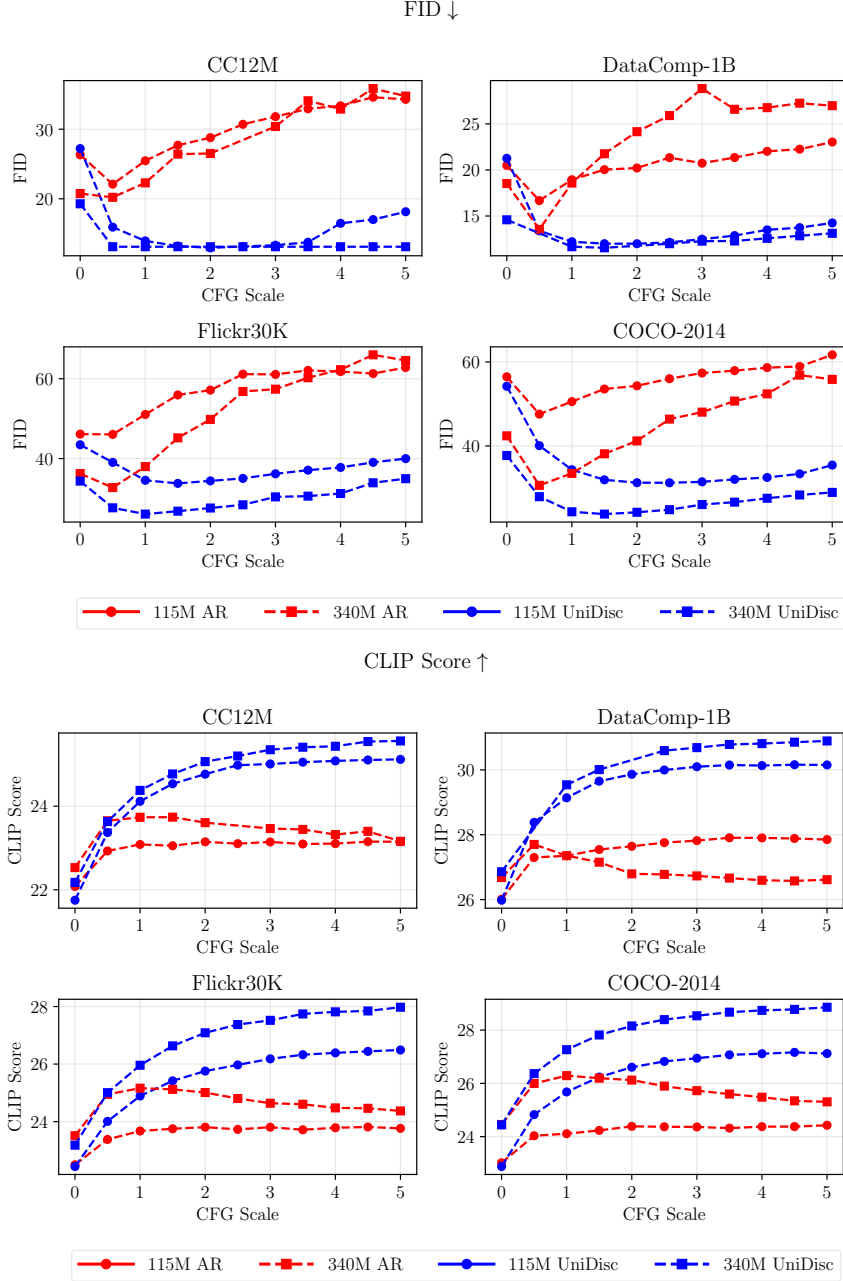


Figure 2.9: **Conditional generation results for both FID and CLIP metrics, across a range of CFG values.** We find that AR is more sensitive to the CFG weighting, with a narrower optimal range.

Quantitative Inpainting Comparison w/autoregressive models

To demonstrate the tradeoff between the pre-training objectives of UniDisc and AR models, we evaluate both models on inpainting. We fine-tune the 340M parameter AR model on a standard set of multimodal datasets (CC12M, Recap-DataComp-1B, LAION 400M) and evaluate UniDisc in a zero shot manner—without any fine-tuning. Specifically, for the AR model, we use a linear masking schedule for the prefix sequence consisting of a randomly

	CC12M	DataComp	Flickr	MS-COCO
Image + Text Perplexity				
Chameleon	541.2	156.8	1254.9	1128.3
UniDisc	494.5	154.8	1115.0	982.2
Image - FID				
Chameleon	30.5	20.49	75.70	70.67
UniDisc	35.78	22.97	88.88	77.43
Text - CLIP				
Chameleon	23.70	26.08	23.70	23.64
UniDisc	25.01	25.98	24.92	25.01

Figure 2.10: Unconditional multimodal generation results for UniDisc and AR baseline at 115M parameters - both models perform similarly.

	CC12M	DataComp	Flickr	COCO
Text to Image - FID				
Chameleon 115M w/o CFG	26.32	20.49	46.13	56.46
Chameleon 340M w/o CFG	20.75	18.53	36.24	42.41
Chameleon 115M w/ CFG (0.5)	22.10	16.68	46.06	47.58
Chameleon 340M w/ CFG (0.5)	20.22	13.55	32.74	30.62
UniDisc 115M w/o CFG	27.22	21.26	43.46	54.21
UniDisc 340M w/o CFG	19.28	14.59	34.37	37.73
UniDisc 115M w/ CFG (1.5)	13.21	12.00	33.79	31.94
UniDisc 340M w/ CFG (1.5)	13.11	11.55	26.83	23.77
Image to Text - CLIP				
Chameleon 115M w/o CFG	22.08	26.01	22.50	23.02
Chameleon 340M w/o CFG	22.53	26.68	23.51	24.46
Chameleon 115M w/ CFG (0.5)	22.93	27.30	23.38	24.03
Chameleon 340M w/ CFG (0.5)	23.65	27.70	24.95	25.99
UniDisc 115M w/o CFG	21.75	25.98	22.44	22.88
UniDisc 340M w/o CFG	22.18	26.86	23.18	24.44
UniDisc 115M w/ CFG (1.5)	24.54	29.65	25.42	26.24
UniDisc 340M w/ CFG (1.5)	24.77	30.01	26.63	27.82

Figure 2.11: Conditional generation results for UniDisc and AR baseline. Our model significantly outperforms the AR model when classifier free guidance is used.

masked text and image pair and then predict and supervise the clean sequence, doubling the overall sequence length. In Figure 2.12, we evaluate at multiple noise levels, showing the degradation in performance as the original sequence is increasingly masked.

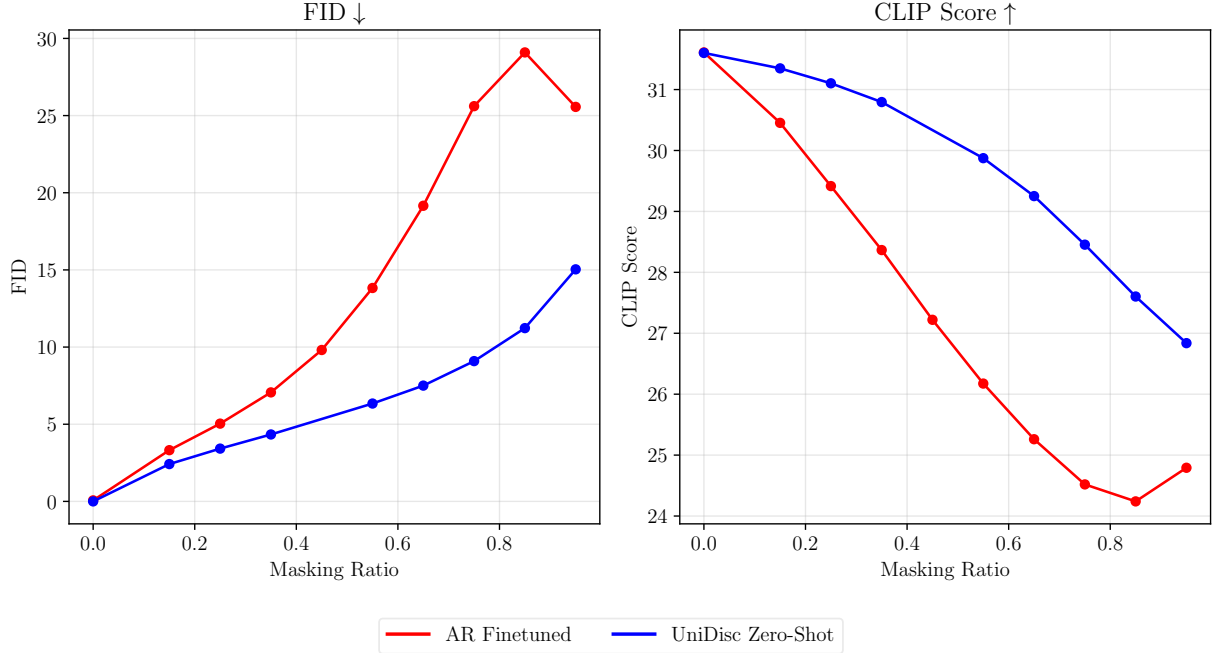


Figure 2.12: We compare UniDisc with an AR model fine-tuned for joint inpainting and evaluate on a subset of DataComp1B.

Discriminative Evaluations

For evaluations on CLEVR-VQA and CLEVR-Ref [Liu+19] we use their respective train-val splits. Note that for CLEVR-VQA and CLEVR-Ref, we do not follow the training scaling factor found in Figure 2.3, we instead train both the models until convergence, i.e. multiple epochs. The small size of these datasets makes it possible to train until convergence. For CLEVR images, we find that none of the existing tokenizers work well, so we fine-tune our own tokenizer on CLEVR images. We use images of 128×128 resolution, with a total sequence length of 320 (256 image tokens and 64 text tokens). For text, we use a standard BERT tokenizer [Dev+19]. In Figure 2.13, we ablate the role CFG and the number of denoising steps play in UniDisc’s retrieval accuracy. While the number of denoising steps in an AR model is fixed to the sequence length, the denoising steps for UniDisc can be much higher.

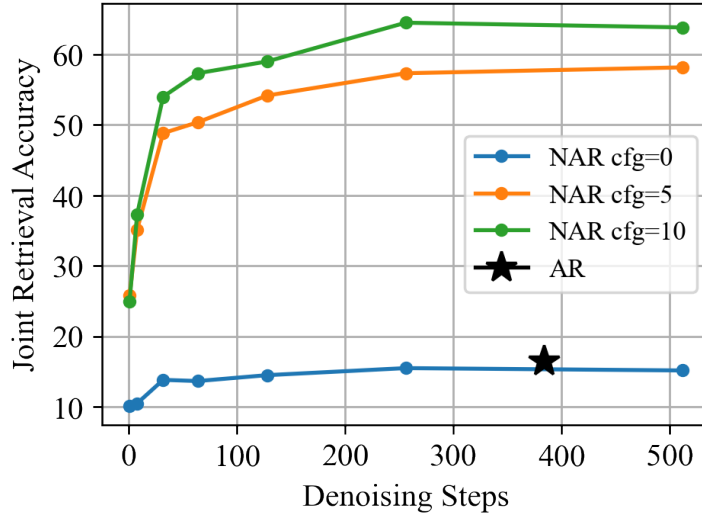


Figure 2.13: **Joint Retrieval Accuracy on DataComp1B.** We outperform AR given the task of retrieving one correct image-text pair out of 16 possible pairs, implying better learnt representations.

2.8.2 Scaling Experiment Details

As in prior experiments, all implementation details are shared between UniDiscand AR training configurations except: (1) causal vs. full attention, (2) masking of the input sequence for UniDisc, and (3) weighting the CE loss as in Equation (2.3).

We use similar hyperparameters as in our small-scale experiments in Section 2.5.1, but repeat them here for clarity. For images, we use lookup-free quantization (LFQ) [Yu+23; Luo+24] with a image resolution of 256×256 , and a downsampling ratio of 16, resulting in a sequence length of 256 image tokens. We use a BPE tokenizer [Tou+23] for text with 128 text tokens, resulting in a total sequence length of 384. We report only non-embedding parameters and data tokenization is identical across all models.

We use a batch size of 512 and use AdamW [LH19] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay $\lambda = 0.05$. We use a max learning rate of $3e-4$ with a linear warmup followed by a cosine decay schedule, ending with zero at the final step for a given training run.

We list all model variants in Table 2.2.

Parameters (M)	n_layers	n_heads	d_model
34	11	6	384
67	11	9	576
116	12	12	768
172	20	12	768
228	20	14	896
343	24	16	1024
484	22	10	1280
543	17	12	1536
622	29	10	1280
713	23	12	1536
826	27	12	1536
1074	26	14	1792
1290	24	16	2048

Table 2.2: Model variants. The FFN hidden size is always 4x the overall d_model

2.8.3 Training Details

Additional Training Implementation Details

We use flash attention for all models except as noted below, using the popular FlashAttention 2 library [Dao23]. For all AR models at inference, we use K/V caching and take advantage of specially optimized functions for this in FlashAttention 2.

2.8.4 Fine-tuning An Autoregressive model for Discrete Diffusion

As we already have a plethora of large-scale AR models [Tou+23; Tea24], it would be useful to have the ability to fine-tune them for a discrete diffusion objective. While the naive method for fine-tuning would be to change the objective function to discrete diffusion while using AR’s pre-trained weights. We find that a better idea is to left-shift the output targets of the diffusion objective such that instead of having the masked token predict its respective visible token, we have the token before the masked token predict it. In this way, we more closely match the original AR next-token prediction objective. In Figure 2.14 we show that this strategy works well and we can effectively fine-tune a pre-trained autoregressive language model using discrete diffusion loss. We demonstrate this result on a 270M parameter language model [Meh+24], OpenELM, which is trained with an AR objective. We compare against training from scratch and training AR without the shift. We find the shifting strategy converges faster.

Large Scaling Training Curve

We show the training curve for the large scale experiments described in [Section 2.5.4](#) in [Figure 2.15](#).

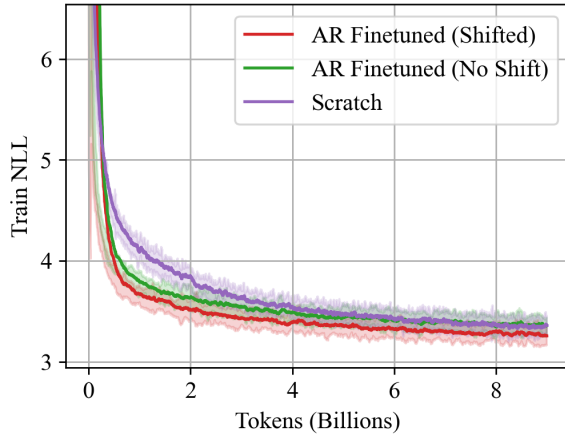


Figure 2.14: Fine-tuning a pre-trained 270M parameter AR model on LM1B.

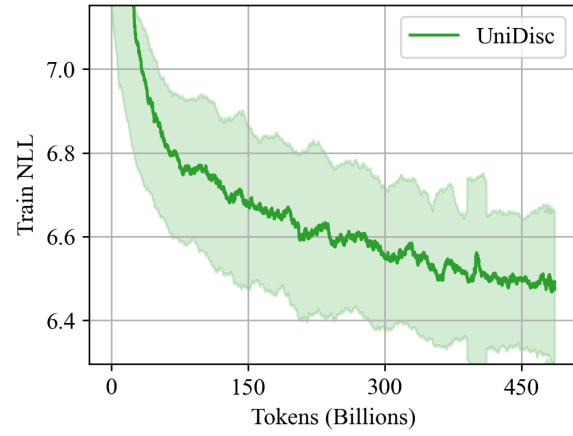


Figure 2.15: Training Loss Curve vs. Tokens on our 1.4B model.

2.8.5 Ablations

We validate our design choices by running small-scale experiments on a subset of our primary dataset, taking 18M image/caption pairs on DataComp1B. We train on lower-resolution images at 128×128 and obtain a 1:1 ratio of text to image tokens, with 64 text and 64 image tokens for a total sequence length of 128, with all other hyperparameters the same as in our primary experiments.

We examine the influence of several design choices for our model in Table 2.3 and reach several conclusions. First, architecture changes to improve training stability—namely adding QK Normalization and using RMSNorm instead of LayerNorm—do not substantially affect convergence in this setting.

Another natural design choice is to parameterize the model such that we provide the modality of a given token to the model. With this relaxation we can drastically reduce the output space and, in theory, simplify the objective for our model. However, we find that this reparametrization only marginally reduces overall perplexity, even at this smaller-scale. We hypothesize that the modality-specific embeddings added to each token allows the model to learn the correct output space with minimal added parameters.

DataComp1B Validation PPL	
UniDisc	93.8
w/o QK Norm	92.7
w/ Zero-linear init	93.8
w/o RMSNorm	93.8
w/o -inf for invalid tokens	94.7
w/o Softmin SNR	109.6
None	111.2

Table 2.3: Ablation w/115M parameter model of QK Norm, zero initialization of linear layers, RMSNorm, setting invalid tokens to $-\infty$ during training and generation, and Softmin SNR.

DataComp1B Validation FID	
UniDisc	11.4
w/cosine noising schedule	11.5
w/o CE loss weighting	11.35
w/discrete time (T=1000)	13.8

Table 2.4: Ablation w/115M parameter model on different objective level decisions such as noising schedule, loss weighting and whether to use discrete time.

2.8.6 Large Scale Qualitative Results

We show additional results on tasks such as joint inpainting, image captioning and image generation. We note that none of these tasks were explicitly trained or optimized for by our model. This is an intrinsic property due to the nature of UniDisc’s unified diffusion based objective. In Figure 2.16 we show standard text-to-image generation and in Figure 2.17 we show standard image-to-text generation. In Figure 2.18 we show zero-shot text-conditioned inpainting, and in Figure 2.19 we show zero-shot *multimodal* inpainting.



A dragon with iridescent scales basking in the warm sunlight on a beach.



Cel shaded, Planet of the Apes character, minimalist icon



A high-tech motorbike crafted from neon-lit circuit boards and scarlet glass, emitting sparks that resemble a swarm of fireflies.



