Analogical Networks: Memory-Modulated In-Context 3D Parsing

Mayank Singh

CMU-RI-TR-22-67

December, 2022

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

Thesis Committee:
Katerina Fragkiadaki, *chair*
Shubham Tulsiani
Gengshan Yang

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

Abstract

Recent advances in the applications of deep neural networks to numerous visual perception tasks have shown excellent performance. However, this generally requires access to large amount of training samples and hence one persistent challenge is the setting of *few-shot learning* (i.e. ability to adapt to new tasks using only a few labeled samples without forgetting the original distribution). In most existing 3D fine-grained parsing related works, a separate parametric neural model is trained to parse each semantic category, which hinders knowledge sharing across objects and few-shot generalization to novel categories. In this thesis, we present Analogical Networks, a model that casts fine-grained 3D visual parsing as analogical inference: instead of mapping input scenes to part labels, which is hard to adapt in a few-shot manner to novel inputs, our model retrieves related scenes from memory and their corresponding part structures, and predicts analogous part structures in the input scene, via an end-to-end learnable modulation mechanism. By conditioning on more than one memory and using this memory as in-context information, compositions of structures are predicted, that mix and match parts from different visual exemplars. This is a memory inspired learning framework for perception parsing tasks that encodes domain knowledge explicitly in a vast collection of memories at different levels of abstraction, in addition to those implicitly encoded as model parameters. We show that Analogical Networks excel at few-shot 3D parsing, where instances of novel object categories are successfully parsed simply by expanding the model's memory, without any weight updates. Analogical Networks outperform existing state-of-the-art detection transformer models, as well as related meta-learning and few-shot learning techniques at part segmentation. We show that part correspondences emerge across memory and input scenes by simply training for a label-free segmentation objective, as a byproduct of the analogical inductive bias.

Acknowledgments

I would like to thank my professors, collaborators, classmates, friends, and family without whom this thesis would not be possible. In particular, I would like to thank my advisor, Prof. Katerina Fragkiadaki for her guidance and support throughout my Master's study. Your dedication and passion for pursuing fundamental research inspired me to focus and pursue relevant problems in computer vision. I would also like to thank Prof. Shubham Tulsiani for guiding me throughout the thesis project. I have learned from the discussions with my advisors about how to approach a problem statement and come up with a methodical set of experiments to further explore the research project.

Besides my advisors, I really appreciate the rest of my committee members. Thank you Gengshan Yang for always being supportive and providing feedback for my thesis work. I would also like to thank my collaborators in our Lab from whom I learned a lot about research work. Thank you to Adam Harley, Ayush Jain, Fish Tung, Gabe Sarch, Hao Zhu, Jing Wen, Jingyun Yang, Mihir Prabhudesai, Nikos Gkanatsios, Paul Schydlo, Wen-Hsuan Chu, Xian Zhou, Yiming Zuo, Yunchu Zhang and Zhaoyuan Fang. It has been a great pleasure to work with you all.

Thank you to all my friends and classmates that made my journey at CMU very much enjoyable. Also, I would like to thank my parents for their unconditional support.

Contents

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

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The dominant paradigm in existing deep visual learning is to train high capacity networks that map input visual observations to task-specific output labels. Despite their success across a plethora of tasks, these models struggle to perform well in *few-shot* settings where only a small set of examples are available for learning. Meta-learning approaches provide one promising solution to this setting by enabling efficient task-specific adaptation of generic models, but this specialization comes at the cost of poor performance on the original tasks as well as the need to adapt separate models for each novel task considered. In this work, we introduce an alternative approach that unifies inference in both few-shot and many-shot settings. Instead of pursuing the prediction of task-specific outputs, we formulate an approach for analogy-driven prediction, where the desired prediction can be dynamically specified at inference given labeled examples from memory.

1.2 Contributions

In his seminal work [3], Moshe Bar argued for the importance of analogies and associations in human reasoning—highlighting how associations of novel inputs to analogous representations in memory can drive perceptual inference. In this work, we

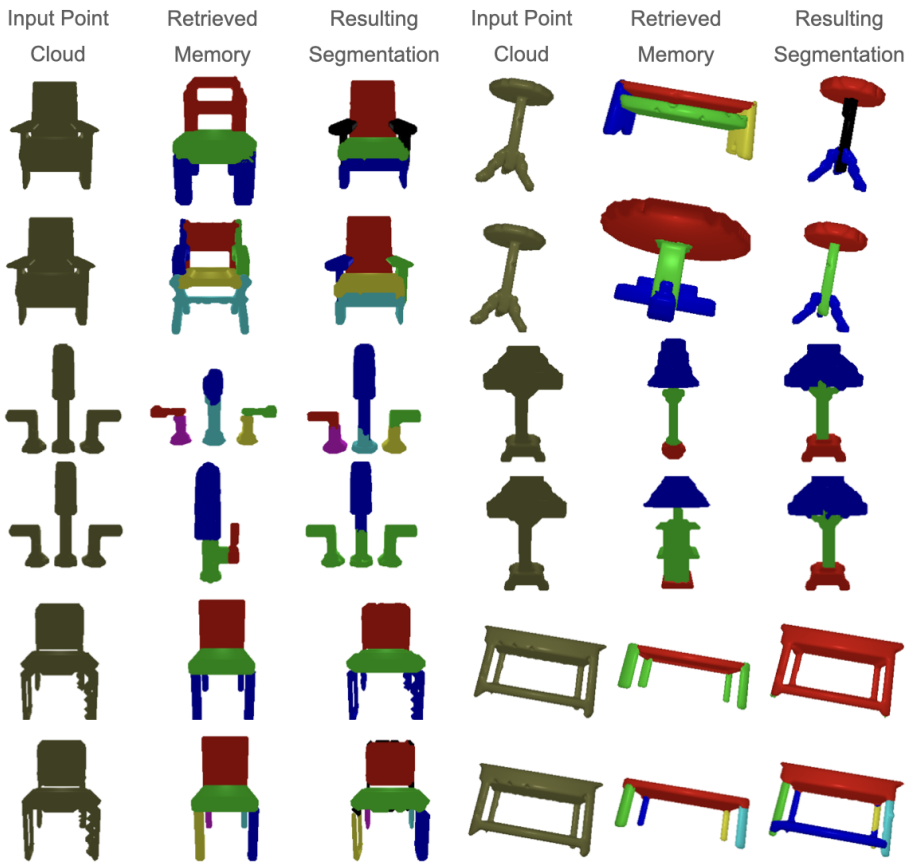

Figure 1.1: **Cross-object part correspondences emerge in Analogical Networks for 3D object segmentation**, without part association or semantic part labelling supervision. One model is trained across multiple object categories and segmentation granularities. Retrieved memories (2nd and 5th columns) modulate segmentation of the input 3D point cloud (1st and 4th columns, respectively). Contextualized memory part embeddings decode analogous parts in the input scene. We indicate corresponding parts between the memory and the input scene with the same color. Parts shown in black in columns 3, and 6 are decoded from scene-agnostic queries, and thus they are not in correspondence to any parts of the memory scene. *Rows 5, 6:* Conditioning the same input point cloud on memories with finer or coarser segmentation results in the segmentation of analogous granularity.

operationalize these insights and propose Analogical Networks, an analogical learning framework for 3D visual parsing. Our approach encodes domain knowledge explicitly in a collection of structured parsed scene memories as well as implicitly in model

parameters. Given an input 3D shape, the model retrieves relevant memories on-the-fly and uses them to modulate inference and segment object parts in the input point cloud. The modulation mechanism operates as top-down guidance for the model to focus on the retrieved memories and infer the *analogous* parts in the input scene. During modulation, the input scene and the retrieved memories are jointly encoded and contextualized via cross-attention operations. The memory part features are then used to decode *corresponding* parts in the input. Instead of mapping the input point cloud to semantic class labels, the model reasons analogically and maps the input to modifications and compositions of past visual memories without any explicit semantic label prediction step. Memory to input scene part associations emerge in our model without association supervision, simply as a byproduct of the analogical inductive biases of the modulation process, as shown in Figure 1.1.

In Analogical Networks, the encoder and modulator parameters are trained end-to-end for the visual parsing task, but the memory retrieval process is not end-to-end differentiable. We devise a novel pre-training scheme where we train the encoder and modulator sub-modules to parse an augmented version of a 3D scene given the original, un-augmented scene as the modulating memory, bypassing the retrieval process. We show this pre-training helps our model, particularly in the few-shot setting.

We test our model on the 3D object segmentation PartNet benchmark of Mo et al. [33]. We compare against state-of-the-art (SOTA) 3D object segmentors, approaches designed specifically for part segmentation, as well as meta-learning and few-shot learning [40] baselines adapted for the task of 3D parsing. Analogical Networks outperform the baselines in the standard many-shot train-test split and particularly shine over the baselines in the few-shot setting: simply by expanding the memory repository with encodings of a few exemplars, and even without any weight updates, Analogical Networks segment novel instances much better than the baselines. We extensively ablate the design choices of our model to quantify the contributions of the modulation architecture, the pre-training scheme, and the memory retrieval. In summary, our contributions are as follows:

1. A mechanism for in-context memory-modulated learning. We introduce a model that, given a set of relevant examples, predicts analogical part structures in the present input scene. The memory context is a natural way to communicate the

task to the model. In this work, a task refers to parsing objects of different categories or segmentation granularity. Analogical Networks cast every task as an analogical correspondence problem to an appropriate set of memories and their labels. This allows us to train **one model across all tasks**, where each inference is modulated with a different set of relevant memories, of appropriate category and segmentation granularity. In contrast, existing neural network models that are trained to map input to output [5, 13, 41, 50, 52, 56, 57] break the general task of point cloud segmentation into sub-tasks and train different networks per object category (e.g., chair, table, etc.) and part label granularity (levels 1, 2, 3 in the PartNet benchmark [33]).

2. Few-shot transfer via analogical prediction. In parametric learning by gradient descent, any change to neural parameters impacts all aspects of the agent's future behavior, leading to catastrophic forgetting [23]. Analogical Networks learn from a few examples through memory expansion without needing to update the weights of the networks, which resolves the interference problems of parametric models [15].

3. Emergent correspondences without semantic supervision. Correspondences emerge across exemplars through the proposed modulation mechanism that uses memory part features as expert queries during part detection in place of learnable scene-agnostic queries used in SOTA transformer visual detectors [5].

1.3 Outline

The thesis is organized as follows: Section 2 discusses the related works in the field of few-shot learning, memory augmented neural networks, 3D instance segmentation, and Neural-symbolic models. Section 3 describes the proposed Analogical Network along with its modules in detail. Section 4 reports in detail about datasets, evaluation setups, and experiments about the Analogical Networks. This Section also explores the ablation experiments for the retriever module of the Analogical Network and shows qualitative results for the scene parsing. Section A describes the implementation and hyper-parameter details used in the experiments, along with additional scene parsing qualitative results for different baselines.

Chapter 2

Related Works

2.1 Few-shot Prediction: Meta-learning and Learning Associations

A key goal for our approach is to enable accurate inference in few-shot settings. Previous approaches [2, 14, 27, 29, 34, 37, 40, 43, 48, 49, 53] that target similar settings can be broadly categorized as relying on either meta-learning or learning better associations. Meta-learning approaches tackle few-shot prediction by learning a generic model that can be efficiently adapted to a new task of interest from a few labelled examples. While broadly applicable, these methods result in catastrophic forgetting of the original task during adaptation and thus require training a new model for each task of interest. Moreover, the goal of learning generic and rapidly adaptable models can lead to sub-optimal performance over the base tasks with abundant data. An alternative approach for the few-shot setting is to learn better associations. For example, the category of a new example maybe inferred by transferring the label(s) from the one (or few) closest samples [40, 42]. While this approach obviates the need for adapting models and can allow prediction in few-shot and many-shot settings, the current approaches are only applicable to global prediction, e.g., image labels. Our work can be viewed as extending such association-based methods to allow predicting fine-grained and generic visual structures using our proposed modulation-based prediction mechanism.

2.2 Memory-Augmented Neural Networks

Santoro *et al.* [38] are equipped with explicit external key-value memories, which they can attend on and retrieve key-value pairs relevant to the input. Memory-augmentation of parametric models permits **fast learning** with few examples, where the data are saved in the explicit memory immediately after their acquisition. Whereas learning via parameter update is slow and requires multiple gradient iterations on de-correlated examples. External memories have recently been used in scaling up language models [4, 24], to alleviate the limited context window of parametric transformers [51], to store knowledge in the form of entity mentions [12], knowledge graphs [11], and question-answer pairs [7]. Memory attention layers in these models are used to influence the computation of transformer layers and have proven critical for factual question-answering and sentence completion over their parametric counterparts.

