



AFRL-RI-RS-TR-2023-195

ACTIVE SIMILARITY LEARNING AND MANIFOLD GRAPHS FOR LEARNING WITH FEW LABELS

GEORGIA TECH RESEARCH INSTITUTE

NOVEMBER 2023

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2023-195 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

NANCY A. SEPULVEDA
Work Unit Manager

/ S /

SCOTT D. PATRICK
Deputy Chief
Intelligence Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE		2. REPORT TYPE		3. DATES COVERED	
NOVEMBER 2023		FINAL TECHNICAL REPORT		START DATE SEPTEMBER 2019	END DATE MAY 2023
4. TITLE AND SUBTITLE ACTIVE SIMILARITY LEARNING AND MANIFOLD GRAPHS FOR LEARNING WITH FEW LABELS					
5a. CONTRACT NUMBER FA8750-19-C-0200		5b. GRANT NUMBER N/A		5c. PROGRAM ELEMENT NUMBER 61101E	
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER R2U1	
6. AUTHOR(S) Zsolt Kira, Chris Rozell, Mark Davenport, and Phillip Odom					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Georgia Tech Research Institute 505 10TH ST Atlanta GA 30318				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RIEA 525 Brooks Road Rome NY 13441-4505			10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI		11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RI-RS-TR-2023-195
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT While significant strides had been made in deep learning in the 5 years subsequent to its resurgence (2012), a large part of this success was through large-scale training of deep neural networks on labeled data. Transfer learning was commonly used, but in reality understanding how to effectively fine-tune pre-trained models (such as those pre-trained on ImageNet [8]) was not well understood. For example, the methods were susceptible to a shift in the data distribution from the source dataset (domain shift) and it was often not clear where to transfer from given a new problem. While early methods involving unsupervised, semi-supervised, domain adaptation, and few-shot learning had been developed (including by this team), they tackled small subsets of real-world problems and in practice often could not beat strong baselines with transfer learning alone [9]. Since then, the landscape has drastically changed, and the methods developed under this program as well as by the community at large have enabled the usage of large-scale unlabeled pre-training combined with semi-supervised learning to significantly increase performance under low-labeled conditions. Their team contributed a significant number of advancements to this body of work, ranging from scientific formulations and understanding of the problems to the development of practical algorithms that performed well under both established academic datasets to the evaluations. Their work viewed these low-labeled problems by explicitly representing the structure of the data manifold using graph representations. Through this lens, they proposed a number of innovations that answer the questions of what to transfer, how to transfer, and where to transfer from. They have specifically developed methods for 1) Leveraging the Manifold-Structure through FeatMatch [13], 2) Leveraging the Manifold-Structure through Manifold Transport Operators [14, 15, 16, 17, 18], 3) Semi-Supervised Clustering and Retrieval for Zero-Shot Learning [19], 4) Low-Labeled Captioning through Retrieval [20, 21], 5) Semi-Supervised Object Detection through Cross-Match , and 6) Advancements in Active Learning.					
15. SUBJECT TERMS Transfer learning, unsupervised learning, semi-supervised learning, domain adaptation, few-shot learning, image classification, object detection					
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR		18. NUMBER OF PAGES 72
19a. NAME OF RESPONSIBLE PERSON NANCY A. SEPULVEDA				19b. PHONE NUMBER (Include area code) N/	

Table of Contents

List of Figures	iii
1.0 Summary	1
2.0 Introduction.....	2
3.0 Methods, Assumptions, and Procedures.....	3
3.1 FeatMatch: Manifold Structure for Semi-Supervised Classification.....	3
3.1.1 Multi-domain Semi-Supervised Learning.....	4
3.1.2 Feature-Based Augmentation and Consistency.....	5
3.1.3 Prototype Selection.....	6
3.1.4 Learned Feature Augmentation.....	7
3.1.5 Consistency Regularization.....	8
3.1.6 Total Loss.....	9
3.2 Leveraging Manifold Structure via Transport Operators.....	9
3.2.1 Variational Autoencoder with Learned Latent Space	10
3.2.2 Variational Sparse Coding	12
3.2.3 Manifold Contrastive Learning	13
3.3 Semi-Supervised Clustering and Retrieval for Zero-Shot Learning	14
3.3.1 Problem Setup of GCD	15
3.3.2 Our Approach.....	16
3.4 Retrieval for Low-Labeled Image Captioning	18
3.4.1 Method	19
3.4.2 Preliminaries	19
3.4.3 Multi-view Hierarchical Ensembling	19
3.4.4 Model Architecture	20
3.4.5 Auxiliary Contrastive Loss	21
3.4.6 Implementation Details	21
3.4.7 Improving the Multi-View Fusion	22
3.5 Semi-Supervised Object Detection through Cross-Match	26
3.5.1 Methodology	28
3.5.2 Problem Definition.....	28
3.5.3 General Framework.....	28
3.5.4 CrossMatch	29
3.5.5 Consistency with Hard Labels.....	29
3.5.6 Consistency with Soft Labels.....	30
3.6 Advancements in Active Learning.....	31
3.6.1 Summary	31
3.6.2 Introduction.....	31
3.6.3 Methods, Assumptions and Procedures	32

4.0 Results and Discussion	35
4.1 FeatMatch: Results	35
4.2 Manifold Transport Operators	36
4.2.1 CelebA Augmentations	36
4.2.2 Variational Inference on Synthetic Dataset.....	37
4.2.3 Learning with Limited Labels using Manifold Augmentations	38
4.3 Semi-Supervised Clustering and Retrieval for Zero-Shot Learning	39
4.3.1 Model architecture details.....	39
4.3.2 Datasets & Evaluation	39
4.3.3 Comparison with the State-of-the-Art.....	40
4.3.4 Analysis	41
4.4 Retrieval for Low-Labeled Image Captioning	43
4.4.1 Initial Method Results.....	43
4.4.2 Improved HAAV Results	44
4.5 Semi-Supervised Object Detection through Cross-Match	45
4.5.1 Results on MS COCO	45
4.5.2 Results on Pascal VOC	46
4.6 Active Learning	46
4.6.1 <i>Info-NN</i> applied to metric learning.....	48
5.0 Conclusions.....	49
6.0 References.....	50
Appendix A - Publications and Presentations.....	62
List of Symbols, Abbreviations, and Acronyms.....	64

List of Figures

Figure 1: Our setup for systematically studying how semantic shift (i.e. different label space) and non-semantic shift (i.e. different modalities). We use a common target set and can investigate either type of shift independently or both at the same time.	3
Figure 2: A prototype recording loop that runs alongside the model training loop. The image features f_{xi} as well as their pseudo labels $\hat{y}_i$ already generated at each iteration of the training loop are collected and recorded in a memory bank as $(f_{xi}, \hat{y}_i)$ pairs. Once the training loop goes over the whole dataset, the recording loop will run K-Means to extract prototypes for each class, update the prototypes for feature-based augmentation, and clear the memory bank.	5
Figure 3: Consistency regularization methods are the most successful methods for semi-supervised learning. The main idea of these methods is to enforce consistency between the predictions of different transformations of an input image. (a) Image- based augmentation method generate different views of an input image via data augmentation, which are limited to operations in the image space as well as operations within a single instance or simple convex combination of two instances. (b) We propose a learned feature-based augmentation that operates in the abstract feature space. The learned feature refinement and augmentation module is capable of leveraging information from other instances, within or outside of the same class.	6
Figure 4: A prototype recording loop that runs alongside the model training loop. The image features f_{xi} as well as their pseudo labels $\hat{y}_i$ already generated at each iteration of the training loop are collected and recorded in a memory bank as $(f_{xi}, \hat{y}_i)$ pairs. Once the training loop goes over the whole dataset, the recording loop will run K-Means to extract prototypes for each class, update the prototypes for feature-based augmentation, and clear the memory bank.	7
Figure 5: Feature-Based Augmentation: The input image features are augmented by attention using extracted prototype features (Eq.1), where the colors of * represent the classes of prototypes. The prototype features are calculated via a weighted sum using the attention weights, concatenated with the image features, and then undergo a fc layer ϕ_a (Eq.2) to produce attention features f_a . Finally, we use the attention features to refine and augment the input image features with a residual connection (Eq.3).	8
Figure 6: Visualization of the encoding of anchor points into the VAELLS latent space.	10
Figure 7: Examples of images decoded from latent vectors sampled from the posterior of models trained on MNIST digits. In each example, the center digit (in the green box) is the encoded digit input and the surrounding digits are from the sampled latent vectors. (a-b) Show samples from VAELLS trained on rotated MNIST digits. This shows that sampling in the VAELLS latent space results in rotations in the sampled outputs.	11
Figure 8: Our proposed approach to variational sparse coding. (a) Standard amortized variational inference (top) results in reparameterized latent features that are not sparse. Our approach (bottom) incorporates a straight-through estimator of the soft-threshold to incorporate exact sparsity. In the backwards pass, we avoid numerical instability by passing the gradient directly from the generator to the inference network. Orange arrows denote a forward pass and blue arrows denote a backwards pass. (b) A visualization of our shifted soft-threshold. We analytically show that this reparameterization leads to a spike-and-slab distribution, explicitly modeling the degree of sparsity in our latent features.	12
Figure 9: Proposed ManifoldCLR system for incorporating Lie group operators into the feature space of a contrastive learner. Given encoded features from a positive pair, manifold feature augmentations $T_\Psi((c_i))$ are applied before computing z_j denote the set of negative pairs in the feature space. We apply the conditional VAE framework to learn coefficient distributions for manifold displacement between two points (encoder network $q_\phi(c z_i, z_j)$) and identity-preserving augmentations from a single point (prior network $p_\theta(c z_i)$).	13
Figure 10: Model Architecture. In stage I (left), we propose a cross-modal retrieval module to retrieve a	

set of contextual text descriptions for the labeled and unlabeled data, generate a view from pooled sentence embedding as complementary information for clustering. In Stage II (right), we concatenate the image view and the text view and use semi-supervised k-means clustering to group seen and unseen classes.	16
Figure 11: A sample of retrieved top-4 most relevant text descriptions from Conceptual Captions (3M) for an image from ImageNet dataset.	17
Figure 12: Two-tier hierarchical cross-attention of the modified transformer decoder layer. For clarity, we only show the operations of the last token $y_{1,t}$	21
Figure 13: HAAV, H ierarchical A ggregation of A ugmented V iews, for image captioning at the step of predicting the word “sofa”. First, heterogeneous views such as detected objects [1], image grid features [2], and text descriptions [3] are generated from the input image by existing methods. We propose to regard these views as augmentations of the input image, and independently encode each view by a shared transformer encoder efficiently. A contrastive loss is incorporated to improve the representation quality of heterogeneous views. Finally, our proposed hierarchical decoder models the effectiveness of each view and adaptively weigh them according to their effectiveness for predicting the current word “sofa”.	22
Figure 14: Model architecture. (Left) An input image is first encoded by some frozen pre-trained models into a set of heterogeneous views $\{v_1, v_2, v_3\}$, each represented by a sequence of d -dimensional tokens. These input views v are regarded as augmentations of the input image, and thus are encoded independently with a shared transformer encoder into $\{u_1, u_2, u_3\}$. (Right) To model the effectiveness of the encoded views u for current word prediction, we propose a hierarchical decoder layer that first aggregates within each encoded view u at the token level with a shared $\text{CrossAttn}_{L_{v1}}$ module, and then aggregates across the encoded views at the view level with a $\text{CrossAttn}_{L_{v2}}$ module. For clarity of illustration, we only show operations for the current $y_{1,t}$ token and not the $y_{1,1:t-1}$ past tokens.	24
Figure 15: An illustration of sparse and weak gradient signals from hard pseudo label. (a): The number of pseudo-label generated per image is much smaller than the number of ground-truth labels, which indicates the supervision signal from pseudo-labels is sparse. (b): In the training process, only a small number of true positive RoIs are sampled with pseudo-labels and a significant amount of foreground RoIs are not sampled (false negative). There are also a number of false positive RoIs that are mis-sampled. This indicates the supervision signal from pseudo-labels is also weak.	27
Figure 16: Overview of CrossMatch . CrossMatch enforces consistency between different scales of feature maps. The teacher model takes weakly augmented images and produces artificial (in both soft and hard form, with a refined OHEM employed for the regions used for soft) with a heuristically selected level of features (denoted by red). The student model takes strongly augmented images and constantly makes predictions with every level of the feature pyramid . The consistency constraint is enforced through both soft and hard pseudo-labels.	28
Figure 17: Visualization of the unified query solicitation framework. Candidate NN queries to be evaluated by the active NN query selection method are formed based on the embeddings learned by the model. These candidate queries are evaluated for informativeness by considering mutual information between the learned embedding and the query response. The oracle then responds to the most informative of these queries.	32
Figure 18: Example of an unlabelled z_i and the nearest labelled neighbors to z_i from each class: $z_i^{(1)}$, $z_i^{(2)}$, $z_i^{(3)}$, $z_i^{(4)}$. In this example, we might expect that the most likely label would be $y_c = 4$, which could be interpreted as a nearest neighbor query response (that $z_i^{(4)}$ is the nearest neighbor).	32
Figure 19: Transformations generated by applying a subset of learned transport operators on three datasets. In each figure, images in the middle column of the image block are the reconstructed inputs and images to the right and left are images decoded from transformed latent vectors in positive and negative directions, respectively. We compare transformations from learned transport operators to transformations	

generated by β -VAE and CAE. The transport operators learn semantically meaningful transformations similar to the disentangled β -VAE representation while maintaining higher reconstruction quality in image outputs.....	36
Figure 20: Paths generated by the application of two learned transport operators with the associated attribute classifier probability outputs for the transformed images. Our model learns operators that correspond to meaningful CelebA dataset attributes.	36
Figure 21: Comparison of transport operators trained with exact and variational sparse inference on synthetic swiss roll dataset. For each method, we plot (a) mean-squared error, (b) cumulative runtime, and (c) f_1 penalty of coefficients. (d,e) Extrapolated paths of the two operators learned by proposed variational sparse coding method. Black dots indicate starting point $\mathbf{x}_i$, with subsequent path found by applying transport operators $\mathbf{T}_{\Psi}(\mathbf{c}_i)\mathbf{x}_i$ with coefficients from a range with fixed step size $\mathbf{c}_i = [-N_c, \dots, N_c]$...	37
Figure 22: Semi-supervised experiments using a frozen backbone and learned, single hidden layer MLP. Comparison methods leverage unlabeled data to encourage consistency in classifier output after applying feature augmentations. Taken over the same 50 data splits used for each method. Variational Lie Group operators (VLGO) refer to the transport operator model with variational sparse coding.	38
Figure 23: Active classification performance comparison on MNIST (left), CIFAR-10 (center) and SVHN (right) datasets.....	48
Figure 24: (Left) An example query in the case of metric learning. An oracle is presented with a reference item and a set of items to report similarity. The green dashed item indicates the oracle’s response. (Right) Performance comparison of Info-NN with baseline active methods and random selection. Our method outperforms all the other methods and NN queries exhibit far superior per-query performance, requiring less interactions with the oracle.....	48

1.0 Summary

While significant strides had been made in deep learning in the 5 years subsequent to its resurgence (2012), a large part of this success was through large-scale training of deep neural networks on *labeled* data. Transfer learning was commonly used, but in reality understanding how to effectively fine-tune pre-trained models (such as those pre-trained on ImageNet [8]) was not well understood. For example, the methods were susceptible to a shift in the data distribution from the source dataset (*domain shift*) and it was often not clear where to transfer from given a new problem. While early methods involving unsupervised, semi-supervised, domain adaptation, and few-shot learning had been developed (including by this team), they tackled small subsets of real-world problems and in practice often could not beat strong baselines with transfer learning alone [9].

Since then, the landscape has drastically changed, and the methods developed under this program as well as by the community at large have enabled the usage of large-scale unlabeled pre-training combined with semi-supervised learning to significantly increase performance under low-labeled conditions. Our team contributed a significant number of advancements to this body of work, ranging from scientific formulations and understanding of the problems to the development of practical algorithms that performed well under both established academic datasets to the evaluations.

Our work viewed these low-labeled problems by *explicitly representing the structure of the data manifold* using graph representations. Through this lens, we proposed a number of innovations that answer the questions of **what to transfer**, **how to transfer**, and **where to transfer from**. We have specifically developed methods for 1) Leveraging the Manifold-Structure through **FeatMatch** [13], 2) Leveraging the Manifold-Structure through **Manifold Transport Operators** [14, 15, 16, 17, 18], 3) **Semi-Supervised Clustering and Retrieval** for Zero-Shot Learning [19], 4) **Low-Labeled Captioning through Retrieval** [20, 21], 5) Semi-Supervised Object Detection through **Cross-Match**, and 6) **Advancements in Active Learning**.

Our methods have been published at top venues including: CVPR, ECCV, NeurIPS ICML, AAAI, UAI, and TMLR. We have also had significant practical impact, and open-sourced a number of repositories related to our papers, which have had hundreds of stars and been used extensively. Finally, our methods have achieved strong results, often one of the top performers, at the independent DARPA evaluations.

2.0 Introduction

While significant strides had been made in deep learning in the 5 years subsequent to its resurgence (2012), a large part of this success was through large-scale training of deep neural networks on *labeled* data. Transfer learning was commonly used, but in reality understanding how to effectively fine-tune pre-trained models (such as those pre-trained on ImageNet[8]) was not well-understood. For example, the methods were susceptible to a shift in the data distribution from the source dataset (*domain shift*) and it was often not clear where to transfer from given a new problem. While early methods involving unsupervised, semi-supervised, domain adaptation, and few-shot learning had been developed (including by this team), they tackled small subsets of real-world problems and in practice often could not beat strong baselines with transfer learning alone [9].

Since then, the landscape has drastically changed, and the methods developed under this program as well as by the community at large have enabled the usage of large-scale unlabeled pre-training combined with semi-supervised learning to significantly increase performance under low-labeled conditions. Our team contributed a significant number of advancements to this body of work, ranging from scientific formulations and understanding of the problems to the development of practical algorithms that performed well under both established academic datasets to the evaluations.

Our work viewed these low-labeled problems by *explicitly representing the structure of the data manifold* using graph representations. Through this lens, we proposed a number of innovations that answer the questions of **what to transfer**, **how to transfer**, and **where to transfer from**. The basis of *what to transfer* is formed by our previous work [10, 11], which shows that *similarity comparisons* (i.e. whether two instances are of the same semantic class or not) rather than labels can form the basis of supervised, semi-supervised, cross-task transfer, and domain adaptation within the same framework and achieve state of art results. For example, when a new unlabeled dataset contains completely new categories, we have shown that feature transfer is not as effective as similarity transfer [10]. In other words, similarity learning can be transferred across domains, and this can also be augmented via unsupervised learning of similarity (e.g. through weak labels such as those obtained from temporal information), meta-learning, etc. Unlike this previous work, which implicitly optimized classifiers given these constraints, we propose to use these similarity comparisons to explicitly represent the connectivity of instances across a graph, i.e. nodes in the graph (representing instances) are connected via edges weighted by their similarity. Given unlabeled and labeled instances from dissimilar domains (TA 1.1) or similar domains (TA 1.2), we then propose to use graph convolutional neural networks (GCNNs) [12] to propagate feature *and* label information across the graph to estimate the class labels for new unlabeled instances in a target domain. This forms the basis of *how to transfer*.

As a summary, we highlight a list of our contributions below:

1. **Leveraging the Manifold-Structure through FeatMatch [13]** - Novel methods for graph (manifold)-based augmentation for semi-supervised image classification.
2. **Leveraging the Manifold-Structure through Transport Operators [14, 15, 16, 17, 18]** - A series of methods for learning manifold transport operators in an efficient manner on real-world data, integrated within semi-supervised learning pipelines to improve performance.
3. **Semi-Supervised Clustering and Retrieval for Zero-Shot Learning [19]** - Novel methods for integrating semi- supervised clustering and retrieval, within large-scale pre-trained feature spaces, to perform effective zero-shot learning.
4. **Low-Labeled Captioning through Retrieval [20, 21]** - Methods for retrieval-based augmentation of images, which can be seen as additional *views* of the data, and efficient and effective architectures for exploiting them for low-labeled image captioning.
5. **Semi-Supervised Object Detection through Cross-Match [22]** - Methods for improving gradient flow in existing semi-supervised object detection methods to significantly improve performance.

6. **Advancements in Active Learning [23]** - Unified view of active learning with connections to deep metric learning, and a novel method Info-NN for selecting queries using information-theoretic criterion.

Our methods have been published at top venues including: CVPR, ECCV, NeurIPS ICML, AAAI, UAI, and TMLR. We have also had significant practical impact, and open-sourced a number of repositories related to our papers, which have had hundreds of stars and been used extensively. Finally, our methods have achieved strong results, often one of the top performers, at the independent DARPA evaluations.

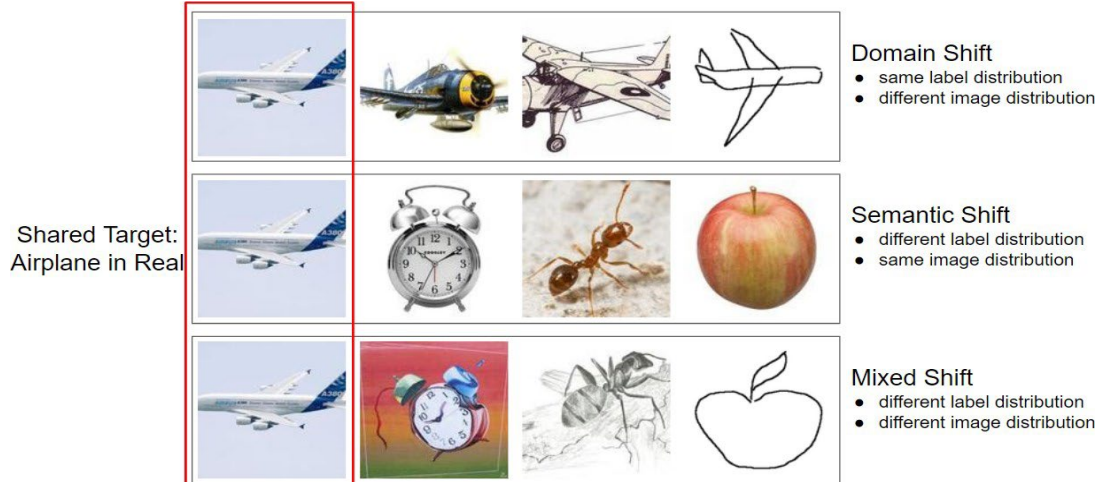


Figure 1: Our setup for systematically studying how semantic shift (i.e. different label space) and non-semantic shift (i.e. different modalities). We use a common target set and can investigate either type of shift independently or both at the same time.

3.0 Methods, Assumptions, and Procedures

In this section, we describe details of our approach including assumptions that underlie them. We specifically break down our methods into the six categories mentioned above.

3.1 FeatMatch: Manifold Structure for Semi-Supervised Classification

In this section, we present our work on our developed manifold-based augmentation techniques [13]. Our focus in the area of semi-supervised classification is two-fold: 1) Advancing results on standard semi-supervised learning datasets and 2) adapting semi-supervised methods and our graph-based feature augmentation to the multi-domain setting. First, we will describe the overall scenario we are setting up. In section 3.1.2 we describe our method and in 4.1 we describe our results that we have obtained for both standard datasets and across data shift.

3.1.1 Multi-domain Semi-Supervised Learning

For the LwLL problems, the unlabeled data may come from a domain different from the domain of the target (sparsely) labeled data. For instance, given a small set of labeled natural images of animals, the large unlabeled set may also contain animal images from paintings, sketches, *etc.* In general, we are interested in the case where there is both semantic shift (i.e. label shift) and non-semantic shift (e.g. different modalities, see Figure 1). Below we describe the setting with only non-semantic shift for which we have preliminary results. We are currently working on investigating either and/or both types of shift.

To investigate the effect of domain shift in the unlabeled set, we proposed a new SSL task based on the DomainNet dataset [24], which contains 345 classes of images coming from six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch.

We use the *Real* domain as our target. Five percent of the data from the Real domain are kept as the target labeled set, and the rest are the target unlabeled set. We select *Painting*, *Sketch*, and *Quickdraw* as shifted domains. To modulate the level of domain shift in the unlabeled data, we propose a parameter r_u that controls the ratio of unlabeled data coming from the target Real domain or the shifted domains. Specifically, r_u percent of target Real unlabeled set is replaced with data uniformly drawn from the shifted domains. By formulating the problem this way, the amount of unlabeled data remains constant. The only factor that affects the performance of the proposed method is the ratio between in-domain data and shifted domain data in the unlabeled set.