A cyberpunk female hacker, dressed in a neon pink hoodie and ripped jeans, poses with a sleek laptop and a futuristic energy blaster, set against a gritty cityscape at night.



a majestic elephant observing a sly fox attempting to steal a ball from a playful raccoon



Watercolor, apples and oranges, unicorn head



Ancient Greek-inspired armor made entirely out of Lego pieces, with a blurred ancient cityscape in the distance



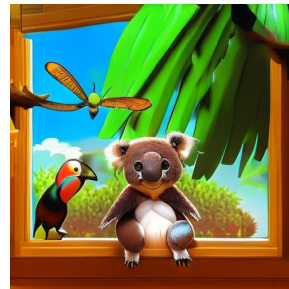
an overflowing crate of assorted summer berries



a sleek, modern train gliding across a snowy mountain pass



a reindeer is sitting on a snowy mountain and has on warm ski goggles



A dragonfly zooms out of a window, a koala climbs up a palm tree, and a toucan sits on a branch.



a medley of autumnal leaves and fallen fruits

Figure 2.16: UniDisc’s ability to generate an image, given unseen text as input.



Figure 2.17: UniDisc’s ability to generate text (captioning), given unseen image as input.

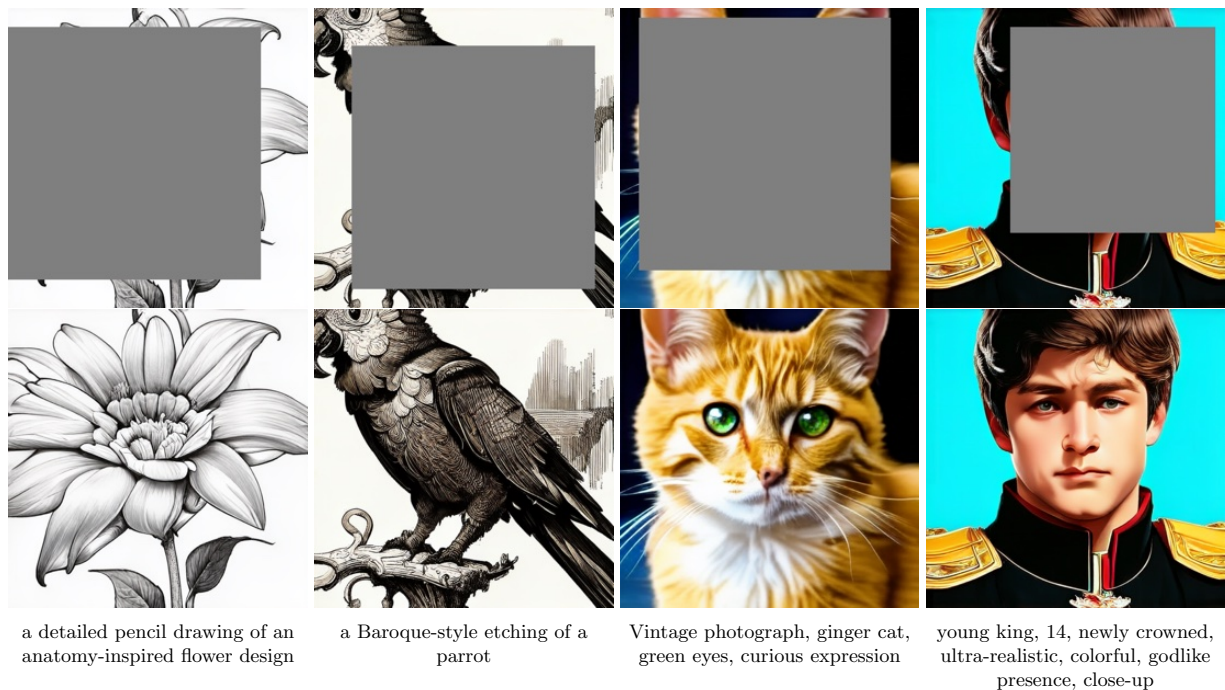


Figure 2.18: Zero-shot text-conditioned inpainting. UniDisc inpaints a masked region given a user-provided text prompt.

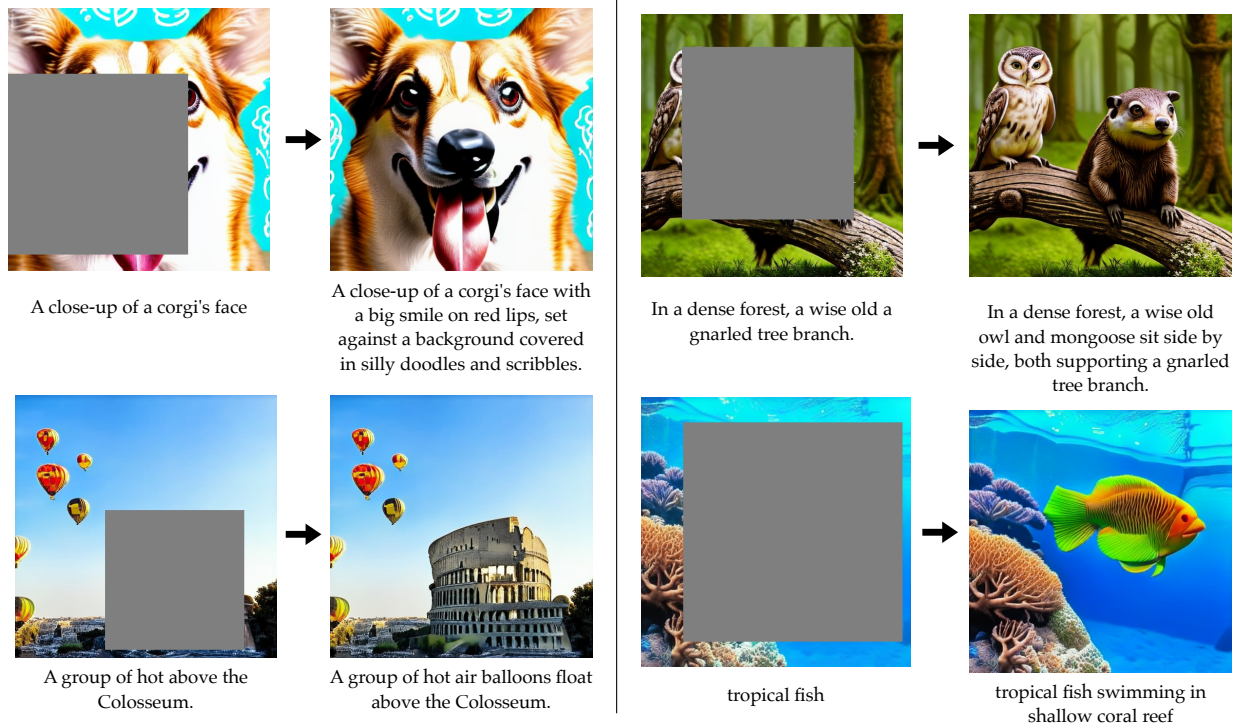


Figure 2.19: Zero-shot multimodal inpainting. UniDisc jointly inpaints in both image and text spaces.

Zero-shot image editing of UniDisc

A clear benefit of diffusion models is the ability to perform zero-shot editing without specific paired data—which is often difficult to obtain. We demonstrate one such method in Figure 2.20, showing that UniDisc can automatically improve a user provided image and caption.

We augment real images by overlaying random objects from the COCO dataset. Similarly, we augment captions by asking an LLM to generate purposely incorrect variations. We then randomly mask the image and text inputs and unmask as described above, automatically removing these undesired image artifacts and generating the correct caption. We adopt a best-of-n sampling strategy with n distinct noise masks. We unroll each generation using the model’s own likelihood to select the best generation.

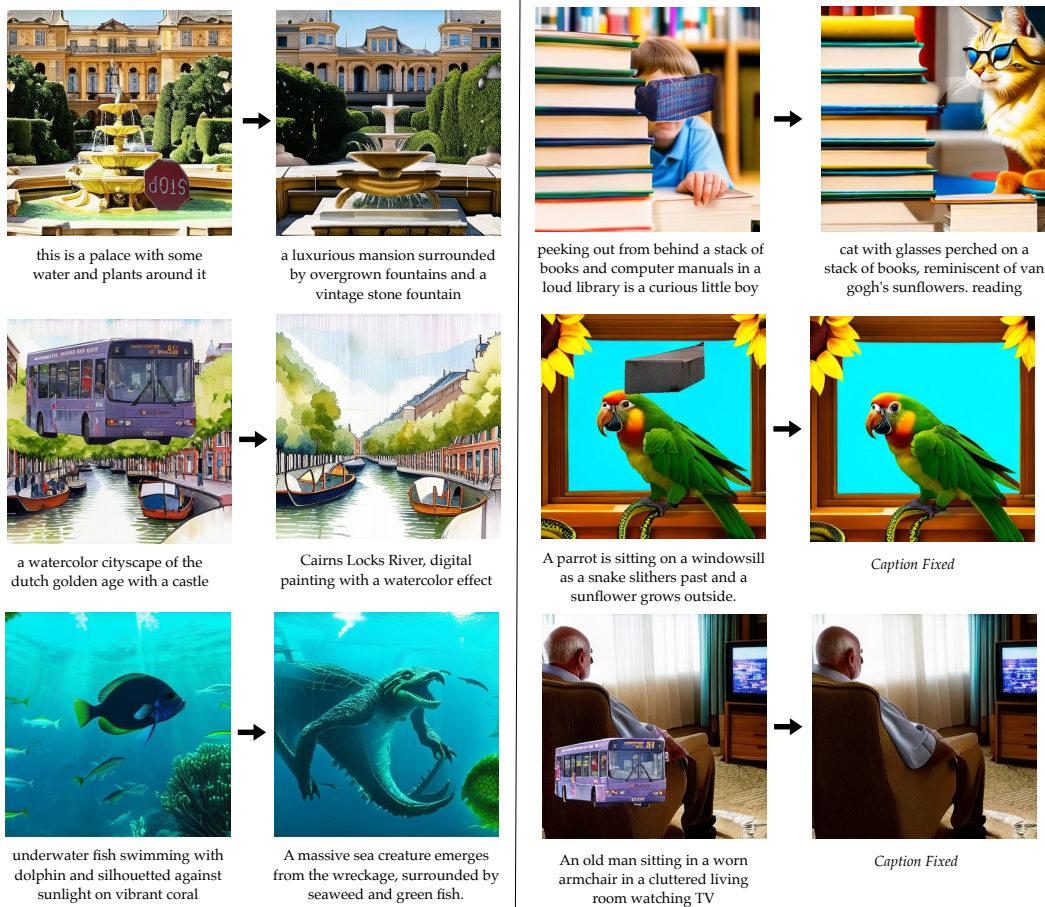


Figure 2.20: Zero-shot multimodal editing. We provide a *clean* image and text pair and UniDisc automatically enhances both the image and text. In the final row, we fix the text and allow only the image to change.

Analyzing the joint image-text generation of UniDisc

In Figure 2.21, we visualize how the model iteratively infills both image and text. This raises the question - does UniDisc follow a certain strategy during generation (for example, generating entire background first then moving to subject or generating text first before image), or does it generate everything at once jointly. To analyze this, we take the final model generated image, semantically segment it (using Grounded SAM 2 in our case) and then see which concepts get generated at what timesteps. This is visualized in Figure 2.22. We find that UniDisc generates all concepts at once proportional to the overall fraction of the image the concept occupies. We also investigated if the UniDisc has any strong positional bias, such as first generating tokens in the middle and radially filling out. However we find no such positional strategy and that UniDisc is positionally invariant. Intuitively, this means that at any denoising step, all positions are equally likely to be decoded.

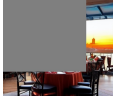
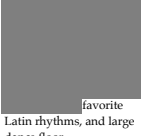
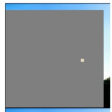
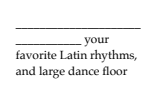
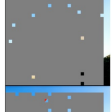
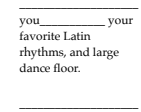
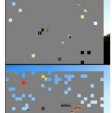
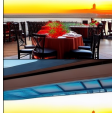
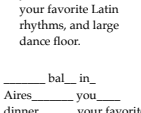
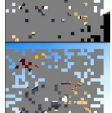
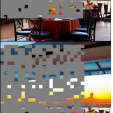
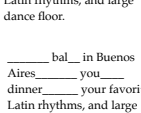
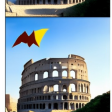
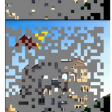
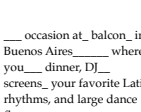
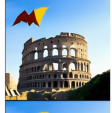
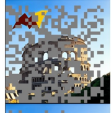
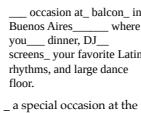
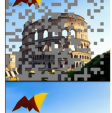
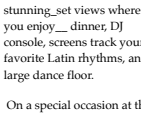
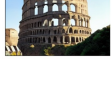
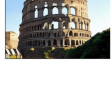
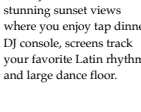
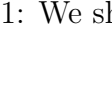
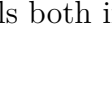
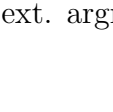
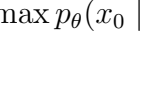
		A group of hot air balloons racing above the Colosseum.		A _____ racing above the Colosseum.		Luxury has invaded Alcatraz Island in the form of a high-end nightclub with fine dining options, live music performances that celebrate your favorite Latin rhythms, and large dance floor.			_____ favorite Latin rhythms, and large dance floor.
Denoising Steps	$\arg \max p_{\theta}(x_0 x_t)$	x_0		$\arg \max p_{\theta}(x_0 x_t)$	x_0				
1		A hot air balloon hovering above the Colosseum.		A _____ above the Colosseum.		A you a a rooftop _____ your favorite Latin rhythms, and large dance floor.			_____ your favorite Latin rhythms, and large dance floor
2		A hot air balloon hovering above the Colosseum.		A _____ above the Colosseum.		You're the the rooftop _____ at sunset, you can the the _____, your favorite Latin rhythms, and large dance floor.			you _____ your favorite Latin rhythms, and large dance floor.
3		A kite with a kite soaring above the Colosseum.		A _____ above the Colosseum.		You you're the rooftop _____ at sunset, you can a romantic dinner party, _____ your favorite Latin rhythms, and large dance floor.			you _____ dinner _____ your favorite Latin rhythms, and large dance floor.
7		A kite with a kite soaring above the Colosseum.		A _____ above the Colosseum.		You you're the on the balcony in Buenos Aires, you sunset, you can a romantic dinner party, _____ your favorite Latin rhythms, and large dance floor.			_____ bal _____ in _____ Aires _____ you _____ dinner _____ your favorite Latin rhythms, and large dance floor.
9		A kite with a kite soaring above the Colosseum.		A _____ above the Colosseum.		You you're the on a balcony in Buenos Aires, you sunset, you can a romantic dinner party, _____ your favorite Latin rhythms, and large dance floor.			_____ bal _____ in Buenos Aires _____ you _____ dinner _____ your favorite Latin rhythms, and large dance floor.
13		A kite with a kite soaring above the Colosseum.		A _____ above the Colosseum.		A special occasion at a balcony in Buenos Aires during a sunset, where you can a dinner, DJ, giant screens displaying your favorite Latin rhythms, and large dance floor.			_____ occasion at _____ bal _____ in Buenos Aires _____ where you _____ dinner, DJ _____ screens _____ your favorite Latin rhythms, and large dance floor.
16		A kite with a kite soaring above the Colosseum.		A _____ above the Colosseum.		Picture a special occasion at a balcony in Buenos Aires during the sunset, where you enjoy a rom dinner, DJing, screens displaying your favorite Latin rhythms, and large dance floor.			_____ occasion at _____ bal _____ in Buenos Aires _____ where you _____ dinner, DJ _____ screens _____ your favorite Latin rhythms, and large dance floor.
18		A kite with a kite soaring above the Colosseum.		A _____ above the Colosseum.		Picture a special occasion at the balcony in Buenos Aires with stunning sunset views where you enjoy a rom dinner, DJ console, screens track your favorite Latin rhythms, and large dance floor.			_____ a special occasion at the bal _____ in Buenos Aires _____ stunning _____ set views where you enjoy _____ dinner, DJ console, screens track your favorite Latin rhythms, and large dance floor.
24		A kite with a kite soaring above the Colosseum.		A _____ above the Colosseum.		On a special occasion at the balcony in Buenos Aires lies stunning sunset views where you enjoy tap dinner, DJ console, screens track your favorite Latin rhythms, and large dance floor.			On a special occasion at the balcony in Buenos Aires lies stunning sunset views where you enjoy tap dinner, DJ console, screens track your favorite Latin rhythms, and large dance floor.

Figure 2.21: We show how UniDisc jointly infills both image and text. $\arg \max p_{\theta}(x_0 | x_t)$

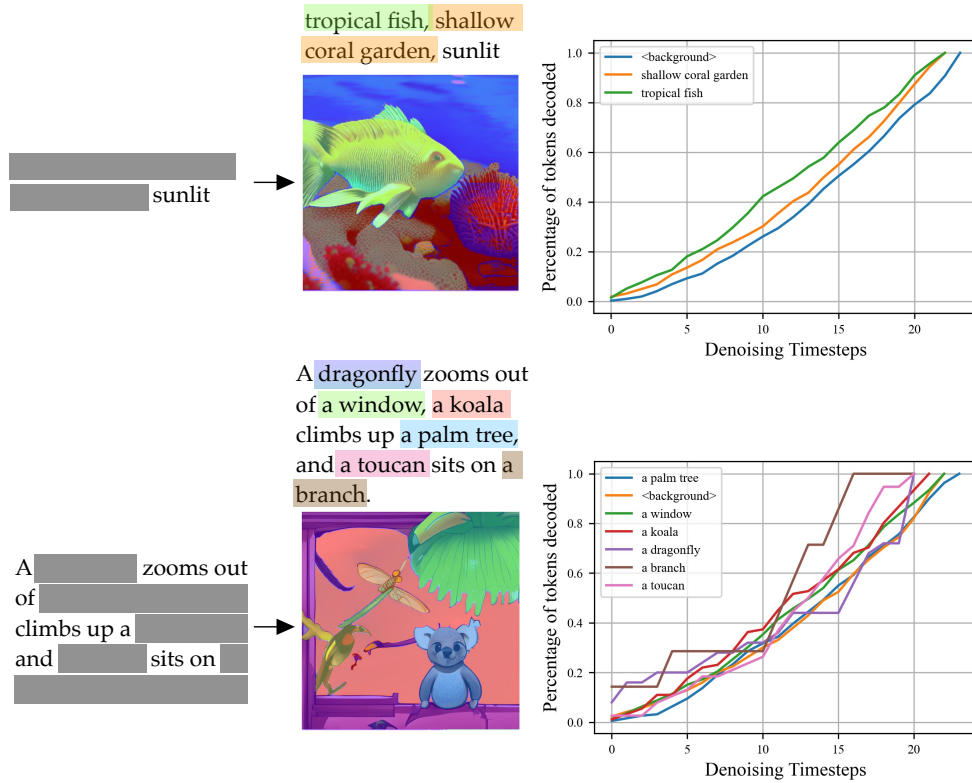


Figure 2.22: We demonstrate that UniDisc uniformly generates all concepts at once.