2.3 3D Instance Segmentation

Instance segmentation in 3D has been traditionally approached as a clustering problem [8, 39]. Point-based methods learn either translation vectors mapping every point to its instance's center [6, 20, 45] or similarities across points [46, 58], followed by one or more stages of clustering. Similarly, [21, 47] oversegment the point cloud into small regions and then merge them into parts. Yu et al. [57] recursively decompose a point cloud into segments of finer resolution. [33, 41] learn representative vectors that form clusters by voting for each point. However, these approaches usually assume a fixed label space and need to train a separate model for each sub-task. In contrast, we employ Detection Transformers [5] for instance segmentation by repurposing the query vectors to act as representative vectors. We extend this set of queries with memory-initialized queries, enabling in-context reasoning. This allows us to train one model across all categories. As our results show, in absence of such memory contextualization, training one model across multiple categories gives inferior results (Table 4.8).

2.4 Neural-Symbolic Models

Analogical Networks represent knowledge explicitly, in terms of structured visual memories, where each one is a graph of part-entity neural embeddings. A structured visual memory can be considered the neural equivalent of a FRAME introduced in [32], "a graph of nodes and their relations for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party", to quote Minsky [32]. FRAME nodes would operate as "slots" to be filled with specific entities, or symbols, in the visual scene. Symbol detection would be carried out by a separate state estimation process such as general-purpose object detectors [59], employed also by recent neuro-symbolic models [30, 54, 55]. In-the-wild detection of symbols (e.g., chair handles, faucet tips, fridge doors) typically fails, which is precisely the reason why these earlier symbolic systems of knowledge, largely disconnected from the sensory input, have not been widely adopted. Analogical Networks take a step towards addressing these shortcomings by including symbol detection as part of the FRAME inference itself, through a top-down modulation that uses the context represented in the FRAME memory graph, to jointly search for multiple entities and localize them in context of one another.

Chapter 3

Approach: Analogical Networks for 3D object parsing

The architecture of Analogical Networks for 3D part segmentation is illustrated in Figure 3.1. It is comprised of two encoders, a retriever and a modulator, as well as a collection of labeled 3D object point clouds of different categories and segmentation granularity as the memory set. The same object with finer or coarser segmentation labeling is represented in different memories.

In a nutshell, Analogical Networks work as follows: First, an encoder encodes the input point cloud and memory point clouds into 1D vectors, and the retriever retrieves the top-k closest memories to the input. A second encoder encodes each retrieved memory into a set of part embeddings by pooling features inside each part, and encodes the input scene into point features. The retrieved memory part embeddings initialize queries, tasked to decode the parts of the input scene. The modulator uses multiple iterations of attention operations to jointly contextualize the input point features and the memory part queries. Queries having high confidence decode the "analogous" parts in the input 3D point cloud. When a memory part query is used to decode an analogous part in the input point cloud, we say that the two parts are put in correspondence by the model. The colors in columns 2, 3 and 5, 6 of Figure 1.1 denote part correspondence between the retrieved memory parts and the input point

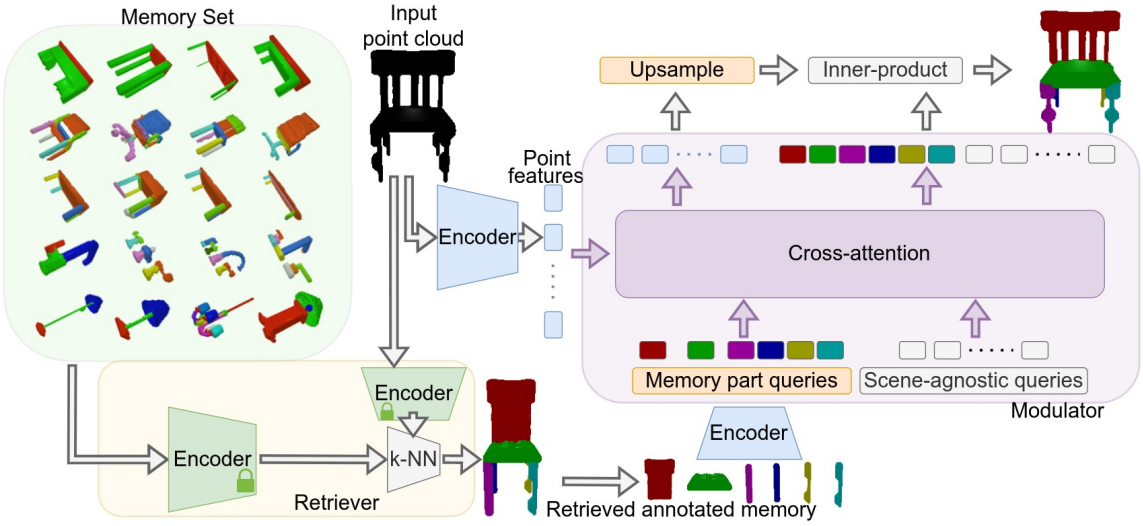

Figure 3.1: **Architecture for Analogical Networks.** Analogical Networks are comprised of an encoder, retriever, and modulator sub-modules. Labeled memories and the (unlabelled) input point cloud are first separately encoded into feature embeddings and the top-k most similar memories to the present input are retrieved. Here for clarity of presentation, we show the case for $k = 1$. Each retrieved memory part embedding initializes a query that is akin to a slot to be "filled" with the analogous part entity in the present scene. These queries are appended to a set of learnable scene-agnostic queries. The modulator contextualizes the queries with the input point cloud through iterative self and cross-attention operations that also update the point features of the input. When a memory part query decodes a part in the input point cloud, we say the two parts are put into analogical correspondence by the model. We color them with the same color to visually indicate this correspondence.

cloud. Note that more than one memory can be retrieved and used, allowing us to mix parts of different memories to explain the current input, as shown in Figure 3 right.

3.1 Encoders

Our model has two encoders, one used for retrieval (green in Figure 3.1) and one for modulation (blue in Figure 3.1). Both encoders have a PointNet++ backbone [35].

The encoder used in retrieval first extracts features for the input point cloud and each memory point cloud and summarizes each memory and input feature cloud

into a 1D vector using average pooling. This encoder is trained with within-instance correspondence pre-training, as explained later.

The encoder used in modulation encodes the input scene S and each retrieved memory scene M to a set of 3D point features. 3D positional encodings are then added to the 3D point features. Each labeled part p of M is then encoded into a 1D feature vector f_p^M by average pooling of its point features. This encoder is trained end-to-end with the modulator, i.e., gradients of the part segmentation objectives back-propagate to the encoder.

3.2 Retriever

Given the memory and input normalized 1D encodings, the top-k memories are retrieved by computing an inner product between the input point cloud feature and memory features.

3.3 Modulator

The modulator sub-network resembles the decoder of a detection transformer (DETR) [5] with two key differences: First, instead of only using a set of scene-agnostic learnable query vectors (64 in our case), it also considers the retrieved memory part features f_p^M (where $M = 1..k$) as queries, i.e., candidates for decoding parts in the input point cloud scene. Second, during the iterative self and cross-attention layers, both the queries and the point features of the input scene are updated, while in DETR, only the queries are updated. We consider 6 layers of self and cross-attention. More details and pseudocode are provided in the appendix (A.0.1).

Lastly, we upsample the point features to the original resolution using convolutional layers [35] and compute an inner product between each query and point feature to compute the segmentation mask for each query. The output of the modulator is a set of N_q segmentation mask proposals and corresponding confidence scores, where N_q is the total number of queries. At training time, these proposals are matched to ground-truth instance binary masks using the Hungarian matching algorithm [5]. For the proposals that are matched to a ground-truth instance, we compute the

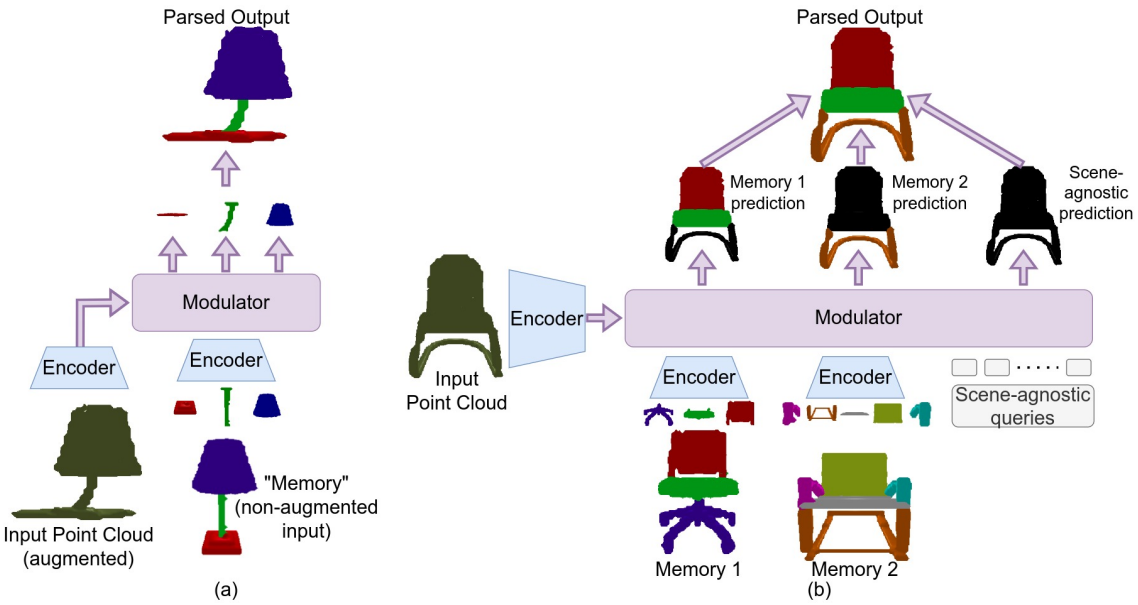

Figure 3.2: *Left:***Within-instance correspondence pre-training.** *Right:* **Multi-memory Analogical Networks.**

segmentation loss, which is a per-point binary cross-entropy loss [9, 45]. We also supervise the confidence score of each query, similar to [5]. The target labels are 1 for the proposals matched with a ground-truth part and 0 for non-matched ones. We found it beneficial to apply these losses after every cross-attention layer in the modulator. At test time, we multiply the per point mask occupancy probability with tiled confidence scores to get a $N_p \times N_q$ tensor (N_p is the number of points); then each point is assigned to the highest scoring query by computing inner product and taking per-point argmax over the queries.

3.4 Within-Instance Correspondence Pre-Training

In Analogical Networks, the retrieval process is not end-to-end differentiable with respect to the downstream scene parsing task. This is because i) we have no ground-truth annotations for retrieval and ii) the retrieved memories are contextualized with the input through dense cross-attention operations, instead of bi-encoder fusion [31]; the former, although a stronger performer than bi-encoder models, is costly to compute for the whole memory set. Thus, our solution is a hard top-k selection of

memories that is not differentiable with respect to the memory feature vectors used by the retriever, as also explained in [7].

We devise a novel pre-training scheme where we pre-train the encoder and modulator parameters independently of the retriever, by augmenting (rotating and deforming [25]) each training 3D point cloud and using the original point cloud as the modulating memory, as shown in the left side of Figure 3.2. In this special case, we actually have ground-truth information regarding the part association between the original and the augmented point cloud. We thus supervise each memory part query to decode the corresponding deformed part in the input cloud and do not use Hungarian matching. We do not have such part association ground truth during regular (cross-instance) training.

We describe the algorithm for within-instance correspondence pre-training of Analogical networks at 1.

3.5 Cross-Instance Training

After within-instance pre-training (Algorithm 1), we maintain two copies of the encoder weights: one copy acts as the encoder of the retriever and is frozen. The other copy is used as the encoder for modulation and is further trained cross-instance, i.e., with the modulating memory being sampled from the top-k retrieved memories. During cross-instance training (Algorithm 2), we further co-train with within-instance training data. We found that in this second training stage, using the part association supervision in within-instance examples does not further contribute to performance, so we use our standard losses and Hungarian matching.

We describe the algorithm for cross-instance correspondence pre-training of Analogical networks at 2.

Chapter 4

Experiments

4.1 Datasets and Evaluation Setup

We test Analogical Networks on PartNet [33], an established benchmark for 3D object segmentation. PartNet contains multiple 3D object instances from 24 object categories, each segmented into three different levels of granularity. We also train and test Analogical Networks on ScanObjectNN [44], which contains noisy and incomplete real-world point clouds. ScanObjectNN contains real world object instances from 15 categories derived from scene datasets (SceneNN [16] and ScanNet [10]). We consider two experimental paradigms:

Many-shot: We split the exemplars of the base categories into train and test sets. The model is tested on segmenting instances of the same categories in the test set. Our base categories for PartNet are Chair, Display, Storage Furniture, Bottle, Clock, Door, Earphone, Faucet, Knife, Lamp, Trash Can, and Vase. For ScanObjectNN dataset, base categories are Bag, Bin, Box, Cabinet, Chair, Desk, Door, Pillow, Shelf, Sink, and Sofa.