We randomly reserve 1% of data from the Real domain as the validation set. The final result is reported on the test set of the Real domain, with the model selected on the reserved validation set. The images are center-cropped and resized to 128x128 resolution, and the model we use is the standard ResNet-18 [25]. There are around 120k training samples, which is more than twice larger than the standard SSL datasets such as CIFAR-10 and CIFAR-100. For a fair comparison, we fix *all* hyper-parameters across experiments of different r_u to truly assess the robustness of proposed methods toward domain shift in the unlabeled data.

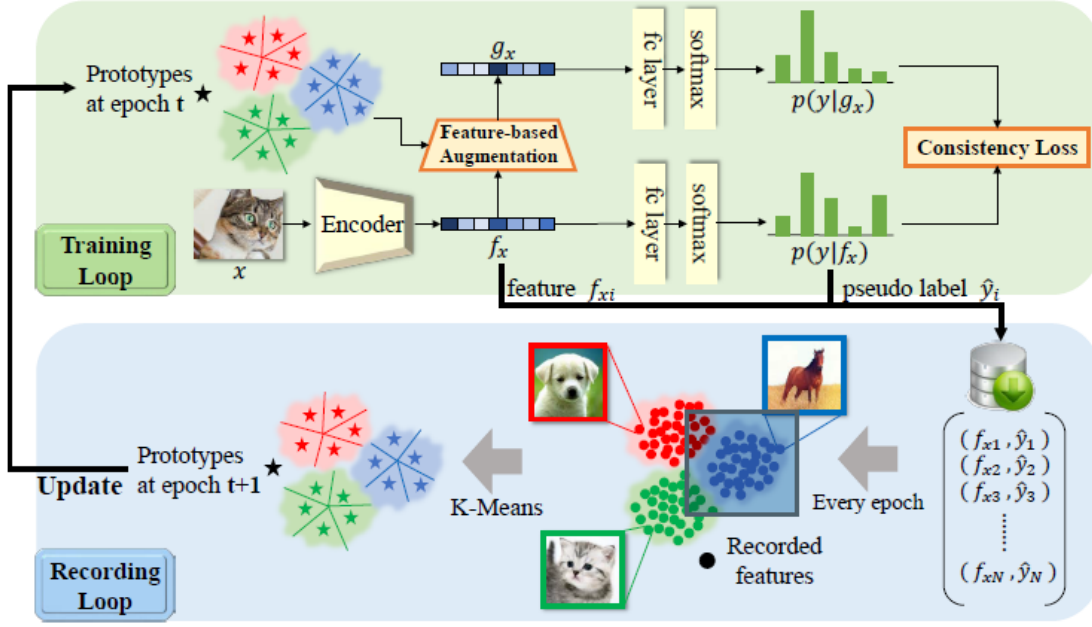


Figure 2: A prototype recording loop that runs alongside the model training loop. The image features f_{xi} as well as their pseudo labels $\hat{y}_i$ already generated at each iteration of the training loop are collected and recorded in a memory bank as $(f_{xi}, \hat{y}_i)$ pairs. Once the training loop goes over the whole dataset, the recording loop will run K-Means to extract prototypes for each class, update the prototypes for feature-based augmentation, and clear the memory bank.

3.1.2 Feature-Based Augmentation and Consistency

As mentioned, during the program we have looked at leveraging our manifold graph methods to perform *complex* data augmentations. Image-based augmentation has been shown to be an effective approach to generate different views of an image for consistency-based SSL methods. However, conventional image-based augmentation has the following two limitations:

1. Operate in image space, which limits the possible transformations to textural or geometric within images.
2. Operate within a single instance, which fails to transform data with the knowledge of other instances, either within or outside of the same class.

Some recent works that utilize Mixup only partially address the second limitation of conventional data augmentation since Mixup operates only between two instances. On the other hand, Manifold Mixup [26] approaches the first limitation by performing Mixup in the feature space but is limited to a simple convex combination of two samples.

We instead propose to address these two limitations simultaneously. We proposed a novel method that refines and augments image features in the abstract feature space rather than image space. To efficiently leverage the knowledge of other classes, we condense the information of each class into a small set of prototypes by performing clustering in the feature space. The image features are then refined and augmented through information propagated from prototypes of all classes. We hypothesize that this feature refinement/augmentation can further improve the feature representations, and these refined features can produce better pseudo-labels than features without the refinement. The feature refinement and augmentation are learned via a lightweight attention network for the representative prototypes and optimized end-to-end with other objectives such as classification loss. A consistency loss can naturally be applied between the prediction from the original features and the refined features to regularize the network as shown in Fig. 3b.

The final model seamlessly combines our novel feature-based augmentation with conventional image-based augmentation for consistency regularization, which applied on data augmented from both sources. Despite the simplicity of the method, we find this achieves significant performance improvement In

summary, we compare our method with other highly relevant SSL works in Table. 1.

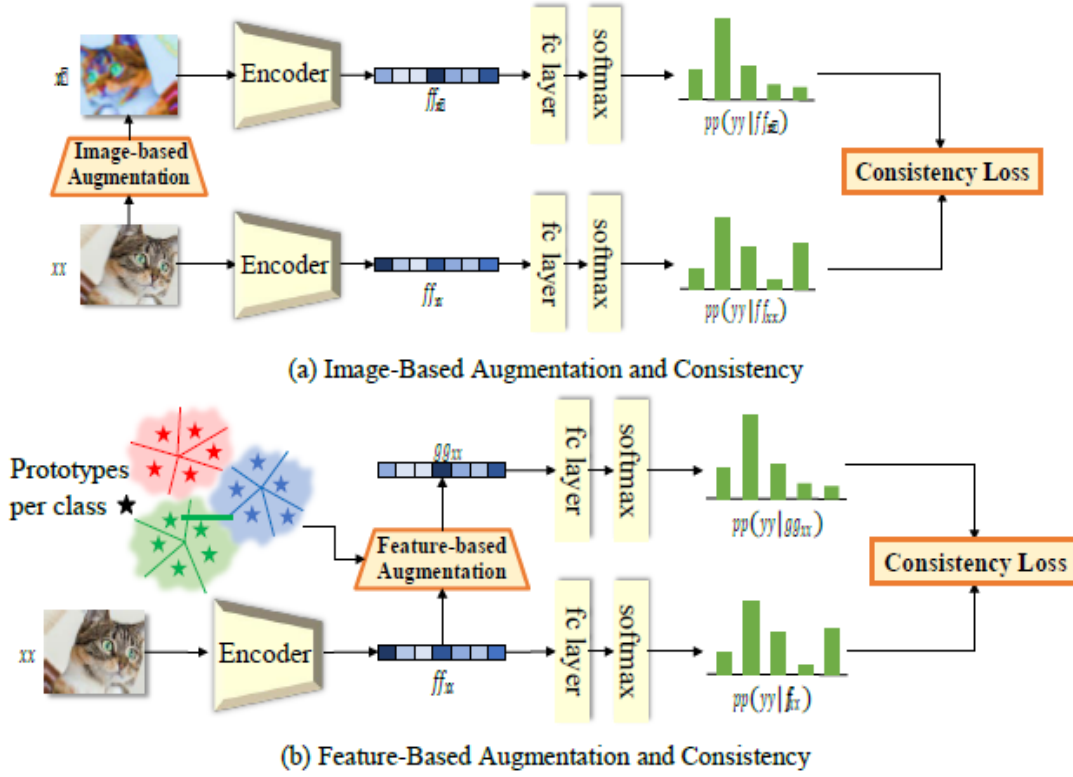


Figure 3: Consistency regularization methods are the most successful methods for semi-supervised learning. The main idea of these methods is to enforce consistency between the predictions of different transformations of an input image. (a) Image-based augmentation method generate different views of an input image via data augmentation, which are limited to operations in the image space as well as operations within a single instance or simple convex combination of two instances. (b) We propose a learned feature-based augmentation that operates in the abstract feature space. The learned feature refinement and augmentation module is capable of leveraging information from other instances, within or outside of the same class.

3.1.3 Prototype Selection.

In order to efficiently leverage the knowledge of other classes for feature refinement and augmentation, we propose to compactly represent the information of each class by clustering in the feature space. To select representative prototypes from the dataset, we propose to use K-Means clustering in the feature space to extract p_k cluster means as prototypes for *each class*. However, there are two technical challenges: (1) in an SSL setting, most images are unlabeled; (2) even if all the labels are available, it is still computationally expensive to extract features of all the images from the entire dataset before running K-Means.

To tackle these issues, as shown in Fig. 4, we collect features f_{xi} and pseudo-labels $\hat{y}_i$ already generated by the network at every iteration of the training loop, *i.e.*, no extra computation needed. In the recording loop, the pairs of pseudo label and features are detached from the computation graph and pushed into a memory bank for later usage. The prototypes are extracted by K-Means at every epoch when we go over the whole dataset. Finally, the feature refinement and augmentation module updates the prototypes with the newly extracted ones in the training loop. Even though the prototypes are extracted from the feature computed from the model a few iterations ago, as training progresses and the model gradually converges, the extracted prototypes fall on the correct cluster and are diverse enough to compactly represent the feature distribution per class. Similar idea is concurrently explored in self-supervised learning by He et al. [32].

3.1.4 Learned Feature Augmentation.

With a set of prototypes selected by the process described above, we propose a learned feature refinement and augmentation module via soft-attention [33] toward the set of selected prototypes. The proposed module refines and augments input image features in the feature space by leveraging the knowledge of prototypes, either within or outside of the same class, as shown in Fig. 5. The lightweight feature refinement and augmentation module composed of three fully connected layers is jointly optimized with other objectives and hence learns a reasonable feature-based augmentation to aid classification.

Table 1: Comparison to other SSL methods with consistency regularization.

	ReMixMatch [27]	MixMatch [28]	Mean Teacher [29]	ICT [30]	PLCB [31]	FeatMatch (Ours)
Feature-Based Augmentation	-	-	-	-	-	✓
Image-Based Augmentation	✓	✓	✓	✓	✓	✓
Temporal Ensembling	✓	✓	✓	-	-	-
Self-Supervised Loss	✓	-	-	-	-	-
Alignment of Class Distribution	✓	-	-	-	✓	-

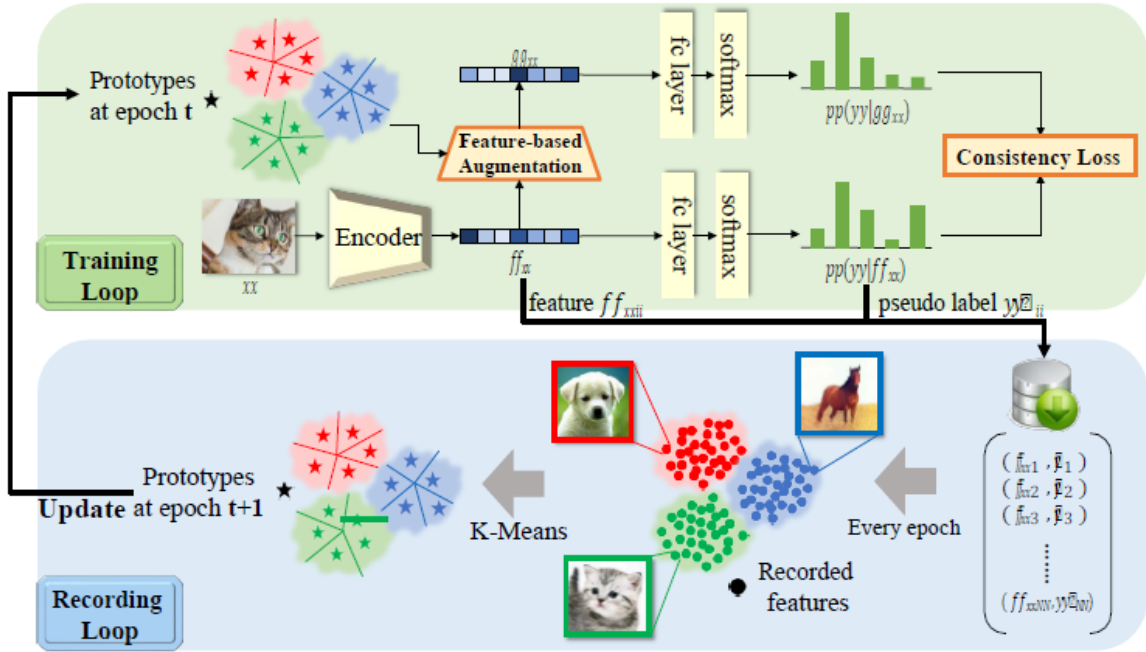


Figure 4: A prototype recording loop that runs alongside the model training loop. The image features f_{xi} as well as their pseudo labels y_i already generated at each iteration of the training loop are collected and recorded in a memory bank as (f_{xi}, y_i) pairs. Once the training loop goes over the whole dataset, the recording loop will run K-Means to extract prototypes for each class, update the prototypes for feature-based augmentation, and clear the memory bank.

Inspired by the attention mechanism [33], each input image feature *attends* to prototype features via attention weights computed by dot product similarity. The prototype features are then weighted summed by the attention weights and then fed back to the input image feature via residual connect for feature augmentation and refinement. Specifically, for an input image with extracted features f_x and the i -th prototype features $f_{p,i}$, we first project them into an embedding space by a learned function ϕ_e as $e_x = \phi_e(f_x)$ and $e_{p,i} = \phi_e(f_{p,i})$ respectively. We compute an attention weight w_i between e_x and $e_{p,i}$ as:

$$w_i = \text{softmax}(e_x^T e_{p,i}), \quad (1)$$

where $\text{softmax}(\cdot)$ normalizes the dot product similarity scores across all prototypes. The information aggregated from the prototypes and passed to the image features for feature refinement and augmentation can then be expressed as a sum of prototype features weighted by the attention weights:

$$f_a = \text{relu}(\phi_a([e_x, \sum_i w_i e_{p,i}])), \quad (2)$$

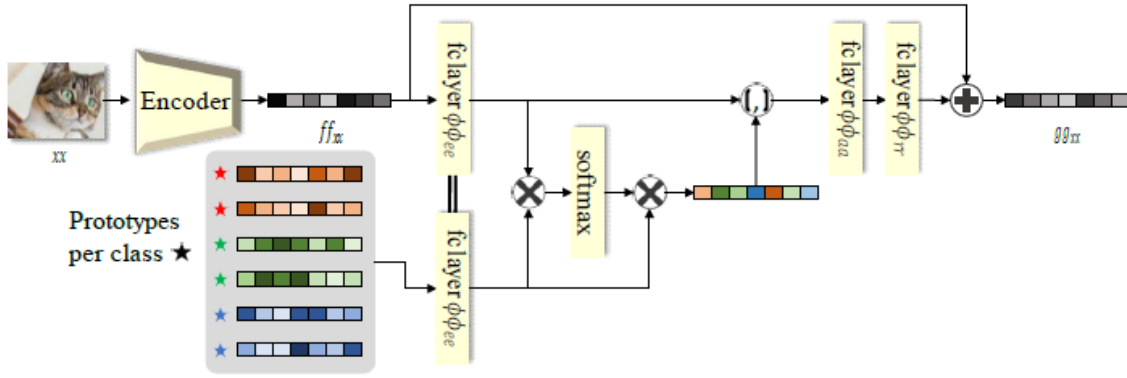


Figure 5: **Feature-Based Augmentation:** The input image features are augmented by attention using extracted prototype features (Eq.1), where the colors of * represent the classes of prototypes. The prototype features are calculated via a weighted sum using the attention weights, concatenated with the image features, and then undergo a fc layer ϕ_a (Eq.2) to produce attention features f_a . Finally, we use the attention features to refine and augment the input image features with a residual connection (Eq.3).

where ϕ_a is a learnable function, and $[\cdot, \cdot]$ is a concatenation operation along the feature dimension. Finally, the input image features f_x is refined via a residual connection as:

$$g_x = \text{relu}(f_x + \phi_r(f_a)), \quad (3)$$

where g_x are the refined features of f_x , and ϕ_r is a learnable function.

The attention mechanism described above can be trivially generalized to multi-head attention as in [33]. In practice, we use multi-head attention, instead of single head for slightly better results. For simplicity, we define the feature refinement and augmentation process $\text{AugF}(\cdot)$ described above as $g_x = \text{AugF}(f_x)$.

3.1.5 Consistency Regularization.

The learned AugF module along with the selected prototypes provides an effective method for feature refinement and augmentation in the feature space, which addresses the limitations of conventional data augmentation methods discussed previously. With the learned feature-based augmentation, we can naturally apply a consistency loss between the prediction of unaugmented image features f_x and augmented features g_x .

However, given a classifier $p = \text{Clf}(f)$, which prediction should we use as pseudo-label, $p_g = \text{Clf}(g_x)$ or $p_f = \text{Clf}(f_x)$? We have investigated this problem and find that AugF is able to *refine* the

input features for better representation, thus generating better pseudo-labels. Therefore, we compute pseudo-label p_g on the refined feature g_x by $p_g = \text{Clf}(g_x)$. The feature-based consistency loss can be computed as:

$$\mathbf{L}_{con} = \mathbf{H}(p_g, \text{Clf}(f_x)) \quad (4)$$

We can easily extend Eq. 4 to work seamlessly with traditional augmentation methods, *i.e.*, traditional data augmentation and Mixup. For simplicity, we will illustrate with only data augmentation, but Mixup can be easily adapted. Inspired by Berthelot et al. [27], we generate a weakly augmented image x and its strongly augmented copy $\hat{x}$. The pseudo-label is computed on the weakly augmented image x that undergoes feature-based augmentation and refinement for better pseudo-labels as $p_g = \text{Clf}(\text{AugF}(\text{Enc}(x)))$. We can then compute two consistency losses on the strongly augmented data $\hat{x}$, one with AugF applied and the other without:

$$\mathbf{L}_{con-g} = \mathbf{H}(p_g, \text{Clf}(\text{AugF}(\text{Enc}(\hat{x})))) \quad (5)$$

$$\mathbf{L}_{con-f} = \mathbf{H}(p_g, \text{Clf}(\text{Enc}(\hat{x}))) \quad (6)$$

Since the pseudo-label p_g is computed on the image undergoing weak data augmentation and feature-based augmentation, the regularization signal of $\mathbf{L}_{con-g}$ and $\mathbf{L}_{con-f}$ comes from both image-based augmentation and feature-based augmentation.

3.1.6 Total Loss

Consistency regularization losses: $\mathbf{L}_{con-g}$ and $\mathbf{L}_{con-f}$ in Eq. 5 and 6 are applied on unlabeled data. For labeled image x with label y , a regular classification loss can be applied:

$$\mathbf{L}_{clf} = \mathbf{H}(y, \text{Clf}(\text{AugF}(\text{Enc}(x)))) \quad (7)$$

Therefore, the total loss can be written as: $\mathbf{L}_{clf} + \lambda_g \mathbf{L}_{con-g} + \lambda_f \mathbf{L}_{con-f}$. Where λ_g and λ_f are weights for $\mathbf{L}_{con-g}$ and $\mathbf{L}_{con-f}$ losses respectively.

To summarize, our key contributions include:

- We developed a learned feature-based refinement and augmentation module to transform input image features in the abstract feature space by leveraging a small set of representative prototypes of all classes in the dataset.
- We proposed a memory bank mechanism to efficiently extract prototypes from images of the entire dataset with minimal extra computations.
- We demonstrated thorough results across four standard SSL datasets and also propose a realistic setting where the unlabeled data partially come from domains shifted from the target labeled set.
- We performed in-depth analysis of the prototype representations extracted and used for each instance, as well as what transformations the proposed feature-based refinement and augmentation module learns.

3.2 Leveraging Manifold Structure via Transport Operators

Throughout the project, we have built upon a class of manifold models [34, 35, 36, 37] that learn the basis of a Lie algebra [38]. Once passed through the matrix exponential, a point in this algebra provides a linear representation of a generator of a Lie group, denoting a displacement along a manifold. We describe this basis as a dictionary of M transport operators $\Psi_m \in \mathbb{R}^{D \times D}$. Given a point on the manifold $\mathbf{x}_i \in \mathbb{R}^D$, the transport operators model an infinitesimal transformation as a weighted sum of the learned

$$\dot{\mathbf{x}}_i = \left(\sum_{m=1}^M \Psi_m c_m \right) \mathbf{x}_i, \quad (8)$$

dictionary

where $\mathbf{c} \in \mathbb{R}^M$ is often modeled with a sparse, factorial prior $p(\mathbf{c}) = \prod_{m=1}^M p(c_m)$ [36, 39]. Given a batch of point pairs sampled nearby on the manifold $(\mathbf{x}_i, \mathbf{x}'_i)$, the dictionary of operators can be learned through an alternating minimization scheme where coefficients $\mathbf{c}$ are first inferred with fixed operators (e.g., using a sparse inference procedure [39]), followed by a gradient step with fixed coefficients to minimize the loss with respect to the operators [36]:

$$\mathcal{L}_m(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{c}) = \|\mathbf{x}'_i - \mathbf{T}_\Psi(\mathbf{c})\mathbf{x}_i\|_2^2, \quad (9)$$

$$\mathbf{T}_\Psi(\mathbf{c}) = \expm\left(\sum_{m=1}^M \Psi_m c_m\right). \quad (10)$$

Learning these operators directly in the data space is often impractical due to dimensionality. As such, later work has extended this model to learning the operators after applying non-linear dimensionality reduction, $\mathbf{z}_i = f_\theta(\mathbf{x}_i) \in \mathbb{R}^d$, where $d \ll D$ [40, 41, 42, 18, 43]. Note that in this setting, the operators describe transformations between latent points and have dimensionality $\mathbb{R}^{d \times d}$. Most of these systems rely on an auto-encoding objective to prevent collapse to a trivial manifold (e.g., projecting every input data to a constant value allows one to learn all operators as identity), but are often limited by the fidelity of the decoder network in their ability to apply manifold transformations.

Furthermore, performing exact sparse inference $\mathbf{c}$ at each training step becomes overly prohibitive as dimensionality d increases. Although fast first-order optimization algorithms have been proposed [44, 18], such methods are not practical in deep learning systems that require upwards of hundreds of thousands of training iterations. Hence, alternative inference methods are necessary when incorporating the operators into high-dimensional feature spaces (e.g., the penultimate layer of a ResNet backbone [25]).

3.2.1 Variational Autoencoder with Learned Latent Space

Our Variational Autoencoder with a Learned Latent Space (VAELLS) [41] model is a VAE that incorporates the transport operator model into the latent space to enable learned manifold structure from the data and appropriate prior in latent space.

For deriving VAELLS, we started with the expanded Evidence Lower Bound (ELBO) from [45]. The ELBO is commonly used as a training objective for a VAE because it is a lower bound on the marginal likelihood of the data which is often intractable:

$$\mathcal{L}(\mathbf{x}_i) = \mathbb{E}_{\mathbf{z}_i \sim q_\phi(\mathbf{z}_i | \mathbf{x}_i)} [\log p_\theta(\mathbf{x}_i | \mathbf{z}_i) + \log p_\theta(\mathbf{z}_i) - \log q_\phi(\mathbf{z}_i | \mathbf{x}_i)] \quad (11)$$

For the likelihood $p_\theta(\mathbf{x} | \mathbf{z})$, we followed prior work and chose an isotropic normal distribution with mean parameterized by neural network $g_\theta(\mathbf{z})$ and fixed variance σ^2 , which is a family of likelihood distributions that has worked well in practice.

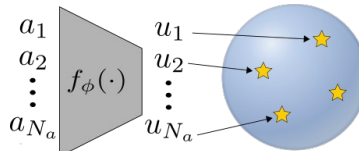


Figure 6: Visualization of the encoding of anchor points into the VAELLS latent space.