Zero-shot length extrapolation of UniDisc

In this section, we demonstrate the ability of UniDisc to perform zero-shot flexible resolution generation thanks to the use of RoPE embeddings on both text and image tokens. UniDisc model was fine-tuned on 512x512 images—resulting in each image using 1024 tokens—but is able to infill at 1024x1024—resulting in 4096 tokens per image—without further training. We demonstrate this in Figure 2.23.

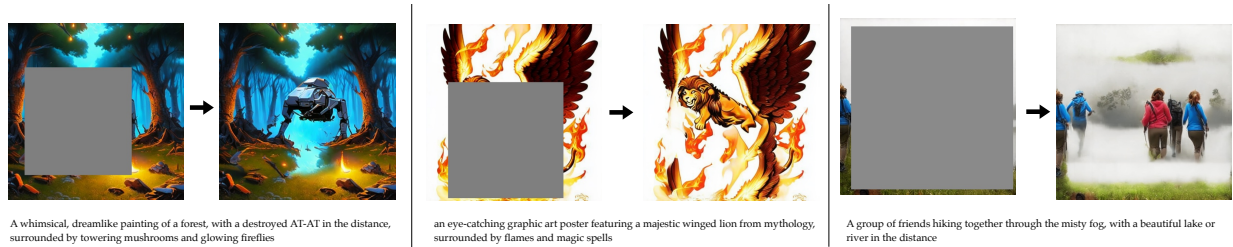


Figure 2.23: We train UniDisc on 512x512 resolution images but demonstrate zero-shot inpainting at 1024x1024.

2.8.7 Large Scale Quantitative Comparisons

Quantitative Generation Comparison with recent mulitmodal models

In Table 2.5, we evaluate UniDisc on the popular GenEval [GHS23] benchmark which looks at how well a generated image adheres to the prompt in terms of a set of predefined attributes (e.g., color, positioning). In Table 2.6, we compare FID on the popular MS-COCO 30K [Che+15] dataset on MJHQ-30K [Li+24a], which contains a higher proportion of highly-aesthetic images.

We also compare to the reported results from UniD3 [Hu+23], which most closely resembles our work.

Method	Sing. Obj.	Two Obj.	Counting	Colors	Position	Color Attr.	Overall
SDv1.5 [Rom+22]	0.97	0.38	0.35	0.76	0.04	0.06	0.43
CoDI [Tan+24]	0.89	0.16	0.16	0.65	0.02	0.01	0.31
Lumina-mGPT [Liu+24]	-	-	-	-	-	-	0.32
UniDisc	0.92	0.47	0.15	0.67	0.13	0.19	0.42

Table 2.5: We evaluate UniDisc on the GenEval [GHS23] benchmark.

Method	MSCOCO-30K FID ↓	MJHQ-30K FID ↓
SDv1.5 [Rom+22]	11.12	-
CoDi [Tan+24]	22.26	19.87
UniD3 [Hu+23] ²	25.11	-
UniDisc (Ours)	23.86	18.67

Table 2.6: We evaluate the 1.4B version of UniDisc on FID. We use evaluate on MS-COCO 30K [Che+15] and MJHQ-30K [Li+24a].

Method	Params	CUB200 FID ↓
UniD3 [Hu+23]	637M	17.38
UniDisc (Ours)	330M	11.03

Table 2.7: We compare our model to [Hu+23] on CUB200.

²Trained only on MS-COCO. Other works listed in this table trained on a broader set of datasets (possibly including MS-COCO). In most cases, training on additional datasets likely harms dataset-specific FID.

2.8.8 Understanding the effect of Classifier Free Guidance (CFG)

In Table 2.11, we observe that CFG is a significant factor in the performance difference between UniDisc and the AR baseline. We hypothesize that this is because CFG is most useful in decoding the first few tokens, with diminishing utility in later tokens. To examine this, we look at intermediate predictions by storing $\arg \max p_{\theta}(x_0 | x_t)$ at each sampling step. As an AR model cannot directly capture this distribution without an intractable rollout, we opt to use the same UniDisc model but with an autoregressive inference strategy, decoding from left to right. This allows us to directly compare the performance of different inference strategies and how they interact with classifier-free guidance.

We visualize this in Figure 2.24, where we visualize the difference between the conditional and unconditional image generated at different percentages of decoded tokens. We notice two things: (a) the difference diminishes as more tokens are decoded and (b) UniDisc consistently has higher distances between the logits than AR, which flattens out more quickly.

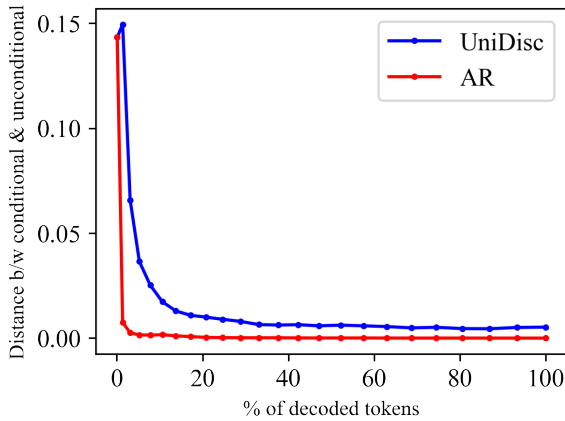


Figure 2.24: L2 distance between unconditional and conditional logits on currently masked tokens as sampling steps increase.

Steps	CLIP Score
[1 – 3]	0.301
[12 – 14]	0.293
[22 – 24]	0.283
All (24)	0.312

Figure 2.25: Comparing CLIP scores by applying CFG only on specific steps. This shows CFG has the most impact on the initial denoising steps (total steps = 24).

Intuitively, this means UniDisc extracts much more discriminating signal from CFG compared to AR. We believe this is because UniDisc has much more flexibility to decode tokens initially based on confidence, compared to AR which is forced to decode in a left to right manner and thus, can course correct quickly and more effectively. This can be seen in Table 2.25, where we selectively apply CFG only on a few steps and notice that CLIP score when CFG is applied on steps 1-3 almost matches applying CFG on all, while applying on the last few steps doesn’t affect things much at all.

Given the differences in CFG between UniDisc and AR models, we conduct a hyperparameter sweep over guidance scales in Figure 2.4. We compute FID and CLIP scores over four datasets, and at both 115/340M parameters. We find that our AR baseline benefits from a weight of $w = 0.5$ but sees far less improvement than UniDisc with CFG. For UniDisc, we choose an overall weight of $w = 1.5$, but note that the CLIP score scales cleanly with the guidance scale, demonstrating the trade-off between visual quality and

prompt adherence.

Finally, in Figure 2.26, we show the effect of CFG on the generated image. We increase the weight of the classifier-free guidance from $w = 0$ to $w = 8$ and observe the effect on the generated image.

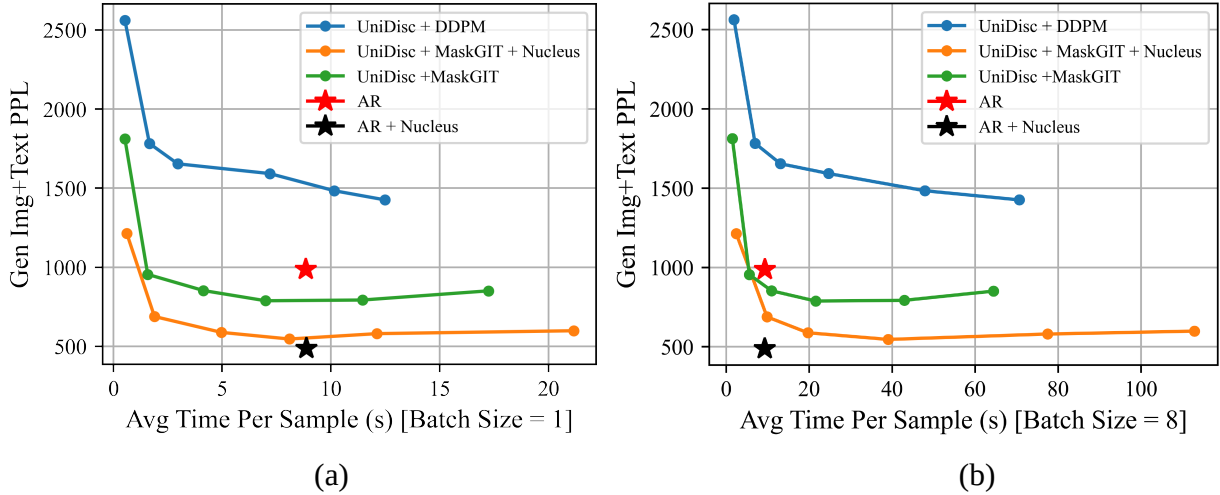


Figure 2.26: We show the effect of classifier-free guidance from left-to-right, starting with $w = 0$, and increasing linearly to $w = 8$ on the right, where output logits are $l_{\text{cfg}} = (1 + w)l_{\text{cond}} + w * l_{\text{uncond}}$.

Caption: "crab meditating, surfboard, orange sun setting, rainbow clouds, zen beach"

2.8.9 Inference: Generation time vs. batch size

We analyze the quality of the generation versus time in Figure 2.27. We make a similar observation as in prior work [Ziv+24; Gat+24] on discrete diffusion, finding that the ability to obtain predictions with varying sampling steps allows lower latencies. However, with current implementations, KV caching in AR models results in higher throughput as the batch size increases. This tradeoff can be explained by looking at the number of function evaluations (NFEs) and the cost of each in both cases. In AR generation w/KV caching, we have a fixed NFE, but each forward pass is substantially less expensive than in the NAR case. In contrast, in NAR, we can use substantially fewer NFEs, but each is more costly. Modern GPUs only reach peak throughput at larger batch sizes [Chi+24b]; as we decrease the batch size, the difference in computation per function evaluation diminishes, resulting in NAR having favorable performance.



Unified 2D and 3D Vision-Language Understanding

3.1 Abstract

Progress in 3D vision-language learning has been hindered by the scarcity of large-scale 3D datasets. We introduce UniVLG, a unified architecture for 2D and 3D vision-language understanding that bridges the gap between existing 2D-centric models and the rich 3D sensory data available in embodied systems. Our approach initializes most model weights from pre-trained 2D models and trains on both 2D and 3D vision-language data. We propose a novel language-conditioned mask decoder shared across 2D and 3D modalities to ground objects effectively in both RGB and RGB-D images, outperforming box-based approaches. To further reduce the domain gap between 2D and 3D, we incorporate 2D-to-3D lifting strategies, enabling UniVLG to utilize 2D data to enhance 3D performance. With these innovations, our model achieves state-of-the-art performance across multiple 3D vision-language grounding tasks, demonstrating the potential of transferring advances from 2D vision-language learning to the data-constrained 3D domain. Furthermore, co-training on both 2D and 3D data enhances performance across modalities without sacrificing 2D capabilities. By removing the reliance on 3D mesh reconstruction and ground-truth object proposals, UniVLG sets a new standard for realistic, embodied-aligned evaluation. Code and additional visualizations are available at univlg.github.io.

3.2 Introduction

Today’s real-world embodied systems rely on depth sensors and egocentric, calibrated camera setups for navigation and interaction with their surroundings [Ahn+22; Chi+24a]. However, despite having access to rich 3D information, these systems predominantly use 2D vision-language models to interpret their sensory video input, rather than leveraging 3D models that incorporate depth and egomotion. At first glance, this reliance on 2D models appears counterintuitive, as prior research has consistently shown that 3D models

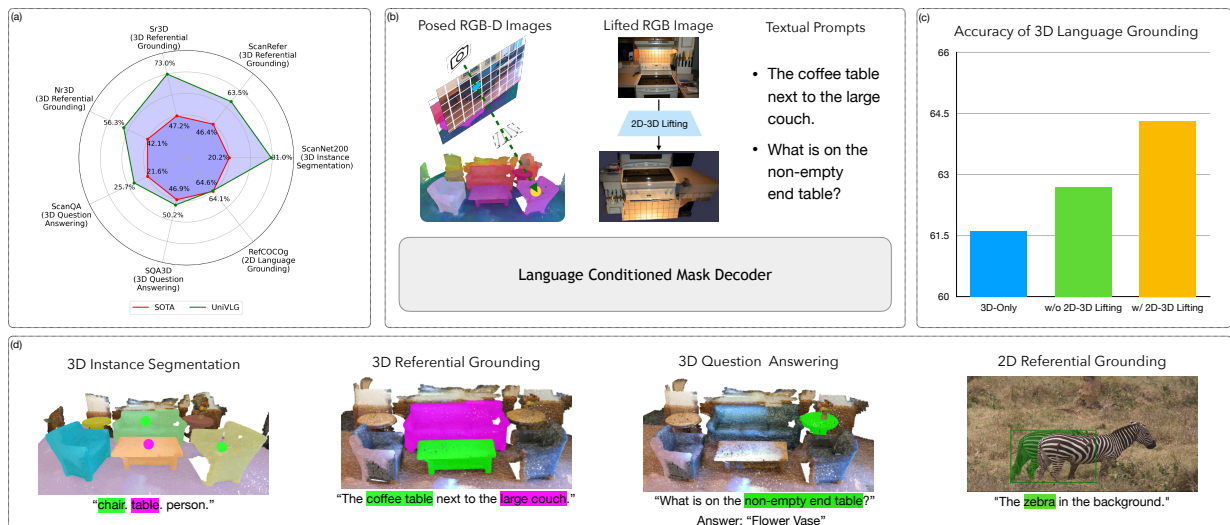


Figure 3.1: **(A)** UniVLG achieves state-of-the-art performance across a range of referential grounding, question answering, and instance segmentation benchmarks. **(B)** UniVLG is a unified model which accepts posed RGB-D sequences, or monocular 2D images which are then lifted to 3D pointmaps. **(C)** UniVLG significantly benefits from joint 2D-3D training, further boosted when all parameters are shared between modalities by using 2D-3D lifting. **(D)** Example task inputs/outputs for UniVLG.

outperform their 2D counterparts when trained on comparable amounts of data [Sid+23; Kun+20; Fan+21; RVK22]. The key limitation, however, is dataset availability: while 2D datasets are vast and well-curated, 3D datasets remain scarce and expensive to annotate [Dai+17; Yes+23]. As a result, there are currently no high-performing, pre-trained 3D encoders capable of processing 3D inputs at the same level as CLIP [Rad+21] does for 2D images. This data imbalance has led to a significant performance gap, ultimately slowing the widespread adoption of 3D models in embodied vision-language systems. Given these challenges, is scaling 3D training data the only viable path to bridging this gap, or are there alternative strategies for making 3D models more effective?

In this chapter, we introduce UniVLG, a unified 2D-3D vision-language model designed to improve 3D understanding by leveraging large-scale 2D data and pre-trained 2D models. UniVLG is trained on both 2D and 3D vision-language tasks, including referential grounding, object detection, and question answering in images and 3D scenes. Unlike models that operate directly on 3D point clouds, UniVLG processes RGB and RGB-D images—natural sensory inputs for embodied agents—and supports both single-view RGB images or multi-view posed RGB-D images. UniVLG processes each image with strong pre-trained 2D backbones, which also constitute the majority of its parameters, and fully leverages their representational power. It discriminates between 2D and 3D purely through the positional encodings of 2D image patch features, which represent the 2D pixel grid locations in images and the 3D (X,Y,Z) coordinates in scenes, similar to [Jai+24]. When training on 2D RGB images, we consider both 2D and 3D processing pathways within UniVLG, by using predicted 3D pointmaps [Wan+24], which further narrows the domain gap between 2D

and 3D input. We further introduce a novel language-conditioned mask decoder, shared across both 2D and 3D input, which predicts segmentation masks by conditioning on both visual features and language instructions to ground objects mentioned in the language input. Segmentation masks serve as a unifying output representation because they involve per-patch predictions, where each patch corresponds to either a 2D pixel or a 3D point. In our experiments, we show that besides unifying the output space, decoding to 3D masks results in significantly more precise predictions, challenging the established paradigm which decode bounding boxes or rely on object proposals. Our model is designed with the goal of sharing all weights across RGB image and RGB-D image sequence processing.

We test UniVLG on established 2D and 3D vision language benchmarks [Ach+20; CCN20]. We find that when trained exclusively on 3D data, UniVLG achieves state-of-the-art performance across all established benchmarks, outperforming prior methods in comparable settings by more than 15%. Furthermore, co-training UniVLG with 2D data enhances its 3D performance even further, both on in-domain and out-of-domain benchmarks. Notably, this improvement does not come at the expense of 2D tasks—UniVLG retains strong performance on 2D referential grounding datasets [Kaz+14] compared to its version which is only trained on 2D referential grounding data. UniVLG directly uses sensor point clouds without any mesh pre-processing of the RGB-D input and without relying on ground-truth bounding box proposals, typically used in existing works and benchmarks [Ach+20]. By benchmarking in these more realistic settings, we hope to encourage future research that aligns more closely with the goals of embodied vision and promotes the progress of 3D vision in practical, real-world scenarios.

In summary, our contributions are:

- **Unified 2D-3D Visual Grounding:** We propose a model that can consume and benefit from both 2D and 3D vision-language data.
- **State-of-the-Art Performance:** UniVLG achieves state-of-the-art performance on in-domain 3D referential grounding benchmarks, including ReferIt3D (SR3D, NR3D) and ScanRefer, outperforming prior methods by a significant margin, while also excelling in out-of-domain 3D referential grounding datasets.
- **Language-Conditioned 3D Mask Decoder:** We propose a novel language-conditioned mask decoder head for 3D referential grounding and show its superior performance over bounding box decoders.
- **Realistic Evaluation Settings:** We benchmark prior methods and UniVLG in realistic, embodied-aligned settings by using sensor-generated instead of mesh-reconstructed point clouds and without relying on ground-truth object proposals.