Few-shot: K examples of novel categories are given, and the model is tasked to segment new examples of these categories. This tests few-shot adaptation. We consider $K = 1$ and $K = 5$ setups. Our novel categories for Partnet dataset are Table, Bed, Dishwasher and Refrigerator. For ScanObjectNN dataset, novel categories are Table, Bed, Display, and Toilet.

Our experiments aim to answer the following questions:

Q1: How do Analogical Networks compare to parametric SOTA neural networks and meta-learning networks for 3D object parsing in the many-shot and few-shot paradigms? (Section 4.2.1)

Q2: How much does weight updating (fine-tuning) over memory expansion help few-shot adaptation in Analogical Networks? (Section 4.2.1)

Q3: How much within-instance correspondence pre-training helps Analogical Networks?

Q4: How does the proposed query-centric modulation compare against more conventional memory-augmented neural networks [51] where memories are attended to but not used directly as decoding slot queries? (Section 4.2.1)

Q5: How well do Analogical Networks learn part-based associations across exemplars without part association or part semantic label supervision? (Section 4.3)

Q6: What is the effect of using different retrieval mechanisms (Section 4.4.1, 4.4.2) and the importance of training using multiple classes (Section 4.5)?

4.2 Instance Segmentation without Semantic labels

Evaluation metrics We use the Adjusted Random Index (ARI) as our segmentation quality metric [36], which is a label-agnostic clustering score ranging from -1 (worst) to 1 (best). ARI calculates the similarity between two point clusters while being invariant to the order of the cluster centers. We compute $100\times$ ARI using the publicly available implementation of [22].

Baselines We compare our model to the following models in the literature: PartNet [33] is an instance segmentation model with the same input encoder as our model. *3D-DETR* is a 3D detection transformer network with the same backbone as our model and similar segmentation prediction head and loss, but without any memory retrieval, akin to the 3D equivalent of a state-of-the-art 2D image segmentor [1, 5]. Note that point features are updated during the decoder layers, the same as in our model, which we found to help performance. *Prototypical Networks* is an adaptation of the episodic prototypical networks for image classification of [40] to the task of

3D object part segmentation. Specifically, given a set of N labeled point clouds, we form average feature vectors for each semantic labeled part and use them as queries to segment points into corresponding part masks.

Contrary to 3D-DETR, Analogical Networks attend on external memories. Contrary to Prototypical Networks, our model considers feature contextualization between memory prototypes and the input scene.

We further consider the following ablated versions of our model; each one aims to evaluate a specific aspect of its design: *Analogical Networks w/o memory part queries* is a model similar to the parametric 3D-DETR baseline augmented with attention to retrieved memories [51]. Instead of using the part memory features as queries, this baseline updates the scene-agnostic query vectors by iteratively attending to the input feature cloud and the memory part features. Different from Analogical Networks, correspondence cannot emerge between a memory and the input scene, since the model does not encode such correspondence explicitly, but only implicitly, in the attention operations. *Analogical Networks single memory* is Analogical Networks with a single modulating memory, in contrast to *Analogical Networks multi-memory* that uses five memories. *Analogical Networks w/o pretrain* is our model without the self-supervised within-instance correspondence objective for training of the modulator and encoder weights.

4.2.1 Many and few-shot 3D object part segmentation

The PartNet benchmark supplies three levels of segmentation per object instance, where level 3 is the most fine-grained. We train and test our model and baselines on all three levels of segmentation. We use a learnable level embedding concatenated to the input for our baselines PartNet and 3D-DETR, as is usually the case in multi-task models [19]. In the many-shot setting, all examples in the training set become part of our model's memory. The retriever has access to the object category. In the few-shot setting, Analogical Networks adapt in two ways i) by expanding the memory of the model with the novel K-shot support examples, and ii) by further adapting the weights via fine-tuning to parse the K examples. In this case, the memory set is only the novel labeled support set instances.

We show quantitative results in Table 4.1. Our conclusions are as follows: **(i)**

Method	Fine-tuned?	Novel Categories		Base Categories
		1-shot ARI (↑)	5-shot ARI (↑)	Many shot ARI (↑)
PartNet	✗	26.0 ± 0.45	26.0 ± 0.45	56.7
	✓	19.4 ± 1.57	29.3 ± 0.68	-
3D-DETR	✗	29.2 ± 0.82	29.2 ± 0.82	70.5
	✓	34.3 ± 2.17	48.6 ± 2.46	-
Analogical Networks w/o memory part queries	✗	48.3 ± 2.07	48.9 ± 1.23	71.3
	✓	51.2 ± 2.64	56.6 ± 2.23	-
Analogical Networks single memory w/o pretrain	✗	36.8 ± 0.92	36.3 ± 1.60	71.8
	✓	46.1 ± 3.61	51.3 ± 1.32	-
Analogical Networks single memory (ours)	✗	53.0 ± 1.26	54.2 ± 1.36	**71.9**
	✓	53.8 ± 1.49	56.3 ± 1.59	-
Analogical Networks multi-memory (ours)	✗	$\mathbf{55.8 \pm 1.35}$	56.4 ± 1.30	70.6
	✓	-	$\mathbf{58.4 \pm 1.52}$	-

Table 4.1: **Results on few-shot and many-shot 3D object segmentation on PartNet** [33]. Our few-shot experiments use four held-out (novel) categories. We report mean and standard deviation for few-shot ARI performance over 10 tasks (each task consists of a different subset of the K-shot support set). **Without any fine-tuning, Analogical Networks outperform 3D-DETR by 26% in the few-shot setup.** Though weight fine-tuning helps both models, it brings a performance boost of 18% for 3D-DETR and only 2% for our model. This means Analogical Networks can generalize and adapt few-shot to new categories by expanding their memory without any weight interference. Even upon fine-tuning, Analogical Networks outperform 3D-DETR by 10% ARI.

Analogical Networks dramatically outperform parametric DETR segmentors in few-shot learning (Q1). While in the base category test set, the two models have similar performance (70.5 versus 70.6), when adapting few-shot to the novel categories, Analogical Networks dramatically outperforms 3D-DETR (48.6 versus 58.4 in 5-shot ARI). **(ii) Analogical networks can adapt few-shot to novel categories without weight updates, simply by memory expansion** (Q2). Indeed, the 5-shot performance of our multi-memory model is very close before and after fine-tuning in the novel categories. **(iii) Multi-memory Analogical Networks outperform single memory Analogical Networks in few-shot adaptation**. **(iv) Within-instance correspondence pre-training helps Analogical Networks to adapt few-shot to the novel categories.** (Q3) **(v) Attending on retrieved memory part features and contextualizing them with the input, as done in Analogical Networks w/o memory part queries, helps over parametric segmentors** (Q4) in few-shot settings. Our interpretation of this interesting result is that attending on relevant memories permits a model to operate in context

and overcome challenges caused by imbalance of examples across categories. Rare instances are not forgotten as they are explicitly saved in the external memory and, upon retrieval and modulation with the input, can steer inference. In parametric-only networks, rare examples are in competition with examples in the mode of the training distribution for their representation in the parameters. Still, this memory-augmented model exhibits much worse performance than our complete model without weight fine-tuning (49% versus 56.4 % ARI).

4.2.2 ARI Performance of PartNet Base Categories

We provide category-specific many-shot ARI scores for our model and baselines on the training categories in Table 4.2. Note that all models in this table are trained on all classes and segmentation levels jointly.

Methods	Base Categories (ARI ↑)											
	Chair	Display	Storage Furniture	Bottle	Clock	Door	Ear phone	Faucet	Knife	Lamp	Trash Can	Vase
PartNet	72.8	80.5	47.1	64.4	37.2	33.0	60.4	60.8	76.3	66.8	26.9	54.4
3D-DETR	76.9	87.4	67.7	79.9	46.8	53.7	75.7	74.1	83.1	79.2	53.6	67.7
Analogical Networks w/o memory part queries	78.9	89.2	71.6	72.3	45.9	53.4	75.9	74.8	84.7	76.8	61.4	70.4
Analogical Networks single memory w/o pretrain	77.8	88.7	71.4	72.6	49.7	51.1	74.8	72.7	84.5	72.7	72.4	72.2
Analogical Networks single memory	79.1	89.1	73.5	69.4	50.7	52.7	74.9	75.9	84.2	78.3	64.4	69.5
Analogical Networks multi-memory	78.2	87.9	74.0	71.2	44.6	56.6	72.5	75.8	84.3	77.3	59.7	64.5

Table 4.2: Category specific ARI scores for base categories of PartNet dataset [33].

4.2.3 ARI Evaluation on ScanObjectNN Dataset [44]

We train and test Analogical Networks on ScanObjectNN [44], which contains noisy and incomplete real-world point clouds. Note that our model is at a disadvantage in this setup because of the inconsistent labeling of ScanObjectNN. For example, as we show in rows 5 and 6 of Figure 4.1, the legs of a chair may be annotated as a single part or multiple parts in the dataset. Although the PartNet dataset provides the "level" information, there is no such information in ScanObjectNN. We empirically find that the label space is more often closer to PartNet's level 1. As a result, the baselines (3D-DETR and PartNet) learn to always predict the most frequent label space. Analogical Networks need to retrieve an object of the same level label space as the expected, which happens at best. This means that Analogical Networks have

higher chances of not "guessing" the expected label space correctly, and thus, even if our predictions are plausible, they are penalized.

Despite this disadvantage, Analogical Networks maintain high performance on both base and novel categories and outperform the baselines PartNet and 3D-DETR . We show qualitative results in Figure 4.1 for base classes and in Figure 4.2 for novel classes. We can see that our predictions are always plausible and consistent with the retrieved memory, even if the expected label space is different.

Method	Fine-tuned?	Novel Categories	Base Categories
		5-shot ARI (↑)	Many shot ARI (↑)
PartNet	✗	28.9 ±0.11	42.6
	✓	37.4 ±2.90	-
3D-DETR	✗	49.1 ± 0.15	58.9
	✓	56.9 ± 4.67	-
Analogical Networks single memory (ours)	✗	**51.3** ± 3.02	**60.1**
	✓	**59.6** ± 4.95	-

Table 4.3: **Results on few-shot and many-shot 3D object segmentation on ScanObjectNN [44].** Our few-shot experiments use four held-out (novel) categories: Table, Bed, Display, Toilet. We use 11 Base categories: Bag, Bin, Box, Cabinet, Chair, Desk, Door, Pillow, Shelf, Sink and Sofa. We report mean and standard deviation for few-shot ARI performance over 10 tasks (each task consists of a different subset of the 5-shot support set).

4.3 Evaluation of Emergent Part Correspondences

In this section, we evaluate the quality of cross-exemplar part correspondences that emerge in Analogical Networks . Our model is trained solely for the segmentation of 3D objects, and no semantic part labels are used during training. The only cross-part association (correspondence) supervision is during pre-training, where the parts in a 3D point cloud are used as queries to decode the corresponding part of the augmented 3D point cloud. Yet, through the use of memory part queries shown in Figure 3.1, our model can "label" input scenes with the labels of the corresponding memory parts. If the parts of visual memories are semantically labeled, those semantic labels will propagate to the input scene. Parts decoded from scene-agnostic queries are not put in correspondence with any memory in our model. We found 80% of 3D points to be

on parts decoded by memory part queries on average.

Here we evaluate the quality of label propagation through emergent correspondences by evaluating few-shot part instance segmentation and semantic segmentation. We compare against *3D-DETR semantic*, which is our 3D-DETR baseline additionally supervised for predicting semantic part labels. Similar to our model, *Prototypical Networks* propagate semantic labels of the prototypical part features. We evaluate Prototypical Networks only for semantic segmentation since it cannot easily produce instance segments: if multiple part instances belong to the same label, this model assumes they belong to the same semantic prototype. We also train Analogical Networks semantic, a version of our model where the scene-agnostic queries are explicitly supervised for semantic part labels.

Evaluation metrics We use mean Panoptic Quality (PQ) [26] for part instance segmentation. We use mean intersection over union (mIoU) for 3D point semantic segmentation.

We show quantitative semantic and instance segmentation results in Table 4.4. Analogical Networks show very high semantic and instance segmentation accuracy via label propagation through memory part queries, **without any supervision (Q5).** We report two sets of numbers for our model. The first penalizes points that belong on parts decoded by scene agnostic queries and assumes they have wrong labels, the second only evaluates segmentation scores on the 3D points on parts decoded by part memory queries.