Our contributions lie in the selection of the variational posterior and prior distribution in the latent space, both of which are built off of the transport operator model. Our variational posterior $q_\phi(\mathbf{z} | \mathbf{x})$ measures the probability of vector $\mathbf{z}$ lying on the manifold in the local neighborhood of the latent encoding of $\mathbf{x}$ and our prior $p_\theta(\mathbf{z})$ is constructed directly from the underlying data manifold. To gain intuition about this prior, imagine a set of N_a anchor points in the data space that correspond to key

locations on the manifold, either sampled to uniformly cover the manifold or selected manually by a practitioner (e.g., selecting several anchors per data class). The data manifold structure is represented by a combination of these anchor points and the learned transport operators that can extrapolate the manifold structure in the latent neighborhood of each of them. Fig. 6 shows a set of anchor points encoded into the latent space to represent the scaffold off of which the manifold structure is built. The prior is defined by the same probabilistic manifold model used in the variational posterior, but starting at each anchor point $\mathbf{a}_i$ rather than $\mathbf{x}$. The overall prior density for $\mathbf{z}_i$ is then defined as

$$p_\theta(\mathbf{z}_i) = \frac{1}{N_a} \sum_{j=1}^{N_a} q_\phi(\mathbf{z}_i | \mathbf{a}_j). \quad (12)$$

We used these newly defined variational posterior and prior distributions to specify the objective function for the VAELLS model. We tested this model on both rotated MNIST data, where we have an intuitive understanding of what the rotational transformations should look like, and naturally varying MNIST. Fig. 7(a-b) shows the decoded outputs of latent vectors sampled from the posterior of the network trained on rotated MNIST digits. In each example, the center image (enclosed in a green box) is the decoded version of the input test sample. The images surrounding each center are decoded outputs of latent posterior samples. The key result is that the VAELLS sampling procedure using the learned transport operator leads to latent space transformations that correspond to rotations in decoded outputs. This verifies that the transport operator corresponds to movement on a learned rotated digit manifold.

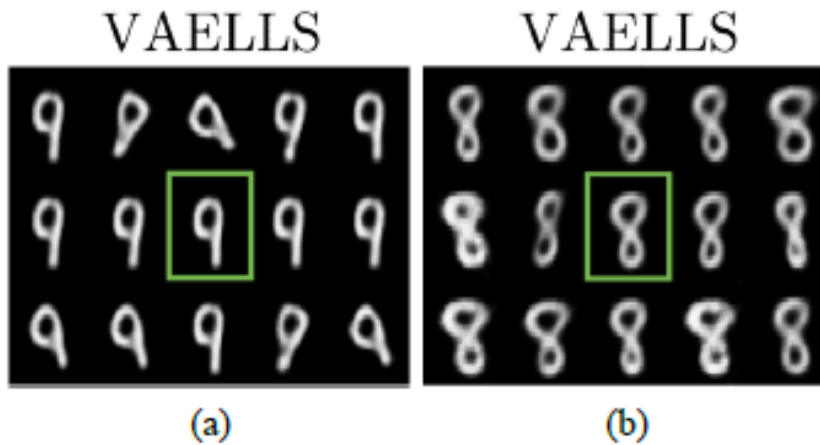


Figure 7: Examples of images decoded from latent vectors sampled from the posterior of models trained on MNIST digits. In each example, the center digit (in the green box) is the encoded digit input and the surrounding digits are from the sampled latent vectors. (a-b) Show samples from VAELLS trained on rotated MNIST digits. This shows that sampling in the VAELLS latent space results in rotations in the sampled outputs.

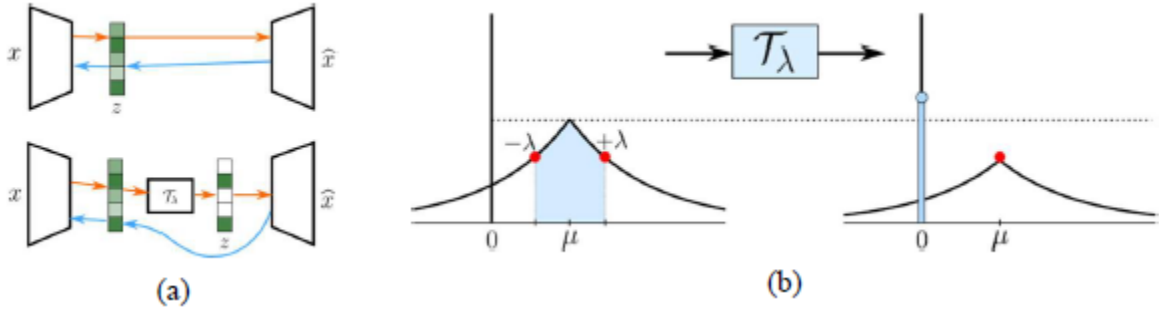


Figure 8: Our proposed approach to variational sparse coding. (a) Standard amortized variational inference (top) results in reparameterized latent features that are not sparse. Our approach (bottom) incorporates a straight-through estimator of the soft-threshold to incorporate exact sparsity. In the backwards pass, we avoid numerical instability by passing the gradient directly from the generator to the inference network. Orange arrows denote a forward pass and blue arrows denote a backwards pass. (b) A visualization of our shifted soft-threshold. We analytically show that this reparameterization leads to a spike-and-slab distribution, explicitly modeling the degree of sparsity in our latent features.

3.2.2 Variational Sparse Coding

Training of the transport operator model requires solving a sparse inference routine at each training step. To infer sparse coefficients, previous work has relied on an iterative procedure that does not scale favorably to higher dimensions [44]. To scale the transport operator model to a feature space of interest, we have developed methods for variational sparse coding [46]. This technique uses a similar method as VAELLS, except in this case the learned latent space is the coefficients of the transport operators $\mathbf{c}$ rather than the activations of a DNN.

Given encoded features of training images $[\mathbf{z}_1, \dots, \mathbf{z}_N]^T \in \mathbb{R}^{N \times d}$, amortized variational inference (AVI) can be applied to train an inference network $q_\phi(\mathbf{c} | \mathbf{z}_i, \mathbf{z}_i^1)$ to learn a distribution over operator coefficients $\mathbf{c}_i \in \mathbb{R}^M$. This distribution can be applied in various machine learning tasks, such learning of the transport operators in Equation 10. To do this, one may employ the variational lower bound (ELBO) as in Equation 11.

Training under this bound requires the practitioner to select a prior $p(\mathbf{c})$ and a distribution family for the posterior $q_\phi(\mathbf{c} | \mathbf{z}_i, \mathbf{z}_i^1)$. We follow the common assumption of independence for both $p(\mathbf{c}) = \prod_{i=1}^M p(c_i)$, $q_\phi(\mathbf{c} | \mathbf{z}_i, \mathbf{z}_i^1) = \prod_{i=1}^M q_\phi(c_i | \mathbf{z}_i, \mathbf{z}_i^1)$.

To parameterize $p(c_i)$, we propose an approach that applies shifted soft-thresholding of samples drawn from a Laplacian distribution. Let $p(s) = \text{Laplace}(\mu, b)$ be a Laplacian pdf with shift μ and scale b . We can draw a sample $s \sim p(s)$ and apply the shifted soft-threshold function:

$$\mathcal{T}(s; \lambda, \mu) = \text{sign}(s - \mu) \max(|s - \mu| - \lambda, 0) + \mathbb{I}[|s - \mu| > \lambda] \mu. \quad (13)$$

For ease of notation, we will use the shorthand $\tau_\lambda(s)$ to refer to $\mathcal{T}(s; \lambda, \mu)$. Intuitively, this function is a soft-threshold around μ , where points mapped to μ are set exactly to zero. This is visualized in Figure 8b. When $\mu = 0$, this function is precisely the soft-threshold function [47].

We apply this shifted soft-threshold to form our final training objective with a reweighted KL term [48]. To handle numerical instability issues, we apply a straight-through estimator [49, 50] of the shifted soft-threshold. This means that we apply τ_λ normally in the forward pass, but skip its differentiation in the backwards pass, as denoted in Figure 8a. Following the notation of [50], we denote by $\text{sg}[\cdot]$ the

stopgradient operator (identity in the forward pass with partial derivatives equal to zero). Our final training objective is thus:

$$\mathbf{c}_i = \mathbf{s}_i + \mathcal{T}_{\lambda_i}(\text{sg}[\mathbf{s}_i]) - \text{sg}[\mathbf{s}_i], \quad (14)$$

$$\mathcal{L} = \mathcal{L}_m + \beta D_{KL}(q_\phi(\mathbf{s} | \mathbf{z}_i, \mathbf{z}_i) || p_\theta(\mathbf{s} | \mathbf{z}_i)) \quad (15)$$

Furthermore, in cases where we wish to sample feature augmentations, we apply the conditional VAE (CVAE) framework with a learned prior $p_\theta(\mathbf{s} | \mathbf{x}_i)$ (written henceforth as $p_\theta(\mathbf{c} | \mathbf{x}_i)$ to indicate thresholded samples).

3.2.3 Manifold Contrastive Learning

Given the variational sparse coding framework, one can learn a manifold describing the natural variations present in a dataset within the representations of a DNN (while also regularizing such representations to match the data manifold). We now present methods to use augmentations from these operators to enrich such representations during contrastive pretraining. We then demonstrate how these augmentations can be used in consistency regularization [51] in downstream tasks like semi-supervised learning.

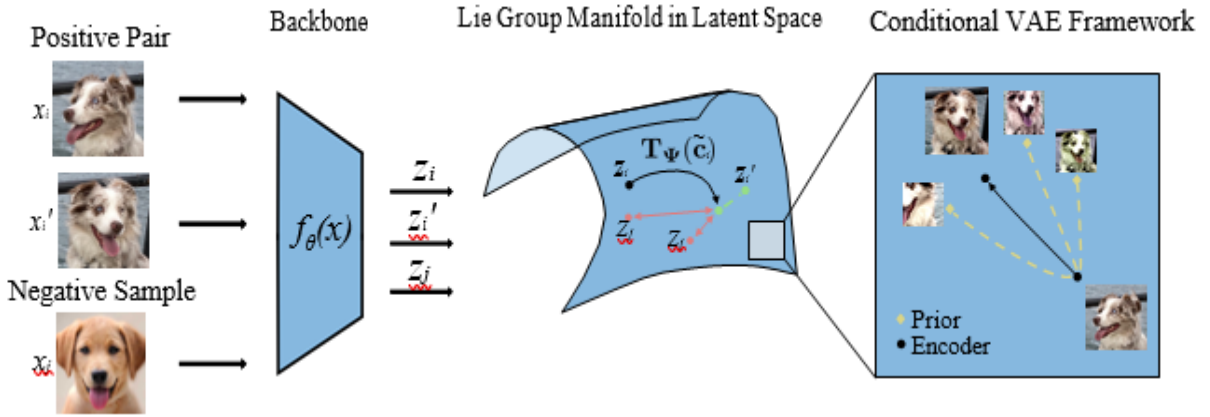


Figure 9: Proposed ManifoldCLR system for incorporating Lie group operators into the feature space of a contrastive learner. Given encoded features from a positive pair, manifold feature augmentations $\mathbf{T}_\Psi(\tilde{\mathbf{c}})$ are applied before computing $\mathbf{z}_j$ denote the set of negative pairs in the feature space. We apply the conditional VAE framework to learn coefficient distributions for manifold displacement between two points (encoder network $q_\phi(\mathbf{c} | \mathbf{z}_i, \mathbf{z}_i)$) and identity-preserving augmentations from a single point (prior network $p_\theta(\mathbf{c} | \mathbf{z}_i)$).

With encoded features $(\mathbf{z}_i, \mathbf{z}_i')$ corresponding to two views of an instance $(\mathbf{x}_i, \mathbf{x}_i')$ (e.g., two augmentations applied to an image), one can sample coefficients from the encoder network $\mathbf{c} \sim q_\phi(\mathbf{c} | \mathbf{z}_i, \mathbf{z}_i')$ to minimize the manifold loss $\mathcal{L}_m$ and learn operators Ψ_m . Furthermore, one can directly sample transformations along the manifold to enrich the number of views that are applied during instance discrimination. To do this, one may sample coefficients $\mathbf{c} \sim p_\theta(\mathbf{c} | \mathbf{z}_i)$ from the learned prior network, applying manifold feature augmentations $\mathbf{z}_i = \mathbf{T}_\Psi(\mathbf{c})$, and contrasting the augmentation point $(\mathbf{z}_i, \mathbf{z}_i')$.

This allows one to both enrich the novel views of an instance, as well as learn coefficient statistics in our prior network that preserve identity. We hypothesize that this latter outcome is a result of negative samples in the contrastive loss, which encourages prior augmentations to remain on the object manifold, visualized on the right in Figure 9.

Thence, the resultant learning objective can be written as:

$$\min_{\Psi, \theta, \phi} \mathcal{L}_{ctt} + \lambda \mathcal{L}_m + \beta \mathcal{L}_{kl}. \quad (16)$$

$$\mathcal{L}_{ctt} = -\log \frac{\exp\left(-\|h_{\theta}(\mathbf{z}_i) - h_{\theta}(\mathbf{z}'_i)\|_2^2/\tau\right)}{\sum_{j \in \mathcal{N}} \exp\left(-\|h_{\theta}(\mathbf{z}_i) - h_{\theta}(\mathbf{z}_j)\|_2^2/\tau\right) + \exp\left(-\|h_{\theta}(\mathbf{z}_i) - h_{\theta}(\mathbf{z}'_i)\|_2^2/\tau\right)}, \quad (17)$$

where Ψ are manifold operator parameters, θ are DNN backbone and prior network parameters, ϕ are coefficient network parameters, and λ and β are loss scaling hyper-parameters. In $_{ctt}$, denotes the set of negative pairs, τ is a temperature hyper-parameter, and h_{θ} is an additional normalized, non-linear projection from the backbone features. All terms and parameters are learned concurrently, end-to-end from random weight initialization. We refer to this system as ManifoldCLR and show that incorporating manifold structure leads to favorable self-supervised properties such as linear separability (see [52] for more details).

3.3 Semi-Supervised Clustering and Retrieval for Zero-Shot Learning

In this work, we detail our work on zero-shot classification leveraging semi-supervised clustering and retrieval [19]. Specifically, we tackle the setting of Novel Category Discovery (NCD) [53], which is a problem highly relevant to the LwLL zero-shot evaluation. The problem involves discovering new categories in unlabeled data, by leveraging pre-training or auxiliary training. Recently, Generalized Category Discovery (GCD) [54] was formalized to make the setting more realistic, where the goal is to jointly discover unknown categories and classify known categories within the unlabeled data. This setting is related to semi-supervised learning but does not assume we know all of the classes in the data [55, 56].

The state-of-the-art methods for this setting utilize self-supervised image pre-training (e.g. DINO [57]) as auxiliary information used to encode the images, after which simple clustering is performed [54]. However, even though self-supervised feature learning can show some out-of-domain generalization [58, 59], it is still a difficult challenge as the features may not be relevant to entirely new categories.

Here, we posit that a key missing element to improve such generalization is a more effective encoding of the semantic relationships between object categories. Recently, aligned multi-modal (vision and language) models have shown to give a remarkable boost in generalization of visual learning, especially when scaled up[60, 61, 62, 63]. These models are learned via alignment of visual and language embeddings through large-scale contrastive training of paired image-text data [63]. Such methods have demonstrated a potential for learning open-world visual concepts, since the textual alignment forces visual features to be nearby similar concepts, and hence new categories can be well-placed in the feature space by the visual encoder.

Given the strong zero-shot results of such models, we therefore propose to first *replace* the uni-modal image encoder with one trained in a multi-modal fashion (CLIP [63]). By itself, this simple modification yields significant performance gains, beating all of the current state of art. Hence, this setting can serve as a simple, but extremely strong, baseline.

However, in just replacing the visual encoder, we discard the text branch of the multi-modal model and thus fail to fully leverage the joint vision-and-language (VL) embedding and its zero-shot generalizability. Furthermore, despite significant gains, the visual encoder from a multi-modal model can still perform poorly when the visual concepts are not well-represented in their training data and are somewhat out-of-distribution.

Here, we propose to *augment* the visual embeddings with *retrieved* textual information. This allows us to better leverage the joint VL embedding and the text encoder as well as provide the ability to extend the contextual knowledge available for clustering unknown and potentially out-of-domain categories and images. Specifically, inspired by prior image captioning works [20], given an image, we retrieve the top- k most relevant text from a large text corpus [64, 65] (which could be from the multi-modal training set itself). We specifically use the alignment between CLIP’s visual encoding (of the image) and textual

encoding (pre-indexed for the text corpus). Our key hypothesis is that such pieces of text, and their encodings, can provide valuable contextual clues for clustering unseen categories. The retrieved top- k text are encoded by CLIP’s text encoder, are mean-pooled, and then concatenated with the CLIP’s visual encoding as the final multi-modal representation for clustering.

We show that our proposed method substantially outperforms the established state of the art across a number of datasets. We specifically expand the set of datasets to include out-of-domain data, DomainNet (a domain adaptation dataset) and Flowers102, a generic image recognition dataset. We perform extensive analysis of what corpus to retrieve from, how much to retrieve, and how to combine (or pool) the resulting embeddings. Crucially, we demonstrate in our ablation studies that the **combination of our two ideas** (using CLIP and retrieving contextual information) is needed to yield strong state of art results. This is because combined clustering of *aligned* embeddings is significantly more effective than clustering individual image and textual embeddings.

In this section, we first introduce the notations and definitions of GCD [54]. Then, we explain how to use CLIP in GCD and introduce our method to tackle this task.

3.3.1 Problem Setup of GCD

As formalized in [54], dataset $\mathcal{D}$ consists of two parts, labeled dataset $\mathcal{D}_L = \{(\mathbf{x}_i, y_i)\}^N \in \mathbf{X} \times \mathbf{Y}_L$ and unlabeled dataset $\mathcal{D}_U = \{(\mathbf{x}_i, \mathbf{y}_i)\}^M \in \mathbf{X} \times \mathbf{Y}_U$, where $\mathbf{Y}_L \subset \mathbf{Y}_U$ which is distinct from NCD [66] that assumes $\mathbf{Y}_L \cap \mathbf{Y}_U = \emptyset$. The goal is to learn a model to group the instances in $\mathcal{U}$ based on information from $\mathcal{L}$. Taking advantage of the recent advances in vision transformers and their remarkable performance in various visual recognition tasks specifically for self-supervised representation learning [57], Vaze *et al.* [54] devise a two-stage training pipeline for the GCD task. First, for representation learning, they jointly fine-tune the representation by performing supervised contrastive learning on the labeled data and unsupervised contrastive learning on all the data.

Let $\mathbf{x}_i$ and $\mathbf{x}_i^l$ be two views with random augmentations of the same image in a mini-batch $\mathcal{B}$. The unsupervised contrastive loss is stated as:

$$\mathcal{L}_i^s = -\frac{1}{|\mathcal{N}(i)|} \sum_{q \in \mathcal{N}(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_q / \tau)}{\sum_n \mathbb{1}_{[n \neq i]} \exp(\mathbf{z}_i \cdot \mathbf{z}_n / \tau)}$$

where $\mathbf{z}_i = h(\mathcal{F}(\mathbf{x}_i))$ is the feature extracted by a backbone $\mathcal{F}(\cdot)$ on the input image $\mathbf{x}_i$ and projected to do the embedding space via a projection head $h(\cdot)$, $\mathbf{z}_i'$ is the feature from another view of the input image $\mathbf{x}_i^l$.

The supervised contrastive loss is stated as

$$\mathcal{L}_i^u = -\log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_i' / \tau)}{\sum_n \mathbb{1}_{[n \neq i]} \exp(\mathbf{z}_i \cdot \mathbf{z}_n' / \tau)}$$

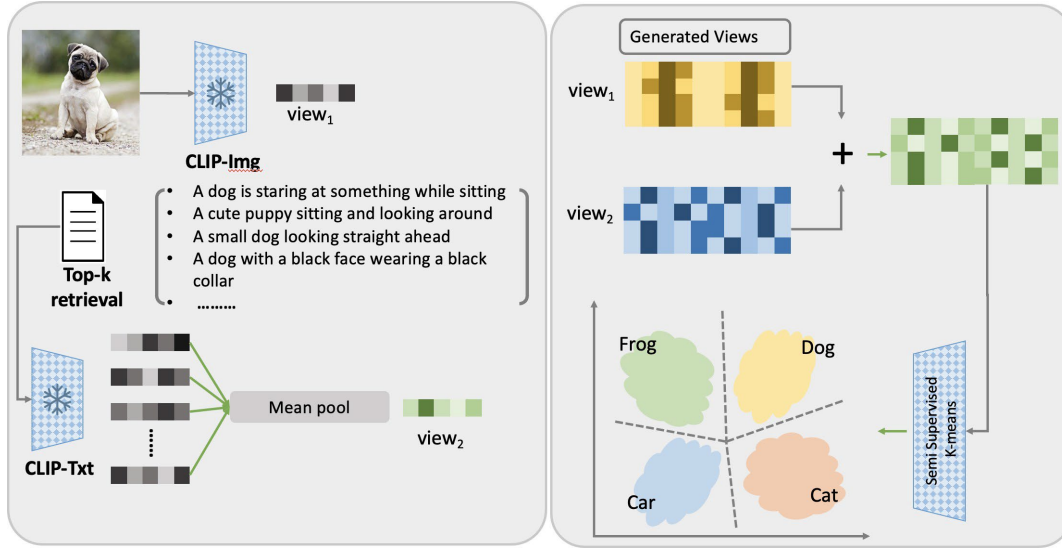


Figure 10: Model Architecture. In stage I (left), we propose a cross-modal retrieval module to retrieve a set of contextual text descriptions for the labeled and unlabeled data, generate a view from pooled sentence embedding as complementary information for clustering. In Stage II (right), we concatenate the image view and the text view and use semi-supervised k -means clustering to group seen and unseen classes.

Where $\mathcal{N}(i)$ denotes the indices of other images having the same label as $\mathbf{x}_i$ in the mini-batch $\mathcal{B}$. Then, the final objective is the combination of the two losses:

$$\mathcal{L}^l = (1 - \lambda) \sum_{i \in \mathcal{B}_L \cup \mathcal{B}_U} \mathcal{L}_i^u + \lambda \sum_{i \in \mathcal{B}_L} \mathcal{L}_i^s$$

where λ is a weight factor and $\mathcal{B}_L, \mathcal{B}_U$ are mini-batches for labeled and unlabeled images respectively. For label assignments, a semi-supervised k -means is proposed, where the overall procedure is similar to k -means [67] However, there is a significant distinction in that semi-supervised k -means takes into account the labeled data in $\mathcal{D}_L$ during the computation of cluster assignment in each step. This means that the samples with labels will always be assigned to the correct cluster, irrespective of their distance to the nearest cluster centroids.

3.3.2 Our Approach

By combining both textual and visual information, language-image models can achieve improved performance in a wide range of tasks, so we propose to leverage CLIP’s zero-shot ability and multi-modal aligned encoders for this setting, and then propose a retrieval-based augmentation.

Using CLIP in General Category Discovery We propose to tackle the GCD task by leveraging the cross-modal joint embedding from CLIP [63]. The CLIP model has two branches: the image branch CLIP-Image and the text branch CLIP-Text that encode image and text into a global feature representation, respectively. CLIP is trained on large-scale image and text pairs *s.t.* paired image and text are pushed together in the embedding space while unpaired ones are pulled apart. Please refer to Figure 10 for the overall architecture. To improve the representation of our data, specifically for both labelled and unlabelled data, we refine the representation by combining two techniques: supervised contrastive learning on the labelled data and unsupervised contrastive learning on all data. We do this by finetuning the representation on our target data simultaneously. CLIP learns image representation by contrasting them with the representations of text description of the image, such as “A photo of a class name”. The text description is called *prompt*, and its design is vital in enhancing CLIP’s performance. However, the unlabeled data contains unseen categories, and we do not have a list of them to use for

prompts. As a result, inspired by recent works in image captioning [20], for the labeled and unlabeled set, we propose to mine a set of text descriptions providing complementary information to the input image from a text corpus. The key hypothesis is that such contextual information, provided as additional “views” of the image, can significantly aid in clustering. To that end, we propose to generate text descriptions for an image as shown in Figure 11 containing details and information of the input image to be mapped into the feature space. Training a separate captioning model to generate text descriptions might be expensive and nontrivial, so for each labeled and unlabeled image, we retrieve the top- k most relevant descriptions from a text corpus, turning this problem into a cross-model retrieval one, which we describe as follows.

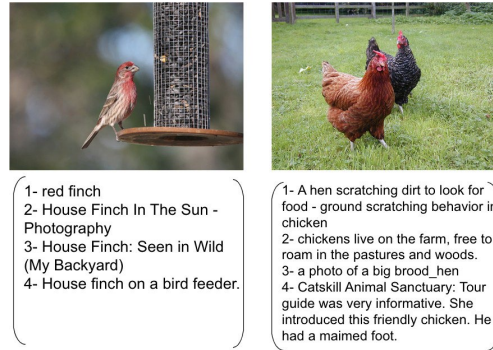


Figure 11: A sample of retrieved top-4 most relevant text descriptions from Conceptual Captions (3M) for an image from ImageNet dataset.