We make our code publicly available at [univlg.github.io](https://github.com/univlg).

3.3 Related Work

Use of 2D data for 3D Visual Language Understanding Tasks Most 3D Visual Language models directly operate over the provided 3D point clouds without using any 2D pre-trained features. SAT-2D [Yan+21] is one of the first 3D visual grounding model

which used 2D visual features during training for aligning 2D and 3D visual features and show significant boost over its versions that do not use 2D features. Recent methods in 3D Question Answering like 3D-LLM [Hon+23] and NaviLLM [Zhe+24a] use multi-view 2D features and pass them to LLMs for decoding answers. However, so far they haven’t been able to successfully address 3D visual grounding tasks. PQ3D [Zhu+24b] uses a combination of several visual backbones, including a 2D based feature backbone from OpenScene [Pen+23b]. Recent work of EFM3D [Str+24] uses 3D feature volumes obtained from lifting 2D image features but only evaluates on the task of 3D object detection and surface reconstruction.

Another related line of research focuses on enhancing 2D vision-language models (VLMs) with 3D reasoning. Spatial-RGPT [Che+24b] and Spatial-VLM [Che+24a] use depth estimation to enrich 2D models with spatial understanding. While these methods focus on improving 2D perception, our approach leverages 2D-to-3D lifting to enhance multi-view 3D reasoning, bridging the gap between 2D and 3D for vision-language grounding.

The closest work to ours is ODIN [Jai+24], which also differentiates between 2D and 3D through positional encodings instead of using separate image and point cloud encoders. However, they only consider the task of object segmentation. UniVLG is inspired by ODIN and innovates over it in the following ways: a) It extends its applicability to referential grounding and question-answering tasks. b) It improves the mask decoder to better incorporate language information. c) It shares all parameters between 2D and 3D pathways, instead of a subset of them by lifting 2D images to 3D pointmaps. With these advancements, UniVLG dramatically outperforms ODIN, and its extension LLaVA-3D [Zhu+24a] on 3D language grounding, demonstrating the importance of its design choices.

3D Visual Grounding Models 3D Visual Grounding Models can be broadly divided into two categories: Two-stage methods and single-stage end-to-end methods. Two stage methods first generate 3D object proposals and then select one proposal out of them. This is the dominant paradigm: InstanceRefer [Yua+21], SAT-2D [Yan+21], ViL3DRel [Che+22a] and recently scaled-up to models of 3D-VisTA [Zhu+23b] and PQ3D [Zhu+24b] which train their model on multiple 3D datasets and tasks. Specifically, 3D-VisTA first pre-trains their model on masked language/object modeling and scene-text matching, and then fine-tunes to downstream several language understanding tasks of interest. PQ3D [Zhu+24b] proposes promptable object queries for 3D scene understanding. While it decodes masks for instance segmentation tasks directly, it follows a 2D stage approach for free-form language grounding and selects a mask from a set of object mask proposals. However, two-stage methods are limited by the failures of the object proposal networks. To overcome this limitation, single-stage methods like 3D-SPS [Luo+22] and BUTD-DETR [Jai+22] directly regress 3D bounding boxes. They achieve strong results, especially on benchmarks like ScanRefer, which do not provide ground-truth proposals. However, they have only been trained on individual tasks and datasets and have not been scaled up yet. In this work, we propose a single-stage end-to-end model that is jointly trained on multiple 3D language understanding tasks, and achieve state-of-the-art results on several benchmarks.

For additional related work, see Appendix (Section-3.8.14).

3.4 Method

We show the architecture of UniVLG in Figure-3.2. The model takes as input a language query, N RGB images of shape $N \times H \times W \times 3$, and an associated 3D pointmap of shape $N \times H \times W \times 3$. The output consists of segmentation masks for each object mentioned in the sentence, a corresponding text span that refers to each segmented object, and optionally, generated text that answers the question. In datasets such as ScanNet, we obtain the 3D pointmap by unprojecting the sensed depth images using the camera parameters and standard pinhole-camera equations. For RGB images from 2D datasets like RefCOCO [Kaz+14], we use a neural 2D-to-3D lifting model [Wan+24], which takes a (monocular) RGB image as input and predicts a 3D pointmap. Note that the 3D pointmap does not need to be metric—in fact, our 3D pointmaps for 2D datasets are represented in relative space.

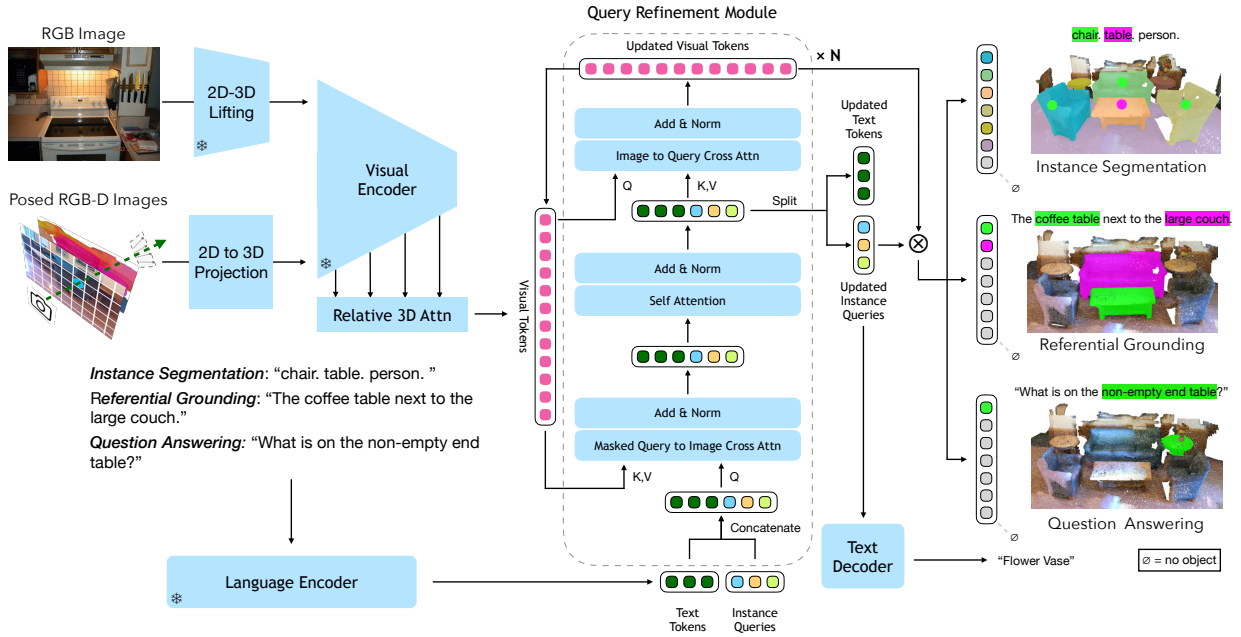


Figure 3.2: **UniVLG Architecture:** A vision language transformer that accepts a language utterance and either (1) a sequence of posed RGB-D images or (2) a monocular RGB image, lifted to 3D (2D to 3D Projection). UniVLG fuses information across vision and language to predict 3D object segments or generate answers. It uses a ViT backbone followed by 3D relative attentions to produce a set of 3D feature tokens. The proposed decoder then iteratively updates a set of learnable queries as well as the 3D feature tokens through token - language - query attentions to decode object segments and match them to noun phrases in the input referential utterance. Masks are decoded through a dot-product between 3D feature tokens and learnable queries. A text decoder predicts answers for the input questions by conditioning on the set of updated object queries.

Visual Encoder: We encode each RGB image independently using DiNO ViT encoder [Oqu+24], and add several 3D attention layers [Jai+24] on top of features from multiple layers. Specifically, we apply 3D k -NN attention with relative positional embeddings to fuse information across the input RGB views. This attention mechanism uses feature maps from the ViT encoder, with 3D pointmaps serving as the positional embed-

dings. Since our attention mechanism is relative, our model does not require a metric pointmap.

Language Encoder: We embed the natural language query using JinaCLIP [Kou+24], generating tokens of shape $M \times F$ where M is the number of tokens and F is the feature dimension.

Language Conditioned Mask Decoder: The mask decoder head takes as input the encoded visual features, their corresponding (relative) 3D coordinates, and the encoded language utterance; it outputs 3D segmentation masks of the mentioned objects and a text span over the encoded language utterance. Our mask decoder head draws inspiration from Mask2Former [Che+22b] and makes important architectural changes to make it suitable for 3D referential grounding.

We initialize a set of M learnable object queries, each responsible for decoding an object instance. We concatenate these object queries with the language tokens along the sequence dimension. We alternate between cross-attention between these and the visual tokens and self-attention among these concatenated queries and text tokens. Instead of using a vanilla cross-attention layer, we follow Mask2Former and use a masked variant where each query only attends to the points falling within the corresponding instance mask predicted by the previous layer. For this operation, we add 3D positional embeddings on the visual features. Next, the visual tokens from the backbone are updated by cross-attending to the updated object and text tokens. Specifically, let $Q^{(0)} \in \mathbb{R}^{M \times D}$ be the initial object queries, $T \in \mathbb{R}^{L \times D}$ be the text tokens, and $V^{(0)} \in \mathbb{R}^{N \times D}$ be the 3D visual tokens. The query refinement process can be described as:

$$\begin{aligned} X^{(0)} &= [Q^{(0)}; T]; \\ X^{(i+1)} &= \text{Norm}(\text{MaskedCrossAttention}(X^{(i)}, V^{(i)}) + X^{(i)}) \\ X^{(i+1)} &= \text{Norm}(\text{SelfAttention}(X^{(i+1)}) + X^{(i+1)}) \\ V^{(i+1)} &= \text{Norm}(\text{CrossAttention}(V^{(i)}, X^{(i+1)}) + V^{(i)}), \end{aligned}$$

where $[\cdot]$ denotes concatenation along the sequence dimension, and i is the layer index. The refined queries after each decoder layer $Q^{(i+1)} = X_{1:M}^{(i+1)}$ are then used for mask prediction with the updated visual features and for language grounding.

We find that *updating visual features* via attention to queries and text tokens *is crucial for 3D-referential grounding*. Open-vocabulary mask decoders, such as those in ODIN [Jai+24] and X-Decoder [Zou+23], which extend Mask2Former’s decoder to accept language tokens, do not update visual features during query refinement as in our method. Although their approach is sufficient for 3D instance segmentation, our experiments show that this choice significantly hinders performance in decoding object masks for 3D referential grounding (Table 3.7). Object2Scene [Zhu+23a], which decodes 3D bounding boxes for referential grounding, finds that only updating queries is sufficient. However, our ablation studies show that while this holds true for bounding box decoding, updating visual features during query refinement is crucial for accurately decoding masks (Table 3.5b).

After attending to text and visual features, the refined object queries decode object segments through a token-wise dot-product with the updated visual features to produce mask logits which are then thresholded to obtain segmentation masks:

$$M_i = \sigma(\text{sigmoid}(Q_i^{(f)} \cdot V^T)), \quad (3.1)$$

where M_i is the mask for the i -th object query, σ is a threshold function, and $\cdot$ denotes dot product.

Text Decoder: Beyond decoding segments, the refined object queries are used as input to the decoder of a pre-trained T5 [Raf+20] decoder to generate answers to questions, following PQ3D [Zhu+24b]. This is useful for question-answering tasks where the output is a text sentence.

3.4.1 Supervision Objective

Mask Loss: We match queries to ground-truth instances using Hungarian Matching [Car+20]. We supervise the matched queries’s predicted masks with both a Binary Cross Entropy (BCE) and Dice loss following Mask2Former [Che+22b].

Text Span Loss: Similar to prior works [Li+22a; Kam+21; Jai+22], we match the predicted 3D object segmentations to the relevant noun phrases in the input utterance through a dot-product between the object queries and the language tokens, generating the distribution G_i over the input text sentence for the i th query:

$$G_i = \text{sigmoid}(f_\phi(Q_i^{(f)}) \cdot f_\theta(T^T)) \quad (3.2)$$

where f_ϕ and f_θ are MLPs, G_i is the grounding distribution for the i -th object query over the input text tokens. We supervise these grounding distributions with a BCE loss, with unmatched queries supervised to have a low probability.

Box Loss: We observe a failure mode in our model where, when trained with the aforementioned objectives, some masks include a small number of distant, unrelated points, or multiple instances of the same object category are predicted by a single object query (see Figure 3.5 in Appendix). To address this, we introduce a novel box loss. This loss computes an enclosing 3D bounding box for each predicted mask and supervises it using standard box prediction losses—L1 and Generalized Intersection-over-Union (GIoU) [Rez+19]—against the ground-truth bounding boxes. We incorporate this box loss as an additional cost in both Hungarian matching and the final loss. This encourages the model to produce more accurate and compact masks, leading to improved downstream performance.

Text Generation Loss: For question answering tasks, our model decodes a text utterance as an output. We supervise the generated text with the ground-truth text answer using standard cross-entropy loss. In summary, our complete loss function is as follows:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{text}}\mathcal{L}_{\text{text}} + \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{gen}}\mathcal{L}_{\text{gen}} \quad (3.3)$$

where $\mathcal{L}_{\text{mask}}$ is the mask loss comprised of binary cross entropy and dice losses, $\mathcal{L}_{\text{text}}$ is the loss for matching the object queries to the mentioned objects in the language sentence, $\mathcal{L}_{\text{box}}$ are the additional bounding box losses described earlier, and $\mathcal{L}_{\text{gen}}$ is the cross-entropy loss over the auto-regressively generated answer (in case of question-answering datasets).

UniVLG shares all learnable parameters between 2D and 3D by leveraging 2D-to-3D lifting strategies to generate pointmaps for 2D datasets, which are seamlessly integrated

into 3D attention layers. Additionally, the mask decoding head unifies the output space between 2D and 3D as per-pixel segmentation masks, enabling the sharing of both loss functions and decoder parameters across modalities.

Implementation details: UniVLG consists of 108M trainable parameters along with a frozen 220M parameter text-encoder [Kou+24] and a 304M parameter image-encoder [Oqu+24]. At test time, we feed all images to our model with a *90-frame* scene taking ~ 1050 ms and ~ 15 GB of VRAM on an A100 GPU. We refer to Section 3.8.2 for further details.

3.5 Experiments

We evaluate our model on 3D and 2D referential grounding, 3D question answering and 3D instance segmentation benchmarks. We train our model on the 3D referential grounding datasets of SR3D, NR3D [Ach+20] and ScanRefer [CCN20] and 3D instance segmentation datasets of ScanNet200 [RLD22] and Matterport3D [Cha+17]. In addition to the 3D datasets, we also train our model on 2D referential grounding datasets with RefCOCO, RefCOCO+ and RefCOCOg [Kaz+14], and 2D image segmentation dataset with COCO [Lin+14]. We present results for two model versions: one trained solely on 3D data (UniVLG-3D-only) and the other trained jointly on both 2D and 3D datasets (UniVLG).

3.5.1 Evaluation on 3D Referential Grounding

Following BUTD-DETR [Jai+22], we test on two evaluation setups: 1. **Det**, where our model and baselines do not have access to ground-truth 3D boxes of objects in the scene, and 2. **GT**, where our model and baselines use ground-truth 3D object proposals provided in the benchmarks.

Additionally, these benchmarks sample point clouds from reconstructed and post-processed meshes instead of directly using the raw point clouds obtained from sensor inputs. As observed in prior works [Jai+24; Kun+20], mesh-sampled point clouds often exhibit fine-grained misalignment with sensor-generated point clouds, which can unfairly disadvantage sensor-based approaches on these benchmarks. To address this, we benchmark our method and prior methods in a more embodied-aligned setting using sensor point clouds directly. We thus evaluate all methods on benchmark-provided point clouds sampled from the post-processed mesh (*Mesh*), and separately retrain and evaluate a subset of methods on sensor point clouds (*Sensor*) obtained by unprojecting posed RGB-D images.

Evaluation Metrics: We use the standard top-1 accuracy metric. For the **Det** setup, a predicted bounding box is considered correct if its intersection over union (IoU) with the ground truth box is higher than a predetermined threshold (we use the standard 0.25, 0.5 and 0.75). As UniVLG predicts masks (instead of axis-aligned bounding boxes), we obtain a bounding box by taking the extents of the mask. For the **GT** setup, we pool visual features inside the given ground-truth masks, and the object queries predict a segmentation mask over the “pooled” feature tokens, one token per object. The prediction is correct if the model selects the feature token corresponding to the ground-truth object.

Baselines: We compare our model against the state-of-the-art two-stage methods of 3D-VisTA [Zhu+23b], PQ3D [Zhu+24b] and concurrent work of LLaVA-3D [Zhu+24a]; and the SOTA single-stage method of BUTD-DETR [Jai+22]. UniVLG uses significantly less 3D

Table 3.1: **Results on 3D language grounding in 3D mesh and sensor point clouds (PC).** We evaluate top-1 accuracy on the official validation set with assuming ground-truth (GT) or without assuming ground-truth proposals (Det).