4.3.1 Evaluation on the Labeled Instance Segmentation Setup

We compare our Analogical Networks semantic model with baselines such as PartNet [33], SGPN [46], PE [58] and Semantic Segmentation-Assisted Instance Feature Fusion (SAIF) [41] for AP_{50} metric in Table 4.5. We observe that our model Analogical Networks semantic outperforms the existing approach by a large margin. Note that our model is at a disadvantage because it is tested without fine-tuning on novel classes (Bed, Dishwasher, Fridge, and Table) and as a result has never seen any examples of these classes at training time, apart from in-context usage of

Method	Novel Category 1-shot		Novel Category 5-shot		Base Category Many-shot	
	mIoU	PQ	mIoU	PQ	mIoU	PQ
3D-DETR semantic	0.33	0.21	0.40	0.30	0.68	0.63
Prototypical Networks	0.39	-	0.40	-	0.42	-
Analogical Networks (ours)	0.51/0.52*	0.43/0.44*	0.56/0.58*	0.49/0.52*	0.69/0.71*	0.65/0.68*
Analogical Networks semantic (ours)	0.51	0.44	0.58	0.51	0.72	0.67

Table 4.4: **Part Semantic and Instance Segmentation performance on few-shot and many-shot 3D object segmentation on PartNet dataset [33].** Few-shot performance is calculated by averaging over the 10 different $K-$shot tasks (each task consists of a different support set, we observed std of ~ 0.02 for all the reported values). * performance is calculated on the subset of input 3D points that were classified by memory part queries (and not by the scene agnostic queries since, in the latter case, semantic labels cannot be propagated.)

labeled memories at test time. In contrast, all other approaches in Table 4.5 use all available training data for each class. Still, we can see that it largely outperforms other methods in three out of four novel classes.

4.4 Retrieval Ablations

4.4.1 Performance under Varying Retrieval Schemes

In the few-shot setting, we evaluate the performance of Analogical Networks under varying memory retrieval schemes in Table 4.6. We compare against a hypothetical *oracle* retriever that can fetch the most helpful (in terms of resulting ARI) memory for each input. Our conclusions are as follows: **(i) Analogical Networks with an oracle memory retriever perform better than Analogical Networks using memories retrieved by the retriever**. This suggests that better training of our retriever or exploring its co-training with the rest of our model could have a significant impact on improving its performance. **(ii) Considering any of the 5-shot exemplars randomly does slightly worse than using memories retrieved by our retriever. (iii) Chamfer distance, (which measures the squared distance of each point in the given point cloud with its nearest neighbor in the other point cloud), performs worse than distance metrics applied on embeddings (cosine, L1). (iv) Using cosine or L1 distance metric results**

	Level	Avg	Bed	Bottle	Chair	Clock	Dish	Disp	Door	Ear	Faucet	Knife	Lamp	Fridge	Stora	Table	Trash	Vase
SGPN	Coarse	52.9	29.8	61.9	72.4	20.3	72.2	89.3	49.0	57.8	63.2	63.2	32.7	50.6	32.9	49.2	56.8	46.6
	Middle	27.1	15.4	-	25.4	-	58.1	-	25.4	-	-	-	21.7	22.1	30.5	18.9	-	-
	Fine	29.3	11.8	45.1	19.4	18.2	38.3	78.8	15.4	35.9	37.8	38.3	14.4	18.2	21.5	14.6	24.9	36.5
	Avg	40.0	19.0	53.5	39.1	19.3	56.2	84.1	29.9	46.9	50.5	50.8	22.9	30.3	28.3	27.6	40.9	41.6
PartNet	Coarse	59.0	48.4	63.6	74.4	42.8	76.3	93.3	52.9	57.7	69.6	58.4	37.2	50.0	45.2	54.2	71.7	49.8
	Middle	36.1	23.0	-	35.5	-	62.8	-	39.7	-	-	-	26.9	35.2	35.0	31.0	-	-
	Fine	36.7	15.0	48.6	29.0	32.3	53.3	80.1	17.2	39.4	44.7	45.8	18.7	26.5	27.5	23.9	33.7	52.0
	Avg	47.1	28.8	56.1	46.3	37.6	64.1	86.7	36.6	48.6	57.2	52.1	27.6	37.2	35.9	36.4	52.7	50.9
PE	Coarse	60.9	51.4	63.1	77.1	41.1	76.9	95.3	61.2	66.5	73.1	76.5	37.1	50.5	47.3	40.3	69.0	48.7
	Middle	39.0	31.0	-	38.6	-	64.2	-	36.9	-	-	-	31.0	37.3	42.0	31.5	-	-
	Fine	39.9	26.2	50.7	34.7	30.2	50.0	82.0	25.7	43.2	55.6	44.4	20.3	31.1	34.2	25.5	37.7	47.6
	Avg	49.7	36.2	56.9	50.1	35.6	63.7	88.7	41.3	54.9	64.4	60.5	29.5	39.6	41.2	32.4	53.4	48.1
SAIF	Coarse	67.4	**54.1**	66.9	84.1	51.2	79.9	97.2	76.8	71.6	79.2	67.8	38.2	57.4	56.4	65.3	79.7	53.8
	Middle	48.1	45.5	-	45.7	-	73.2	-	52.0	-	-	-	30.9	48.2	53.3	36.2	-	-
	Fine	47.7	40.9	55.9	38.2	37.1	**56.5**	87.4	41.3	53.7	59.1	48.8	21.7	44.1	44.0	28.9	51.3	54.6
	Avg	57.0	**46.8**	61.4	56.0	44.2	69.9	92.3	56.7	62.7	69.2	58.3	30.3	49.9	51.2	43.5	65.5	54.2
ours	Coarse	**82.0**	47.9	**87.6**	**95.7**	**78.0**	**86.3**	**97.6**	**80.9**	**85.5**	**85.0**	**75.3**	**71.8**	**85.5**	**72.0**	**88.3**	**85.7**	**89.4**
	Middle	**63.6**	**46.4**	-	**72.1**	-	**76.1**	-	**63.2**	-	-	-	**66.3**	**56.8**	**70.4**	**57.5**	-	-
	Fine	**63.7**	**43.8**	**69.0**	**68.6**	**52.1**	55.2	**92.6**	**56.3**	**59.2**	**73.2**	**62.0**	**55.8**	**43.0**	**68.3**	**54.2**	**79.0**	**87.7**
	Avg	**72.2**	46.0	**78.3**	**78.8**	**65.0**	**72.5**	**95.1**	**66.8**	**72.3**	**79.1**	**68.6**	**64.6**	**61.7**	**70.2**	**66.6**	**82.3**	**88.5**

Table 4.5: Part instance segmentation AP_{50} performance of the test set on PartNet [33]. We report performance across 3 level of segmentation levels for Analogical Networks semantic (ours) and compare with baselines PartNet [33], SGPN [46], PE [58] and SAIF [41] that train a separate model for each category. We report the results for other baselines as mentioned in SAIF [41].

in similar ARI performance. This is probably because we do not optimize for the retrieval task explicitly. Thus the feature space is not regularized for a specific distance metric. This however could be a more important choice if we train the retriever end-to-end with the modulator.

4.4.2 Qualitative Performance of the Retriever

We show qualitative results of the retriever on multiple classes, both seen (Figure 4.4) during training and unseen (Figure 4.5). We observe the following: **(i) The retriever considers fine-grained object similarities and not only class information.** To illustrate this, we include two examples for the "Chair" and "Earphone" classes in Figure 4.4, as well as the "Bed" and "Refrigerator" classes in Figure 4.5. Different instances of the same category retrieve very different memories that share both structural and semantic similarities with the respective input point cloud. **(ii) The**

Method	Fine-tuned?	Modulating Memory	Novel Category: 5-shot ARI (↑)
Analogical Networks single memory (ours)	✗	Random	52.2 ± 0.78
		Retriever (cosine)	54.2 ± 1.36
		Retriever (L1)	54.4 ± 0.98
		Retriever (Chamfer)	52.7 ± 0.73
		Oracle	61.9 ± 1.22
Analogical Networks single memory (ours)	✓	Random	53.5 ± 1.28
		Retriever (cosine)	56.3 ± 1.59
		Retriever (L1)	55.5 ± 1.10
		Retriever (Chamfer)	54.0 ± 0.62
		Oracle	62.3 ± 0.66

Table 4.6: **Ablations on ARI segmentation performance under varying retrieval schemes for 5-shot on 4 novel categories**.

retriever generalizes to novel classes, not seen during training, as shown in Figure 4.5.

For ScanObjectNN dataset, we visualize top-4 retrieval results in Figure 4.3.

4.5 The Effect of Training on Multiple Classes

One advantage of Analogical Networks is that it can adapt in different contexts simply by retrieving memories of different object classes or segmentation granularity. This allows our model to i) generalize few-shot to novel classes **without fine-tuning** (i.e., without updating model parameters) and ii) improve many-shot performance across all classes using a **single model**. We quantitatively show this in Table 4.8, where we compare against the parametric models PartNet [33] and 3D-DETR . We train each model under three setups: i) only on "Chair", which is the most common category (see Table 4.7, approximately half of our training examples fall into this category), ii) only on "Faucet", which is a class with relatively few examples (Table 4.7), iii) on all classes. We test the performance of each variant on the two selected classes as well as the four novel classes we use throughout our paper.

We can observe the following patterns: **(i) Models trained on a single category usually fail to generalize to other classes** even after fine-tuning, despite their strong performance on the training class. **(ii) Analogical Networks trained on all classes outperform category-specific models on all tested setups. This**

Base Categories												Novel Categories			
Chair	Display	Storage Furniture	Bottle	Clock	Door	Ear phone	Faucet	Knife	Lamp	Trash Can	Vase	Table	Bed	Dishwasher	Refrigerator
6323	928	2269	436	554	225	228	648	327	2207	321	1076	8218	194	181	187

Table 4.7: **Number of samples per category in the PartNet dataset [33].** Note that each sample has annotations for three levels of segmentation granularity.

Method	Fine-tuned?	Novel Categories	Chair	Faucet
		5-shot ARI (↑)	ARI (↑)	ARI (↑)
PartNet trained on "Faucet"	✗	6.8 ± 0.20	17.1	69.4
	✓	23.6 ± 1.59	-	-
PartNet trained on "Chair"	✗	21.6 ± 0.14	75.1	24.1
	✓	24.1 ± 2.89	-	-
PartNet	✗	26.0 ± 0.45	72.8	60.9
	✓	29.3 ± 0.68	-	-
3D-DETR trained on "Faucet"	✗	14.7 ± 0.16	33.2	72.1
	✓	25.1 ± 2.3	-	-
3D-DETR trained on "Chair"	✗	26.2 ± 0.08	78.3	42.8
	✓	46.3 ± 2.85	-	-
3D-DETR	✗	29.2 ± 0.82	72.8	70.8
	✓	48.6 ± 2.46	-	-
Analogical Networks single memory (ours) trained on "Faucet"	✗	16.2 ± 0.74	37.8	71.2
	✓	35.7 ± 2.43	-	-
Analogical Networks single memory (ours) trained on "Chair"	✗	38.9 ± 1.84	78.4	41.8
	✓	54.9 ± 1.13	-	-
Analogical Networks single memory (ours)	✗	$\mathbf{54.2} \pm 1.36$	**79.5**	**75.7**
	✓	$\mathbf{56.3} \pm 1.59$	-	-

Table 4.8: **Comparison of single-category trained and multi-category trained models. Analogical Networks both do better within a category *and* generalize better when trained across all categories, while the baselines do better within a category if trained *only* with that category.** The baselines lack in-context learning, and specialization in a category fights generalization across categories. For Analogical Networks , specialization and generalization objectives align. They do better in both many-shot and few-shot when trained with diverse data.

is not true for the other baselines. Both PartNet and 3D-DETR are better when trained on a single class and tested on the same class. Analogical Networks generalizes better with more diverse data, as the model learns to use labeled memory as input context.

4.6 Limitations - Future directions

Below we present a set of future directions that are necessary for Analogical Networks to scale beyond the segmentation of single object scenes.

(i) The retriever in Analogical Networks currently operates over whole object memories, and it is not end-to-end differentiable with respect to the downstream task. Sub-object part-centric memory representations would permit fine-grained retrieval of visual memory scenes. We further plan to explore alternative supervision for the retriever module inspired by works in the language domain [17, 18]. **(ii)** Scaling Analogical Networks to segmentation of complete, multi-object 3D scenes in realistic home environments requires scaling up the size of memory collection. It would further necessitate bootstrapping fine-grained object part annotations, missing from 3D scene datasets [10], by transferring knowledge of object part compositions from PartNet. Such semi-supervised fine-grained scene parsing is an exciting avenue for future work. **(iii)** So far we have assumed the input to Analogical Networks to be a complete 3D point cloud. However, this is hardly ever the case in reality. Humans and machines need to make sense of single view, incomplete and noisy observations. Extending Analogical Networks with generative heads that detect not only analogous parts in the input but also in-paint or complete missing parts is a direct avenue for future work.