Description database: The description database is an organized collection of textual descriptions relevant to an image, and we select the top- k most pertinent ones. Several options can be used, and we show results across annotations from databases such as Conceptual Captions (3M) [68], Conceptual Captions (12M) [69], MS COCO [70], and LION [71]. We don’t perform any rigorous processing and simply collect all the captions.

Text description retrieval: Given a query of an image, the goal is to retrieve the top- k most relevant text descriptions from the description database. To this end, we propose to exploit the cross-modal joint embedding from CLIP [63] for this cross-modal retrieval task. Specifically, we use CLIP-Text to encode all the descriptions in the description database as the search key. The image is encoded by CLIP-Image into a query. We then search in the description database for the text descriptions with the top- k highest cosine similarity scores. Some examples of the top-4 results are shown in Figure 11.

Multi-view generation for clustering: The general approach for our feature vector extraction and view generation framework is illustrated in Figure 10 (Stage I). Given an image and a set of text descriptions, an image view (feature vector) is generated by encoding it using the CLIP image encoder, then using the CLIP text encoder, we encode the set of text descriptions, pool embeddings, and generate a view (sentence embedding) using mean pooling. Finally, the feature vectors of the image and text (views) are concatenated and projected into CLIP latent space, and clustering is performed directly in it.

Label assignment with semi-supervised k-means clustering: Given the image view and the text view we concatenate the feature vectors and apply semi-supervised k -means clustering following [54] to group the unlabeled data into seen and unseen classes. The semi-supervised k -means is a transformation of the traditional k -means method into a constraint-based algorithm, where the number of clusters k is assumed known. This will involve requiring that the D_L data instances are assigned to their appropriate clusters based on their ground-truth labels. The first set of centroids $|Y_L|$ for D_L in semi-supervised k -means are obtained using actual class labels. The second set of centroids for the additional number of new classes $|Y_U \setminus Y_L|$ are obtained from D_U using k -means++ [72], but only within the constraint of D_L centroids. During the process of updating and assigning centroids, instances from the same class in D_L are always grouped together, whereas instances in D_U can be assigned to any cluster based on their

distance to various centroids. After the algorithm converges, each instance in D_U can be given a cluster label.

3.4 Retrieval for Low-Labeled Image Captioning

In this section, we detail our work to support low-labeled image captioning by leveraging both retrieval and effective architectures to fuse the resulting multiple views of data [20, 21]. We specifically focus on an important research problem that has remained unanswered: *how should we leverage and ensemble the information from heterogeneous views?* One naive approach [73, 74, 75, 76] is to train $|V|$ (number of views) models for each view and average the predictions from these models. Clearly, this method is computationally expensive and completely ignores the difference in the usefulness of each view by ensembling the views with equal weights. Another method [77, 78, 3] is to concatenate the views, each represented by a sequence of tokens, into one single long sequence and feed the concatenated sequence to the transformer model [33], hoping that the transformer is capable of learning a good strategy to leverage and ensemble the views internally. Although transformer models may have the capacity and flexibility to leverage the information from all views jointly by a transformer encoder, and ensemble the views by learned attention, it has also been shown in the vision transformer community [79] that transformer models are notoriously data-hungry. On the popular MS-COCO [80] image captioning benchmark with less than 0.6 M training examples, we posit that we should introduce some structural priors to aid the learning of the transformer model [81, 82].

To solve this, we leverage a probabilistic interpretation of image captioning models to derive appropriate architectural biases. Specifically, multi-view hierarchical ensembling (illustrated in Fig. 13) emerges naturally in the derivation. Grounded on the derived probability model, at the step of predicting the next word, the model first generates heterogeneous *views* of the input image by different pre-trained models. These views are then encoded *separately*. Given the separately encoded views and previously generated words (e.g. “Black bags sitting on top of a” in Fig. 13), the model learns to ensemble the views, putting more attention on views that provide more helpful information for generating the next word. Finally, the next word is generated from the ensembled view. Our proposed strategy is different from previous works in that the views are encoded and ensembled hierarchically: the model first encodes information *within* the views, and then ensembles *across* the encoded views conditioned on previously generated words.

Since the views are encoded separately in the proposed multi-view hierarchical ensembling and each view contains partial information of the input image, we also propose to regard each view as an augmentation of the input image. To the best of our knowledge, we are the first to propose the idea of generating and regarding heterogeneous multi-modal views, obtained only from the input image, as augmentation in vision and language. The key difference here is that we assume each view contain partial information of the input image, and is treated individually and encoded separately, rather than concatenated and encoded jointly. Data augmentation increases data diversity and has achieved great success in low-label settings such as semi-supervised learning [28, 51, 13] and self-supervised learning [83, 84, 85, 86]. This is particularly important for training data-hungry transformer models. Furthermore, by regarding each view as an augmentation, we can naturally incorporate contrastive learning to help the representation learning of heterogeneous views. In contrastive learning, given different views (random augmentations) of input images, the views from the same image are pulled together in an embedding space while those from different images are pushed apart. It has been shown [85, 87, 88] that contrastive loss is beneficial in low-label settings.

3.4.1 Method

As mentioned, current methods for image captioning typically adopt a single-view formulation, either by encoding the input image into only a single view or by concatenating multiple views into one single view. In Sec. 3.4.2, we derive a probabilistic model of image captioning and show that multi-view ensembling emerges naturally in the derivation. We compare these two formulations and point out the potential issues of the single-view formulation. In Sec. 3.4.3, we describe optimization challenges of multi-view ensembling and propose an approximation to make the problem tractable. We then propose a concrete implementation of multi-view ensembling with a transformer encoder-decoder architecture in Sec. 3.4.4. Lastly, in Sec. 3.4.5, in addition to the standard loss for word prediction, we propose for the views to be regarded as augmentations of the input image, and thus we introduce a contrastive loss [83, 84] to regularize the representation of the views.

3.4.2 Preliminaries

Current auto-regressive image captioning models that generate text descriptions $\mathbf{y}$ from a give image $\mathbf{x}$ can be formulated as follows:

$$p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|\mathbf{x}, y_{1:t-1}) = \prod_i \sum_j p(\mathbf{v}_j|\mathbf{x}, y_{1:t-1}) p(y_i|\mathbf{v}_j, \mathbf{x}, y_{1:t-1}) \quad (18)$$

where $\mathbf{v}_j$ are different *views* of the input image $\mathbf{x}$ that encode relevant information from $\mathbf{x}$ for image captioning. The most common approach is to encode $\mathbf{x}$ into a set of objects by a pre-trained object detector. Other options include object tags, global image features, and image grid features. Recent work proposed cross-modal textual descriptions for the input image [3]. Specifically, they use CLIP to retrieve the top- k most relevant text descriptions of sub-regions of the input image.

In the right-most term of Eq. 18, we can see that the model predicts the next token y_t from different views separately as $p(y_t|\mathbf{v}_j, \cdot)$, and then ensembles the predictions by the weights $p(\mathbf{v}_j|\cdot)$. Nevertheless, in Eq. 19, most current works deviate from this paradigm either by (1) assuming there is only one view for describing the image, or by (2) concatenating all views into a long sequence as a single view.

For (1), since the view is generated by a frozen pre-trained model such as an object detector, some important information such as scene information and object relationships may not have been properly encoded. Thus, it is important to encode $\mathbf{x}$ from different perspectives and generate the final caption by considering these views altogether. For (2), suppose we use a transformer model; although it is possible to let the transformer model learn how to combine the views internally, explicitly exposing the structural prior of ensembling the predictions made separately from each view in Eq. 18 is beneficial to the learning of the model. Furthermore, in terms of computational complexity, suppose we have v views and $s = \max_j |S_j|$ represents the longest sequence length among the views, concatenating the views into a long sequence as the input to the transformer model leads to $O((v s)^2) = O(v^2 s^2)$. On the other hand, by ensembling the predictions made separately from each view in Eq. 18, the model has lower complexity of $O(v s^2)$, reducing the complexity by a factor of v .

3.4.3 Multi-view Hierarchical Ensembling

In this section, we aim to propose a tractable learning strategy for multi-view hierarchical ensembling in Eq. 18. If we naively conduct maximum likelihood optimization of Eq. 18, the model could get trapped in a local optimum quickly. Specifically, to optimize Eq. 18, the model is optimized to increase the value of $p(\mathbf{v}_j)$ for those $p(y_t|\mathbf{v}_j, \cdot)$ with higher values. Likewise, the model is optimized to increase the value of $p(y_t|\mathbf{v}_j, \cdot)$ for those $p(\mathbf{v}_j|\cdot)$ with higher values. Eventually, the model quickly gets trapped in a local optimum.

To overcome this optimization challenge, we propose to ensemble in the features space rather than in the

probability space as in Eq. 18:

$$p(y|x) = \prod_t \sum_j p(v_j|x, y_{1:t-1})p(y_t|v_j, y_{1:t-1}) \equiv \prod_t \sum_j \alpha_j f(v_j) \quad (20)$$

$$\simeq \prod_t f\left(\sum_j \beta_j v_j\right) = \prod_t p\left(y_t \mid y_{1:t-1}, \sum_j \beta_j v_j\right) \quad (21)$$

In Eq. 20, we rewrite Eq. 18 as $f(\cdot)$ combined by a sum weighted by α . Assume $f(\cdot)$ has sufficient capacity, we approximate Eq. 20 by learning a new set of weights β and turning the weighted sum of $f(\cdot)$ into $f(\cdot)$ of weighted sum in Eq. 21. Intuitively, in Eq. 21, the views are first processed *separately* into v_j , and the model then learns to ensemble the information *across* the processed views, and finally uses the ensembled view to generate the next word. If we look at the gradient of the i -th word

$$\frac{\partial f(\Sigma)}{\partial \theta} = \frac{\partial f(\Sigma)}{\partial \Sigma} \frac{\partial \Sigma}{\partial \theta} = f'(\Sigma) \left(\sum_j \frac{\partial \beta_j}{\partial \theta} v_j + \beta_j \frac{\partial v_j}{\partial \theta} \right) = \sum_j f'(\Sigma) \beta'_j v_j + f'(\Sigma) \beta_j v'_j, \quad (22)$$

where $\Sigma = \sum_j \beta_j v_j$ is the ensembled view. We can see in Eq. 22 that the final gradient is the product of the gradients from $f(\Sigma)$, and the gradient from β_j and v_j . Intuitively, this indicates that β_j and v_j are also optimized toward the direction of learning a better ensembled view, which prevents the model from blindly selecting one view and trapped into a local optimum quickly.

3.4.4 Model Architecture

In this section, we give a concrete implementation of the proposed multi-view hierarchical ensembling in Eq. 21 with a transformer model. Based on the derivation in Sec. 3.4.2, here are two desired properties of the implementation: (1) Each view is encoded and processed separately into v without cross-talk. (2) β should be conditioned on previously generated words $y_{1:t-1}$ because it plays the same role as $p(v_j|x, y_{1:t-1})$ in the derivation of Eq. 21.

To satisfy desiderata (1), each view is encoded by a *shared* three-layer transformer encoder *separately*. Tokens in one view cannot attend to tokens in another and thus no cross-talk between views during encoding. The views of an input image can be (a) a sequence of objects predicted by a pre-trained object detector, (b) flattened image grid features encoded by a pre-trained image encoder, and (c) cross-modal textual context of different granularity (whole, five-crops, and nine-crops).

To satisfy desiderata (2) for the ensembling weight β , we propose to modify the transformer decoder layer by introducing two-tiered hierarchical cross-attention as shown in Fig. 12. Specifically, given input $\tilde{y}_l = [\tilde{y}_{l,1}, \tilde{y}_{l,2}, \dots, \tilde{y}_{l,t}]$ of previously generated t words at the l -th decoder layer and separately encoded views $[v_1, v_2, \dots, v_N]$, we first encode y_l with masked self-attention into $\hat{y}_l = [\hat{y}_{l,1}, \hat{y}_{l,2}, \dots, \hat{y}_{l,t}] = \text{Self-Attn}(y_l)$ such that $y_{l,i}$ does not attend to $y_{l,j}$ for $i < j$. For the clarity of illustration, we will only show the operations on $\hat{y}_{l,t}$. We perform first-tier cross-attention to ensemble tokens *within* each view as $\tilde{v}_i = \text{Cross-Attn}_{L_{v1}}(\hat{y}_{l,t}, v_i, v_i)$, where $\text{Cross-Attn}(Q, K, V)$ takes query, key, and value as input and output a sequence same shape as the query. The output of token-level ensembling are collected into $\tilde{v} = [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_N]$. We then perform second-tier cross-attention to ensemble *across* views as $\hat{y}_{l,t} = \text{Cross-Attn}_{L_{v2}}(\hat{y}_{l,t}, \tilde{v}, \tilde{v})$. Ensembling weight β is modeled as the attention weight within $\text{Cross-Attn}_{L_{v2}}$ with $\hat{y}_{l,t}$ as query, which self-attends to $y_{1:t-1}$. Therefore, β is conditioned on $y_{1:t-1}$ and desiderata (2) is satisfied. The final output is $y_{l+1,t} = \text{MLP}(\hat{y}_{l,t})$.

3.4.5 Auxiliary Contrastive Loss

Since the output of the model remains the same, we can train with the standard cross entropy loss or the SCST loss for the generated words. We term this loss $\mathcal{L}_{\text{word}}$. Next, we propose to regard the views as *augmentations* of the input image, which is enabled by the fact that they are encoded separately as described above. We therefore add a contrastive loss $\mathcal{L}_{\text{con}}$ across views to help the representation learning of the encoded heterogeneous views. Note that this is only possible if we encode the views separately and regard each view as an augmentation of the input image.

Implementation-wise, to learn the representation of each view, a learnable [CLS] token is prepended to each view before sending into the transformer encoder. The encoded views from the same input image are positive pairs and the views from different images are negative pairs. To further augment the views, we also apply dropout along the channel, sequence, and view dimensions, with drop rate p_c , p_s , and p_v , respectively. It has been shown by Gao et al. [89] and Liu et al. [90] that simple augmentation introduced by dropout is sufficient for learning a good sentence-level representation. For channel-wise dropout, each channel across views and sequences are randomly zero-ed out with probability p_c . For sequence-wise dropout, each token, which is a d -dimensional feature vector, across views are randomly zero-ed out with probability p_s . For view-wise dropout, each view, which is a matrix of shape $S \times d$ where S is the sequence length of a view, is randomly removed with probability p_v .

The final loss is thus $\mathcal{L} = \mathcal{L}_{\text{word}} + \lambda \mathcal{L}_{\text{con}}$.

3.4.6 Implementation Details

We integrate the proposed multi-view hierarchical ensembling into the $\mathcal{M}^2$ image captioning model [74]. For the contrastive loss, we adapt MoCo-v2 [91] with an exponential moving average (EMA) transformer encoder and a memory buffer of size 10000. The weight for contrastive loss λ_{con} is tuned and set to 0.05 for cross-entropy training and 0.2 for SCST training. The drop rates p_c , p_s , and p_v are all set to 0.1 following Gao et al. [89] without tuning. To generate views of the input image, we use (1) objects detected by Faster-RCNN pre-trained on Visual Genome [1], (2) flattened image grid features from CLIP ViT-B/32 pre-trained model [63], (3) whole, five-crops, and nine-crops cross-modal textual contexts (Txt-Ctx) retrieved by CLIP ViT-L/14 pre-trained model [3]. Note that all models for generating the views are frozen so that all views are pre-computed offline. Unlike Shen et al. [2] that trains the view encoders jointly, our method reduces the computation substantially and achieves much better performance.

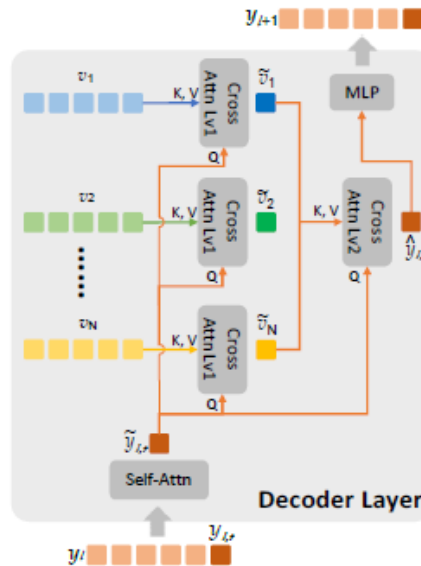


Figure 12: Two-tier hierarchical cross-attention of the modified transformer decoder layer. For clarity, we only show the operations of the last token $y_{i,t}$.

3.4.7 Improving the Multi-View Fusion

In this section, we build on the methods above to improve the integration of information from the retrieved text, images, and bounding boxes. We specifically look at the problem of *how to efficiently and effectively leverage these heterogeneous views for caption generation*? For **efficiency**, three factors are particularly important: computation, parameter count, and label efficiency. State-of-art VL and image captioning models are typically a transformer encoder-decoder model [33],

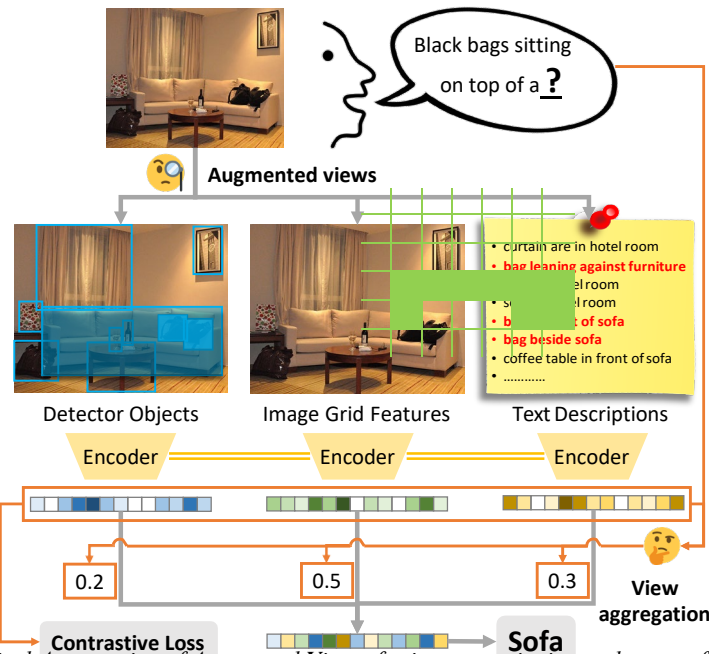


Figure 13: HAAV, **H**ierarchical **A**ggregation of **A**ugmented **V**iews, for image captioning at the step of predicting the word “sofa”. First, heterogeneous views such as detected objects [1], image grid features [2], and text descriptions [3] are generated from the input image by existing methods. We propose to regard these views as augmentations of the input image, and independently encode each view by a shared transformer encoder efficiently. A contrastive loss is incorporated to improve the representation quality of heterogeneous views. Finally, our proposed hierarchical decoder models the effectiveness of each view and adaptively weigh them according to their effectiveness for predicting the current word “sofa”.

which has undesirable quadratic computational complexity with respect to the input sequence size. Therefore, as more views are incorporated, each represented by a sequence of tokens, we should carefully manage the computation and model size. Moreover, on the medium-scale MS-COCO image captioning benchmark [92] (0.6M training samples), we should take label efficiency into consideration when training the data-hungry [79] transformer model to avoid negative effects such as overfitting. For **effectiveness**, different views contain some shared and some complementary information of the input image. Therefore, it is important to model the effectiveness of views and adaptively weigh them according to their effectiveness for predicting each word. Take image captioning in Figure 13 as an example, when predicting the current word “sofa”, for the incomplete caption “black bags sitting on top of a ?”, if say, the view of detected objects fails to detect sofa in the input image, the captioning model should down-weigh the less effective view of detected objects and rely on other more effective views that properly encode the information about sofa. With these considerations in mind, we propose HAAV, **H**ierarchical **A**ggregation of **A**ugmented **V**iews. In HAAV, given a set of heterogeneous views of the input image from existing works such as detected objects [1], image grid features [2], and text descriptions [3], we propose to (1) regard heterogeneous views as augmentations of the input image, and (2) devise a hierarchical decoder layer to account for the effectiveness of heterogeneous views.

For (1), by regarding views as augmentations, we naturally choose to use a *shared* transformer encoder to encode each view *independently*. Compared to methods that concatenate all views into a long sequence as input [77, 78, 3], where the computational complexity scales up quadratically with respect to the number of views, our method scales up linearly. Compared to models that use entire models per view [73, 74, 75, 76] or methods that encode each view with unshared encoders [93, 94, 95, 96], our method is more parameter efficient. Furthermore, data augmentation increases data diversity and thus improves data efficiency, which is particularly important for training data-hungry transformer models. Last but not least, by regarding heterogeneous views generated by different pre-trained models as augmentations of the input image, we incorporate a contrastive loss in a novel way to help representation learning of heterogeneous views and increase data efficiency [85, 87, 88]. Different from how other VL methods [63, 97, 98, 93, 99] incorporate a contrastive loss, our formulation does **not** require annotated pairs (e.g. human annotated image-caption pairs in MS-COCO or image-text pairs scraped from the internet [100]) and can work with unlabeled image-only data to achieve better performance.

Also crucially, for (2) we devise a hierarchical decoder layer, which modifies the standard transformer decoder layer by introducing two-tiered cross-attention modules. The hierarchical decoder first aggregates *within* each view at the token level and then aggregates *across* views at the view level. By introducing this hierarchical aggregating structure, we can better model the effectiveness of views and adaptively weigh them according to their effectiveness. For example, in the Experiment section, we show that if we add noise to a certain view or mask out a prominent region of the input image, the proposed hierarchical decoder indeed down-weigh the noised and masked view when generating words and captions.

In image captioning, given heterogeneous views of the input image from existing works (see Figure 13) such as detected objects [1], image grid features [2], and text descriptions [3], we aim to **efficiently** and **effectively** leverage these views for the target image captioning (IC) task. Following the successful paradigm of image captioning [1, 74, 3], an input image x is first encoded by frozen pre-trained models into heterogeneous views $\{v_1, v_2, \dots, v_{|V|}\}$, each v represented by a sequence of d -dimensional tokens. These views v are then fed into an IC transformer encoder-decoder model, where the IC-encoder encodes important information from v into u for the later decoding step, and the IC-decoder generates captions y conditioned on the encoded views u . In the rest of the paper, since the input image is encoded into v by some frozen pre-trained models from existing works and is not the focus of this paper, for the sake of brevity, we use “input views” or simply “views” for v , “encoded views” for u , and “encode” for the encoding process by the IC-encoder.

In Section 3.4.7 we revisit other successful paradigms in heterogeneous view image captioning: methods that use a separate entire model per view [73, 74, 75, 76] and concatenation of views [77, 78, 3]. By comparing the pros and cons of these two methods, we aim to propose our method that can efficiently account for the effectiveness of input heterogeneous views for caption generation. In Section 3.4.7, we first propose to regard input views v as augmentations of the input image, and thus naturally choose to encode each v *independently* with a *shared* IC-encoder into u . This formulation is more efficient in terms of computation, parameter, and label. It also allows us to add a contrastive loss in a novel way to help representation learning of the encoded views u , and increase label efficiency. In Section 3.4.7, we then propose a *hierarchical decoder layer* to first aggregate *within* each encoded view u at the token level and then aggregate *across* all u at the view level. By introducing this hierarchical aggregating structure, the decoder better models the effectiveness of encoded views u , and adaptively weighs them according to their effectiveness. The overall model architecture is illustrated in Figure 14.

Heterogeneous Views Image Captioning

We start by reviewing two closely related formulations in existing works: (1) combining separate models for each view [73, 74, 75, 76] and (2) concatenation of input views [77, 78, 3]. For (1), $|V|$ captioning models are trained independently for each input view. The word probability predicted by these models is averaged as the final prediction. For (2), all input views, each represented by a sequence of tokens, are concatenated along the sequence dimension to form a single long view. The concatenated view is then fed into a single image captioning model for caption generation.