Method	SR3D				NR3D				ScanRefer		
	Acc @25 (Det)	Acc @50 (Det)	Acc @75 (Det)	Acc (GT)	Acc @25 (Det)	Acc @50 (Det)	Acc @75 (Det)	Acc (GT)	Acc @25 (Det)	Acc @50 (Det)	Acc @75 (Det)
ReferIt3DNet [Ach+20]	27.7	-	-	39.8	24.0	-	-	-	26.4	16.9	-
ScanRefer [CCN20]	-	-	-	-	-	-	-	-	35.5	22.4	-
InstanceRefer [Yua+21]	31.5	-	-	48.0	29.9	-	-	-	40.2	32.9	-
LanguageRefer [Roh+22]	39.5	-	-	56.0	28.6	-	-	-	-	-	-
Mesh SAT-2D [Yan+21]	35.4	-	-	57.9	31.7	-	-	-	44.5	30.1	-
PC BUTD-DETR [Jai+22]	52.1	-	-	67.0	43.3	-	-	54.6	52.2	39.8	-
3D-VisTA [Zhu+23b]	56.5	51.5	42.8	76.4	47.7	42.2	35.5	65.1	51.0	46.2	36.7
LLaVA-3D [Zhu+24a]	-	-	-	-	-	-	-	-	54.1	42.2	-
PQ3D [Zhu+24b]	62.0	55.9	46.2	79.7	52.2	45.0	37.6	66.7	56.7	51.8	43.3
ODIN [Jai+24]	38.1	29.3	23.1	-	31.6	20.8	15.8	-	43.1	33.4	26.2
Sensor BUTD-DETR [Jai+22]	43.3	28.9	6.58	-	32.2	19.4	3.64	-	42.2	27.9	6.53
PC 3D-VisTA [Zhu+23b]	47.2	43.2	36.1	61.4	42.1	37.4	32.0	54.2	46.4	42.5	36.3
UniVLG-3D-only (Ours)	71.6	63.8	49.4	78.9	52.5	43.3	34.2	65.8	60.7	53.2	42.6
UniVLG (Ours)	73.0	64.8	51.8	-	56.3	48.0	37.6	-	63.5	56.4	46.0

training data than prior SOTA 3D referential grounding models. For example, 3D-VisTA [Zhu+23b] trains on the previously mentioned 3D datasets that we use but also includes 3RScan (1500 scenes) [Wal+19], Objaverse (700k objects) [Dei+22], and additional text sentences on ScanNet generated using GPT-3 (see Table-3 of 3D-VisTA). Similarly, PQ3D adds the Multi3DRefer [ZGC23] and Scan2Cap datasets [Che+20], but also utilizes a point encoder that was trained on all 3D-VisTA datasets. All two-stage baselines assume access to ground-truth proposals at test-time in the SR3D and NR3D benchmarks; hence we re-evaluate them with predicted boxes coming from a SoTA object detector, Mask3D [Sch+23]. We also re-train 3D-VisTA and BUTD-DETR with sensor point clouds. Despite our best efforts, we could not manage to re-train PQ3D with sensor point clouds due to their use of multiple backbones, and multi-stage training strategies. We also compare our model with ODIN [Jai+24] trained for 3D language grounding using their architecture but our grounding losses. The 3D referential grounding results are presented in Table 3.1, from which we find:

UniVLG outperforms prior methods, regardless of data selection, on all setups which do not assume GT boxes. Even without our joint 2D training strategy—and with less 3D data than prior methods—UniVLG-3D-only significantly outperforms all prior methods. It dramatically outperforms alternative single stage models, such as BUTD-DETR, on the stricter IoU threshold of 0.75, thanks to predicting masks instead of bounding boxes—as we demonstrate later in Table 3.5c. In the GT setup as well, UniVLG significantly outperforms 3D-VisTA and closely matches the performance of the recent work of PQ3D in the setup where PQ3D uses mesh point clouds, while UniVLG operates over sensor point clouds.

Co-training UniVLG with 2D and 3D data enhances 3D performance across all 3D vision-language grounding benchmarks, demonstrating that leveraging 2D data during

training provides additional benefits beyond initializing the model with pre-trained 2D weights (row-5, sensor PC, Table 3.8).

Performance of all prior SOTA models drop with sensor point cloud as input and without assuming GT boxes: Both single-stage methods like BUTD-DETR and two-stage methods like 3D-VisTA have a performance drop of 5-15% when using sensor RGB-D point clouds as input instead of mesh point-clouds. The sensor point cloud and mesh point clouds have fine-grained misalignment, resulting in this drop. Shifting from ground-truth box proposals to a more realistic setup of using predicted box proposals from a SOTA detector results in a drop of 15-20% in accuracy. Nonetheless, even when UniVLG uses sensor pointclouds (which as we showed above result in a 5-15% accuracy drop on these benchmarks), it still outperforms the baselines that use *mesh* point clouds as input.

Qualitative results of UniVLG are in Figure 3.3 (Appendix).

3.5.2 Evaluation on Out-of-Domain 3D Referential Grounding

We evaluate our model and baselines on LX3D[†], an out-of-domain 3D language grounding dataset that spans ScanNet [Dai+17], ScanNet++ [Yes+23], ARKitScenes [Bar+21], and HM3D [Yad+23] (details in Section 3.8.3). LX3D allows us to assess the robustness of our model on new scenes, camera capture systems, and language instructions. We show the results of our model, both a 3D-only variant and our full model w/2D data + lifting in Table 3.2. We find that our model outperforms prior methods on these out-of-domain datasets, achieving strong performance across the board.

3.5.3 Evaluation on 3D Question Answering

We test UniVLG on ScanQA [Azu+22] and SQA3D [Ma+22] question answering benchmarks. ScanQA [Azu+22] focuses on spatial relations. Alongside question-answer pairs, the dataset includes annotations for the objects referenced in the question, and we supervise our model to predict these in addition to generating the answer. SQA3D [Ma+22] provides pairs of situation descriptions and questions regarding embodied scene understanding, navigation, common sense and multi-hop reasoning, such as, “*looking for some food in the fridge*”, “*which direction should i go?*” and the task is to generate the correct answer (“*right*”).

Table 3.2: **Out-of-Domain 3D Referential Grounding** Acc@25 in Det. From left-to-right, ScanNet++, HM3D, ARKitScenes, ScanNet (GT), ScanNet (SAMPro3D). See Section 3.8.3 for details.

Model	SN++	HM3D	ARKit	SN-GT	SN-SAM
UniVLG-3D-only	42.4	49.7	64.6	77.0	50.9
UniVLG	42.8	51.9	65.1	79.9	52.5

[†]LX3D is contributed by a concurrent anonymous submission and is not part of our contributions.

Table 3.3: **Results on 3D Visual Question Answering** on official validation sets. We evaluate top-1 exact match accuracy (EM@1).

	Method	ScanQA	SQA3D
Mesh PC	3D-LLM (BLIP2-flanT5) [Hon+23]	20.5	–
	PQ3D [Zhu+24b]	21.0	47.0
	3D-VisTA [Zhu+23b]	22.1	47.5
	NaviLLM [Zhe+24a]	23.9	–
Sensor PC	3D-VisTA [Zhu+23b]	21.6	46.9
	UniVLG (Ours)	25.7	50.2

Evaluation Metrics: We use the established Exact Match (EM@1) metric, which measures if the generated answer matches either of the two answer candidates provided by ScanQA, or the single ground-truth answer provided by SQA3D. We also report results with additional metrics in Appendix (Table 3.11).

Baselines: We compare against the LLM based methods of 3D-LLM [Hon+23] and NaviLLM [Zhe+24a] which use BLIP2-flanT5 [Li+23b] and Vicuna-7B [Pen+23a] as their answer generation heads. We also compare with 3D-VisTA [Zhu+23b] and PQ3D [Zhu+24b] which use small decoder heads like T5-small [Raf+20], similar to our approach. We show results in Table 3.3 on the validation sets of these benchmarks. UniVLG outperforms all prior baselines on both benchmarks. We found that using sensor point clouds vs mesh point clouds does not result in a significant difference in performance in these benchmarks, likely because the models are evaluated on text generation instead of localization of objects as in 3D referential grounding and segmentation benchmarks.

Additionally, we show results on 3D instance segmentation in Section 3.8.1.

3.5.4 Evaluation on 2D Referential Grounding

We also evaluate UniVLG on the 2D Referential Grounding benchmarks [Kaz+14] (Table 3.4). We train two versions of our model: UniVLG (2D only), which is trained exclusively on 2D datasets, and UniVLG (2D-3D), which is trained on both 2D and 3D data. Our results show that co-training with 3D data does not degrade the performance of the version trained solely on 2D data. This demonstrates that it is indeed possible to train a single model for both 2D and 3D tasks. As we show in our experiments, this approach leads to significant improvements in 3D performance without negatively affecting 2D performance. In this work, due to our focus on improving 3D vision-language grounding and resource constraints, we did not train our model on additional 2D datasets, which is common in prior work. Scaling up these models with more 2D data and studying its impact on 3D vision-language grounding is a promising avenue for future research.

Table 3.4: **Results on val sets of 2D Ref. grounding datasets**

	RefCOCO	RefCOCO+	RefCOCOg
LAVT [Yan+22] (B)	72.7	62.4	61.2
ReSTR [Kim+22]	67.2	55.7	54.5
X-Decoder (T) [Zou+23]	–	–	61.9
X-Decoder (B) [Zou+23]	–	–	64.5
X-Decoder (L) [Zou+23]	–	–	64.6
UniVLG (2D only)	69.4	61.3	64.0
UniVLG (2D-3D)	69.2	61.3	64.1

Table 3.5: **Analysis of Box Head vs Mask Head** on ScanRefer Dataset with Acc@25 if not otherwise stated.

(a) Parametric vs Non-parametric Query			(b) Updating Visual Features			(c) Results at Various IoU Thresholds		
Query Type	Box Head	Mask Head	Feat Attn	Box Head	Mask Head		Acc@25	Acc@75
Param	23.9	54.4	✓	33.9	54.4	Box Head	34.5	1.1
Non-param	34.5	43.9	✗	34.5	41.5	Mask Head	54.4	33.2

Table 3.6: **Analysis of 2D training strategies** Acc@25 in DetSetup

Model		Avg Accuracy	SR3D	NR3D	ScanRefer
UniVLG-3D-only		61.6	71.6	52.5	60.7
UniVLG-2D-3D(w/o 3D lifting)	2D-	62.7	73.0	53.5	61.5
UniVLG-2D-3D(w/ lifting)	2D-3D	64.3	73.0	56.3	63.5

Table 3.7: **Ablations** Acc@25 in DetSetup

Model		Avg Accuracy	SR3D	NR3D	ScanRefer
UniVLG		61.0	67.1	55.7	60.2
w/o mask decoder	w/ box decoder	39.3	38.9	33.2	45.7
w/o feature attn		36.9	38.0	30.0	42.8
w/o pretrained 2D weights		53.4	54.3	49.1	56.9
w/o mask bounding box loss		56.8	64.3	49.5	56.7

3.5.5 Ablations

We ablate a series of design choices of our model on referential grounding datasets of SR3D, NR3D, and ScanRefer on Table 3.6 and Table 3.7; and on ScanRefer dataset in Table 3.5. We have the following conclusions:

1. Lifting 2D datasets to 3D improves 3D performance. In Table 3.6, we compare three variants of our model: one trained only on 3D data, one trained with 3D data and 2D images without lifting them to 3D (where the 3D layers are skipped for 2D inputs, following [Jai+24]), and our proposed approach of lifting 2D images to 3D pointmaps. We observe that incorporating 2D data improves performance in both scenarios, but our approach of lifting 2D images to 3D achieves the best results. In Section 3.8.11, we show in a more controlled setting that training without using 3D pointmaps—by skipping the 3D layers—results in significant overfitting to individual 2D-3D domains.

2. Decoding boxes is inferior to decoding segmentations. Shifting from decoding segmentation masks to decoding bounding boxes hurts performance (row 2 of Table 3.7), especially in tight IoU thresholds IoU@0.75, shown in Table 3.5c.

3. Visual tokens updating through attending to language and queries during mask decoding is essential for good performance in 3D referential grounding, as shown in row 3 of Table 3.7. This is potentially because the mask decoding head relies on dot-products of queries and features to predict masks; and thus having both object queries and visual features to be very well distinguished for different instances of the same object is crucial. This design choice is unique to mask decoding heads, as we show in Table 3.5b.

Box-decoding models work similarly well irrespective of updating the visual tokens with language and object tokens. This variant is very close to ODIN’s open vocabulary head, which also lacks such attention operations, and as we show it does not work well for referential language grounding.

4. 2D feature pretraining dramatically improves performance as shown in row 4 of Table 3.7.

5. The predicted mask bounding box loss helps significantly as shown in row 5 of Table 3.7.

6. Non-Parametric Queries are crucial for decoding boxes prediction, while parametric queries work well for decoding segments. There are two popular choices for object queries: *Parametric Queries* which are scene-independent learnable vectors, initialized from scratch, and are updated via attention. *Non-Parametric Queries*, which are scene-dependent, and are typically initialized by doing Farthest Point Sampling on the input point clouds and encoding the corresponding xyz locations as query positional embeddings and corresponding features as query feature embeddings. Using non-parametric queries are crucial for box-decoding heads, while both queries work similarly well for mask-decoding heads (Table 3.5a). Box-decoding heads need to regress raw XYZ coordinates in 3D space; the search space is large and sparse—as most of it is empty—and parametric queries have difficulty handling such free space, as already mentioned in 3DETR [MGJ21]. Mask decoding uses dot-product between queries and visual tokens coming from the 3D backbone, and thus does not need to reason about 3D free space.

3.6 Conclusion

We present UniVLG, a vision-language model that integrates 2D and 3D data to address data scarcity in 3D vision-language learning. By leveraging pre-trained 2D features, 2D-to-3D lifting strategies, and a novel mask decoder head, UniVLG significantly outperforms prior methods in realistic embodied 3D vision settings while maintaining strong 2D understanding. Our extensive ablations validate key design choices: (1) Mask decoding is superior to box decoding, and each proposed component is crucial for its success. (2) Pre-trained 2D features improve performance, and co-training 3D vision-language tasks with 2D data provides additional gains. (3) Incorporating 2D-to-3D lifting strategies further enhances 3D understanding when training with 2D data. More broadly, our findings suggest that scaling 3D data is not the only path forward—leveraging 2D data and pre-trained features can effectively enhance 3D reasoning. We hope UniVLG inspires further research into vision-language models that bridge 2D and 3D for real-world applications.

3.7 Acknowledgements

This chapter was authored jointly with Ayush Jain and contributed to by: Yuzhou Wang, Sergio Arnaud, Ada Martin, Alexander Sax, Franziska Meier, Katerina Fragkiadaki.

The CMU authors were supported by an NSF Career award, ONR award N00014-23-1-2415, AFOSR Grant FA9550-23-1-0257, and DARPA No. HR00112490375 from the

U.S. DARPA Friction for Accountability in Conversational Transactions (FACT) program. Ayush Jain is supported in part by the CMU Robotics Vision Fellowship. The authors express gratitude to Nikolaos Gkanatsios, Wen-Hsuan Chu, Gabriel Sarch, Mathew Bronars, and Naitik Khandelwal for their valuable feedback on the early draft of this work. Special thanks to Kyle Genova for several insightful discussions during the early stages of this project’s development. The authors also thank Cortex team at Meta for valuable feedback and discussions.

3.8 Appendix

3.8.1 Evaluation on 3D Instance Segmentation

Table 3.8: **Evaluation on 3D Instance Segmentation Benchmarks.** (S) and (M) denotes models trained on sensor and mesh point clouds respectively.

(a) ScanNet200				(b) Matterport3D			
Model		mAP	mAP25	Input	Model	mAP	mAP25
Closed Vocabulary	Mask3D [Sch+23] (S)	15.5	24.3	Closed Vocabulary	Mask3D [Sch+23] (S)	2.5	10.9
	Mask3D [Sch+23] (M)	27.4	42.3		Mask3D [Sch+23] (M)	11.3	23.9
	PQ3D (closed) [Zhu+24b] (M)	27.0	46.3		ODIN [Jai+24] (S)	14.5	36.8
	QueryFormer [Lu+23a] (M)	28.1	43.4	Language Prompted	UniVLG (Ours) (S)	10.7	24.5
	MAFT [Lai+23] (M)	29.2	43.3		Prompted UniVLG-2D-3D (Ours) (S)	13.4	29.6
	ODIN [Jai+24] (S)	31.5	53.1				
Language Prompted	PQ3D (open) [Zhu+24b] (M)	20.2	32.5				
	UniVLG-3D-only (Ours) (S)	27.9	46.1				
	UniVLG (Ours) (S)	31.0	52.1				

We test UniVLG on 3D segmentation benchmarks of ScanNet200 [RLD22] and Matterport3D [Cha+17] for instance segmentation tasks. These benchmarks have a fixed vocabulary of objects (200 classes in ScanNet200 and 160 classes in Matterport3D). SOTA models like ODIN [Jai+24] and Mask3D [Sch+23] train and evaluate in this fixed vocabulary setup by predicting a distribution over the fixed set of classes and supervising with softmax losses. PQ3D [Zhu+24b] evaluates in a language-prompted setup where they supply object names, one object at a time, and gather predictions for all objects in the vocabulary. They compare with a closed-vocabulary version of their model, and find that their language-prompted version is about 7% worse than their closed vocabulary version due to ambiguities in class names confusing CLIP (eg. “chair” and “armchair”; “table” and “desk” are different categories in ScanNet200). We follow PQ3D and evaluate our model in the language-prompted setup. The input to the model is a concatenation of all object classes of the benchmark as a long sentence (eg: “chair. table. sofa. bed. ...”). While PQ3D cannot predict multiple object classes simultaneously, and hence have to supply one object at a time, our model can simultaneously decode masks for all objects mentioned in the sentence. The results are shown in Table-3.8 on the official validation splits of these benchmarks. We observe that UniVLG outperforms PQ3D in the language-prompted evaluation setup on ScanNet200.

3.8.2 Additional Implementation details

UniVLG consists of 108M trainable parameters along with a frozen 220M parameter text-encoder [Kou+24] and a 304M parameter image-encoder [Oqu+24]. For ablations in Table 3.7 and 3.5, we use a 88M parameter Swin [Liu+21] image-encoder. We train in data-parallel across 32 A100 80G GPUs with an effective batch size of 64. We use ScanEnts3D [Abd+23] version of ScanRefer [CCN20] and Referit3D [Ach+20] which provides object annotations for all noun words in the language sentence. During training, we process

either a sequence of N posed RGB-D images, or a single RGB image. During training, the model processes either a sequence of N posed RGB-D images or a single RGB image. For 2D images, we apply a 2D-to-3D lifting strategy with a 50% probability. When lifted, the images pass through all 2D-3D layers; otherwise, they remain in 2D space, skipping the 3D attention layers. At test time, we retain 2D images in their original space to prevent noise from predicted 3D pointmaps from impacting 2D performance. For 3D scenes, we compute CLIP embeddings for all images and captions and use this to select 5 relevant frames, with an additional 10 frames coming from Furthest-Point-Sampling (FPS) in the CLIP embedding space, for a total of 15 frames. At test time, we feed all images in a scene to our model. For validation results, we perform span prediction to identify the primary subject from a given utterance. We found that prompting an LLM [Dub+24] with examples specific to a given dataset to result in better performance compared to traditional NLP libraries. We use Jina-CLIP [Kou+24] as the text-encoder, as it supports arbitrary input-length. We jointly train our model on all datasets, with text generation loss only active in question answering datasets. Our method provides for fast inference, with a *90-frame* scene taking $\sim 1050\text{ms}$ and $\sim 15\text{GB}$ of VRAM on an A100 GPU.