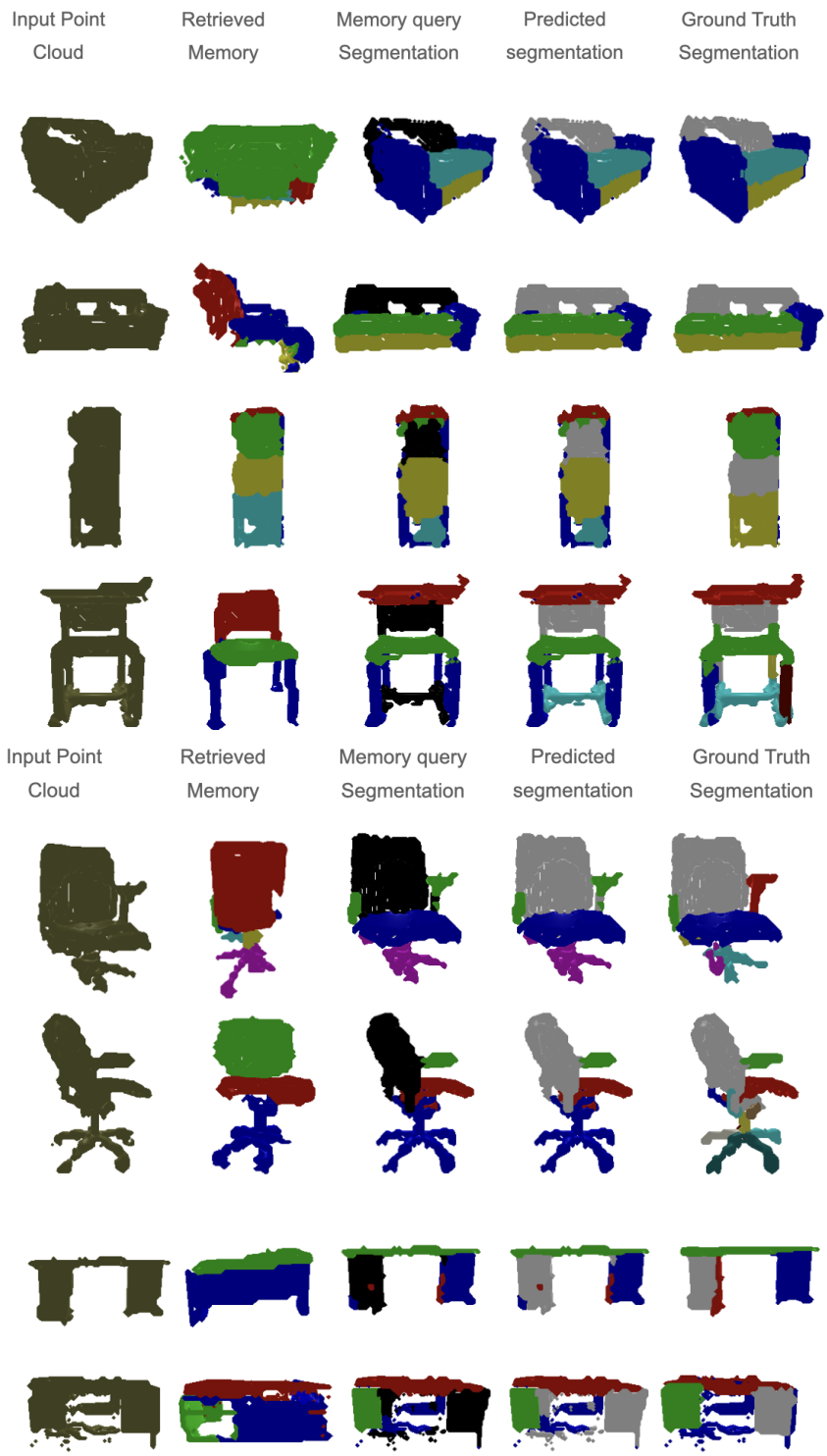

Figure 4.1: Results on base category samples from ScanObjectNN [44] using Analogical Networks.

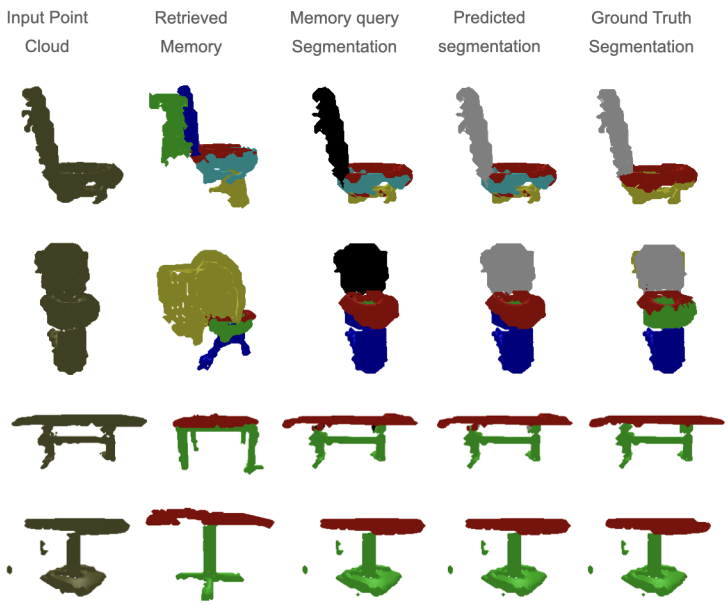

Figure 4.2: Results on novel category samples from ScanObjectNN [44] using Analogical Networks.

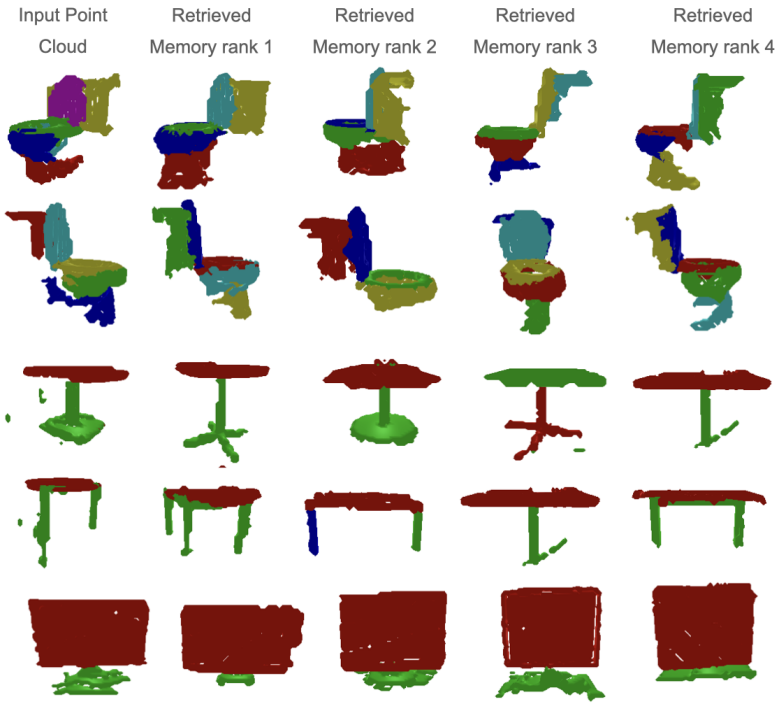

Figure 4.3: Top-4 retrieved results for the input point cloud from ScanObjectNN [44] dataset.

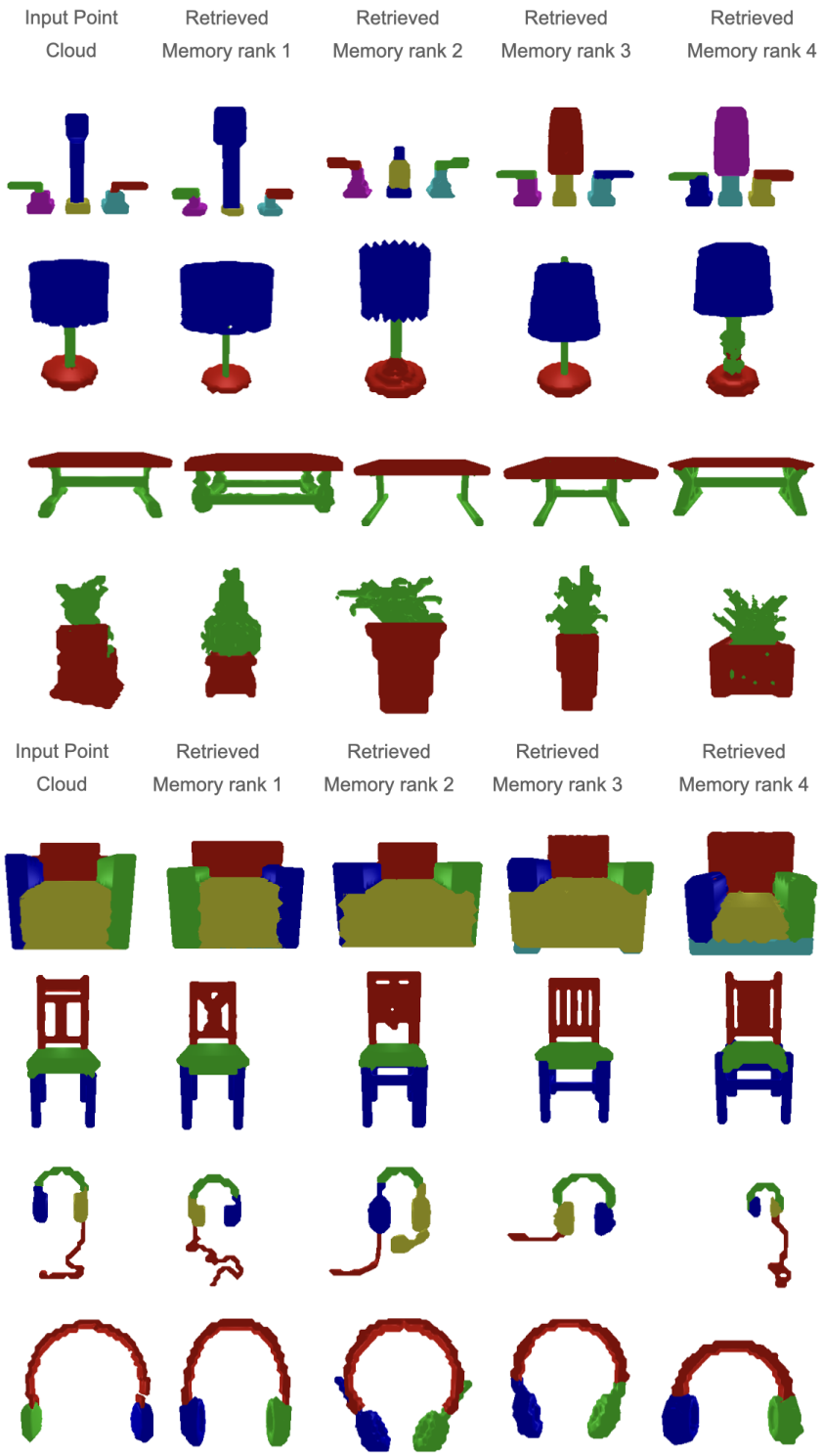

Figure 4.4: Top-4 retrieved results for each input point cloud. Examples from base classes of PartNet [33] dataset. Note that instances of the same category can retrieve different memories, focusing on structural similarity and not only semantics.

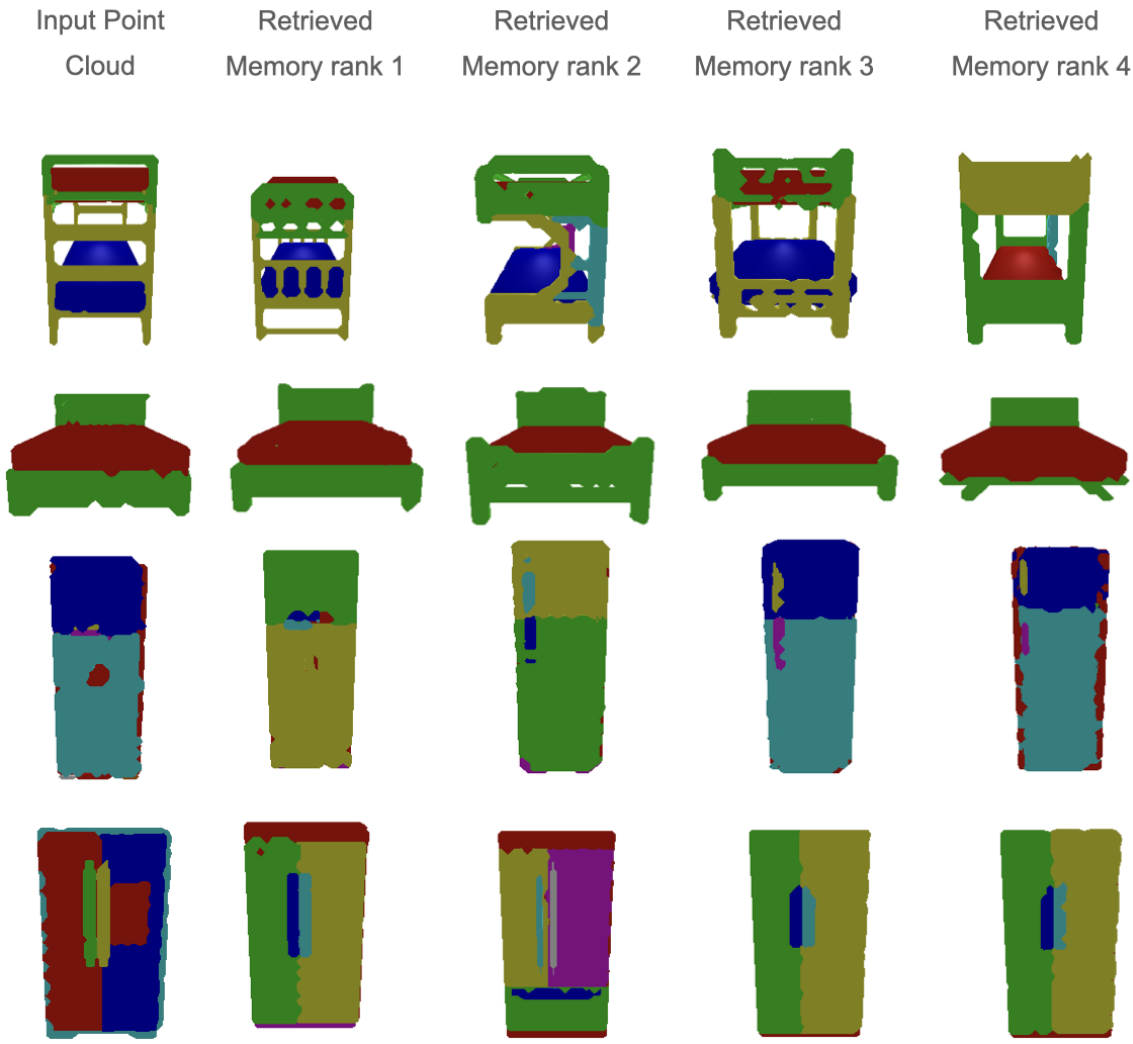

Figure 4.5: Top-4 retrieved results for each input point cloud. Examples from novel classes of PartNet [33] dataset. Note that instances of the same category can retrieve different memories, focusing on structural similarity and not only semantics. This behavior generalizes to novel classes as well, even if the model has never seen such geometries before.