These two formulations are two extreme cases. In (1), views are encoded and decoded independently, and

the effectiveness of each view is not accounted for to generate the final prediction. The computational complexity scales linearly, but the number of parameters also scales linearly with respect to V . In contrast, in (2), views are encoded and decoded jointly, and the effectiveness of views for caption generation is modeled internally by the attention module within the encoder and decoder layers. The number of parameters remains constant, but this formulation has undesirable quadratic computational complexity with respect to $|V|$. We ask: *can we find a middle ground between these two formulations that enjoys the advantages of both?* Specifically, we would like to model the effectiveness of views for caption generation in (2), but at the same time maintain linear computational complexity in (1) and a constant number of parameters in (2).

Heterogeneous View Encoding

We first observe that each input view v is generated from the input image x and thus contains partial information of x . This means that input views v can be regarded as augmentations of the input image x [101] that describe x from different perspectives. Just like how we use data augmentation in computer vision, we would not concatenate all of the augmented images into one single image before sending it into the model or use different encoders to encode different augmented images. Therefore, we propose to decouple the roles of the IC-encoder and the IC-decoder. Each v is encoded *independently* using a *shared* IC-encoder into u efficiently for the later decoding step. The IC-decoder then models the effectiveness of encoded views u and adaptively weighs them according to their effectiveness. We will focus on the IC-encoder in this section and the IC-decoder in the next section.

By encoding each view independently, the computational complexity scales up linearly ($O(V)$), which is computationally efficient. By encoding each view with a shared IC-encoder, the model size stays constant ($O(1)$), which is parameter efficient. Data augmentation also increases data diversity and thus improves label efficiency. Furthermore, since we regard heterogeneous input views as augmentations of the input image, we propose to add a contrastive loss [83, 84, 86, 85] to improve the representation quality of encoded heterogeneous views u . In contrastive learning, augmented views from the same input image are positive pairs and those from different images are negative pairs. The representation of positive pairs

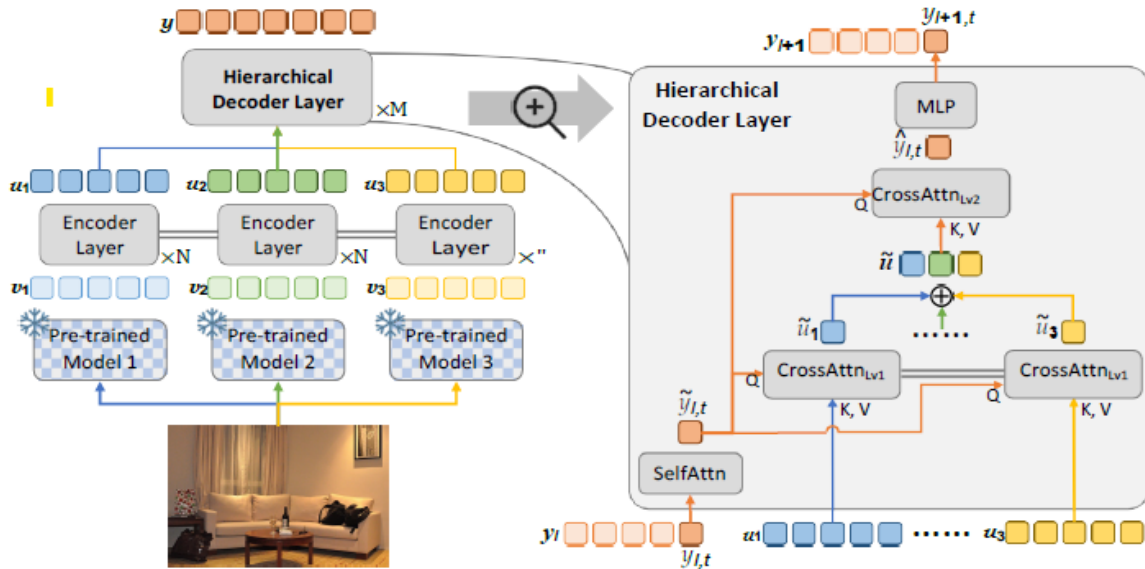


Figure 14: Model architecture. (Left) An input image is first encoded by some frozen pre-trained models into a set of heterogeneous views $\{v_1, v_2, v_3\}$, each represented by a sequence of d -dimensional tokens. These input views v are regarded as augmentations of the input image, and thus are encoded independently with a shared transformer encoder into $\{u_1, u_2, u_3\}$. (Right) To model the effectiveness of the encoded views u for current word prediction, we propose a hierarchical decoder layer that first aggregates within each encoded view u at the token level with a shared CrossAttn_{Lv1} module, and then aggregates across the encoded views at the view level with a CrossAttn_{Lv2} module. For clarity of illustration, we only show operations for the current $y_{t,t}$ token and not the $y_{t,t-1}$ past tokens.

is pulled together in an embedding space while the representation of negative pairs is pushed apart. To further increase diversity across heterogeneous views for the contrastive loss [86, 101, 51, 13], we add channel-wise and sequence-wise dropout during training. For channel-wise dropout, each channel across views and sequences is randomly zero-ed out with probability p_c . For sequence-wise dropout, each token, which is a d -dimensional feature vector, across views is randomly zero-ed out with probability p_s . Note that our goal is not to propose a new contrastive loss here but to incorporate an existing one in a novel way in the VL setting. Compared to how most other VL pre-training methods incorporate a contrastive loss [63, 97, 98, 93, 99], where paired annotations such as image and text pairs annotated by human or scraped from the internet are required, our formulation only requires input image *without* paired annotations to construct positive and negative pairs.

Heterogeneous View Decoding

Given the encoded views u from the IC-encoder, in this section we shift our attention to the IC-decoder to account for the effectiveness of u for predicting each word. Specifically, the IC-decoder should adaptively weigh each encoded view u according to their effectiveness at each word prediction step conditioned on previously generated words and all other encoded views u . For example, in Figure 13, when predicting the current word “sofa” for the incomplete sentence “Black bags sitting on top of a ?”, if say, the object detector fails to detect sofa, the decoder should down-weight the view of detected objects and should leverage other views that properly encode sofa information to correctly predict the next word. On the other hand, we spent great efforts in Section 3.4.7 to encode the views efficiently. Therefore, the proposed decoding strategy should be at least as efficient as the proposed encoding strategy.

We propose a **hierarchical decoder layer** shown in Figure 14, which modifies the standard transformer decoder layer by introducing the two-tiered cross-attention structure. First, current word $y_{l,t}$ is contextualized with all previously generated words $y_{l,1:t-1}$ into $\tilde{y}_{l,t}$ by the SelfAttn module. Next, to model the effectiveness of each encoded view for current word prediction, the hierarchical decoder aggregates *within* each encoded view u independently at the token level by a shared CrossAttn_{Lv1} module. The outputs of CrossAttn_{Lv1} from all views are collected into $\tilde{u}$. Finally, to adaptively weigh all the views according to their effectiveness for current word prediction, the hierarchical decoder then aggregates *across* views $\tilde{u}$ at the view level by the CrossAttn_{Lv2} module. We also add a view-wise dropout to randomly drop an encoded view u with probability p_v during training to encourage the decoder to better leverage all encoded views instead of focusing on a few certain ones. By introducing this hierarchical aggregation structure, we can better model the effectiveness of the encoded views u by CrossAttn_{Lv1} and adaptively weigh them by CrossAttn_{Lv2}.

In terms of computation and parameter efficiency, the hierarchical decoder layer uses a shared CrossAttn_{Lv1} module to first aggregate within each view. The computational complexity scales up linearly, and the model size is constant with respect to $|V|$. It then aggregates across views using a CrossAttn_{Lv2} module with linear computational complexity, and constant model size with respect to $|V|$. Overall, the computational complexity and model size does not exceed our proposed view encoding strategy in Section 3.4.7.

Implementation-wise, to learn the representation of each view, a learnable [CLS] token is prepended to each view before sending into the transformer encoder. The encoded views from the same input image are positive pairs and the views from different images are negative pairs. To further augment the views, we also apply dropout along the channel, sequence, and view dimensions, with drop rate p_c , p_s , and p_v , respectively. It has been shown by Gao et al. [89] and Liu et al. [90] that simple augmentation introduced by dropout is sufficient for learning a good sentence-level representation. For channel-wise dropout, each channel across views and sequences are randomly zero-ed out with probability p_c . For sequence-wise dropout, each token, which is a d -dimensional feature vector, across views are randomly zero-ed out with probability p_s . For view-wise dropout, each view, which is a matrix of shape $S \times d$ where S is the sequence length of a view, is randomly removed with probability p_v . The final loss is thus $\mathcal{L} = \mathcal{L}_{\text{word}} + \lambda \mathcal{L}_{\text{con}}$.

Implementation Details

Our proposed HAAV can be easily incorporated into existing transformer encoder-decoder IC models. In this paper, we choose the state-of-art Xmodal-Ctx [3] as our base model. Following the conventional training procedure [74, 77, 102], the model is first trained with cross-entropy loss and then fine-tuned with SCST [75] loss, which optimizes CIDEr score by reinforcement learning (note this is done by all compared methods). The dropout rates p_c , p_s , and p_t are all set to 0.1 following [89] without tuning. We reuse the heterogeneous views from previous methods without modifications, including detected objects from [1], CLIP ViT-B/32 [63] image grid features from [2], and text descriptions of image regions from [3]. For the contrastive loss, we prepend a learnable [CLS] token for each input view v before sending it into the IC-encoder and use the encoded [CLS] token as a view-level representation for computing the contrastive loss. We use MoCo-v2 [91] with an exponential moving average (EMA) transformer encoder and a memory buffer to compute the contrastive loss. More implementation details and hyper-parameters can be found in the supplementary.

3.5 Semi-Supervised Object Detection through Cross-Match

In this section, we detail our methods for semi-supervised object detection, by leveraging a denser gradient signal from pseudolabels [22]. A great deal of work, including by our group, has been done on semi-supervised image classification. One popular line of research for SSL can be summarized as generating artificial labels on unlabeled images and feeding those unlabeled images with artificial labels through loss functions to train the models. For instance, pseudo-labelling [103, 104, 105, 106] (also known as self-training [107, 108]) produces labels generated from the model’s predictions themselves. Typically, confidence-based thresholding is used to filter out predictions that are low-confidence and hence likely to be incorrect. Consistency regularization [109, 29, 28, 27, 110, 51] applies different perturbations to inputs or models and enforces predictions under these perturbations to be consistent. The consistency constraint is usually applied through soft labels (*i.e.*, distillation) though it is also possible to do it through hard labels [51] where predictions are converted into hard (one-hot) labels.

Although great success has been achieved for SSL in the image classification context using these methods, translating these techniques to SSL for object detection has only recently begun to be explored [7, 111, 112, 113, 114]. In translating these techniques to object detection, the typical process used to generate pseudo-labels is extracting a dense set of bounding box predictions, followed by confidence thresholding and non-maximum suppression (NMS).

However, a direct translation of pseudo-labeling does not fully consider the characteristics of the object detector training. For example, since hard decisions are made based on probability-based thresholding, only a small number of bounding boxes make it through this thresholding to eventually become pseudo-labels. Some of these pseudo-labels may still be incorrect and, more importantly, Region of Interests (RoIs) that are highly overlapped with false negatives (*i.e.*, boxes with real objects that are filtered out) will be treated incorrectly as background or get ignored during the training process. As a result, the overall gradient signal produced by such pseudo-labeling is *weak* (*i.e.*, there will only be a few positives, and several sources of error could result in degradation of the signal) and *sparse* (*i.e.*, across the image and objects).

In this work, we present our method, CrossMatch, to address these problems by seeking to strengthen the gradient signal in two ways. First, we identify a key source of rich information already inherent in many object detection approaches:

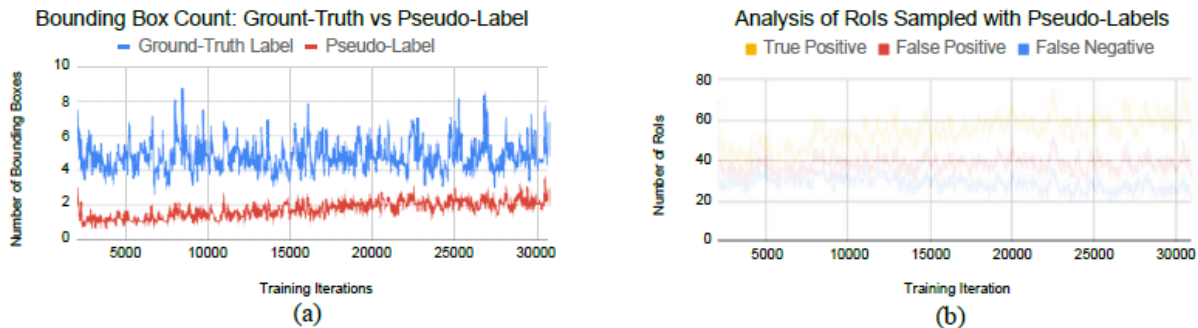


Figure 15: An illustration of sparse and weak gradient signals from hard pseudo label. (a): The number of pseudo-label generated per image is much smaller than the number of ground-truth labels, which indicates the supervision signal from pseudo-labels is sparse. (b): In the training process, only a small number of true positive RoIs are sampled with pseudo-labels and a significant amount of foreground RoIs are not sampled (false negative). There are also a number of false positive RoIs that are mis-sampled. This indicates the supervision signal from pseudo-labels is also weak.

multi-scale feature extraction. For example, the Feature Pyramid Network [115] generates a set of features at multiple scales for each image. Even though the region proposal network [116] uses the entire feature pyramid to generate RoIs at multiple scale, the network only assigns one chosen level of the feature pyramid to RoIs for class-level prediction. We argue that other levels of the feature pyramid can be viewed as feature representations at different scales and is a non-negligible resource for diverse representation to train the network under semi-supervised settings. Thus, we propose a novel method that enforces consistency between multiple scales. In contrast to the default level-assignment approach, CrossMatch makes predictions with multiple levels of features from the feature pyramid and computes losses with those predictions accordingly. For inference, we still use the default heuristic-based method for feature selection so that no additional computation overheads are generated. To the best of our knowledge, our proposal is the first to use such multi-scale losses in the context of semi-supervised learning.

Second, unlike prior work [113], which attempts to completely replace hard pseudo-labels with soft probability targets, CrossMatch enforces multi-scale consistency through both soft and hard labels. We argue that combining soft targets with hard labels can be an effective approach to alleviate the weak and sparse supervision signals from hard pseudo-labels. Our approach adopts the teacher-student [29, 7] and strong-weak augmentation [51, 7, 112, 113] framework and train the student network with both hard and soft artificial labels. Since soft labels can be easily obtained for all RoIs without matching them to ground-truth bounding boxes, the student network can effectively distills meaningful knowledge through dense RoIs from the teacher model. To prevent the gradients from being dominated by easy background RoIs, we use a refined design of OHEM [117] to perform RoI sampling for training with soft labels. More importantly, "false-negative RoIs" are assigned to a meaningful target during optimization.

We evaluated our method on MS-COCO [80] and Pascal VOC [118] following the experimental protocol used in state-of-the-art literature [111]. It is worth mentioning that our method achieves 16.81 and 20.63 mAP on Faster-RCNN ResNet-50-FPN architecture with only 0.5% and 1% labeled images respectively on MS-COCO, which significantly outperforms other state-of-the-art methods by more than 2 mAP. In summary, the contributions of this paper are as follows:

- We propose a novel method to train the student network with multi-scale consistency. This practice provides diverse representations of features at multiple scales to better leverage unlabeled images. To the best of our knowledge, this is the first work that leverages multi-scale consistency in semi-supervised object detection.
- Based on our analysis (Figure 15), we propose to enforce the multi-scale consistency through the combination of soft and hard labels to alleviate the limitation of only using hard pseudo-labels.
- Our method achieves state-of-the-art performance particularly when labeled data is limited. It significantly outperforms other methods with 0.5%, 1%, 2% and 5% labeled images for MS-COCO.

3.5.1 Methodology

3.5.2 Problem Definition

The problem of semi-supervised object detection is defined as follows: we are given a training set of labeled images $D_s = \{x^s, y^s\}_{i=1}^{N_s}$ and a set of unlabeled images $D_u = \{x^u\}_{i=1}^{N_u}$, where N_s and N_u are the size of labeled and unlabeled set respectively. Typically, we expect $N_s \ll N_u$. The label y^s contains information about both object categories and coordinates of bounding boxes.



Figure 16: Overview of **CrossMatch**. CrossMatch enforces consistency between different scales of feature maps. The teacher model takes weakly augmented images and produces artificial (in both soft and hard form, with a refined OHEM employed for the regions used for soft) with a **heuristically selected level of features** (denoted by red). The student model takes strongly augmented images and constantly makes predictions with **every level of the feature pyramid**. The consistency constraint is enforced through both soft and hard pseudo-labels.

To train an object detector under this setting, the general form of loss is decoupled into a supervised loss ($\mathcal{L}_s$) and an unsupervised loss ($\mathcal{L}_u$):

$$\mathcal{L} = \mathcal{L}_s + \beta \mathcal{L}_u \quad (23)$$

where β is a tunable weight scaling for unsupervised loss.

3.5.3 General Framework

Our method adopts the teacher-student framework [29, 7, 113] which has been proven to be an effective approach in semi-supervised learning. Specifically, we maintain two copies of the network weights standing for **Teacher** and **Student**. The student network is updated based on both the supervised loss and unsupervised loss as shown in Equation 23, whereas the teacher model is only updated by computing the exponential moving average (EMA) of the student model weights:

$$\theta_{teacher} \leftarrow a\theta_{teacher} + (1 - a)\theta_{student}. \quad (24)$$

Another important aspects of recent success in semi-supervised learning for both image classification [51, 110] and object detection [7, 112, 113, 119] is the Weak-Strong Augmentation Scheme. When incorporating this scheme with our Teacher-Student framework, weakly augmented images are passed to the teacher model for more reliable artificial labels and strongly augmented images are used as inputs to the student model for higher diversity of input images.

Since the model is usually underfit in the initial training phase, we allow the model to warm-up for a few iterations, which is known as the Burn-In stage in Unbiased Teacher [7]. During the Burn-In stage, the student network is trained with only supervised loss and no EMA is applied on the teacher model. At the end of Burn-In stage, the teacher model is initialized by the student model weights and updated by EMA only in the rest of training. Following Faster R-CNN [116], the supervised loss has four major components: the RPN classification loss $\mathcal{L}_{cls}^{rpn}$, the RPN regression loss $\mathcal{L}_{reg}^{rpn}$, the RoI classification loss $\mathcal{L}_{cls}^{roi}$, and the RoI regression loss $\mathcal{L}_{reg}^{roi}$:

$$\mathcal{L}_s = \mathcal{L}_{cls}^{rpm}(x, y) + \mathcal{L}_{reg}^{rpm}(x, y) + \mathcal{L}_{cls}^{roi}(x, y) + \mathcal{L}_{reg}^{roi}(x, y). \quad (25)$$

where x and y are input images and the ground-truth annotations (class labels and bounding boxes) respectively.

3.5.4 CrossMatch

To address the issue of sparse and weak gradient signal in pseudo-labeling methods (Figure 15), we introduce CrossMatch. The intuition behind our method is that we can propagate gradient signals through multiple paths from multi-scale methods. Specifically, our proposed method for semi-supervised object detection enforces consistency *between different levels of the feature pyramid generated by the feature pyramid network (FPN)* [115]. See Figure 16 for illustration.

The FPN generates a set of multi-scale feature maps $\{P_2, P_3, P_4, P_5, P_6\}$, which are used in both stages of two-stage detectors¹. The region proposal network (RPN) uses these feature maps to produce multi-scale region of interests (RoIs). Once the RoIs are collected, two-stage detectors like Faster-RCNN [116] assigns only one particular level of features to each RoI determined by a heuristic function (Equation 26) based on spatial dimensions of RoIs (and ignores other levels):

$$k = \lfloor k_0 + \log_2(\sqrt{wh}/224) \rfloor \quad (26)$$

We argue that different levels of features on the feature pyramid can be viewed as feature representations at different scales and therefore propose a novel method to enforce consistency between different scales. The fact that this includes smaller scales, but potentially in a soft manner (see Section 3.5.6), would result in a denser (but still weak) signal that can be used for semi-supervised training. Specifically, the student network makes predictions using each level of $\{P_2, P_3, P_4, P_5\}$ and the teacher network generates artificial labels using the features determined by the heuristic function. Although we can also use all scales for the teacher, the heuristic function ensures the appropriate scale is used to generate reliable artificial labels. We empirically found that using all scales for the teacher is computationally intensive and did not improve performance. The unsupervised loss function is computed by averaging the loss between each of student predictions and artificial labels, which is described in the next sections.

Unlike prior work that applies the consistency constraint with either soft labels [109, 29, 28, 27] or hard labels [51], and in order to further densify the gradient signal, CrossMatch enforces consistency between different scales by using both soft and hard labels. We show in experiments that this is a crucial design for semi-supervised object detection especially with limited labeled data.

3.5.5 Consistency with Hard Labels

In this section, we describe the training process of enforcing cross-scale consistency with hard artificial labels.

Each unlabeled image is first weakly augmented and forwarded to the teacher model for hard artificial label generation. The teacher model assigns features to RoIs based on equation 26 as discussed earlier. Notice that RoIs with different spatial dimension may still be assigned to different levels but ultimately only one particular level is matched with one RoI. We apply the confidence thresholding to filter out low-confident RoIs and use non-maximum suppression (NMS) (a standard post-processing technique for anchor-based detectors) to exclude highly overlapped RoIs. This process helps reduce confirmation bias and error accumulation from noisy hard artificial labels.

After label generation, the same image, which is transformed by strong augmentation operations, is forwarded to the student model. The training process is similar to train the model with labeled data except that the student model makes predictions using each level of the feature from the feature pyramid. The loss is computed in the exact same manner as the supervised loss where the supervision signal passes through both RPN and RoI heads of the student network. Notice that we do not include the

bounding box regression loss with hard labels because we found it did not help the training process as it is more difficult to obtain and filter accurate regression thresholds. This finding is consistent to prior work [7] and we therefore exclude the regression loss in the training procedure.

$$\mathcal{L}_p = \mathcal{L}_{cls}^{rpm}(x, y) + \mathcal{L}_{cls}^{roi}(x, y) \quad (27)$$

As pointed out by prior work [7], the foreground-background imbalance issue in supervised object detection [117, 120] still exists in semi-supervised object detection. Following existing work [7], we use multi-class focal loss [120] in replacement of vanilla cross-entropy loss for RoI-Level predictions.

While this process is similar to pseudo-label methods in prior work [112, 113, 7, 119], in our method the hard labels are generated with one certain level of features and the loss is computed with student output across multiple levels of features. This approach introduces a consistency constraint between different scales and we demonstrate the its efficacy in the experiment section.

¹ P_6 is often excluded in the second stage.

3.5.6 Consistency with Soft Labels

Limitation with Hard Labels Even though hard labelling has achieved recent success in semi-supervised object detection, there are still possible limitations to this approach. The standard object detection training process involves matching RoIs to labels. RoIs that are matched with labels (when the intersection-over-union score is above the threshold) are sampled as positive examples and the rest is treated as backgrounds (or ignored). However, due to confidence thresholding, it is possible that no (or inaccurate) psuedo-labels are generated in some object regions. Unlike semi-supervised image classification where the low-confident images are simply excluded from training, RoIs that are highly overlapped with those false-negative regions can be mistakenly treated as backgrounds which can be a potential source of erroraccumulation.

To tackle this problem, we propose to combine soft labels with hard labels. For each unlabeled image, we first use the Teacher’s region proposal network (RPN) to produce a list of RoIs $\{R_i\}_{i=1}^N$ where N is the total number of proposals. This is a crucial difference from the hard label branch as the supervision signal from soft labels only passes through RoI heads of the student model. We found the Teacher’s RPN is more accurate in terms of generating proposals and thus yields to better performance.

Next, for each RoI R_i , Student and Teacher make their own probability predictions on strongly and weakly augmented input images respectively. Similar to the training process with hard labels, the student network makes predictions using all levels of features from the feature pyramid whereas the teacher model only uses heuristically selected levels. The consistency loss is then computed as the KL Divergence between each of the student’s class probability distribution and the teacher’s probability distribution (p^k and p_t respectively):

$$\mathcal{L}_c = \sum_{k=0}^M D_{KL}(p_s^k | p_t) \quad (28)$$

Where p_s^k stands for the student prediction made with feature level k and M is the total number of levels of the feature pyramid.