3.8.3 More details on Out-of-domain LX3D datasets

Note: This collection of these out-of-domain datasets is not our contribution, but comes from a concurrent anonymous submission. We include the following details for the information of the reader.

The data is collected using an off-the-shelf annotation interface which allows for selecting rectangular regions on 2D videos and associating them with text. 3D instance masks are projected into 2D and shown alongside the RGB camera view. For ScanNet-GT [Dai+17], HM3D [Yad+23] and ScanNet++ [Yes+23], ground truth instance masks are used that are provided by the dataset. For ARKitScenes [Bar+21], and ScanNet-SAM, instance masks produced by SAMPro3D [Xu+23] are used. Annotators have the ability to conglomerate multiple instance masks into a single object – this proved necessary for SAMPro3D segmentations, which tended to break large objects into many subparts. This is in line with ScanNet’s instance segmentation procedure in which annotators stitch together an automatically-generated oversegmentation of a 3D scene.

In addition to grounding the target object, annotators were asked to provide object grounding for all other nouns in the phrase they provided. For SAMPro3D masks, annotators could select any object they wished as a target. In later iterations of the task, for ScanNet and ScanNet++, a mask was highlighted in white with at least 1 distractor to serve as the target object for description. This is in line with prior literature which observed that the 3D referential grounding task is most challenging in the high-distractor case [CCN20].

While it was found that annotations on ScanNet and ScanNet++ were relatively reliable from a first pass, considerable mask/bounding box noise was observed when collecting annotations on SAMPro3D instance masks. To address this, a separate validation task was created. In this task, annotators saw a video similar to the video provided in the generation task, but only the selected masks for a particular language sample were highlighted. The

referential expression description was also shown. This improved the quality of SAMPro3D samples substantially, but it also drastically slowed the collection rate.

Annotation setup The annotation setup largely mirrors ScanRefer [CCN20] – annotators simply describe objects in a one-step fashion rather than the two-player game format of NR3D [Ach+20]. A 2D video interface was used for the task, projecting 3D instance masks to 2D. Annotators can write full referring expressions and associate particular instance masks with particular tokens in the expressions. They may also agglomerate two or more instances into one.

When ground-truth instance masks are available (as in ScanNet and ScanNet++), these are used as the basis for annotation. When they are not available, instance masks are generated using SAMPro3D [Xu+23].

Decomposed by scene dataset, LX3D contains:

1. **ScanNet**: 4,470 language annotations covering 130 venues and 1038 objects for validation.
2. **ScanNet++**: 3,774 language annotations covering 50 venues and 1,303 objects for validation.
3. **ARKitScenes**: 1,416 language annotations covering 714 venues and 2,322 objects for validation.

3.8.4 Effect of Fine-tuning 2D backbones in UniVLG

We study the effect of fine-tuning the 2D backbones on in-domain and out-of-domain performance. We train two versions of UniVLG with swin backbones, one with fine-tuning and the other without fine-tuning. For training, we use SR3D and NR3D, and evaluate on the validation sets of SR3D, NR3D (in-domain) and ScanRefer (out-of-domain). The results of the experiments are shown in Table-3.9. We find that both models work similarly well, both in-domain and out-of-domain.

Table 3.9: **Effect of Fine-tuning 2D backbones of UniVLG** for Acc@25 in DetSetup. SR3D and NR3D are in-domain and ScanRefer is out-of-domain

Model		SR3D	NR3D	ScanRefer
UniVLG w/ fine-tune		65.6	52.7	54.4
UniVLG w/o finetune		66.7	52.0	54.5

3.8.5 Performance with different backbones

We demonstrate that the performance can scale with the strength of the backbone. Specifically we use a DINOv2 [Oqu+24] backbone consisting of 1.1B parameters, scaling over 5x

compared to the Swin backbone. To achieve high-performance during training, we freeze the backbone, although we note that it is possible that additional performance could be obtained with efficient fine-tuning techniques such as LoRA [Hu+21]. In Table-3.10, we find that adding this backbone boosts performance on all 3 language grounding datasets, with substantial margins of 4.5%, 1.9%, and 3.4% @ 0.25 on SR3D, NR3D, and ScanRefer respectively.

Table 3.10: Ablation of visual backbones on 3D language grounding. We evaluate top-1 accuracy on the official validation set without assuming ground-truth proposals (Det).

Method	SR3D			NR3D			ScanRefer		
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
	@25	@50	@75	@25	@50	@75	@25	@50	@75
	(Det)	(Det)	(Det)	(Det)	(Det)	(Det)	(Det)	(Det)	(Det)
UniVLG (Swin)	67.1	58.7	46.4	50.6	41.9	32.2	57.3	49.8	40.2
UniVLG (DINOv2)	71.6	63.8	49.4	52.5	43.3	34.2	60.7	53.2	42.6

3.8.6 Additional Metrics on ScanQA Dataset

We report additional standard metrics used by ScanQA benchmark in Table-3.11.

3.8.7 Visualizations of UniVLG on Referential Grounding Datasets

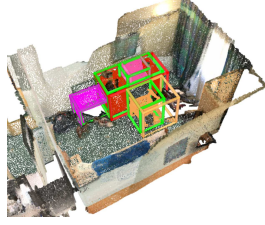
We show the visualization of UniVLG on 3D referential grounding in Figure-3.3 and on 2D referential grounding in Figure-3.4.

3.8.8 Visualization of common failure modes of UniVLG

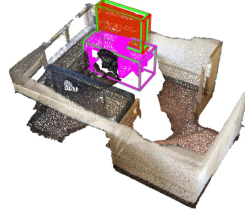
We identify three systematic failure modes in our model, illustrated in Figure-3.5.

- **Inclusion of distant outlier points in the predicted masks:** In the first image of Figure-3.5, while UniVLG accurately predicts the object, it also mistakenly includes some distant points in the mask. This leads to a larger bounding box during the mask-to-bounding box conversion in post-processing, negatively affecting accuracy metrics. Although our proposed box loss mitigates this issue, it doesn't fully resolve it.
- **Multiple instances of the same object being segmented together:** As shown in the middle image of Figure-3.5, UniVLG predicts both beds as a single output. Incorporating attention to language and queries helps reduce such errors, though they still persist. Our box loss also aids in addressing this issue.
- **Failures in language understanding** as seen in the third image of Figure-3.5.

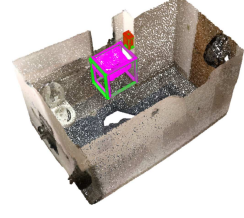
ScanRefer



"The desk is to the right of the bench.
The desk has a laptop on top of it."

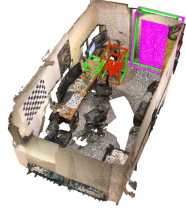


"The shelf is on top of the desk.
The shelf is a white rectangle."



"The wall soap dispenser. the
dispenser is above the sink."

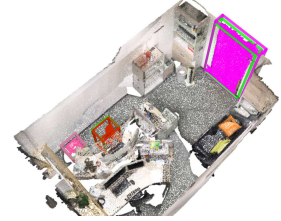
SR3D



"The chair that is in the center
of the door and the laptop."

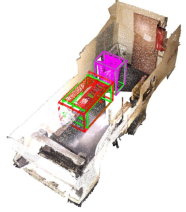


"Find the plant that is on the end table."

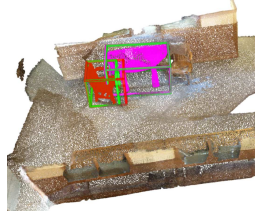


"The chair that is in the middle
of the door and the backpack."

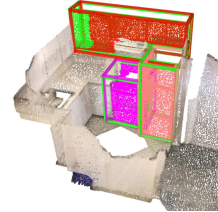
NR3D



"The table closest to the green tall trash can."



"Facing the half table the left chair."



"These are the kitchen cabinets
that are located directly above
the stove and refrigerator."

Figure 3.3: **Visualizations of UniVLG on 3D Referential Grounding Datasets of ScanRefer, SR3D, and NR3D** The red masks indicate UniVLG’s prediction for the target object, and pink masks indicate its predictions for the anchor object. Green masks and boxes indicate ground-truth target and anchor objects.

The first two failure modes are specific to mask-decoding architectures, and similar issues have been noted by Mask3D [Sch+23] in their 3D instance segmentation tasks. Box-decoding architectures, on the other hand, generally avoid these problems. Nevertheless, we find that mask-decoding architectures offer significant advantages in other aspects, such as more accurate and fine-grained segmentation, making them valuable despite these challenges.

3.8.9 Performance analysis with pose and depth noise

To analyze the performance of UniVLG under sensor noise we conduct two experiments to model error in both pose and depth. For the pose error experiment, we add gaussian noise to the translation and rotation components of every camera pose in a scene. Similarly, for the depth error experiment, we add gaussian noise uniformly to the depth map. When each depth map is unprojected, the resulting point cloud becomes misaligned and performance

RefCOCO



white car on the left



yellow and blue truck second from left



center case on floor with squares

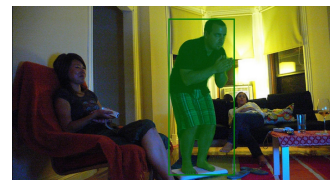
RefCOCO+



boy with psn shirt

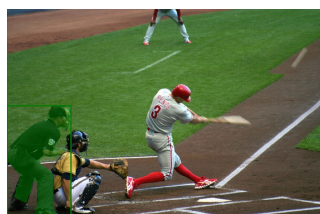


black cat under piped



guy standing playing wii fit

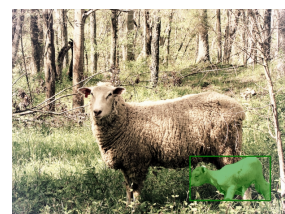
RefCOCOg



referee dressed in a black shirt



small cow laying down

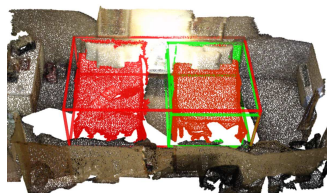


the white little lamb

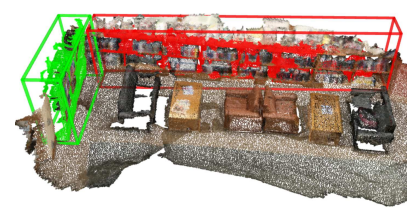
Figure 3.4: **Visualizations of UniVLG on 2D Referential Grounding Datasets of RefCOCO, RefCOCO+, and RefCOCOg:** The green masks indicate predictions of UniVLG.



"The door is the one that is on the right when facing the two doors on the same wall."



"The bed closest to the wall heater."



"The taller bookshelf."

Figure 3.5: **Systematic failure modes of UniVLG:** Green boxes and masks are ground-truth, red masks and boxes are UniVLG’s predictions.

decreases. We use relative pose error as defined in [Stu+12].

We compare the robustness of UniVLG to prior state-of-the-art single-stage method of BUTD-DETR [Jai+22]. We chose a single-stage method as our baseline, since multi-stage methods like PQ3D [Zhu+24b] and 3D-VisTA [Zhu+23b] rely on several external models, and use pre-processed intermediate outputs from them for their inference. This makes it

Table 3.11: **Extra Metrics on ScanQA validation set**

	Method	EM	BLEU-1	ROUGE	METEOR	CIDEr
Mesh PC	3D-LLM [Hon+23]	20.5	39.3	35.7	14.5	69.4
	PQ3D [Zhu+24b]	21.0	-	-	-	-
	3D-VisTA [Zhu+23b]	22.5	32.0	35.5	13.8	69.1
	NaviLLM [Zhe+24a]	23.0	-	38.4	15.4	75.9
Sensor PC	3D-VisTA [Zhu+23b]	21.6	30.1	34.1	13.2	65.3
	UniVLG (Ours)	25.7	36.1	40.0	15.2	78.5

harder to fairly run comparisons directly on the point cloud input. As shown in Figure-3.6, UniVLG is highly robust to both types of noise. At a mean error of 0.2, UniVLG impressively maintains a Top1@0.25 IoU accuracy of 66.7%.

In the pose error case, the model must understand the misaligned point cloud and cannot simply ignore the spurious points. However, UniVLG still shows impressive robustness with substantially less degradation compared to BUDT-DETR.

We believe a great portion of robustness comes from reliance on 2D pre-trained features and 2D layers in the network. Despite the noise in the depth and pose estimation, they still operate over the clean RGB images. Additionally, our 3D layers use local and relative attentions, which additionally contribute to the robustness.

3.8.10 Miscellaneous Details

Frame Sampling To improve training efficiency, we opt to train on only a subset of available frames in each scene. This is critical, not only for reducing the average cost per step, but also in ensuring that the computations and memory is fixed per-step, allowing us to maximize the batch size and prevent waiting between GPUs in DDP. We initially tested a random selection strategy where each caption was paired with $N = 15$ images from a given scene, with each scene originally containing around 90 frames. However, this change means that some sets of frames may no longer align with a given caption (i.e., the referenced object may not be visible in the selected frames). By simply ignoring instances where this mismatch occurs, this strategy performed remarkably well overall.

However, to improve performance further, we sought to specifically include frames that were relevant to the caption (e.g., given "the red chair", we want to make sure all chairs in the scene are included), without biasing the model and causing a train/test distribution gap. To do this, we computed the CLIP embeddings of all scene frames, as well as the text embeddings of all captions. For a given caption, we select 5 relevant frames using the cosine similarity of these embeddings, ensuring that referenced entities are part of the selected frames. Moreover, to ensure diversity of frames and prevent the aforementioned bias, we select the remaining 10 frames using Furthest-Point-Sampling (FPS) on the CLIP image embedding space. We found fewer than 15 frames to slightly reduce performance, but that increasing beyond this point had no benefit.

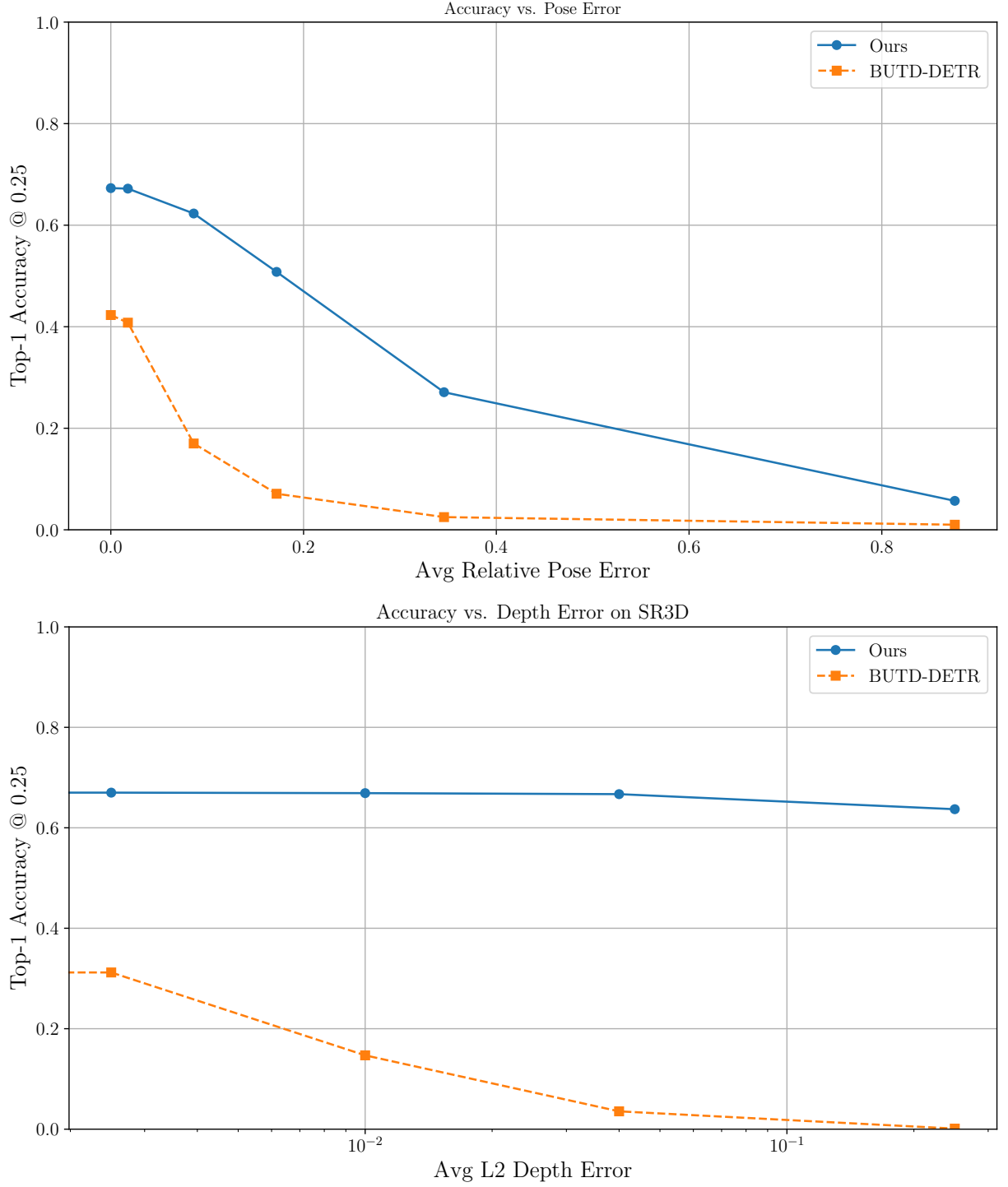


Figure 3.6: We analyze the performance of UniVLG and BUTD-DETR on SR3D as the pose and depth error increases. We add gaussian noise to the pose and raw depth which affects the unprojected point cloud that both models observe.

Language Encoder As the original CLIP model only supported a maximum of 77 tokens, we opted to use Jina-CLIP which uses RoPE embeddings and is trained on 512 tokens.

Matching Cost Tuning the Hungarian matching cost weights turned out to be critical for 3D referential grounding. In prior works [Jai+24; Che+22b], mask cost weights are typically higher than class cost weights. We find that it is important to have high class cost weights in referential grounding. This difference in costs is only required in Hungarian matching, we do not observe much benefits in changing the loss weights from what was used in the prior works.