Chapter 5

Conclusions

We presented Analogical Networks, a model for associative 3D visual parsing that encodes domain knowledge – (1) explicitly in a set of past labeled exemplars and (2) implicitly in model parameters. Analogical Networks does not follow the traditional approach of mapping visual scenes to semantic labels and instead puts forward an analogical in-context paradigm of linking input to past scenes and their parts labels. Given an input scene, the model retrieves a set of labeled memory point clouds and contextualizes those with the input through attention operations. The contextualized memory part features decode *corresponding* parts in the input point cloud. By framing visual parsing as analogical correspondence, Analogical Networks can be trained and share knowledge across multiple tasks. Here, each task is communicated via the features and labels of appropriate conditioning memories that act as the 'context' for prediction. One-shot, few-shot, or many-shot learning is treated uniformly by conditioning to the appropriate set of memories, whether taken from a single, few, or many memory exemplars and inferring analogous parses. We showed Analogical Networks outperform SOTA baselines in both many and few-shot parsing evaluation settings. We further showed part correspondences emerge across exemplars without supervision as a by-product of the analogical inductive bias.

Appendix A

Ablations

The Appendix is organized as follows: In Section A.0.1, we provide implementation details and the pseudo-code for training Analogical Networks. In section A.0.2, we provide extensive qualitative visual object parsing results for single and multi memory variants of Analogical Networks .

A.0.1 Implementation Details and Pseudo Code

For both stages of training (i.e. within-instance correspondence pre-training and cross-instance training), we use AdamW optimizer [28] with an initial learning rate of 1e−4 and batch size of 8. We train the model for 40 epochs within-instance and 25 cross-instance with a learning rate schedule decay of 0.5 at every 30^{th} epoch. For few-shot fine-tuning/evaluation, we use AdamW optimizer with an initial learning rate of 1e−5 and batch size of 4. We fine-tune it for 30 epochs and we report the performance across 10 different task runs (each task has a different set of K support samples). We describe Analogical Networks' training details in the following pseudo-code for within (Algorithm 1) and cross-instance training (Algorithm 2) respectively.

Algorithm 1: Pseudo code for within-instance correspondence pre-training of Analogical Networks

```
# S: input point cloud, M: memory point cloud, Np: numbers of points in S or M, N: sub-sampled points
     , C: number of feature channels, P: number of parts in M
# augment: a sequence of standard 3D point cloud augmentations
# pc_encoder: point cloud encoder
# Xp(M): ground-truth label assignment of points in parts, copied directly from the memory
# part_encoder: Computes the part features using mean pooling
# pos_encode: Adds positional encoding
# upsampler: Upsamples point cloud
# Segmentation_Loss: Cross entropy loss to assign each point to the matched query.

for S in dataloader: # load a batch with B samples
    M = S # the memory is the un-augmented version
    S = augment(S) # the input is augmented
    # S : B x Np x 3 and M: B x Np x 3
    # Compute point features
    F^S = pc_encoder(S) # B x N x C
    F^M = pc_encoder(M) # B x N x C

    # Initialize memory part queries
    f^M = part_encoder(F^M) # B x P x C

    # Add positional embedding
    x = pos_encode(F^S)
    y = pos_encode(f^M) # B x P x C

    Loss = 0
    # Do multiple layers of modulation using Self-Attn and Cross-Attn
    for layer in num_layers:
        x = Self-Attn(x) # B x N x C
        y = Self-Attn(y) # B x P x C
        x = Cross-Attn(x, y) # B x N x C
        y = Cross-Attn(y, x) # B x P x C

        X = upsampler(x) # B x Np x C
        point_query_similarity = matmul(normalize(X), normalize(y.T)) # B x Np x P

        Loss += Segmentation_Loss(argmax(point_query_similarity, -1), Xp(M))


    # optimizer step
    loss.backward()
    optimizer.step()
```

Algorithm 2: Pseudo code for cross-instance training of Analogical Networks

```
# S: input point cloud, M: retrieved memory point cloud, Np: numbers of points in S or M, N: sub-
    sampled points, C: number of feature channels, P: number of parts in M
# Q: number of learnable scene-agnostic queries
# pc_encoder: point cloud encoder
# Xp_Hungarian: Hungarian matched label assignment of points in S to queries
# yp_Hungarian: Hungarian matched label assignment of queries to GT parts in S
# part_encoder: Computes the part features using mean pooling
# pos_encode: Adds positional encoding
# upsampler: Upsamples point cloud
# Objectness_loss: Binary cross entropy loss to decide which queries (scene-agnostic+learnable) would
     be responsible for decoding parts
# Segmentation_Loss: Cross entropy loss to assign each point to the hungarian matched query.

for S,M in dataloader: # load a batch with B samples
    # S : B x Np x 3 and M: B x Np x 3
    # Compute point features
    F^S = pc_encoder(S) # B x N x C
    F^M = pc_encoder(M) # B x N x C

    # Initialize memory part queries
    f^M = part_encoder(F^M) # B x P x C

    # Add positional embedding
    x = pos_encode(F^S)
    y = Concatenate(pos_encode(f^M), scene_agnostic_queries) # B x (P + Q) x C

    Loss = 0
    # Do multiple layers of modulation using Self-Attn and Cross-Attn
    for layer in num_layers:
        x = Self-Attn(x) # B x N x C
        y = Self-Attn(y) # B x P x C
        x = Cross-Attn(x, y) # B x N x C
        y = Cross-Attn(y, x) # B x P x C

        X = upsampler(x) # B x Np x C
        point_query_similarity = matmul(normalize(X), normalize(y.T)) # B x Np x (P + Q)

        Loss += Segmentation_Loss(armax(point_query_similarity, -1), Xp_Hungarian) + Objectness_loss(y
            , yp_Hungarian)


    # optimizer step
    loss.backward()
    optimizer.step()
```

A.0.2 Qualitative Parsing Results for Single-Memory and Multi-Memory Analogical Networks

In this section, we show our model's qualitative results for object parsing. In the Figures A.1, A.2, A.3, A.4, A.5, A.6, 4.1, 4.2 we use the following 5-column pattern:

- Unlabelled input point cloud
- Memory used for modulation
- Object parsing generated using only the memory part queries (not the scene-agnostic queries). In this column, regions that are not decoded by a memory part query are colored in black.
- Final predicted segmentation parsing using both memory-initialized queries and scene-agnostic queries. Regions that are colored in black in the third column but colored differently in the fourth column are decoded by scene-agnostic queries.
- Input point cloud's ground truth segmentation at the granularity level of the memory.

We qualitatively show the emergence of part correspondence between retrieved memory (column 2) and the input point cloud parsed using memory queries (column 3). Parts having the same color in columns 2 and 3 demonstrate correspondence, i.e. a part in column 2 decodes the part with the same color in column 3. Analogical Networks promote correspondence of parts on both base (Figure A.1 bottom and A.2) and novel (Figure A.1 top and A.3) categories. This correspondence is semantic but also geometric, as can be seen in Figure A.5. When multiple memories are available, Analogical Networks mix and match parts of different memories to parse the input. Furthermore, we show parsing results for Analogical Networks single memory w/o pretrain in Figure A.6. We observe that all of memory part query are inactive in the parsing stage. This demonstrates the utility of within-instance pre-training, as without this pre-training part correspondence does not emerge, as shown in Figure A.6. Lastly, we show that Analogical Networks generalize to noisy and incomplete point clouds in ScanObjectNN.

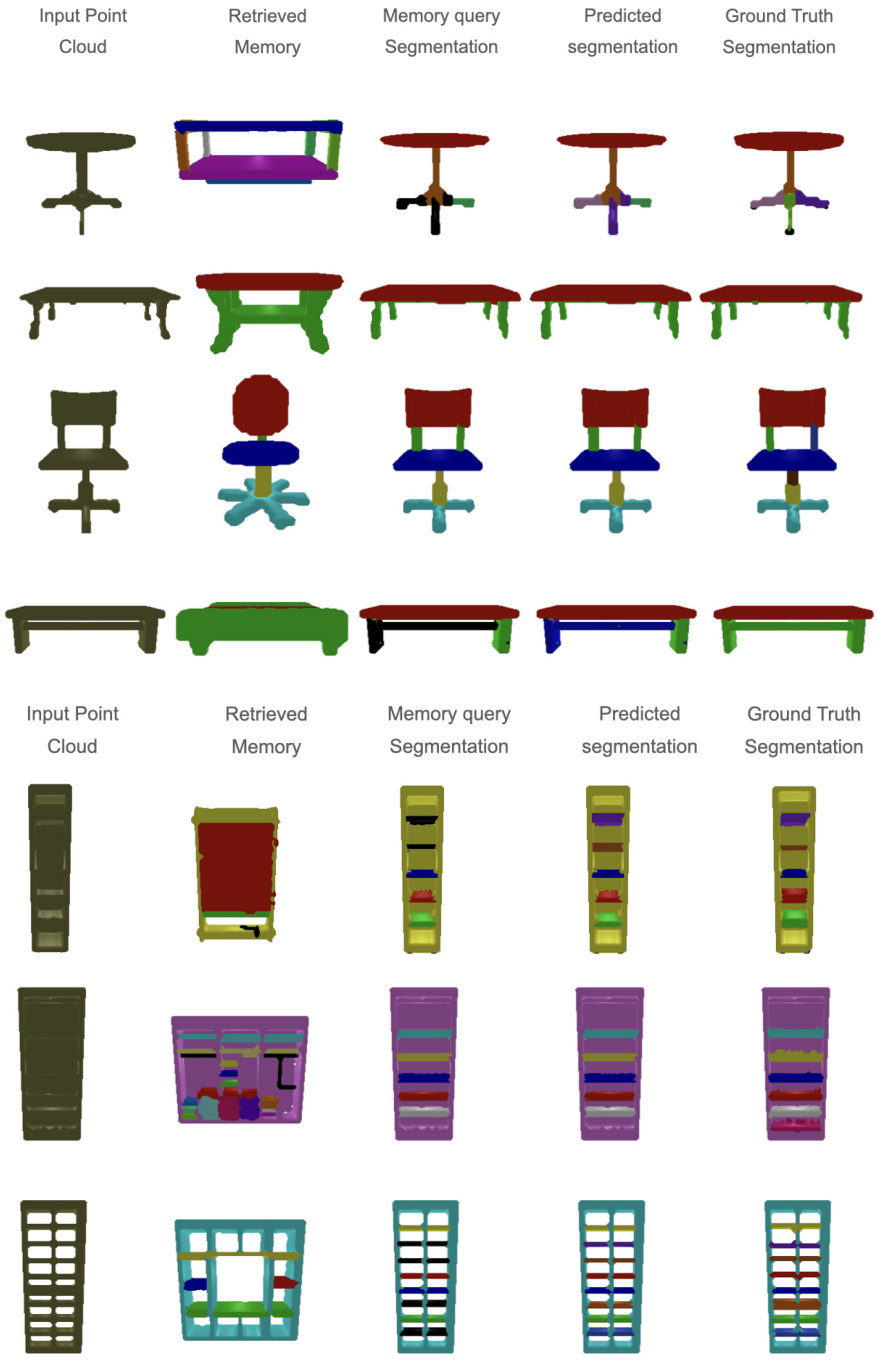

Figure A.1: Qualitative object parsing results using Analogical Networks.