Since the region proposal network typically generates a large number of RoIs and most of them are easy background examples, the imbalance issue also exists in soft labels. To alleviate the problem, we adopt Online Hard Example Mining (OHEM) [117] as the sampling strategy in our soft label branch. The general idea is to sample RoIs based on the loss values: we compute the consistency loss for all RoIs but only select top-K of them with largest loss values for back-propagation where K is a tunable hyper-parameter. We make two refinements on the original form of OHEM in our scenario. First, we do not apply NMS scored by loss values because we find doing so significantly hurts the model performance. Moreover, selecting a small portion of RoIs with large loss values affects the stability of training in the initial phase. Therefore, we apply a ramp-down process on the value of K from the total number of RoIs

to the desired value of K in a few iterations.

Note that the OHEM sampling is applied between student predictions from each level of features and the teacher prediction; it is possible that different RoIs are sampled for different levels of student features. Therefore, the total number of RoIs involved in the soft label branch may be greater than K .

3.6 Advancements in Active Learning

3.6.1 Summary

One of the ways to learn good models using less labels is by using active learning to select the most informative samples to acquire labels for. Based on some prior work in our group [121, 122], we developed a new method to select samples actively with the underlying goal of improving the embedding using information theoretic metrics. This method, being primarily dependent on the embeddings of the dataset, is agnostic to the task being considered. Hence our method, InfoNN is a unified framework for AL which can be used in any problem where a key component is learning the embedding of the dataset. We demonstrated the versatility of the proposed method by experimenting with both supervised classification and deep metric learning and exhibiting improved performance [23].

3.6.2 Introduction

A defining feature of modern machine learning is the reliance on large volumes of human-labeled data. Perhaps the most prominent example is the existence of massive hand-labelled image datasets, but the task of acquiring large amounts of human-provided data is nearly ubiquitous in machine learning. However, such data is not free; it is often tedious and expensive to gather a sufficient number of query responses to satisfy data hungry machine learning models. Active learning (AL) [123] seeks to mitigate this issue by carefully selecting only the most informative samples to be labelled. More generally, AL attempts to identify the most informative queries to pose to an oracle. These queries can include asking for a class label or rating, or more general relational queries such as the similarity (or dissimilarity) of different items.

Classification relies on models (e.g., neural networks) with the ability to learn good representations of the data. Active classification focuses on how to best solicit labels for unlabelled data points, with many modern approaches either implicitly or explicitly relying on the representations learned to determine the most informative label. Methods that use metrics based on the predicted class probabilities, such as uncertainty [124] or consistency [125], implicitly rely on such representations, whereas core-set based approaches [126, 127] directly use learned representations to select diverse samples. Thus, if we seek the most informative labels with respect to improving the learned representation of our classification model, it enables our method to be task agnostic. Here, we present a unified framework, which is made feasible by a novel type of similarity query that can be applied to any problem which involves learning a representation of the data.

Specifically, we consider the *nearest neighbor* (NN) query, which, given a reference data point r , asks an *oracle* (e.g., a human expert) to select the most similar point from among a set of C alternatives $t^1, t^2, \dots, t^C$. We denote this a length C NN query. With the goal of minimizing the required number of queries, we adapt an *active* query selection strategy to this query type. We take an information theoretic approach and estimate the gain in *mutual information* (conditioned on previous query responses) as the criteria for selecting the most informative query, an approach that we dub *Info-NN*.

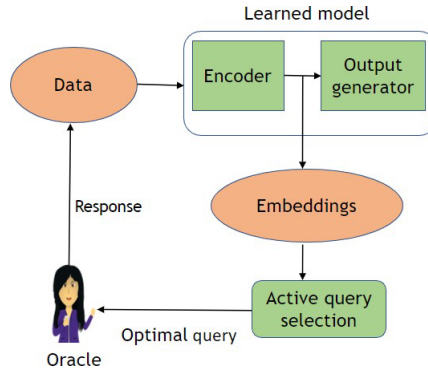


Figure 17: Visualization of the unified query solicitation framework. Candidate NN queries to be evaluated by the active NN query selection method are formed based on the embeddings learned by the model. These candidate queries are evaluated for informativeness by considering mutual information between the learned embedding and the query response. The oracle then responds to the most informative of these queries.

3.6.3 Methods, Assumptions and Procedures

In this section, we provide an overview of our proposed generalized query framework. Specifically, we show that in any classification setting where a latent representation of the data is learned, querying an oracle for a class label can be re-formulated as soliciting the oracle’s feedback for an NN query, allowing us to draw the connection to metric learning. We also present Info-NN, an active method of selecting NN queries using information theoretic criterion.

Formally, a NN query $Q_i = r_i \cup T_i$ of length C consists of a reference data point r_i and a set of data points $T_i = \{t_i^1, t_i^2, \dots, t_i^C\}$, from which the oracle picks the point most similar to the reference r_i . Let $Y_i \in \{1, 2, \dots, C\}$ be the random variable indicating the oracle’s response to the i^{th} query. When $Y_i = c$, this indicates that the oracle selected $t_c \in T_i$ as the most similar to the reference r_i .

3.6.4 Classification as an NN query problem

We approach AL for classification by selecting labels that will improve the feature representation, as most modern classification techniques (e.g., neural networks) can be interpreted as learning an embedding that enables simple linear classifiers to be effective. We do this via an analogy in which obtaining the class label for an unlabelled sample is equivalent to a particular NN query response.

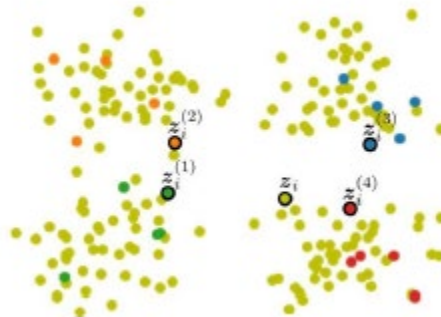


Figure 18: Example of an unlabelled z_i and the nearest labelled neighbors to z_i from each class: $z_i^{(1)}$, $z_i^{(2)}$, $z_i^{(3)}$, $z_i^{(4)}$. In this example, we might expect that the most likely label would be $y_c = 4$, which could be interpreted as a nearest neighbor query response (that $z_i^{(4)}$ is the nearest neighbor).

Consider a dataset $\mathcal{X} = \{x_i\}_{i=1}^N$ consisting of points belonging to C classes, $\{y_i\}_{i=1}^N \in \{1, 2, \dots, C\}$. We assume access to an initial labelled, $\mathcal{L} = \{x_i, y_i\}_{i=1}^J$ and unlabelled, $\mathcal{U} = \{x_i, y_{j+1}\}_{i=J+1}^N$ set of samples. Let $\mathcal{Z} = \{z_i\}_{i=1}^N$ represent initial estimates of the embeddings for the dataset according to a model learned on an initial set of labelled samples. Now suppose we want to choose a new point x_{j+1} from $\mathcal{U}$ whose label y_{j+1} we will obtain. For any x_i in $\mathcal{U}$, consider its embedding z_i in the feature space and the nearest neighbor to z_i from $\mathcal{L}$ for each class, i.e.,

$$z_i^{(c)} = \arg \min_{z_l \in \mathcal{L}_c} \|z - z_l\|_2,$$

where $\mathcal{L}_c = \{z_l: (x_l, y_l) \in \mathcal{L}, y_l = c\}$. An example of an unlabelled z_i and the nearest labelled neighbors to z_i from each class is illustrated in Fig. 18.

Note that if the embedding that we have learned does a reasonable job of representing similarity (as it pertains to the task of classification), then we would expect that the most likely label for z_i would correspond to the class c for which $z_i^{(c)}$ is closest to z_i . Thus, we can interpret the label y_i as a response to the nearest neighbor query in which z_i is the reference to which $z_i^{(1)}, z_i^{(2)}, \dots, z_i^{(C)}$ are compared. (For computational reasons, one may choose to not use all C nearest neighbors in practice.) Because this NN query response reveals information about the relative locations of items in the learned representation, retraining the classification model with the new oracle response should improve the representation. **This is the key idea behind our approach: select NN queries (or equivalently, points to label) that result in the best improvement of the embedding.**

3.6.5 Unified framework for active classification and metric learning

This view of active classification gives rise to a unified framework which can be used in either active classification or active Deep Metric Learning (DML): from a pool of candidate NN queries, choose the most informative query to ask the oracle, then re-train the model to incorporate the newly acquired query response. **Despite each problem seemingly requiring fundamentally different oracle responses (similarity information vs. labels), both problems can be tackled utilizing NN queries, and thus, the same active query selection strategy.** The main difference is the pool of candidate queries. In active DML, we can query the oracle for similarity information about any set of items, whereas in active classification, the pool of candidate NN queries is restricted to queries that contain one item from every class. This pool of candidate queries is formed by setting every z_i corresponding to an $x_i \in \mathcal{U}$ as the reference point, and finding (up to) C nearest neighbors of differing classes. **A critical feature of this unified framework is that it does not depend on which measure of “informativeness” is used. This allows for a practitioner to plug-in their desired active query selection criteria without making any modifications to the framework,** as depicted in Fig. 17. In our experiments, we select the queries that maximize *mutual information* for both active DML and classification experiments.

3.6.6 Active query selection via Info-NN

Observation model. To model the oracle’s response, we use a Plackett-Luce (PL) model [128] which is an extension of the triplet model commonly used with similarity comparisons [129]:

$$P(Y_i = c) = \frac{(D_{ic}^2 + \mu)^{-1}}{\sum_{j=1}^C (D_{ij}^2 + \mu)^{-1}} \quad (29)$$

Table 2: Comparison on CIFAR-100 and mini-imageNet. Numbers represent error rate in three runs. For fair comparison, we use the same model as other methods: CNN-13 for CIFAR-100 and ResNet-18 for mini-ImageNet.

Method	CIFAR-100		mini-ImageNet	
	# Labeled samples		# Labeled samples	
	4,000	10,000	4,000	10,000
Π -model [131]	-	39.19 ± 0.36	-	-
SNTG [132]	-	37.97 ± 0.29	-	-
SSL with Memory [133]	-	34.51 ± 0.61	-	-
Deep Co-Training [134]	-	34.63 ± 0.14	-	-
Weight Averaging [135]	-	33.62 ± 0.54	-	-
Mean Teacher [29]	45.36 ± 0.49	36.08 ± 0.51	72.51 ± 0.22	57.55 ± 1.11
Label Propagation [106]	43.73 ± 0.20	35.92 ± 0.47	70.29 ± 0.81	57.58 ± 1.47
PLCB [31]	37.55 ± 1.09	32.15 ± 0.50	56.49 ± 0.51	46.08 ± 0.11
FeatMatch (Ours)	31.06 ± 0.41	26.83 ± 0.04	39.05 ± 0.06	34.79 ± 0.22

where D_{ic} denotes the distance between the embeddings of r_i and t^c , and μ is a parameter set by the user. This model captures uncertainty in the oracle responses as well as uncertainty in our current estimate of the embedding (and hence distances). The parameter μ is indicative of our confidence in the distances. Note that even though we use this model in our query selection strategy, we do *not* require that query responses are generated according to the PL model.

Active query selection criteria. The main idea behind our selection strategy is to select queries that are maximally informative about the embedding while avoiding ones that do not provide new information. This goal is achieved by using mutual information between the embedding and a query as a measure of the informativeness of the query. Let $y^{n-1} = y_1, y_2, \dots, y_{n-1}$ denote the set of all responses obtained after $n-1$ queries. We denote Y_n to be the random variable corresponding to the oracle’s response to query Q_n . Now consider the mutual information between the embedding $\mathbf{Z}$ and the response Y_n :

$$I(\mathbf{Z}; Y_n | y^{n-1}) = H[\mathbf{Z} | y^{n-1}] - \mathbb{E}_{Y_n}(H[\mathbf{Z} | Y_n, y^{n-1}]). \quad (30)$$

This quantity measures how much information the response to query Q_n would provide about the embedding, conditioned on the fact that we have already acquired the responses y^{n-1} to the previous queries. This is exactly what we would like to use to select informative queries, but computing this quantity in the above form is computationally expensive. To compute this in a naïve manner we would need to find the estimate of the embedding for every possible response to the query and compute the entropies of these estimates in the high dimensional embedding space. Fortunately, using an approach similar to [130], we can use the symmetry of mutual information to re-write (30) as

$$I(Y_n; \mathbf{Z} | y^{n-1}) = H[Y_n | y^{n-1}] - \mathbb{E}_{\mathbf{Z}}(H[Y_n | \mathbf{Z}, y^{n-1}]). \quad (31)$$

We can now compute entropies in the response space, which is usually much smaller than the embedding space. This second form of mutual information also provides an interesting insight about the selection strategy. The first term, which denotes the entropy of the predicted response, encourages the selection of queries which are highly uncertain for the current estimate of the embedding. The second term denotes the expected entropy of the responses predicted by the individual samples from the distribution over the embedding estimate and encourages queries for which the individual samples are fairly confident. This simultaneously avoids the acquisition of redundant queries and queries for which the oracle response is

likely to be uncertain.

4.0 Results and Discussion

4.1 FeatMatch: Results

Standard SSL Datasets. We conduct experiments on commonly used SSL datasets: SVHN [136], CIFAR-10 [137], CIFAR-100 [137], and mini-ImageNet [138] (we only show the latter larger datasets here). Following the standard approach in SSL,

Table 3: Comparison between the image-based baseline with our proposed feature-based augmentation method on Domain-Net with 1) unlabeled data coming from the same domain as the labeled target ($r_u = 0\%$), and 2) half of unlabeled data coming from the same domain as the labeled target and the other half from shifted domains ($r_u = 50\%$). Numbers are error rates across 3 runs.

Method (5% labeled samples)	$r_u = 0\%$	$r_u = 50\%$
(Semi-supervised) Baseline	56.63 ± 0.17	65.82 ± 0.07
FeatMatch (Ours)	40.66 ± 0.60	54.01 ± 0.66
Supervised baseline (5% labeled samples, lower bound,)	77.25 ± 0.52	
Supervised baseline (100% labeled samples, upper bound)	31.91 ± 0.15	

we randomly choose a certain number of labeled samples as a small labeled set and discard the labels for the remaining data to form a large unlabeled set. Our proposed method is tested under various amounts of labeled samples. SVHN is a dataset of 10 digits, which has about 70k training samples. CIFAR-10 and CIFAR-100 are natural image datasets with 10 and 100 classes respectively. Both dataset contains 50k training samples. For mini-ImageNet, we follow [106, 31] to construct the mini-ImageNet training set. Specifically, given a predefined list of 100 classes [138] from ILSVRC [139], 500 samples are selected randomly for each class, thus forming a training set of 50k samples. The samples are center-cropped and resized to 84x84 resolution. We then follow the same standard procedure and construct a small labeled set and a large unlabeled set from the 50k training samples.

Note that we have also tested these methods under SSL under domain shift, where the unlabeled data can come from a different distribution (in this case only non-semantic shift), described in Section 3.1.1.

Results. We first show our results on CIFAR-100 and mini-ImageNet with 4k and 10k labels in Table 2. Our method consistently improves over state of the arts by large margins, with about absolute 5% on CIFAR-100 with 4k labels and 17% on mini-ImageNet with 4k labels.

In Table 3, we show our results on the larger dataset of DomainNet setting, which contains unlabeled data coming from other shifted domains. It can be clearly seen that in the setting of $r_u = 50\%$, where 50% of the unlabeled data are coming from other shifted domains, the performance drops by a large margin compared with the setting of $r_u = 0\%$, where all the unlabeled data are coming from the same domain as the target labeled set. Nevertheless, our proposed feature-based augmentation method improves over supervised baseline by absolute 36% error rate when $r_u = 0\%$ and 23% when $r_u = 50\%$. When compared to the conventional image-based augmentation baseline, we improved by 12% when $r_u = 50\%$ and 16% when $r_u = 0\%$.

4.2 Manifold Transport Operators

4.2.1 CelebA Augmentations

We first show results in unsupervised learning of natural variations in a dataset using the manifold autoencoder (MAE) framework [40, 18]. To effectively train the MAE, we must select training point pairs that are close on the data manifold. We do this for each training point by selecting nearest neighbors in the intermediary layer of a ResNet18 classifier pretrained on ImageNet. This approach is known as perceptual similarity [140] and has successfully been applied for style transfer and image-to-image translation. We find that perceptual similarity is effective in denoting point pairs that result in semantically meaningful transformations without labels, as depicted in Figure 19.

In this figure we also compare to other methods that learn unsupervised transformations using a similar network architecture. The β -VAE [141] encourages disentanglement of latent dimensions by reweighting the KL-divergence term in the VAE objective. We note how this model learns similar transformations as the MAE, at the cost of sacrificing reconstruction quality. Additionally, the contractive autoencoder (CAE) [142] paths coincide with directions of motion on a linear tangent plane at each latent vector.

To quantify the learned MAE transformations, we utilize a classifier trained on the attribute labels from CelebA images. Denoted in Figure 20, it can be seen that the applied augmentations result in changes in the classifier output for attributes that coincide with the semantic meaning of each transformation.

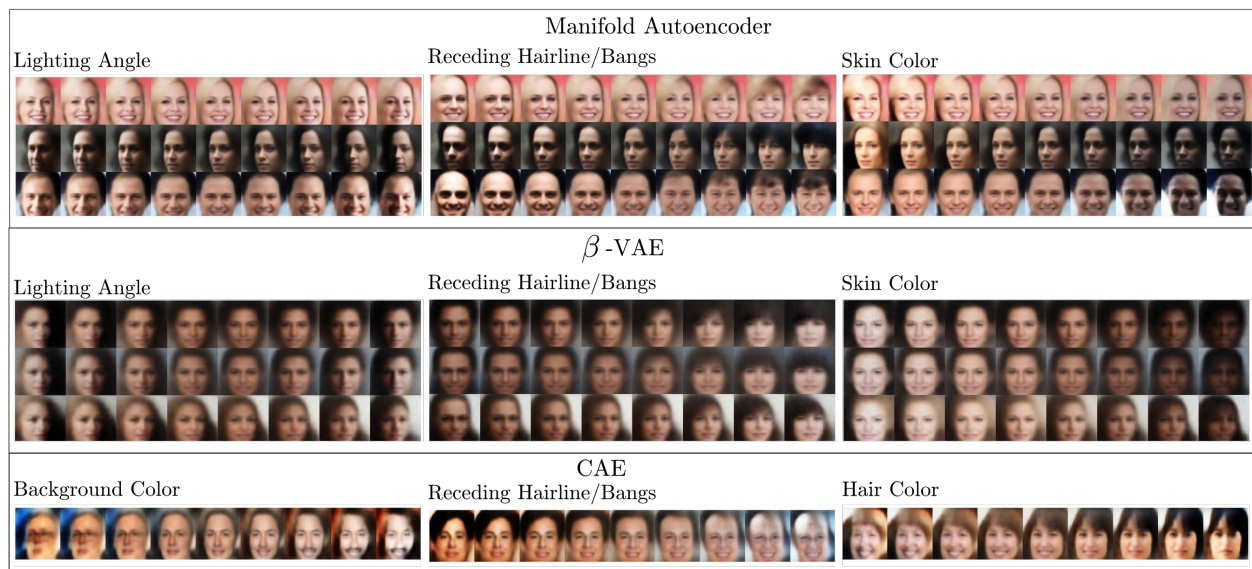


Figure 19: Transformations generated by applying a subset of learned transport operators on three datasets. In each figure, images in the middle column of the image block are the reconstructed inputs and images to the right and left are images decoded from transformed latent vectors in positive and negative directions, respectively. We compare transformations from learned transport operators to transformations generated by β -VAE and CAE. The transport operators learn semantically meaningful transformations similar to the disentangled β -VAE representation while maintaining higher reconstruction quality in image outputs.

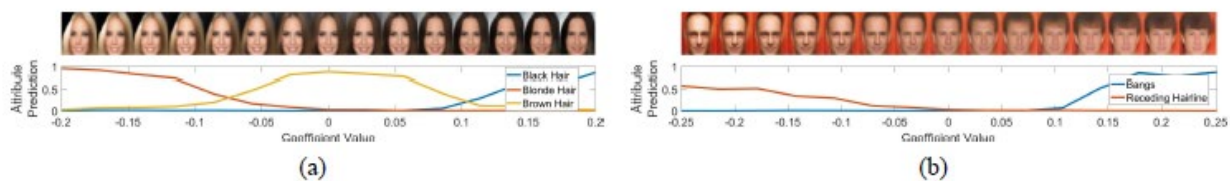


Figure 20: Paths generated by the application of two learned transport operators with the associated attribute classifier probability outputs for the transformed images. Our model learns operators that correspond to meaningful CelebA dataset attributes.

4.2.2 Variational Inference on Synthetic Dataset

To measure the effectiveness and speed-up of the variational sparse coding model, we evaluate on a synthetic dataset, comparing to transport operators trained using the procedure proposed in [36]. We train on 5000 points generated on a swiss roll manifold, using the fast iterative shrinkage-thresholding (FISTA) algorithm [44] for exact sparse inference as a baseline. At each iteration, we select 500 points and sample one of 60 nearest neighbors as the point pair. In total, we train for 1000 epochs and measure three key metrics, (1) the mean-squared error between transported point pairs m , (2) the cumulative runtime, and (3) the f_1 penalty to of coefficients to measure sparsity. For the baseline, we set $M = 6$, $\zeta = 0.6$, and $\gamma = 1e^{-3}$.

We compare FISTA to variational methods using both a standard Laplacian prior [143] and thresholded samples from a Laplacian [46], evaluating at different sample counts J . We use the same hyper-parameters as the FISTA baseline with a fixed prior and $\beta = 5e3$. The results are shown in Figure 21, where three main conclusions can be drawn: (1) as the number of samples drawn from the encoder increases (see [46] for details), variational methods match FISTA in MSE, (2) thresholding gives the same MSE as a standard Laplacian with much lower f_1 loss, (3) variational methods are approximately 16x faster in training walltime than FISTA. By adopting these changes, our team has been able to scale the transport operator model to datasets and dimensionalities that were not possible before. Furthermore, it can be seen that after training the two non-zero operators learned under the thresholded Laplacian can effectively generate other points on the swiss roll.

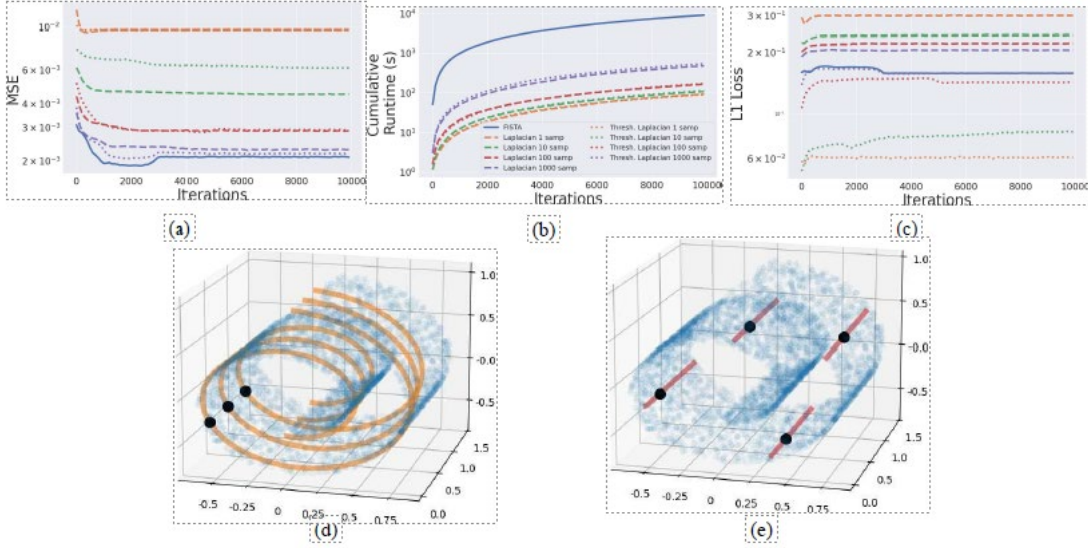


Figure 21: Comparison of transport operators trained with exact and variational sparse inference on synthetic swiss roll dataset. For each method, we plot (a) mean-squared error, (b) cumulative runtime, and (c) f_1 penalty of coefficients. (d,e) Extrapolated paths of the two operators learned by proposed variational sparse coding method. Black dots indicate starting point $\mathbf{x}_i$, with subsequent path found by applying transport operators $\mathbf{T}\Psi(\mathbf{c}_i)\mathbf{x}_i$ with coefficients from a range with fixed step size $\mathbf{c}_i = [-N_c, \dots, N_c]$.