Mask Decoder design Unlike prior works [Jai+24; Che+22b], where object queries attend to visual features across multiple resolutions, we attend only to the highest-resolution feature scale. We explored multi-scale variants of our model but struggled to achieve good results, primarily due to difficulties in properly updating multi-scale visual feature maps with object queries and language tokens. Single-scale mask decoding approach is suboptimal for backbones like Swin, where attending to high-resolution feature maps is computationally expensive. However, with ViT backbones, where all feature maps share the same resolution, this limitation is less pronounced.

Box Loss As described in the main text, we employ the box loss to enhance the sharpness of the predicted masks. The box loss penalizes outlier points predicted by the mask head, which might not receive significant penalties through the mask loss alone. We incorporate the box loss into both Hungarian matching and the final loss used for backpropagation. Initially, we hypothesized that applying the box loss solely in Hungarian matching would suffice, as the bounding box is derived only from the extreme points of the predicted mask, resulting in sparse gradients during backpropagation. However, empirical results demonstrate that explicitly including the box loss in the final loss is beneficial. Notably, we use bounding box loss exclusively for 3D datasets and not for (lifted) 2D datasets. This distinction arises because lifted 3D point clouds can be noisy, potentially leading to inaccurate 3D bounding boxes.

3.8.11 2D-3D Generalization Test

Table 3.12: 2D-3D Generalization Test

Model	3D Sup. Classes	2D Sup. Classes
UniVLG	72.6	53.8
UniVLG w/o 2D-to-3D lifting	71.4	0.0
UniVLG (Upper-Bound)	69.7	84.2

We study the generalization between 2D and 3D domains using the 3D Instance Segmentation benchmark in the AI2-THOR simulator [Kol+17]. AI2-THOR consists of 120

object classes and provides posed RGB-D images of its 3D environments. We split these classes into two disjoint subsets: (a) 3D Supervised Classes (30) and (b) 2D Supervised Classes (90).

We train UniVLG in a 2D-3D setting where, for multi-view RGB-D inputs, the model is tasked with detecting objects from the 3D Supervised Classes subset. For single RGB image inputs, it is tasked with detecting objects from the 2D Supervised Classes subset. At test time, the model is evaluated in a multi-view posed RGB-D setup on all 120 object classes to assess its generalization across the two subsets. Results are presented in [Table 3.12](#).

We find that when UniVLG does not lift 2D images to 3D—simply skipping 3D attention layers in 2D batches, similar to ODIN [\[Jai+24\]](#) (row 2 in [Table 3.12](#))—performance on 2D Supervised Classes drops to nearly zero. This suggests that without depth-based transformation, the model fails to learn from 2D supervision. However, when UniVLG incorporates depth information, its performance on 2D Supervised Classes becomes non-trivial, indicating that it successfully learns from 2D supervision and applies this knowledge in the 3D domain at test time.

In row 3, we report results for UniVLG trained in 3D on all 120 classes, representing an upper bound on its performance given full 3D supervision. We observe that UniVLG in row 1 remains significantly below this upper bound, highlighting the need for further improvements in 2D-to-3D generalization.

3.8.12 Discussion: Decoding Masks vs. Boxes

Decoding 3D bounding boxes has its advantages. For instance, datasets like Arkit3DScenes [\[Bar+21\]](#) and Aria [\[Str+24\]](#) only provide supervision for 3D boxes, making box decoding more favorable in such scenarios. However, recent methods such as Box2Mask [\[Chi+22\]](#) demonstrate that segmentation predictions can be effectively supervised using bounding box annotations alone, suggesting that the lack of mask labels may not pose a significant limitation. Despite this, we observe failure cases in mask predictions, such as outlier points being segmented in 3D or multiple instances of the same object being predicted as the answer (see [Figure 3.5](#) in Appendix). These errors result in oversized or incorrect bounding boxes when masks are converted to boxes in a post-processing step, indicating the need for further research to address these issues in mask-decoding heads.

On the other hand, decoding masks offers significant benefits. As our experiments show, predicting masks leads to better performance compared to decoding boxes. Moreover, models like ODIN and UniVLG aim to unify 2D and 3D perception tasks, and mask prediction provides a consistent interface across these modalities. Masks represent per-point segmentation of pixels or points, whether in 2D or 3D, whereas box decoding requires separate prediction heads for 2D (4D outputs) and 3D (6D outputs), complicating unification efforts. Additionally, box-decoding heads are sensitive to normalization requirements for 3D scenes, such as mean-centering or scaling within a specific range (e.g., 0 to 1). Indoor and outdoor datasets, which vary significantly in 3D extents, often require separate decoder heads, further complicating training. In contrast, mask-decoding heads rely on cosine similarity between pixel/point features and object queries, making them more robust to variations in scene normalization and context. For example, UniVLG maintains strong performance

even when input scenes are translated by 1000 meters, despite not being explicitly trained with translation augmentations. Furthermore, annotating 3D masks is easier for humans than annotating 3D boxes, especially with recent advances in interactive segmentation methods like SAM [Kir+23], which simplify the process of creating accurate mask annotations. This robustness, simplicity, and annotation efficiency make mask-decoding heads a preferable choice for unifying 2D-3D perception tasks.

Table 3.13: Mask mAP evaluation on 3D language grounding. We evaluate top-1 accuracy on the official validation set without assuming ground-truth proposals (Det).

Method	SR3D			NR3D			ScanRefer		
	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc	Acc
	@25	@50	@75	@25	@50	@75	@25	@50	@75
	(Det)	(Det)	(Det)	(Det)	(Det)	(Det)	(Det)	(Det)	(Det)
UniVLG	75.4	69.2	54.4	59.9	52.3	39.4	66.8	60.8	48.0

3.8.13 Evaluation with Mask based Intersection Over Union

While standard evaluations on existing 3D language grounding benchmarks rely on bounding box Intersection Over Union (IoU) for computing accuracy, we additionally report results using mask-based IoU for accuracy computation. The corresponding results are presented in Table 3.13.

3.8.14 Additional Related Work

Language Understanding Benchmarks Vision Language Grounding is the task of localizing the objects mentioned in a language utterance in a given 2D or 3D scene. In the 2D domain, this task is primarily benchmarked on RefCOCO/+g datasets [Kaz+14] with humans annotating language instructions on top of COCO images. In the 3D community, this task is primarily studied in the popular benchmarks of SR3D [Ach+20] containing programmatically generated sentences, and NR3D [Ach+20] and ScanRefer [CCN20], containing human-annotated sentences, and 3D scenes from the ScanNet [Dai+17] dataset. The original benchmarks of SR3D and NR3D provide access to ground-truth bounding boxes of all objects in the scenes as input, and the task is to select the correct bounding box that corresponds to the language sentence. Most methods operate under this assumption, except for BUTD-DETR [Jai+22], which proposed directly predicting 3D bounding boxes instead of selecting from the available proposals. We follow BUTD-DETR and report results without assuming access to ground-truth boxes. The ScanRefer benchmark is similar to NR3D but does not provide ground-truth boxes as input. Recently, ScanQA [Azu+22] and SQA3D [Ma+22] introduced 3D Question Answering Benchmarks. ScanQA focuses on spatial relations. SQA3D [Ma+22] provides pairs of situation descriptions and

questions regarding embodied scene understanding, navigation, common sense and multi-hop reasoning.

3D Question Answering and Captioning For 3D question answering and captioning, approaches like PQ3D [Zhu+24b] and 3D-Vista [Zhu+23b] use small text generation heads on top of their language-contextualized features or queries to decode answers. Other approaches like 3D-LLM [Hon+23] and NaviLLM [Zhe+24a] condense the visual scene features into a set of latent vectors and pass it to large pre-trained LLMs like BLIP2-flan5 [Li+23b] or Vicuna-7B-v0 [Pen+23a]. However, unlike 3D-Vista and PQ3D, they either get significantly poor performance on 3D referential grounding tasks (3D-LLM) or skip evaluating in that setup (NaviLLM). In this work, we follow PQ3D and 3DVista’s approach and use a small text generation head, mainly for its simplicity.

Sensor vs Mesh Point Clouds in 3D benchmarks: All the 3D benchmarks use point clouds derived from the 3D meshes provided by ScanNet [Dai+17]. These meshes were constructed using several steps of post-processing over the raw sensor RGB-D data (which takes minutes-to-hours). In this work, we propose the first 3D language grounding model that operates directly over sensor RGB-D point clouds. For fair comparison, we benchmark other prior works with sensor point clouds as inputs, and show the benefits of using 2D pre-trained features for 3D language understanding tasks. These post-processing steps include mesh reconstruction and camera pose estimation, as well as several manual post-processing steps. These processes create fine-grained misalignments between the reconstructed mesh and the sensor RGB-D stream, resulting in drop in performance for methods operating over sensor RGB-D streams instead of the mesh point clouds, as also shown by prior works [RVL22; Kun+20; Jai+24]. This discourages the use of sensor RGB-D streams and thus the 2D features pre-trained on internet scale data. Using sensor point clouds directly is an emerging idea in the community, further bolstered by the recent introduction of datasets like EmbodiedScan [Wan+23] which also use sensor data directly instead of using meshes.

Conclusions and Future Directions

In conclusion, this work shows that unifying modeling strategies can lead to substantive improvements in vision-language models. First, we show that unified vision-language discrete diffusion offers compelling advantages in controllability, multimodal inpainting, and inference efficiency over traditional autoregressive methods. Moreover, we show that unified modeling of 2D and 3D data for vision-language tasks common to embodied agents provides a significant boost in 3D task performance by increasing the available size and diversity of training data. We hope that this thesis inspires future progress in extending the success of large-scale language modeling to new tasks and domains.

Bibliography

This bibliography contains 139 references.

- [Abd+23] Ahmed Abdelreheem, Kyle Olszewski, Hsin-Ying Lee, Peter Wonka, and Panos Achlioptas. *ScanEnts3D: Exploiting Phrase-to-3D-Object Correspondences for Improved Visio-Linguistic Models in 3D Scenes*. 2023. arXiv: [2212.06250](https://arxiv.org/abs/2212.06250) [cs.CV]. URL: <https://arxiv.org/abs/2212.06250>.
- [Ach+20] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. “ReferIt3D: Neural Listeners for Fine-Grained 3D Object Identification in Real-World Scenes”. In: *Proc. ECCV*. 2020.
- [Ahn+22] Michael Ahn et al. *Do As I Can, Not As I Say: Grounding Language in Robotic Affordances*. 2022. arXiv: [2204.01691](https://arxiv.org/abs/2204.01691) [cs.R0]. URL: <https://arxiv.org/abs/2204.01691>.
- [Ala+22] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. “Flamingo: a visual language model for few-shot learning”. In: *Advances in neural information processing systems* 35 (2022), pp. 23716–23736.
- [Aus+21] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. “Structured Denoising Diffusion Models in Discrete State-Spaces”. In: (2021). arXiv: [2107.03006](https://arxiv.org/pdf/2107.03006) [astro-ph.IM]. URL: <http://arxiv.org/pdf/2107.03006>.
- [Azu+22] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. “Scanqa: 3d question answering for spatial scene understanding”. In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 19129–19139.
- [Bac+24] Roman Bachmann, Oguzhan Fatih Kar, David Mizrahi, Ali Garjani, Mingfei Gao, David Griffiths, Jiaming Hu, Afshin Dehghan, and Amir Zamir. “4m-21: An any-to-any vision model for tens of tasks and modalities”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 61872–61911.
- [Bar+21] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. “Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data”. In: *arXiv preprint arXiv:2111.08897* (2021).
- [Car+20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. “End-to-End Object Detection with Transformers”. In: *Proc. ECCV*. 2020.

- [CCN20] Dave Zhenyu Chen, Angel Chang, and Matthias Nießner. “ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language”. In: *Proc. ECCV*. 2020.
- [Cha+17] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. “Matterport3d: Learning from rgb-d data in indoor environments”. In: *arXiv preprint arXiv:1709.06158* (2017).
- [Cha+21] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. *Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts*. 2021. arXiv: [2102.08981](https://arxiv.org/abs/2102.08981) [cs.CV]. URL: <https://arxiv.org/abs/2102.08981>.
- [Cha+22] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. “Maskgit: Masked generative image transformer”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11315–11325.
- [Cha+23] Huiwen Chang et al. *Muse: Text-To-Image Generation via Masked Generative Transformers*. 2023. arXiv: [2301.00704](https://arxiv.org/abs/2301.00704) [cs.CV]. URL: <https://arxiv.org/abs/2301.00704>.
- [Che+15] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. *Microsoft COCO Captions: Data Collection and Evaluation Server*. 2015. arXiv: [1504.00325](https://arxiv.org/abs/1504.00325) [cs.CV]. URL: <https://arxiv.org/abs/1504.00325>.
- [Che+20] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. *Scan2Cap: Context-aware Dense Captioning in RGB-D Scans*. 2020. arXiv: [2012.02206](https://arxiv.org/abs/2012.02206) [cs.CV].
- [Che+22a] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. “Language conditioned spatial relation reasoning for 3d object grounding”. In: *Advances in neural information processing systems* 35 (2022), pp. 20522–20535.
- [Che+22b] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. “Masked-attention Mask Transformer for Universal Image Segmentation”. In: 2022.
- [Che+24a] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. *SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities*. 2024. arXiv: [2401.12168](https://arxiv.org/abs/2401.12168) [cs.CV]. URL: <https://arxiv.org/abs/2401.12168>.
- [Che+24b] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. *SpatialRGPT: Grounded Spatial Reasoning in Vision Language Models*. 2024. arXiv: [2406.01584](https://arxiv.org/abs/2406.01584) [cs.CV]. URL: <https://arxiv.org/abs/2406.01584>.
- [Chi+22] Julian Chibane, Francis Engelmann, Tuan Anh Tran, and Gerard Pons-Moll. “Box2mask: Weakly supervised 3d semantic instance segmentation using bounding boxes”. In: *European conference on computer vision*. Springer. 2022, pp. 681–699.

- [Chi+24a] Hao-Tien Lewis Chiang et al. *Mobility VLA: Multimodal Instruction Navigation with Long-Context VLMs and Topological Graphs*. 2024. arXiv: [2407.07775](https://arxiv.org/abs/2407.07775) [cs.R0]. URL: <https://arxiv.org/abs/2407.07775>.
- [Chi+24b] Krishna Teja Chitty-Venkata, Siddhisanket Raskar, Bharat Kale, Farah Ferdous, Aditya Tanikanti, Ken Raffanetti, Valerie Taylor, Murali Emani, and Venkatram Vishwanath. *LLM-Inference-Bench: Inference Benchmarking of Large Language Models on AI Accelerators*. 2024. arXiv: [2411.00136](https://arxiv.org/abs/2411.00136) [cs.LG]. URL: <https://arxiv.org/abs/2411.00136>.
- [Dai+17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. “Scannet: Richly-annotated 3d reconstructions of indoor scenes”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5828–5839.
- [Dao23] Tri Dao. “FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning”. In: *The Twelfth International Conference on Learning Representations*. Oct. 2023. (Visited on 11/13/2024).
- [Dei+22] 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: *arXiv preprint arXiv:2212.08051* (2022).
- [Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: [1810.04805](https://arxiv.org/abs/1810.04805) [cs.CL]. URL: <https://arxiv.org/abs/1810.04805>.
- [Din+21] Ming Ding et al. “CogView: Mastering Text-to-Image Generation via Transformers”. In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan. 2021, pp. 19822–19835. URL: <https://proceedings.neurips.cc/paper/2021/hash/a4d92e2cd541fca87e4620aba658316d-Abstract.html>.
- [Dri+23] Danny Driess et al. “PaLM-E: An Embodied Multimodal Language Model”. In: *International Conference on Machine Learning*. 2023. URL: <https://api.semanticscholar.org/CorpusID:257364842>.
- [Dub+24] Abhimanyu Dubey et al. “The Llama 3 Herd of Models”. In: *ArXiv abs/2407.21783* (2024). URL: <https://api.semanticscholar.org/CorpusID:271571434>.
- [Ess+24] 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: *Forty-first International Conference on Machine Learning*. 2024.
- [Fan+21] Zhaoyuan Fang, Ayush Jain, Gabriel Sarch, Adam W. Harley, and Katerina Fragkiadaki. *Move to See Better: Self-Improving Embodied Object Detection*. 2021. arXiv: [2012.00057](https://arxiv.org/abs/2012.00057) [cs.CV]. URL: <https://arxiv.org/abs/2012.00057>.

- [Gad+24] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, et al. “Datacomp: In search of the next generation of multimodal datasets”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [Gat+24] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. *Discrete Flow Matching*. Nov. 2024. arXiv: [2407.15595](https://arxiv.org/abs/2407.15595). (Visited on 11/13/2024).
- [GH24] Ishaan Gulrajani and Tatsunori B Hashimoto. “Likelihood-based diffusion language models”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [GHS23] Dhruva Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. *GenEval: An Object-Focused Framework for Evaluating Text-to-Image Alignment*. 2023. arXiv: [2310.11513](https://arxiv.org/abs/2310.11513) [cs.CV]. URL: <https://arxiv.org/abs/2310.11513>.
- [Han+23] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. “Efficient diffusion training via min-snr weighting strategy”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 7441–7451.
- [Heu+17] 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: *Advances in neural information processing systems* 30 (2017).
- [HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: [2006.11239](https://arxiv.org/abs/2006.11239) [cs.LG]. URL: <https://arxiv.org/abs/2006.11239>.
- [Hof+22] Jordan Hoffmann et al. *Training Compute-Optimal Large Language Models*. 2022. arXiv: [2203.15556](https://arxiv.org/abs/2203.15556) [cs.CL]. URL: <https://arxiv.org/abs/2203.15556>.
- [Hon+23] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. “3d-llm: Injecting the 3d world into large language models”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 20482–20494.
- [Hoo+] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. “Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions”. In: *Advances in Neural Information Processing Systems* 34 (), pp. 12454–12465. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf.
- [HS22] Jonathan Ho and Tim Salimans. “Classifier-Free Diffusion Guidance”. In: (2022). arXiv: [2207.12598](https://arxiv.org/abs/2207.12598) [astro-ph.IM]. URL: <http://arxiv.org/pdf/2207.12598>.
- [Hu+21] 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: *arXiv preprint arXiv:2106.09685* (2021).