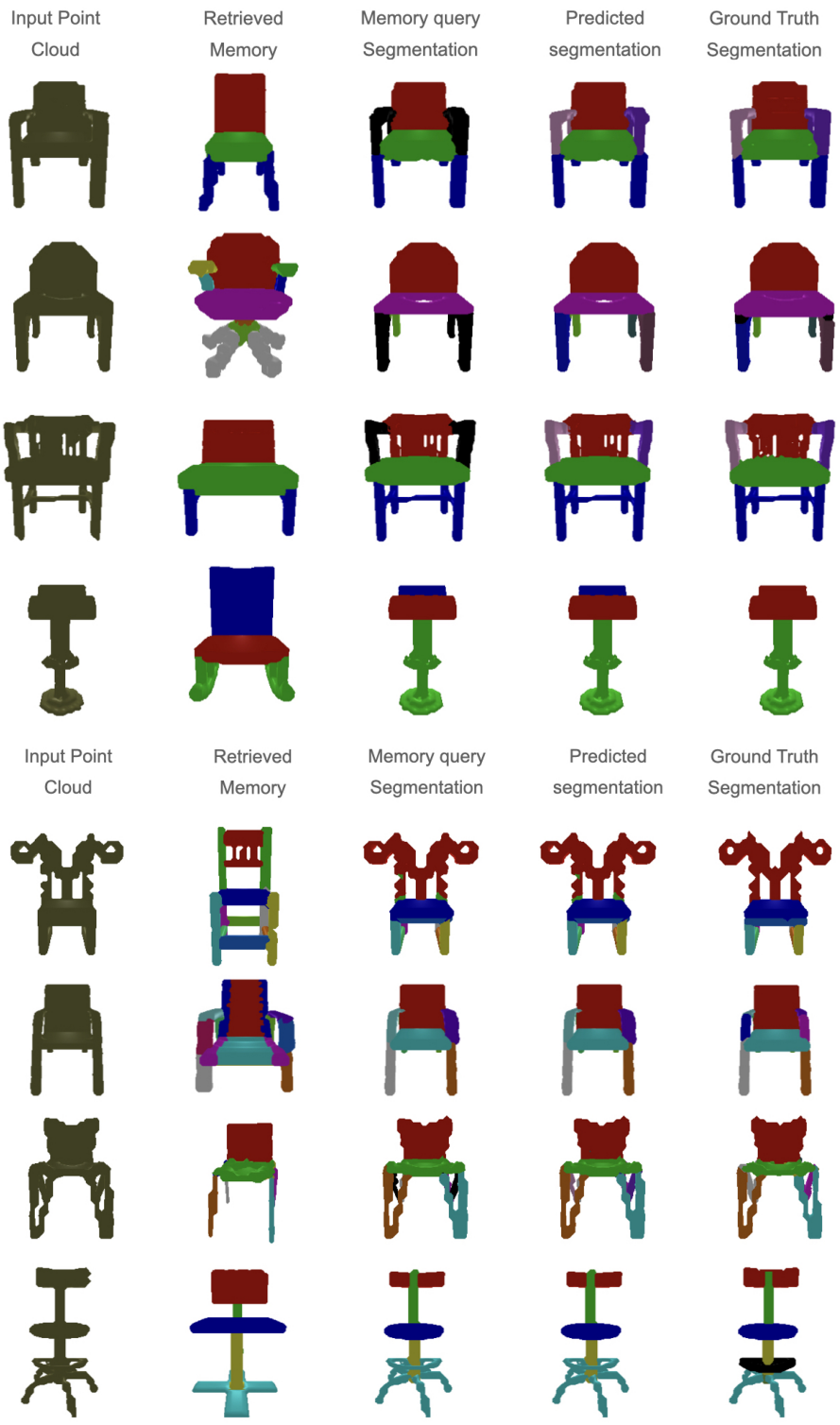

Figure A.2: Qualitative object parsing results for Analogical Networks.

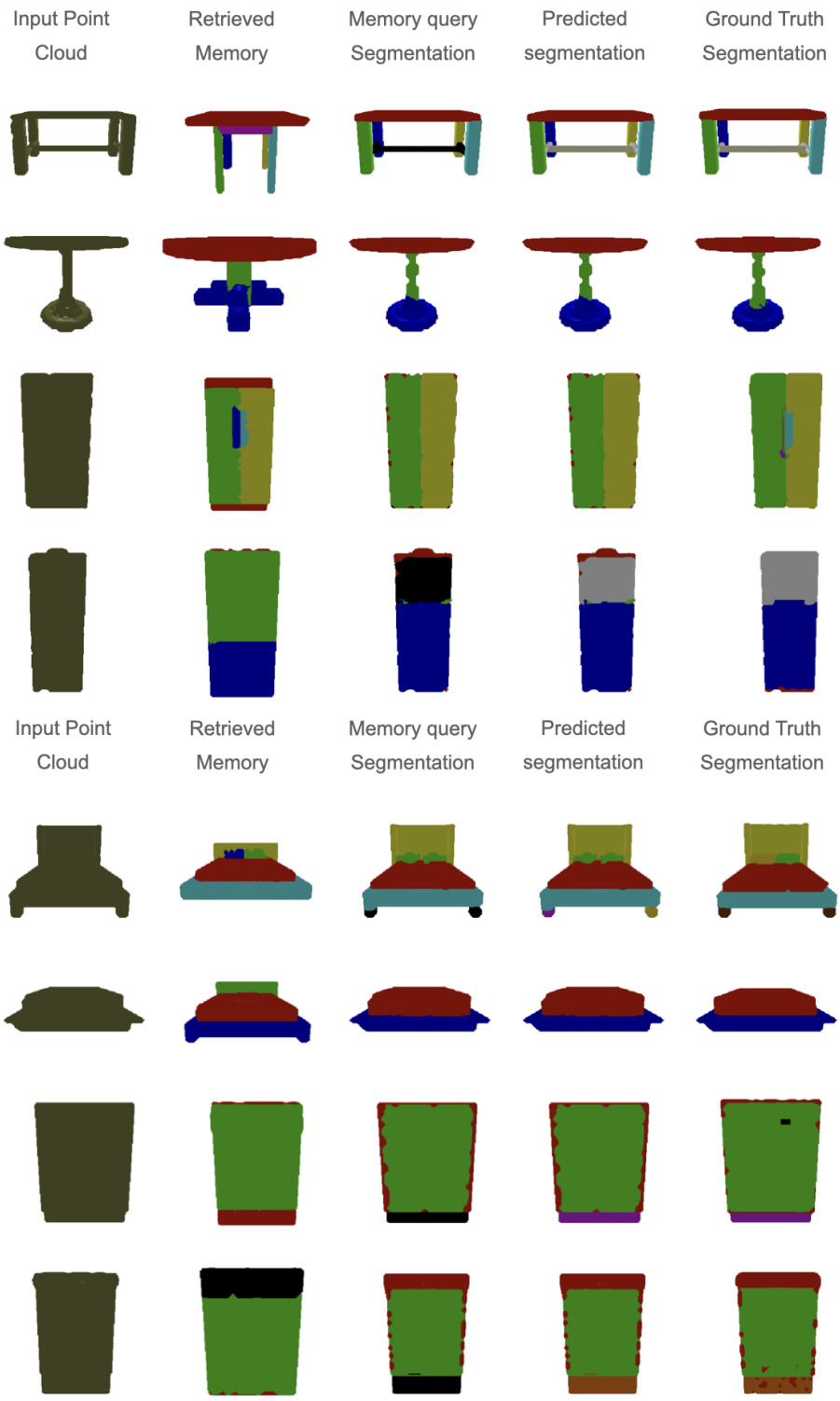

Figure A.3: Qualitative results on novel category samples from PartNet dataset [33] using Analogical Networks **without fine-tuning**.

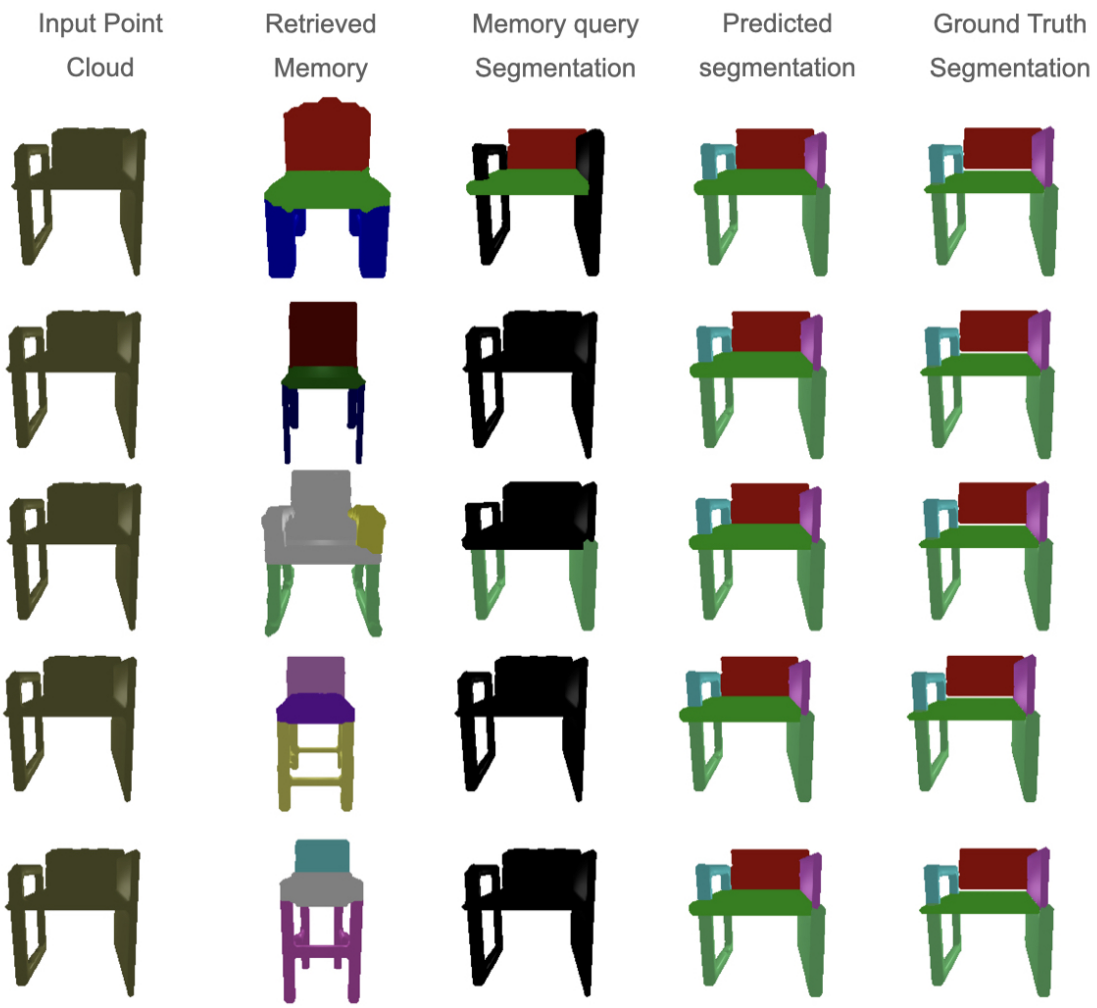

Figure A.4: Modulation using multi-memory Analogical Networks. The model takes as input 5 different memories simultaneously and then parses the object. Each row shows the effect of a different memory. All memories decode simultaneously and we show which part each one decodes in the third column. In the fourth column we show the combined predictions of all memories and scene-agnostic queries.