Table 4: Average percent improvement in semi-supervised accuracy over baseline over 50 splits with varying datasets and methods for incorporating feature augmentations.

Method	CIFAR10	STL-10	TinyImageNet
Pseudo-labeling	$3.99 \pm 1.88\%$	$0.34 \pm 4.90\%$	$1.25 \pm 0.28\%$
Manifold Mixup ICT	$-1.46 \pm 0.48\%$	$-3.26 \pm 1.11\%$	$-0.01 \pm 0.11\%$
FeatMatch	$5.08 \pm 1.20\%$	$1.35 \pm 0.50\%$	$0.26 \pm 0.15\%$
<u>Transport Operators</u>	$8.61 \pm 2.57\%$	$5.47 \pm 2.37\%$	$1.58 \pm 0.30\%$

4.2.3 Learning with Limited Labels using Manifold Augmentations

To highlight the down-stream benefits of generating manifold feature augmentations in the ManifoldCLR system, we evaluate on a semi-supervised task using five labels per class and consistency regularization [51] in Figure 22. For each dataset, we freeze the weights of a backbone network after contrastive pre-training and train a single hidden-layer MLP with different methods for feature augmentations. Given labeled features (z^l, y^l) and unlabeled features z^u , we train a classifier r_θ using the following loss:

$$\min_{\theta} \frac{1}{B^l} \sum_{i=1}^{B^l} H(q_i^l, y_i^l) + \frac{1}{N^u} \sum_{b=1}^{B^u} \mathbb{1}[q_b^u \geq \tau] H(\tilde{q}_b^u, \tilde{y}_b^u).$$

Here, B^l and B^u are the batch size of the labeled and unlabeled dataset, respectively. $q_i^l = r_\theta(z_i^l)$ and $q_b^u = r_\theta(z_b^u)$ are the normalized logits from the classifier corresponding to class probabilities for the labeled and unlabeled features, respectively.

$N^u = \sum_{b=1}^{B^u} \mathbb{1}[q_b^u \geq \tau]$ is the number of unlabeled examples with confidence above τ and $H(\cdot)$ is the cross-entropy loss. We follow the methodology of [51] and use the class predictions of unlabeled data of which the classifier is sufficiently confident as pseudo-labels $\tilde{y}_b^u = \arg \max q_b^u$ for predictions on augmented examples $\tilde{q}_b^u = r_\theta(\tilde{z}_b^u)$. To get augmentations, we apply Lie group transformations $\tilde{z}_b^u = \mathbf{T}_\Psi(\tilde{\mathbf{c}}_b)z_b^u$ using coefficients sampled from learned prior $\tilde{\mathbf{c}}_b \sim p_\theta(\mathbf{c} | z_b^u)$. We focus our comparison to other methods for incorporating feature augmentations.

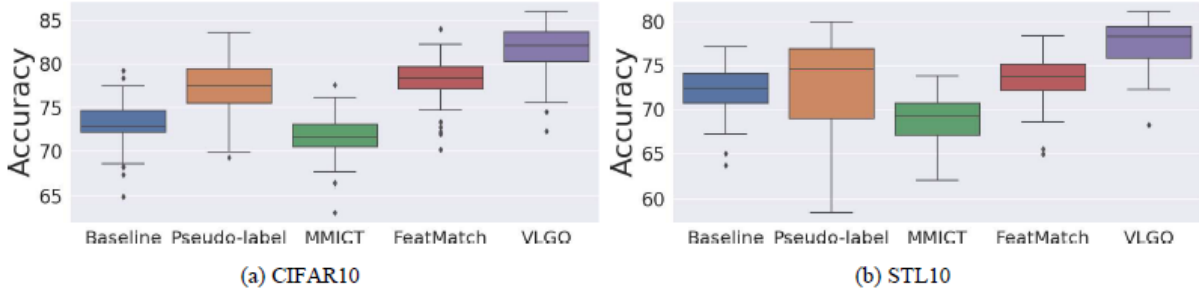


Figure 22: Semi-supervised experiments using a frozen backbone and learned, single hidden layer MLP. Comparison methods leverage unlabeled data to encourage consistency in classifier output after applying feature augmentations. Taken over the same 50 data splits used for each method. Variational Lie Group operators (VLGO) refer to the transport operator model with variational sparse coding.

We train each dataset and method with $B^l = 32$, $B^u = 480$ randomly sampled examples at each iteration, using a single hidden-layer MLP with 2048 hidden units. We train with the AdamW optimizer with a fixed learning rate and weight decay of $5e-4$. For CIFAR10 and STL10, we set $\tau = 0.95$. For TinyImagenet, we set $\tau = 0.7$. Furthermore, we evaluate using an exponential moving average (EMA) of our classifier weights at each training step with weight 0.999. We train each method for 5,000 iterations, except FeatMatch [13] which we train for 10,000 iterations to give the attention head sufficient iterations to learn meaningful augmentations.

To account for high variance in trial performance (e.g., certain splits of five labels per class are more challenging than others), we present the result with respect to percent improvement over a baseline that only uses a standard cross-entropy loss on labeled features in Table 4. We compare to pseudo-labeling without augmentations [144], FeatMatch [13], and Manifold Mixup interpolation consistency training (MMICT) [145]. For the former, we use all the labeled data as the prototypes for the attention-based

feature augmentation module with four attention heads. For the latter, we apply a random linear interpolation sampled from $U(0, 1)$ between two randomly sampled unlabeled points and apply an MSE loss between the Mixup of logits predicted by the EMA model and the logits of the interpolated features. This method is meant to compare the benefit of learning non-linear paths with the transport operators. Among these methods, it can be seen that the variational transport operators provide the most benefit in Figure 22. We find that the transport operators provide the most benefit in the case of few labels, especially on data splits where the baseline performs poorly.

4.3 Semi-Supervised Clustering and Retrieval for Zero-Shot Learning

4.3.1 Model architecture details

CLIP [63] has two encoders, CLIP-Image and CLIP-Text which are pre-trained transformer models for image and text. CLIP-Text is a base transformer model consisting of 12 layers, a hidden size of 768, and the final linear projection layer produces a representation vector of size 512. CLIP-Image is a hybrid ViT-Base model (which is the same as the DINO-trained model used for fair comparison) consisting of 12 stacked layers, with a convolutional layer in the beginning for feature extraction. For a given image, a total of 49 embedding vectors with a hidden size of 768 are generated, and to match the output of the CLIP-Text encoder; the output hidden state is projected from 768 to 512 dimensions. We fine-tune the last block of the vision transformer starting with a learning rate of $5e-5$ decaying it over time using a cosine annealed schedule. We train the model for 100 epochs using batches of size 128 and set the value of λ to 0.25 in the loss function above. Tuning and testing is done on a separate validation set to select the best hyperparameters.

4.3.2 Datasets & Evaluation

We evaluate the performance of our method on both generic image classification and fine-grained datasets. Following [54], we selected CIFAR-10/100 [149], ImageNet-100 [8], and Flowers102 [150] as the generic image classification datasets. We use CUB-200 [151], Stanford Cars [152], and FGVC-Aircraft [153] as fine-grained datasets. We also experiment with a challenging domain adaptation dataset DomainNet (Sketch) [5]. We split the training data into two parts, a labeled dataset and an unlabeled dataset by dividing all classes equally into seen classes and unseen ones, then sampling 50% images from

Table 5: Comparative results on generic image recognition datasets

Classes	CIFAR10			CIFAR100			ImageNet-100			Flowers-102		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
RankStats+ [146]	46.8	19.2	60.5	58.2	77.6	19.3	37.1	61.6	24.8	-	-	-
UNO+ [147]	68.6	98.3	53.8	69.5	80.6	47.2	70.3	95.0	57.9	-	-	-
GCD [54]	91.5	97.9	88.2	73.0	76.2	66.5	74.1	89.8	66.3	74.1	82.4	70.1
GCD w/ CLIP	95.9	97.0	95.8	84.2	83.1	82.3	79.3	94.6	71.1	67.8	82.3	60.5
XCon [148]	96.0	97.3	95.4	74.2	81.2	60.3	77.6	93.5	69.7	-	-	-
Ours	96.6	97.2	96.4	85.2	85.0	85.6	84.0	95.5	78.2	76.3	88.6	70.2

the seen classes as unlabeled data so that the unlabeled set D_U contains images from both seen classes and unseen classes, while the labeled set only contains seen classes. The splits are summarized in Table 6.

Table 6: Our dataset splits in the experiments. $(|Y_L|, |Y_U|)$ correspond to the number of classes in the labeled and unlabeled sets respectively. $(|D_L|, |D_U|)$ is the number of images for each set.

	CIFAR10	CIFAR100	CUB-200	SCARS
$ Y_L $	5	80	100	98
$ Y_U $	10	100	200	196
$ D_L $	12.5k	20k	1.5k	2.0k
$ D_U $	37.5k	30k	4.5k	6.1k
	ImageNet-100	DomainNet(Sketch)	Flowers-102	FGVC-Aircraft
$ Y_L $	50	172	51	50
$ Y_U $	10	345	102	100
$ D_L $	31.9K	10.1k	255	1.7k
$ D_U $	95.3k	38k	765	5k

Evaluation Metric To measure the performance of our model, we use the clustering accuracy (ACC) defined as below.

$$ACC = \max_{p \in P(Y_U)} \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{\{y_i = p(\hat{y}_i)\}}$$

where P is the set of all permutations that matches the model’s predictions $\hat{y}_i$ and the ground truth labels y_i using the Hungarian method [154] and N is the total number of images in the unlabeled set. Following [54], we use the metric on three different sets, ‘All’ which refers to the entire unlabeled set D_U , ‘Old’ referring to instances in D_U belonging to classes in Y_L , and ‘New’ referring to instances in D_U belonging to $Y_U \setminus Y_L$.

4.3.3 Comparison with the State-of-the-Art

We start by comparing our method with the SOTA methods on both generic image classification, fine-grained image classification, and domain adaptation benchmarks. RankStats+ [146] and UNO+ [147] are two methods modified from two competitive baselines for NCD and adopted to the GCD setting. XCon [148] is a method that targets fine-grained datasets in the GCD setting, lastly, GCD w/ CLIP is our proposed use of the GCD method with CLIP image encoder in lieu of DINO. The results on generic image recognition benchmarks are shown in Table 5. On all the datasets we experimented with, our method shows the best performance across most of the categories, often improving upon previous works with large margins. On ImageNet-100, CIFAR100, and Flowers102, our method outperforms the other methods on all subsets ‘All’, ‘Old’, and ‘New’, reinforcing the idea that our dual usage of multi-modal models boosts performance compared to vision only models. On the fine-grained image classification benchmarks, our results are presented in Table 7. We show the best performance of our method on all categories ‘All’, ‘Old’, and ‘New’ for most datasets while achieving comparable results for FGVC-Aircraft dataset. This indicates that our method is effective for fine-grained category discovery. On the domain adaptation classification front, our methods show the best results across all subsets ‘All’, ‘Old’, and ‘New’ on the DomainNet dataset, which indicates that our method is much more robust to distribution shift than standard ImageNet pre-trained models.

Table 7: Comparative results on SSB [4] and DomainNet [5]

Classes	Stanford Cars			FGVC-Aircraft			DomainNet (Sketch)			CUB-200		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
RankStats+ [146]	28.3	61.8	12.1	26.9	36.4	22.2	-	-	-	33.3	51.6	24.2
UNO+ [147]	35.5	70.5	18.6	40.3	56.4	32.2	-	-	-	35.1	49.0	28.1
GCD [54]	39.0	57.6	29.9	45.0	41.1	46.9	45.2	50.4	43.3	51.3	56.6	48.7
GCD w/ CLIP	62.8	85.2	52.0	43.7	52.8	39.2	52.7	74.2	43.7	59.7	76.1	51.5
XCon [148]	40.5	58.8	31.7	47.7	44.4	49.4	-	-	-	52.2	54.3	51.0
Ours	70.6	88.2	62.2	50.0	56.6	46.5	55.2	75.5	47.4	62.8	77.1	55.7

Table 8: Ablation study on CIFAR-10 with various amount of labeled samples.

Experiment	Image-Based Augmentation	Feature-Based Augmentation	$\mathcal{L}_{con-f}$	$\mathcal{L}_{con-g}$	$\mathcal{L}_{con}$	#Labeled samples		
						250	1,000	4,000
Baseline	✓	-	-	-	✓	19.55 ± 1.58	9.04 ± 1.00	6.08 ± 0.16
w/o $\mathcal{L}_{con-f}$	✓	✓	-	✓	-	18.57 ± 3.19	8.38 ± 0.35	6.09 ± 0.16
w/o $\mathcal{L}_{con-g}$	✓	✓	✓	-	-	8.19 ± 1.74	6.07 ± 0.46	5.16 ± 0.30
FeatMatch (Ours)	✓	✓	✓	✓	-	7.90 ± 0.49	5.94 ± 0.16	5.00 ± 0.21

4.3.4 Analysis

We analyze the contribution of certain aspects of our methodology through a rigorous ablation study. Specifically, we highlight the significance of the following components of the approach: whether language supervision can result in vision models with transferable representation versus classic image-only models, the effect of the number of texts k retrieved per image on the accuracy of the model, retrieved text quality, and CLIP image encoder ViT backbone with and without finetuning.

How important is language supervision in this setting? Table 8 shows the effect of language on the clustering task. The *Image Encoder* column represents different types of vision transformer backbones. GCD is a finetuned ViT-B-16 backbone with DINO [57] pre-trained weights from GCD [54] and CLIP [63] is a finetuned pre-trained ViT-B-16 backbone. The *Knowledge* columns indicate whether we are clustering vision-only features or vision and text features combined. We record the accuracy of the model across all categories, All, Old, and New for three datasets, then average them for each combination of dataset, image encoder, and knowledge. As shown, the results indicate that CLIP image and text outperform image-only by a large margin, confirming that language does help in this setting compared to image-only models. We note that while using CLIP as an encoder without retrieval is an extremely strong baseline, our retrieval mechanism further improves performance by significant margins e.g. almost 4% on All and almost 6% specifically on Old.

How important is the descriptiveness of retrieved captions? Text descriptions in typical datasets can vary in terms of how they relate to the image. Ideally, we want to encode salient objects in the image that are meaningful in representation learning for object recognition tasks. The learned representations for contrastive models are governed by the text transformer (captions for CLIP), suggesting that text descriptions that describe the contents of a scene in an image will improve transferability in the CLIP model. We verify this hypothesis and quantify the *descriptiveness* of a caption using multiple caption data sources. We perform top-4 cross-modal retrieval from Conceptual Captions (3M) [68], Conceptual Captions (12M) [69], and COCO [70], and LION [71], then record the accuracy of the model for each data corpus on *All*, *Old*, and *New* subsets averaged for each knowledge database.

Table 9 shows the results of the model on three datasets CIFAR100, Stanford Cars, and DomainNet(Sketch). Previous work in linguistics has shown that captions that are descriptive (meant to replace an image) are different from those that are complementary or give additional information and context. Contrary to LAION and Conceptual Captions (12M) which usually contain information

complementary to the image, Conceptual Captions (3M) and MS COCO are more descriptive due to the strict annotation process. We use a score given by CLIP of a caption matching its corresponding image in our cross-modal retrieval, and according to the results, the hypothesis does not align with our subjective assessment, at least for the datasets tested. We posit that the descriptiveness of the captions retrieved from a corpus and the size of the knowledge database, as well as diversity of captions, all play a role.

Table 9: Accuracy of the model using different knowledge databases as a source of text descriptions

Dataset	Knowledge DB	All	Old	New
CIFAR-100	CC-12M	85.9	85.0	88.1
CIFAR-100	CC-3M	82.8	82.6	83.2
CIFAR-100	MSCOCO	85.1	85.5	84.2
CIFAR-100	LAION-400M	82.0	82.6	80.8
CIFAR-100	LAION-5B	82.5	83.4	80.6
Stanford Cars	CC-12	70.9	89.3	62.0
Stanford Cars	CC-3M	63.8	85.1	53.5
Stanford Cars	MSCOCO	62.4	85.5	51.2
Stanford Cars	LAION-400M	66.1	86.7	56.1
Stanford Cars	LAION-5B	71.2	89.4	64.5
Sketch	CC-12	54.7	74.6	47.4
Sketch	CC-3M	55.2	76.8	47.6
Sketch	MSCOCO	55.2	78.2	47.2
Sketch	LAION-400M	53.8	76.1	45.3
Sketch	LAION-5B	54.6	77.3	46.9
Average	CC-12M	70.5	83.0	65.8
Average	CC-3M	67.3	81.5	61.4
Average	MSCOCO	67.7	83.1	60.9
Average	LAION-400M	67.3	81.8	60.7
Average	LAION-5B	69.4	83.0	64.0

How many captions do we need to retrieve for each image? We probe how the variability of captions within a caption database affects our model transfer capabilities. There are a number of ways to annotate an image as shown in Figure 11. In each corpus, captions vary in terms of how an object is described e.g. "train" or "railcar", and which part of the image the focus is on, e.g. "cloud" or "bird". The focus, lexical, and style variation in captioning could confuse the model and make it push image-text pairs apart instead of pulling them together. We examine the sensitivity of our model to the number of captions per image (top-k), averaging accuracy across three datasets, CIFAR100, Stanford Cars, and DomainNet (Sketch), and we chose to limit retrieval of captions to Conceptual Captions (12M)[69]. Figure 11 suggests that variability in dataset captions can hurt the accuracy of the model. They suggest that some of the captions might not contain useful information making the model accuracy plateau or even reduce after a certain number.

Does CLIP need finetuning? One of the most impressive aspects of the CLIP model is its performance in zero-shot learning, classifying objects it has never seen before, based on their descriptions in natural language. In this experiment, we probe CLIP’s performance in the GCD setting without performing any finetuning. Table 10 shows our results for a CLIP model finetuned versus a model without finetuning on three datasets, CIFAR100, Stanford Cars, and Sketch with a finetuned CLIP outperforming a non-finetuned CLIP model. Recent studies have shown that CLIP finetuning might distort its pretrained representation leading to unsatisfactory performance, but our results show that it can be finetuned with the right hyperparameter choices, challenging the notion that CLIP is not suitable for finetuning.

4.4 Retrieval for Low-Labeled Image Captioning

4.4.1 Initial Method Results

We train our proposed method with cross-entropy loss first and then optimize the CIDEr score with SCST loss [75] following recent image captioning works [74, 77, 102]. The results are shown in Tab. 11. It is worth noting that our method does *not* pre-train the transformer image captioning model on external large-scale image-and-text datasets and instead trains the model from scratch only on the MS-COCO [80] image captioning dataset. In this setting, our proposed method outperforms previous SoTA Xmodal-Ctx [3] by 2.6% in CIDEr and 1.5% in BLEU-4. On the other hand, when compared with models with transformer pre-training, our method, despite being only trained on 10x less data, achieves comparable performance.

Table 10: Results for finetuned CLIP vs. not finetuned

Dataset	Finetuned CLIP	All	Old	New
CIFAR-100	N	68.7	71.1	63.7
CIFAR-100	Y	85.2	85.0	85.6
Stanford Cars	N	65.8	78.9	59.5
Stanford Cars	Y	70.6	88.2	62.2
Sketch	N	51.6	60.0	48.5
Sketch	Y	55.2	75.5	47.4
Average	N	62.0	70.0	57.2
Average	Y	70.3	82.9	65.1

Table 11: Image captioning results on the test set of MS-COCO Karpathy split [6].

Method	Transformer Pre-training	B-4	M	C	S
VLP [155]	Y (3M)	39.5	29.3	129.3	23.2
Oscar [77]	Y (6.5M)	40.5	29.7	137.6	22.8
VinVL [78]	Y (8.8M)	40.9	30.9	140.4	25.1
AoANet huang2019attention	N	38.9	29.2	129.8	22.4
M ² cornia2020meshed	N	39.1	29.1	131.2	22.6
CLIP-ViL [2]	N	40.2	29.7	134.2	23.8
Xmodal-Ctx [3]	N	39.7	30.0	135.9	23.7
Ours	N	41.1	30.1	138.5	23.9

Table 12: Image captioning results on the test set of MS-COCO Karpathy split [6]. Since our HAAV is trained from scratch, we separately compare methods with large-scale transformer pre-training on the top block, and those trained from scratch on the bottom block. Our HAAV outperforms previous trained-from-scratch methods by a large margin and achieves comparable or often better performance compared to methods with large-scale transformer pre-training.

Method	Pretrain Data	B-4	M	C	S
VLP [155]	3M	39.5	29.3	129.3	23.2
X-VLM [156]	16M	40.4	-	139.3	-
Oscar [77]	6.5M	40.5	29.7	137.6	22.8
VinVL [78]	8.8M	40.9	30.9	140.4	25.1
GIT [157]	4M	41.3	30.4	139.1	24.3
ViTCap [158]	4M	41.2	30.1	138.1	24.1
SCST [75]	None	34.2	26.7	114.0	-
Up-Down [1]	None	36.3	27.7	120.1	21.4
AoANet [73]	None	38.9	29.2	129.8	22.4
M ² [74]	None	39.1	29.1	131.2	22.6
CLIP-ViL [2]	None	40.2	29.7	134.2	23.8
Xmodal-Ctx [3]	None	39.7	30.0	135.9	23.7
HAAV (ours)	None	41.0	30.2	141.5	23.9

4.4.2 Improved HAAV Results

In Table 12, we show the results of our HAAV on the test set of MS-COCO Karpathy split [6]. It is worth noting that HAAV is **not** pre-trained on external image-and-text corpus and instead is trained from scratch only on the MS-COCO image captioning dataset [92]. Therefore, for a fair comparison, we separate these two different settings in Table 12, where methods on the top block use pre-training while those on the bottom are trained from scratch. In the same trained-from-scratch setting, our HAAV outperforms previous state-of-art Xmodal-Ctx [3] by 5.6% in CIDEr and 1.3% in BLEU-4. On the other hand, when compared with methods with transformer pre-training, our HAAV, despite being only trained on MS-COCO, achieves comparable or often better performance.

To further demonstrate the label efficiency of HAAV, we train it on the Karpathy split [6]

of Flickr30K, which only has 0.15M training data (4x smaller than MS-COCO), and show the results in Table 13. The results of other methods are taken from [161]. We can see that our HAAV outperforms previous state-of-art ORT substantially by 12.9% in CIDEr and 4.3% in BLEU-4. The significant improvement may come from the label efficiency of our method, which is particularly important when trained on the smaller-scale dataset of Flickr30K.

Table 13: Image captioning results on a smaller-scale Flickr30K Karpathy split [6]. We also demonstrate semi-supervised learning (SSL) for image captioning with labeled data from Flickr30K and unlabeled data from MS-COCO. Compared to previous state of the arts, our HAAV achieves significant performance improvement, which may come from the label efficiency of our method.

Method	B-4	M	C	S
Show & Tell [102]	21.5	18.3	41.7	12.2
Show, Attend & Tell [159]	23.6	19.2	49.1	13.3
Up-Down [1]	28.3	21.6	63.3	15.9
M ² [74]	29.8	22.4	68.4	16.2
ORT [160]	30.1	22.8	68.8	16.9
HAAV (ours)	34.3	24.6	81.7	18.0
HAAV + SSL (ours)	34.3	25.1	85.6	19.0

4.5 Semi-Supervised Object Detection through Cross-Match

Implementation Details. Our implementation of CrossMatch is based on Detectron2 [162]. For fair comparison, we follow previous work[7, 112, 113, 111] to use Faster-RCNN [116] with ResNet-50-FPN backbone. The backbone weights are initialized with ImageNet-pretrained weights. Unless otherwise noted, all our experiments are conducted with batch size of 16 (the ratio between labeled and unlabeled data is 1:1). We follow Unbiased Teacher for data augmentation operations. We use random horizontal flip for weak augmentation, and strong augmentation involves color jittering, grayscale, Gaussian blur, and cutout [163]. We do not include further augmentation operations such as geometric transformation as in STAC or MixUp [164] and Mosaic [165] as in Instant-Teaching. More details of hyper-parameters can be found in Appendix A.2.

All of our experiments are conducted with 8 Nvidia-V100 GPUs. Though CrossMatch has some additional computation overheads (during training only) due to the multi-level consistency, the training process can be completed within 3 days (about two times longer than vanilla for 180k iterations), which is acceptable in most scenarios.