- [Hu+23] Minghui Hu, Chuanxia Zheng, Zuopeng Yang, Tat-Jen Cham, Heliang Zheng, Chaoyue Wang, Dacheng Tao, and Ponnuthurai N. Suganthan. “Unified Discrete Diffusion for Simultaneous Vision-Language Generation”. In: *The Eleventh International Conference on Learning Representations*. 2023.
- [Jae+21] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. “Perceiver io: A general architecture for structured inputs & outputs”. In: *arXiv preprint arXiv:2107.14795* (2021).
- [Jai+22] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. “Bottom up top down detection transformers for language grounding in images and point clouds”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 417–433.
- [Jai+24] Ayush Jain, Pushkal Katara, Nikolaos Gkanatsios, Adam W Harley, Gabriel Sarch, Kriti Aggarwal, Vishrav Chaudhary, and Katerina Fragkiadaki. “ODIN: A Single Model for 2D and 3D Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 3564–3574.
- [JCG24] Priyank Jaini, Kevin Clark, and Robert Geirhos. *Intriguing properties of generative classifiers*. 2024. arXiv: [2309.16779 \[cs.CV\]](https://arxiv.org/abs/2309.16779). URL: <https://arxiv.org/abs/2309.16779>.
- [Kam+21] Aishwarya Kamath, Mannat Singh, Yann André LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. “MDETR - Modulated Detection for End-to-End Multi-Modal Understanding”. In: *Proc. ICCV*. 2021.
- [Kap+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. “Scaling Laws for Neural Language Models”. In: *CoRR* abs/2001.08361 (2020). arXiv: [2001.08361](https://arxiv.org/abs/2001.08361). URL: <https://arxiv.org/abs/2001.08361>.
- [Kaz+14] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. “Refer-ItGame: Referring to Objects in Photographs of Natural Scenes”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 787–798. DOI: [10.3115/v1/D14-1086](https://doi.org/10.3115/v1/D14-1086). URL: <https://aclanthology.org/D14-1086/>.
- [Kim+22] Namyup Kim, Dongwon Kim, Cuiling Lan, Wenjun Zeng, and Suha Kwak. *ReSTR: Convolution-free Referring Image Segmentation Using Transformers*. 2022. arXiv: [2203.16768 \[cs.CV\]](https://arxiv.org/abs/2203.16768). URL: <https://arxiv.org/abs/2203.16768>.
- [Kir+23] 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: *arXiv preprint arXiv:2304.02643* (2023).
- [Kol+17] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al.

- “Ai2-thor: An interactive 3d environment for visual ai”. In: *arXiv preprint arXiv:1712.05474* (2017).
- [Kou+24] Andreas Koukounas, Georgios Mastrapas, Michael Günther, Bo Wang, Scott Martens, Isabelle Mohr, Saba Sturua, Mohammad Kalim Akram, Joan Fontanals Martínez, Saahil Ognawala, et al. “Jina CLIP: Your CLIP Model Is Also Your Text Retriever”. In: *arXiv preprint arXiv:2405.20204* (2024).
- [Kun+20] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. “Virtual multi-view fusion for 3d semantic segmentation”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*. Springer. 2020, pp. 518–535.
- [LA22] Adam Dahlgren Lindström and Savitha Sam Abraham. *CLEVR-Math: A Dataset for Compositional Language, Visual, and Mathematical Reasoning*. 2022. DOI: [10.48550/ARXIV.2208.05358](https://doi.org/10.48550/ARXIV.2208.05358). URL: <https://arxiv.org/abs/2208.05358>.
- [Lai+23] Xin Lai, Yuhui Yuan, Ruihang Chu, Yukang Chen, Han Hu, and Jiaya Jia. “Mask-attention-free transformer for 3d instance segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 3693–3703.
- [LH19] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. arXiv:1711.05101 [cs, math]. Jan. 2019. URL: <http://arxiv.org/abs/1711.05101> (visited on 11/10/2022).
- [Li+22a] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. “Grounded language-image pre-training”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10965–10975.
- [Li+22b] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. *Diffusion-LM Improves Controllable Text Generation*. 2022. arXiv: [2205.14217](https://arxiv.org/abs/2205.14217) [cs.CL]. URL: <https://arxiv.org/abs/2205.14217>.
- [Li+23a] Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. *Your Diffusion Model is Secretly a Zero-Shot Classifier*. 2023. arXiv: [2303.16203](https://arxiv.org/abs/2303.16203) [cs.LG]. URL: <https://arxiv.org/abs/2303.16203>.
- [Li+23b] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models”. In: *International conference on machine learning*. PMLR. 2023, pp. 19730–19742.
- [Li+24a] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. “Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation”. In: *arXiv preprint arXiv:2402.17245* (2024).
- [Li+24b] Xianhang Li, Haoqin Tu, Mude Hui, Zeyu Wang, Bingchen Zhao, Junfei Xiao, Sucheng Ren, Jieru Mei, Qing Liu, Huangjie Zheng, et al. “What If

- We Recaption Billions of Web Images with LLaMA-3?” In: *arXiv preprint arXiv:2406.08478* (2024).
- [Lin+14] 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: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.
- [Liu+19] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan Yuille. *CLEVR-Ref+: Diagnosing Visual Reasoning with Referring Expressions*. 2019. arXiv: [1901.00850 \[cs.CV\]](https://arxiv.org/abs/1901.00850). URL: <https://arxiv.org/abs/1901.00850>.
- [Liu+21] 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: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.
- [Liu+23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. *Visual Instruction Tuning*. 2023. arXiv: [2304.08485 \[cs.CV\]](https://arxiv.org/abs/2304.08485). URL: <https://arxiv.org/abs/2304.08485>.
- [Liu+24] Dongyang Liu, Shitian Zhao, Le Zhuo, Weifeng Lin, Yu Qiao, Hongsheng Li, and Peng Gao. “Lumina-mGPT: Illuminate Flexible Photorealistic Text-to-Image Generation with Multimodal Generative Pretraining”. In: *arXiv preprint arXiv:2408.02657* (2024).
- [LME] Aaron Lou, Chenlin Meng, and Stefano Ermon. “Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution”. In: *Forty-first International Conference on Machine Learning*.
- [Lu+22] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Anirudha Kembhavi. “Unified-io: A unified model for vision, language, and multi-modal tasks”. In: *The Eleventh International Conference on Learning Representations*. 2022.
- [Lu+23a] Jiahao Lu, Jiacheng Deng, Chuxin Wang, Jianfeng He, and Tianzhu Zhang. “Query Refinement Transformer for 3D Instance Segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 18516–18526.
- [Lu+23b] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. “Unified-IO 2: Scaling Autoregressive Multimodal Models with Vision, Language, Audio, and Action”. In: *arXiv preprint arXiv:2312.17172* (2023).
- [Luo+22] Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. “3d-sps: Single-stage 3d visual grounding via referred point progressive selection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16454–16463.
- [Luo+24] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. “Open-MAGVIT2: An Open-Source Project Toward Democratizing Auto-regressive Visual Generation”. In: *arXiv preprint arXiv:2409.04410* (2024).

- [Ma+22] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. “Sqa3d: Situated question answering in 3d scenes”. In: *arXiv preprint arXiv:2210.07474* (2022).
- [Meh+24] Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, et al. “OpenELM: An Efficient Language Model Family with Open-source Training and Inference Framework”. In: *arXiv preprint arXiv:2404.14619* (2024).
- [MGJ21] Ishan Misra, Rohit Girdhar, and Armand Joulin. “An end-to-end transformer model for 3d object detection”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 2906–2917.
- [Miz+23] David Mizrahi, Roman Bachmann, Oguzhan Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. “4m: Massively multimodal masked modeling”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 58363–58408.
- [Nie+24] Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. *Scaling up Masked Diffusion Models on Text*. 2024. arXiv: [2410.18514](https://arxiv.org/abs/2410.18514) [cs.AI]. URL: <https://arxiv.org/abs/2410.18514>.
- [Oqu+24] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2024. arXiv: [2304.07193](https://arxiv.org/abs/2304.07193) [cs.CV]. URL: <https://arxiv.org/abs/2304.07193>.
- [Pen+23a] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. “Instruction tuning with gpt-4”. In: *arXiv preprint arXiv:2304.03277* (2023).
- [Pen+23b] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. “Openscene: 3d scene understanding with open vocabularies”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 815–824.
- [Plu+16] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. *Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models*. 2016. arXiv: [1505.04870](https://arxiv.org/abs/1505.04870) [cs.CV]. URL: <https://arxiv.org/abs/1505.04870>.
- [Pra+23] Mihir Prabhudesai, Tsung-Wei Ke, Alexander C. Li, Deepak Pathak, and Katerina Fragkiadaki. *Diffusion-TTA: Test-time Adaptation of Discriminative Models via Generative Feedback*. 2023. arXiv: [2311.16102](https://arxiv.org/abs/2311.16102) [cs.CV]. URL: <https://arxiv.org/abs/2311.16102>.
- [Rad+21] 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*. PMLR. 2021, pp. 8748–8763.
- [Raf+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. “Exploring the limits

- of transfer learning with a unified text-to-text transformer”. In: *Journal of machine learning research* 21.140 (2020), pp. 1–67.
- [Rez+19] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. “Generalized intersection over union: A metric and a loss for bounding box regression”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 658–666.
- [RLD22] David Rozenberszki, Or Litany, and Angela Dai. “Language-grounded indoor 3d semantic segmentation in the wild”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 125–141.
- [RM23] Sai Saketh Rambhatla and Ishan Misra. *SelfEval: Leveraging the discriminative nature of generative models for evaluation*. 2023. arXiv: [2311.10708](https://arxiv.org/abs/2311.10708) [cs.CV]. URL: <https://arxiv.org/abs/2311.10708>.
- [Roh+22] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. “Languagerefer: Spatial-language model for 3d visual grounding”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 1046–1056.
- [Rom+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.
- [RVK22] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. “Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 2397–2406.
- [RVL22] Damien Robert, Bruno Vallet, and Loic Landrieu. “Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5575–5584.
- [Sah+24] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. “Simple and Effective Masked Diffusion Language Models”. In: *arXiv preprint arXiv:2406.07524* (2024).
- [Sch+23] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. “Mask3D: Mask Transformer for 3D Semantic Instance Segmentation”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 8216–8223.
- [Seh+24] Vikash Sehwal, Xianghao Kong, Jingtao Li, Michael Spranger, and Lingjuan Lyu. “Stretching Each Dollar: Diffusion Training from Scratch on a Micro-Budget”. In: *arXiv preprint arXiv:2407.15811* (2024).
- [Shi+24] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. “Simplified and Generalized Masked Diffusion for Discrete Data”. In: *arXiv preprint arXiv:2406.04329* (2024).
- [Sid+23] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kotschieder. “Panoptic lifting for 3d scene

- understanding with neural fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 9043–9052.
- [Sin+24] Vasu Singla, Kaiyu Yue, Sukriti Paul, Reza Shirkavand, Mayuka Jayawardhana, Alireza Ganjdanesh, Heng Huang, Abhinav Bhatele, Gowthami Somepalli, and Tom Goldstein. *From Pixels to Prose: A Large Dataset of Dense Image Captions*. 2024. arXiv: [2406.10328](https://arxiv.org/abs/2406.10328) [cs.CV]. URL: <https://arxiv.org/abs/2406.10328>.
- [Soh+] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: (), pp. 2256–2265. URL: <http://proceedings.mlr.press/v37/sohl-dickstein15.pdf>.
- [Son+20] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2020. arXiv: [2011.13456](https://arxiv.org/abs/2011.13456) [cs.LG].
- [Str+24] Julian Straub, Daniel DeTone, Tianwei Shen, Nan Yang, Chris Sweeney, and Richard Newcombe. *EFM3D: A Benchmark for Measuring Progress Towards 3D Egocentric Foundation Models*. 2024. arXiv: [2406.10224](https://arxiv.org/abs/2406.10224) [cs.CV]. URL: <https://arxiv.org/abs/2406.10224>.
- [Stu+12] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. “A benchmark for the evaluation of RGB-D SLAM systems”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 573–580. DOI: [10.1109/IRoS.2012.6385773](https://doi.org/10.1109/IRoS.2012.6385773).
- [Su+23] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. 2023. arXiv: [2104.09864](https://arxiv.org/abs/2104.09864) [cs.CL]. URL: <https://arxiv.org/abs/2104.09864>.
- [Sun+23] Keqiang Sun et al. *JourneyDB: A Benchmark for Generative Image Understanding*. 2023. arXiv: [2307.00716](https://arxiv.org/abs/2307.00716) [cs.CV]. URL: <https://arxiv.org/abs/2307.00716>.
- [Sun+24] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. “Autoregressive Model Beats Diffusion: Llama for Scalable Image Generation”. In: *arXiv preprint arXiv:2406.06525* (2024).
- [Tan+24] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. “Any-to-any generation via composable diffusion”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [Tea+23] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. “Gemini: a family of highly capable multimodal models”. In: *arXiv preprint arXiv:2312.11805* (2023).
- [Tea24] Chameleon Team. “Chameleon: Mixed-Modal Early-Fusion Foundation Models”. In: *arXiv preprint arXiv:2405.09818* (2024). DOI: [10.48550/arXiv.2405.09818](https://doi.org/10.48550/arXiv.2405.09818). URL: <https://github.com/facebookresearch/chameleon>.
- [Thr+22] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. *Winoground: Probing Vision and Language*

- Models for Visio-Linguistic Compositionality*. 2022. arXiv: [2204.03162](https://arxiv.org/abs/2204.03162) [cs.CV]. URL: <https://arxiv.org/abs/2204.03162>.
- [Ton+24] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. “Cambrian-1: A fully open, vision-centric exploration of multimodal llms”. In: *arXiv preprint arXiv:2406.16860* (2024).
- [Tou+23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. “Llama 2: Open foundation and fine-tuned chat models”. In: *arXiv preprint arXiv:2307.09288* (2023).
- [Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. Tech. rep. arXiv:1706.03762. arXiv:1706.03762 [cs] type: article. arXiv, Dec. 2017. DOI: [10.48550/arXiv.1706.03762](https://doi.org/10.48550/arXiv.1706.03762). URL: <http://arxiv.org/abs/1706.03762> (visited on 05/21/2022).
- [Wal+19] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. “Rio: 3d object instance re-localization in changing indoor environments”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7658–7667.
- [Wan+23] Tai Wang et al. *EmbodiedScan: A Holistic Multi-Modal 3D Perception Suite Towards Embodied AI*. 2023. arXiv: [2312.16170](https://arxiv.org/abs/2312.16170) [cs.CV]. URL: <https://arxiv.org/abs/2312.16170>.
- [Wan+24] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. “Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision”. In: *arXiv preprint arXiv:2410.19115* (2024).
- [Wor+23] Mitchell Wortsman, Peter J Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, et al. “Small-scale proxies for large-scale transformer training instabilities”. In: *arXiv preprint arXiv:2309.14322* (2023).
- [Xu+23] Mutian Xu, Xingyilang Yin, Lingteng Qiu, Yang Liu, Xin Tong, and Xiaoguang Han. *SAMPro3D: Locating SAM Prompts in 3D for Zero-Shot Scene Segmentation*. 2023. arXiv: [2311.17707](https://arxiv.org/abs/2311.17707) [cs.CV]. URL: <https://arxiv.org/abs/2311.17707>.
- [Yad+23] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. “Habitat-matterport 3d semantics dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 4927–4936.
- [Yan+21] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. “SAT: 2D Semantics Assisted Training for 3D Visual Grounding”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1856–1866.

- [Yan+22] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip H. S. Torr. *LAVT: Language-Aware Vision Transformer for Referring Image Segmentation*. 2022. arXiv: [2112.02244](https://arxiv.org/abs/2112.02244) [cs.CV]. URL: <https://arxiv.org/abs/2112.02244>.
- [Yes+23] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. “Scannet++: A high-fidelity dataset of 3d indoor scenes”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 12–22.
- [Yu+23] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. “Language Model Beats Diffusion—Tokenizer is Key to Visual Generation”. In: *arXiv preprint arXiv:2310.05737* (2023).
- [Yua+21] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. “InstanceRefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1791–1800.
- [ZGC23] Yiming Zhang, ZeMing Gong, and Angel X Chang. “Multi3drefer: Grounding text description to multiple 3d objects”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 15225–15236.
- [Zhe+24a] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. “Towards learning a generalist model for embodied navigation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 13624–13634.
- [Zhe+24b] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. *Masked Diffusion Models are Secretly Time-Agnostic Masked Models and Exploit Inaccurate Categorical Sampling*. 2024. arXiv: [2409.02908](https://arxiv.org/abs/2409.02908) [cs.LG].
- [Zho+24] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. “Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model”. In: (2024). arXiv: [2408.11039](https://arxiv.org/abs/2408.11039) [astro-ph.IM]. URL: <http://arxiv.org/pdf/2408.11039>.
- [Zhu+23a] Chenming Zhu, Wenwei Zhang, Tai Wang, Xihui Liu, and Kai Chen. “Object2scene: Putting objects in context for open-vocabulary 3d detection”. In: *arXiv preprint arXiv:2309.09456* (2023).
- [Zhu+23b] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. “3d-vista: Pre-trained transformer for 3d vision and text alignment”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 2911–2921.
- [Zhu+24a] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. *LLaVA-3D: A Simple yet Effective Pathway to Empowering LMMs with 3D-*

- awareness*. 2024. arXiv: [2409.18125](https://arxiv.org/abs/2409.18125) [cs.CV]. URL: <https://arxiv.org/abs/2409.18125>.
- [Zhu+24b] Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. “Unifying 3D Vision-Language Understanding via Promptable Queries”. In: *arXiv preprint arXiv:2405.11442* (2024).
- [Zhu+24c] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Lirui Zhao, Fu-Yun Wang, Zhanyu Ma, et al. “Lumina-Next: Making Lumina-T2X Stronger and Faster with Next-DiT”. In: *arXiv preprint arXiv:2406.18583* (2024).
- [Ziv+24] Alon Ziv, Itai Gat, Gael Le Lan, Tal Remez, Felix Kreuk, Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. *Masked Audio Generation Using a Single Non-Autoregressive Transformer*. Mar. 2024. arXiv: [2401.04577](https://arxiv.org/abs/2401.04577). (Visited on 11/13/2024).
- [Zou+23] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. “Generalized decoding for pixel, image, and language”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 15116–15127.
- [ZS19] Biao Zhang and Rico Sennrich. “Root mean square layer normalization”. In: *Advances in Neural Information Processing Systems* 32 (2019).