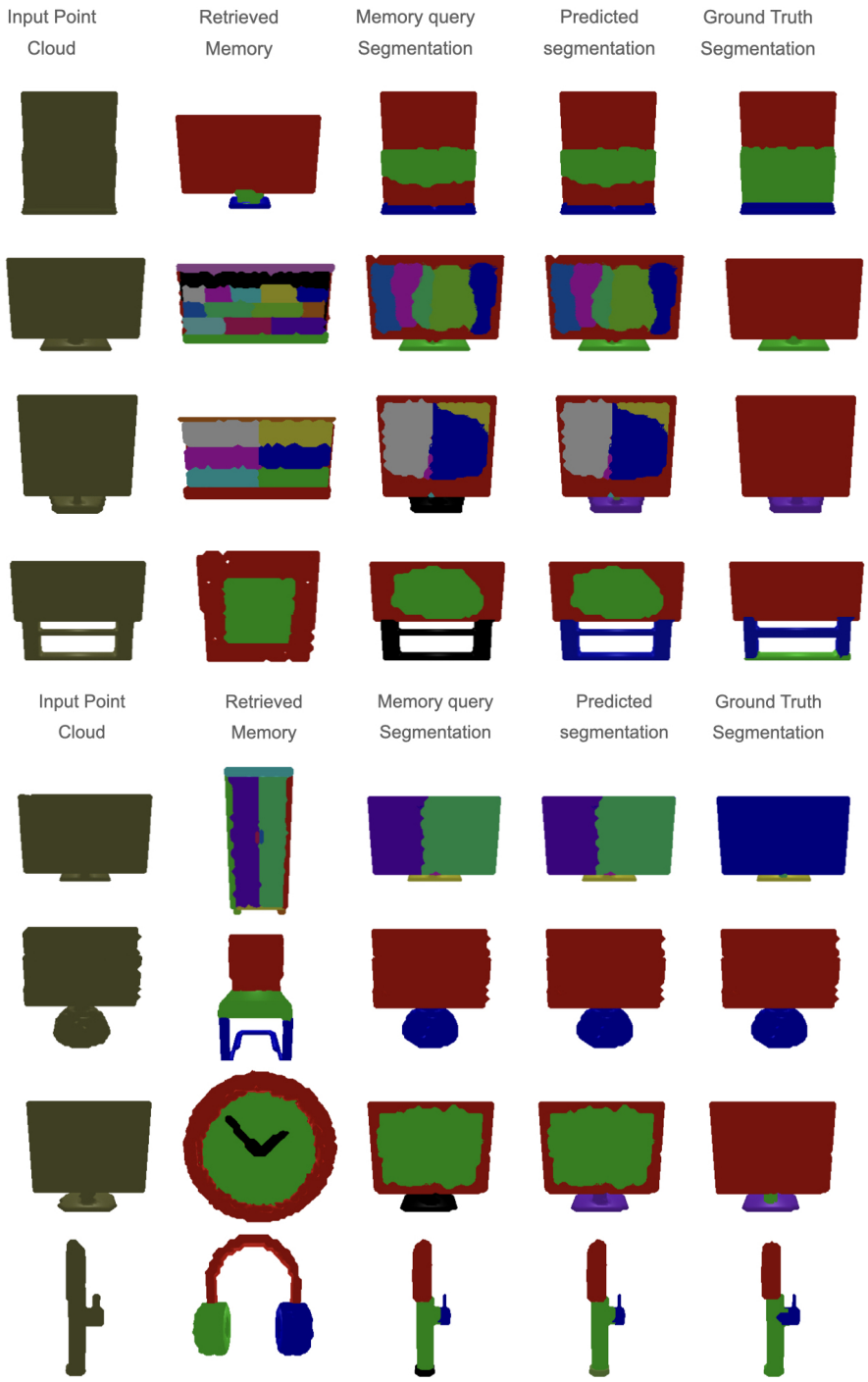

Figure A.5: To qualitatively evaluate the effect of modulation in parsing, we modulate the input point cloud with a different category object and show its corresponding object parsing that is predicted by Analogical Networks. The model is able to generalize geometric correspondences across instances of different classes, e.g. display and clock.

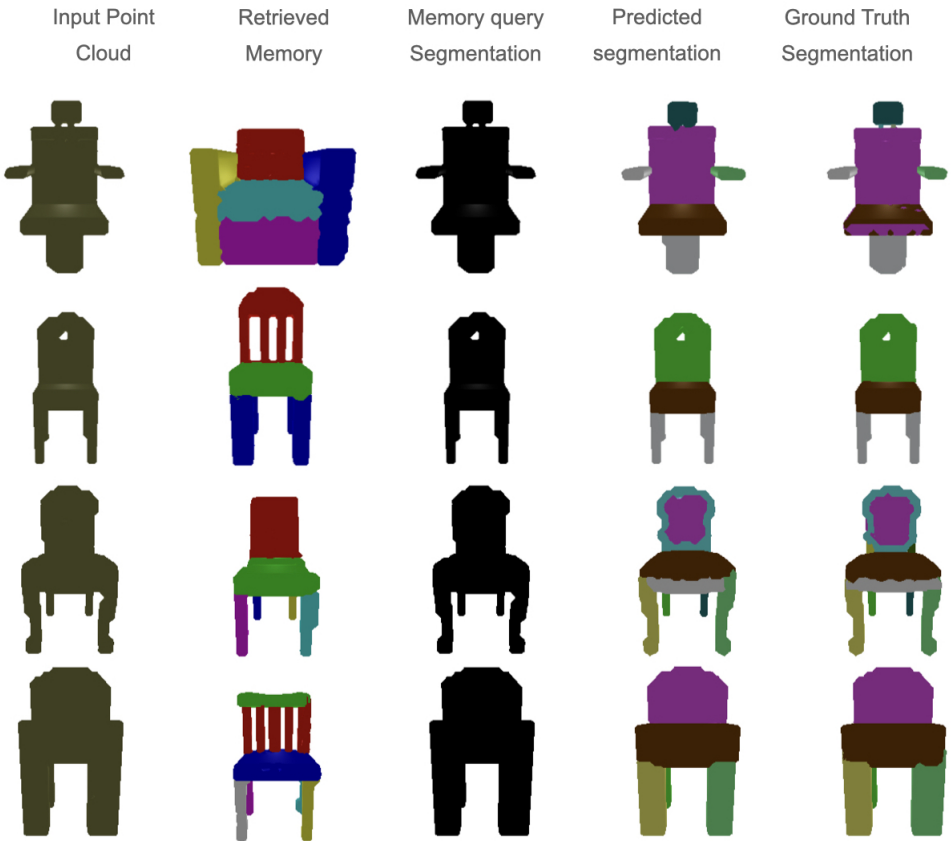

Figure A.6: We show the parsing of input point cloud using Analogical Networks single memory w/o pretrain. Most regions are black in column 3, denoting that memory part queries do not decode anything and everything is being decoded by scene-agnostic queries. This highlights the role of within-instance pre-training for the emergence of part correspondence.

Bibliography

[1] Ali Athar, Jonathon Luiten, Alexander Hermans, Deva Ramanan, and Bastian Leibe. Hodor: High-level object descriptors for object re-segmentation in video learned from static images, 2021. URL https://arxiv.org/abs/2112.09131. 4.2

[2] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A Efros. Visual prompting via image inpainting. *NeurIPS*, 2022. 2.1

[3] Moshe Bar. The proactive brain: Using analogies and associations to generate predictions. *Trends in Cognitive Sciences*, 11(7):280–289, 2007. doi: 10.1016/j.tics.2007.05.005. 1.2

[4] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. *CoRR*, abs/2112.04426, 2021. URL https://arxiv.org/abs/2112.04426. 2.2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *Proc. ECCV*, 2020. 1.2, 2.3, 3.3, 4.2

[6] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15447–15456, 2021. 2.3

[7] Wenhu Chen, Pat Verga, Michiel de Jong, John Wieting, and William Cohen. Augmenting pre-trained language models with qa-memory for open-domain question answering, 2022. URL https://arxiv.org/abs/2204.04581. 2.2, 3.4

[8] Xiaobai Chen, Aleksey Golovinskiy, and Thomas A. Funkhouser. A benchmark for 3d mesh segmentation. *ACM Trans. Graph.*, 28:73, 2009. 2.3

[9] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel clas-

sification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 3.3

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017. URL http://arxiv.org/abs/1702.04405. cite arxiv:1702.04405. 4.1, 4.6

[11] Rajarshi Das, Ameya Godbole, Ankita Naik, Elliot Tower, Robin Jia, Manzil Zaheer, Hannaneh Hajishirzi, and Andrew McCallum. Knowledge base question answering by case-based reasoning over subgraphs, 2022. URL https://arxiv.org/abs/2202.10610. 2.2

[12] Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Fei Sha, and William Cohen. Mention memory: incorporating textual knowledge into transformers through entity mention attention. *CoRR*, abs/2110.06176, 2021. URL https://arxiv.org/abs/2110.06176. 2.2

[13] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey E. Hinton, and Andrea Tagliasacchi. Cvxnets: Learnable convex decomposition. *CoRR*, abs/1909.05736, 2019. URL http://arxiv.org/abs/1909.05736. 1.2

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/finn17a.html. 2.1

[15] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020. 1.2

[16] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *International Conference on 3D Vision (3DV)*, 2016. 4.1

[17] Gautier Izacard and Edouard Grave. Distilling Knowledge from Reader to Retriever for Question Answering. In *ICLR 2021 - 9th International Conference on Learning Representations*, Vienna, Austria, May 2021. URL https://hal.archives-ouvertes.fr/hal-03463398. 4.6

[18] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models, 2022. URL https://arxiv.org/abs/2208.03299. 4.6

[19] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: zero-shot task generalization with robotic imitation learning. *CoRR*, abs/2202.02005, 2022. URL https:

//arxiv.org/abs/2202.02005. 4.2.1

[20] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4866–4875, 2020. 2.3

[21] R. K. Jones, Aalia Habib, and Daniel Ritchie. Shred: 3d shape region decomposition with learned local operations. *ArXiv*, abs/2206.03480, 2022. 2.3

[22] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019. 4.2

[23] Prakhar Kaushik, Alex Gain, Adam Kortylewski, and Alan L. Yuille. Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping. *CoRR*, abs/2102.11343, 2021. URL https://arxiv.org/abs/2102.11343. 1.2

[24] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models, 2019. URL https://arxiv.org/abs/1911.00172. 2.2

[25] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J Kim. Point cloud augmentation with weighted local transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 548–557, 2021. 3.4

[26] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4.3

[27] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 142–158. Springer, 2020. 2.1

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. A.0.1

[29] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 2.1

[30] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *CoRR*, abs/1904.12584, 2019. URL http://arxiv.org/abs/1904.12584. 2.4

[31] Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. *CoRR*, abs/2103.16553, 2021. URL https://arxiv.org/abs/2103.16553. 3.4

[32] M. Minsky. A framework for representing knowledge. In R. J. Brachman and H. J. Levesque, editors, *Readings in Knowledge Representation*, pages 245–262. Kaufmann, Los Altos, CA, 1985. URL http://web.media.mit.edu/~minsky/papers/Frames/frames.html. 2.4

[33] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. (document), 1.2, 2.3, 4.1, 4.2, 4.1, 4.2, 4.3.1, 4.4, 4.5, 4.5, 4.7, 4.4, 4.5, A.3

[34] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 622–631, 2019. 2.1

[35] Charles Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. NIPS*, 2017. 3.1, 3.3

[36] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971. 4.2

[37] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018. 2.1

[38] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1842–1850. JMLR.org, 2016. 2.2

[39] Oana Sidi, Oliver Matias van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *Proceedings of the 2011 SIGGRAPH Asia Conference*, 2011. 2.3

[40] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf. 1.2, 2.1, 4.2

[41] Chunyu Sun, Xin Tong, and Yang Liu. Semantic segmentation-assisted instance feature fusion for multi-level 3d part instance segmentation. *Computational Visual Media*, 2022. (document), 1.2, 2.3, 4.3.1, 4.5

[42] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2.1

[43] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2.1

[44] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1588–1597, 2019. (document), 4.1, 4.2.3, 4.2.3, 4.3, 4.1, 4.2, 4.3

[45] Thang Vu, Kookhoi Kim, Tung Minh Luu, Xuan Thanh Nguyen, and Chang-Dong Yoo. Softgroup for 3d instance segmentation on point clouds. *ArXiv*, abs/2203.01509, 2022. 2.3, 3.3

[46] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. (document), 2.3, 4.3.1, 4.5

[47] Xiaogang Wang, Xun Sun, Xinyu Cao, Kai Xu, and Bin Zhou. Learning fine-grained segmentation of 3d shapes without part labels. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10271–10280, 2021. 2.3

[48] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. *CoRR*, abs/1803.08035, 2018. URL `http://arxiv.org/abs/1803.08035`. 2.1

[49] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020. 2.1

[50] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. PQ-NET: A generative part seq2seq network for 3d shapes. *CoRR*, abs/1911.10949, 2019. URL `http://arxiv.org/abs/1911.10949`. 1.2

[51] Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers, 2022. URL `https://arxiv.org/abs/2203.08913`. 2.2, 4.1, 4.2

[52] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *CoRR*, abs/1906.01140, 2019. URL http://arxiv.org/abs/1906.01140. 1.2

[53] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 763–778. Springer, 2020. 2.1

[54] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic VQA: disentangling reasoning from vision and language understanding. *CoRR*, abs/1810.02338, 2018. URL http://arxiv.org/abs/1810.02338. 2.4

[55] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. *CoRR*, abs/1910.01442, 2019. URL http://arxiv.org/abs/1910.01442. 2.4

[56] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J. Guibas. GSPN: generative shape proposal network for 3d instance segmentation in point cloud. *CoRR*, abs/1812.03320, 2018. URL http://arxiv.org/abs/1812.03320. 1.2

[57] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1.2, 2.3

[58] Biao Zhang and Peter Wonka. Point cloud instance segmentation using probabilistic embeddings. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8879–8888, 2021. (document), 2.3, 4.3.1, 4.5

[59] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision, 2022. URL https://arxiv.org/abs/2201.02605. 2.4