4.5.1 Results on MS COCO

We compare our method with other methods on MS COCO [80] dataset. We randomly sample different percentage of labeled training data as the labeled set and treat the rest of the data as the unlabeled set. All of our experiments are conducted with 5 runs with different sampling seed and we report the average and standard deviation of these runs. We also re-ran Unbiased Teacher [7] with batch size of 16 for fair comparison using their publicly released codebase since all other methods report their results with batch size 16. Results are evaluated on MS COCO 2017 *val* set.

Table 14 shows the results comparing between CrossMatch and other state-of-the-art methods on MS

COCO. Our method consistently outperforms other methods with 0.5, 1, 2, 5% of labeled training data. It is worth noting that CrossMatch works particularly well with limited labeled data. Specifically, CrossMatch achieves 16.81 mAP and 20.63 mAP with only 0.5% and 1% labeled data respectively, which advances previous state-of-the-art results by around 2 mAP. In contrast, Humble Teacher [113] has the best result with 10% but does not perform well with low percentages of labeled images.

Table 14: Experimental results on COCO-standard comparing with other state-of-the-art methods. We reproduced the results of Unbiased Teacher using the officially released codebase. *: results are re-implemented by [7].

	COCO-standard				
	0.5%	1%	2%	5%	10%
Supervised	6.83 $\pm$ 0.15	9.05 $\pm$ 0.16	12.70 $\pm$ 0.15	18.47 $\pm$ 0.22	23.86 $\pm$ 0.81
CSD [166]*	7.41 $\pm$ 0.21 (+0.58)	10.51 $\pm$ 0.06 (+1.46)	13.93 $\pm$ 0.12 (+1.23)	18.63 $\pm$ 0.07 (+0.16)	22.46 $\pm$ 0.08 (-1.40)
STAC [111]	9.78 $\pm$ 0.53 (+2.95)	13.97 $\pm$ 0.35 (+4.92)	18.25 $\pm$ 0.25 (+5.55)	24.38 $\pm$ 0.12 (+5.86)	28.64 $\pm$ 0.21 (+4.78)
Humble Teacher [113]	-	16.96 $\pm$ 0.38 (+7.91)	21.72 $\pm$ 0.24 (+9.02)	27.70 $\pm$ 0.15 (+9.23)	31.61 $\pm$ 0.28 (+7.74)
Instant Teaching [112]	-	18.05 $\pm$ 0.15 (+9.00)	22.45 $\pm$ 0.15 (+9.75)	26.75 $\pm$ 0.05 (+8.28)	30.40 $\pm$ 0.05 (+6.54)
ISMT [114]	-	18.88 $\pm$ 0.74 (+9.83)	22.43 $\pm$ 0.56 (+9.73)	26.37 $\pm$ 0.24 (+7.9)	30.53 $\pm$ 0.52 (+6.67)
Unbiased Teacher [7]	14.36 $\pm$ 0.09 (+7.53)	18.33 $\pm$ 0.19 (+9.28)	22.23 $\pm$ 0.21 (+9.53)	26.65 $\pm$ 0.31 (+8.18)	29.56 $\pm$ 0.24 (+5.7)
CrossMatch (In the manuscript)	16.81 $\pm$ 0.26 (+9.98)	20.63 $\pm$ 0.29 (+11.58)	23.98 $\pm$ 0.08 (+11.28)	28.01 $\pm$ 0.18 (+9.54)	30.72 $\pm$ 0.08 (+6.86)
CrossMatch (Updated)	17.33 $\pm$ 0.18 (+10.50)	21.53 $\pm$ 0.39 (+12.48)	24.74 $\pm$ 0.21 (+12.04)	28.77 $\pm$ 0.24 (+10.3)	31.78 $\pm$ 0.18 (+7.92)

4.5.2 Results on Pascal VOC

We further compare CrossMatch with other methods on Pascal VOC. Following prior work [166, 111], we use VOC07 as the labeled set and VOC12 (plus COCO-20) as unlabeled set. Since AP_{50} is indicated as a saturated metric by existing work [167], we further include $AP_{50:95}$ as an additional metric for comparison. Our results are evaluated on VOC07 *test* set.

As shown in Table 15, CrossMatch outperforms CSD, Unbiased Teacher and ISMT in terms of both AP_{50} and $AP_{50:95}$, which indicates its generalizability across multiple datasets. CrossMatch achieves 8.24 and 9.87 absolute mAP improvement over the supervised baseline when using VOC12 and VOC12 plus COCO20 as unlabeled set respectively.

Notice that we only include methods that have the similar supervised baselines to ours for fair comparison. Since the semi-supervised setting for Pascal VOC uses the entire VOC07 dataset as labeled data, the training settings of supervised baseline can significantly impact the results of semi-supervised methods. Most of other methods with different baselines do not explicitly mention their training settings or publish their code making it hard to reproduce their supervised baselines.

4.6 Active Learning

We perform experiments on active image classification in a supervised setting, using NN queries to acquire labels iteratively. Info-NN-distances is used in these experiments.

Label selection and experimental framework. To select samples using Info-NN, for every unlabelled sample, we form the corresponding nearest neighbor query and compute an estimate of the information gain provided by that query. We then request a label for the unlabelled sample corresponding to the most informative query. A simple batch extension of our query acquisition strategy, which performs a clustering of the unlabelled samples in the embedding space and selects the most informative samples from every cluster, is used in the experiments. *Info-NN-C* means the batch variant of Info-NN was used to

select NN queries of length C . We use Euclidean distances between the features learned by the last hidden layer to compute distances for the probability model. We experimented with the length of the queries and illustrate plots for the best performing values. We plot the median and 25% and 75% quantiles over 3 trials.

We conduct experiments on the MNIST, CIFAR-10 and SVHN datasets using CNNs to demonstrate the performance of our active learning method with supervised classification. The experiments on MNIST have an initial balanced labelled set of 30 samples, chosen at random and an acquisition batch of size 10 is used. For CIFAR-10 and SVHN, we start with initial balanced labelled sets of 5000 and 3000 respectively, and acquire batches of size 5000 and 3000. We compare the performance of Info-NN with BatchBALD [168], K -Center [126], MaxEntropy, and Random methods.

Experiment results. While our method outperforms all the baselines on MNIST, on CIFAR-10 and SVHN, it performs almost on par with MaxEntropy. A possible explanation here is that the embeddings learned might be of lower quality. The embeddings directly affect the informativeness and diversity measures in our method, thereby impacting the quality of labels.

Table 15: Results on VOC comparing with other methods.

	Backbone	Labeled	Unlabeled	AP_{50}	$AP_{50:95}$
Supervised (ours)	ResNet50-FPN	VOC07	None	76.3	42.6
CSD [166]	ResNet101-R-FCN			74.70 (+2.07)	-
STAC	ResNet50-FPN			77.45 (+1.15)	44.64 (+2.04)
ISMT [114]	ResNet50-FPN			77.23 (+4.6)	46.23 (+4.1)
Unbiased Teacher [7]	ResNet50-FPN	VOC07	VOC12	75.95 (+2.32)	48.86 (+6.73)
Instant Teaching	ResNet50-FPN			79.2 (+2.9)	50.0 (+7.4)
Humble Teacher	ResNet50-FPN			80.94 (+4.64)	53.04 (+10.44)
CrossMatch (Ours)	ResNet50-FPN			81.48 (+5.62)	55.36 (+12.76)
CSD [166]	ResNet101-R-FCN			75.10 (+2.47)	-
STAC	ResNet50-FPN			79.08 (+1.15)	46.01 (+2.04)
ISMT [114]	ResNet50-FPN		VOC12	77.75 (+5.12)	49.59 (+7.46)
Unbiased Teacher [7]	ResNet50-FPN	VOC07	+	77.52 (+4.89)	49.71 (+7.58)
Instant Teaching	ResNet50-FPN		<i>COCO20cls</i>	79.9 (+3.6)	50.8 (+8.2)
Humble Teacher	ResNet50-FPN			81.29 (+4.99)	54.41(+11.81)
CrossMatch (Ours)	ResNet50-FPN			82.62 (+6.32)	56.52 (+13.92)

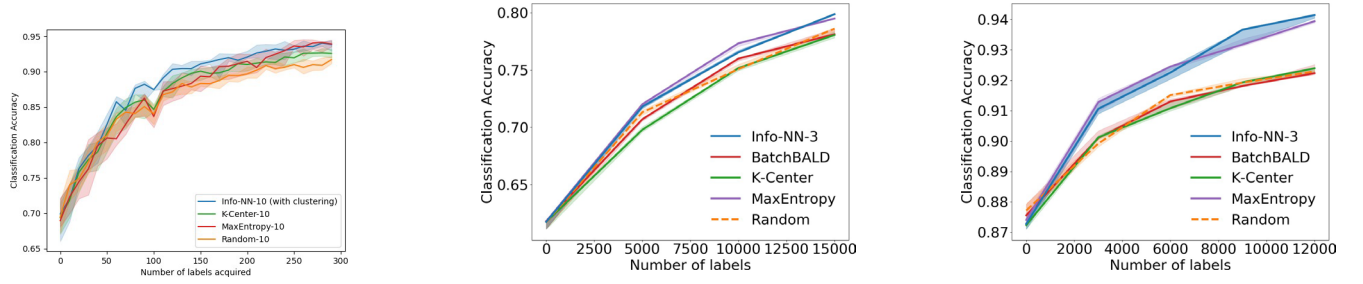


Figure 23: Active classification performance comparison on MNIST (left), CIFAR-10 (center) and SVHN (right) datasets.

chosen. Using pre-trained networks instead could result in better quality embeddings. We do not explore this direction here but it is an interesting avenue for future work.

4.6.1 Info-NN applied to metric learning

As an example, we applied the *Info-NN* framework to active metric learning. Learning an embedding or representation of the data that accurately reflects similarity between items is the goal of metric learning. We directly query an oracle for perceptual similarity information, as is done in [169], where triplets of the form “Is item B or item C more similar to item A?” are actively selected for learning an embedding of items. Active deep metric learning (DML) builds on this idea by finding the most informative queries to ask the oracle.

We conducted experiments on a food dataset consists of 72, 148 crowdsourced triplets gathered for 73 different food items [170]. We utilize a neural network to learn an embedding that matches the oracle’s responses to similarity queries. Because a length C NN query can be decomposed into C 1 triplets, we utilize a triplet loss [171]. To measure the performance of our embedding learning algorithm, we use *triplet generalization accuracy*, which records the fraction of test triplets whose ordering is consistent with the learned embedding. A sample result is illustrated in Fig. 24.

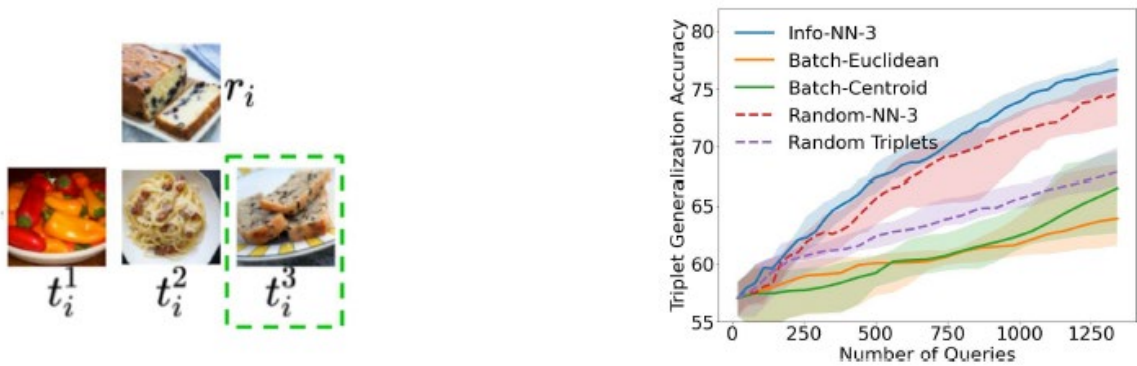


Figure 24: (Left) An example query in the case of metric learning. An oracle is presented with a reference item and a set of items to report similarity. The green dashed item indicates the oracle’s response. (Right) Performance comparison of Info-NN with baseline active methods and random selection. Our method outperforms all the other methods and NN queries exhibit far superior per-query performance, requiring less interactions with the oracle.

5.0 Conclusions

Across this program, we have developed a range of methods tackling zero-shot, semi-supervised, and low-labeled image classification, image captioning, and object detection. Our methods have been strongly cited and influenced the field, while at the same time producing real-world impact through artifacts such as open-source code. Specifically, we developed the following methods:

1. **Leveraging the Manifold-Structure through FeatMatch [13]** - Novel methods for graph (manifold)-based augmentation for semi-supervised image classification.
2. **Leveraging the Manifold-Structure through Transport Operators [14, 15, 16, 17, 18]** - A series of methods for learning manifold transport operators in an efficient manner on real-world data, integrated within semi-supervised learning pipelines to improve performance.
3. **Semi-Supervised Clustering and Retrieval for Zero-Shot Learning [19]** - Novel methods for integrating semi-supervised clustering and retrieval, within large-scale pre-trained feature spaces, to perform effective zero-shot learning.
4. **Low-Labeled Captioning through Retrieval [20, 21]** - Methods for retrieval-based augmentation of images, which can be seen as additional *views* of the data, and efficient and effective architectures for exploiting them for low-labeled image captioning.
5. **Semi-Supervised Object Detection through Cross-Match [22]** - Methods for improving gradient flow in existing semi-supervised object detection methods to significantly improve performance.
6. **Advancements in Active Learning [23]** - Unified view of active learning with connections to deep metric learning, and a novel method Info-NN for selecting queries using information-theoretic criterion.

During this time, other advancements have also changed the field. Specifically, large-scale pre-trained models (i.e. Foundation Models) have recently shown that the range of methods we have utilized (combinations of self-supervised/contrastive learning, clustering, pseudo-labeling, and data selection) can be scaled up to produce emergent properties including strong performance and robustness to out-of-distribution data. Such advancements have generated a great deal of excitement, and indeed pose significant new opportunities to even further push the envelope on low-labeled learning.

6.0 References

- [1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom- up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.
- [2] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.
- [3] Chia-Wen Kuo and Zsolt Kira. Beyond a pre-trained object detector: Cross-modal textual and visual context for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [4] Poojan Oza and Vishal M. Patel. C2ae: Class conditioned auto-encoder for open-set recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2302–2311, 2019.
- [5] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.
- [6] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.
- [7] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. *arXiv preprint arXiv:2102.09480*, 2021.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [9] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.
- [10] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *International Conference on Learning Representations (ICLR)*, 2018.
- [11] Yen-Chang Hsu and Zsolt Kira. Neural network-based clustering using pairwise constraints. *International Conference on Learning Representations (ICLR) workshop*, 2016.
- [12] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [13] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi- supervised learning. In *European Conference on Computer Vision*, pages 479–495. Springer, 2020.
- [14] John Lee, Max Dabagia, Eva Dyer, and Christopher Rozell. Hierarchical optimal transport for multimodal distribution alignment. *Advances in neural information processing systems*, 32, 2019.

- [15] Marissa Connor and Christopher Rozell. Representing closed transformation paths in encoded network latent space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3666–3675, 2020.
- [16] Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In *International Conference on Artificial Intelligence and Statistics*, pages 2359–2367. PMLR, 2021.
- [17] Kion Fallah and Christopher J Rozell. Variational sparse coding with learned thresholding. *ICML*, 2022.
- [18] Marissa Connor, Kion Fallah, and Christopher Rozell. Learning identity-preserving transformations on data mani- folds. In *Transactions on Machine Learning Research*, 2023.
- [19] Rabah Ouldnooghi, Chia-Wen Kuo, and Zsolt Kira. Clip-gcd: Simple language guided generalized category discovery. *arXiv preprint arXiv:2305.10420*, 2023.
- [20] Chia-Wen Kuo and Zsolt Kira. Beyond a pre-trained object detector: Cross-modal textual and visual context for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17969–17979, 2022.
- [21] Chia-Wen Kuo and Zsolt Kira. Haav: Hierarchical aggregation of augmented views for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11039–11049, 2023.
- [22] Zhuoran Yu, Yen-Cheng Liu, Chih-Yao Ma, and Zsolt Kira. Crossmatch: Improving semi-supervised object detection via multi-scale consistency. 2021.
- [23] Namrata Nadagouda, Austin Xu, and Mark A Davenport. Active metric learning and classification using similarity queries. In *Uncertainty in Artificial Intelligence*. PMLR, 2023.
- [24] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [26] Vikas Verma, Alex Lamb, Christopher Beckham, Aaron Courville, Ioannis Mitliagkis, and Yoshua Bengio. Manifold mixup: Encouraging meaningful on-manifold interpolation as a regularizer. *stat*, 1050:13, 2018.
- [27] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remix- match: Semi-supervised learning with distribution alignment and augmentation anchoring. In *Proc. International Conference on Learning Representations (ICLR)*, 2020.
- [28] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.
- [29] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

- [30] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.
- [31] Eric Arazo, Diego Ortego, Paul Albert, Noel E O’Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv preprint arXiv:1908.02983*, 2019.
- [32] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [34] Rajesh Rao and Daniel Ruderman. Learning Lie Groups for Invariant Visual Perception. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1998.
- [35] Xu Miao and Rajesh P. N. Rao. Learning the Lie Groups of Visual Invariance. *Neural Computation*, 19(10):2665–2693, October 2007.
- [36] Benjamin J Culpepper and Bruno A Olshausen. Learning transport operators for image manifolds. In *Advances in Neural Information Processing Systems*, pages 423–431, 2009.
- [37] Jascha Sohl-Dickstein, Ching Ming Wang, and Bruno A Olshausen. An unsupervised algorithm for learning lie group transformations. *arXiv preprint arXiv:1001.1027*, 2010.
- [38] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, volume 222 of *Graduate Texts in Mathematics*. Springer International Publishing, Cham, 2015.
- [39] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, December 1997.
- [40] Marissa Connor and Christopher Rozell. Representing closed transformation paths in encoded network latent space. In *AAAI Conference on Artificial Intelligence*, 2019.
- [41] Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2359–2367. PMLR, 13–15 Apr 2021.
- [42] Diane Bouchacourt, Mark Ibrahim, and Stéphane Deny. Addressing the topological defects of disentanglement via distributed operators, 2021.
- [43] Romain Cosentino, Randall Balestriero, Richard Baraniuk, and Behnaam Aazhang. Deep autoencoders: From understanding to generalization guarantees. 2020.
- [44] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, January 2009.
- [45] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2013.

- [46] Kion Fallah and Christopher J Rozell. Variational sparse coding with learned thresholding. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6034–6058. PMLR, 17–23 Jul 2022.
- [47] David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, September 1994.
- [48] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK. page 22, 2017.
- [49] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv:1308.3432 [cs]*, August 2013. arXiv: 1308.3432.
- [50] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *arXiv:1711.00937 [cs]*, May 2018. arXiv: 1711.00937.
- [51] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.
- [52] Kion Fallah, Alec Helbling, Kyle Johnsen, and Christopher Rozell. Manifold contrastive learning with variational lie group operators. In Submission, 2023.
- [53] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. *ArXiv*, abs/1711.10125, 2018.
- [54] Sagar Vaze, K. Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2022.
- [55] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20, 2006.
- [56] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *ArXiv*, abs/2001.07685, 2020.
- [57] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021.
- [58] Umar Khalid, Ashkan Esmaeili, Nazmul Karim, and Nazanin Rahnavard. Rodd: A self-supervised approach for robust out-of-distribution detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 163–170, 2022.
- [59] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Xiaodong Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019.

- [60] Andreas Furst, Elisabeth Rumetshofer, Viet-Hung Tran, Hubert Ramsauer, Fei Tang, Johannes Lehner, David P. Kreil, Michael Kopp, Günter Klambauer, Angela Bitto-Nemling, and Sepp Hochreiter. Cloob: Modern hopfield networks with infoloob outperform clip. *ArXiv*, abs/2110.11316, 2021.
- [61] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Ney Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.
- [62] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Super-vision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *ArXiv*, abs/2110.05208, 2022.
- [63] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [64] Yunchao Gong, Liwei Wang, Micah Hodosh, J. Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 2014.
- [65] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task. *Journal of Artificial Intelligence Research*, 2013.
- [66] K. Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8400–8408, 2019.
- [67] J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.
- [68] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.
- [69] Soravit Changpinyo, Piyush Kumar Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3557–3567, 2021.
- [70] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- [71] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *ArXiv*, abs/2111.02114, 2021.
- [72] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA '07*, 2007.
- [73] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4634–4643, 2019.
- [74] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020.

- [75] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024, 2017.
- [76] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 499–515, 2018.
- [77] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.
- [78] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021.
- [79] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [80] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [81] 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*, pages 10012–10022, 2021.
- [82] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021.
- [83] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.
- [84] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.
- [85] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020.
- [86] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.
- [87] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

- [88] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021.
- [89] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [90] Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1459, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [91] Jiayi Ji, Yunpeng Luo, Xiaoshuai Sun, Fuhai Chen, Gen Luo, Yongjian Wu, Yue Gao, and Rongrong Ji. Improving image captioning by leveraging intra-and inter-layer global representation in transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1655–1663, 2021.
- [92] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.
- [93] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34, 2021.
- [94] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021.
- [95] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1439–1449, 2021.
- [96] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *Advances in Neural Information Processing Systems*, 33:25–37, 2020.
- [97] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.
- [98] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.
- [99] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. Vision-language pre-training with triple contrastive learning. 2022.
- [100] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

- [101] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning?, 2020.
- [102] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.
- [103] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.
- [104] Eric Arazo, Diego Ortego, Paul Albert, Noel E O’Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.
- [105] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.
- [106] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019.
- [107] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.
- [108] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. 2005.
- [109] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2017.
- [110] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.
- [111] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.
- [112] Qiang Zhou, Chaohui Yu, Zhibin Wang, Qi Qian, and Hao Li. Instant-teaching: An end-to-end semi-supervised object detection framework. *arXiv preprint arXiv:2103.11402*, 2021.
- [113] Yihe Tang, Weifeng Chen, Yijun Luo, and Yuting Zhang. Humble teachers teach better students for semi-supervised object detection.
- [114] Qize Yang, Xihan Wei, Biao Wang, Xian-Sheng Hua, and Lei Zhang. Interactive self-training with mean teachers for semi-supervised object detection.
- [115] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
- [116]

- [117] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.
- [118] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.
- [119] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- [120] Yandong Li, Di Huang, Danfeng Qin, Liqiang Wang, and Boqing Gong. Improving object detection with selective self-supervised self-training. In *European Conference on Computer Vision*, pages 589–607. Springer, 2020.
- [121] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [122] Gregory Canal, Andy Massimino, Mark Davenport, and Christopher Rozell. Active embedding search via noisy paired comparisons. In *International Conference on Machine Learning*, pages 902–911. PMLR, 2019.
- [123] G. Canal, S. Fenu, and C. Rozell. Active ordinal querying for tuplewise similarity learning. In *Proc. AAAI Conf. on Artificial Intelligence (AAAI)*, Honolulu, Hawaii, February 2019.
- [124] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.
- [125] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. Sydney, Australia, 2017.
- [126] Mingfei Gao, Zizhao Zhang, Guo Yu, Serkan ÖArık, LarryDavis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. Online, August 2020.
- [127] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. Vancouver, Canada, April 2018.
- [128] Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. Vancouver, Canada, December 2019.
- [129] Heather Turner, Jacob van Etten, David Firth, and Ioannis Kosmidis. Introduction to plackettluce, 2018.
- [130] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. Bellevue, WA, June 2011.
- [131] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.
- [132] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016.

- [133] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8896–8905, 2018.
- [134] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Semi-supervised deep learning with memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–283, 2018.
- [135] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–152, 2018.
- [136] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Improving consistency-based semi-supervised learning with weight averaging. *arXiv preprint arXiv:1806.05594*, 2, 2018.
- [137] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [138] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- [139] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.
- [140] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [141] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
- [142] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.
- [143] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *International Conference on Machine Learning*, 2011.
- [144] Gabriel Barello, Adam S. Charles, and Jonathan W. Pillow. Sparse-Coding Variational Auto-Encoders. preprint, Neuroscience, August 2018.
- [145] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.
- [146] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation Consistency Training for Semi-Supervised Learning. *arXiv:1903.03825 [cs, stat]*, May 2019. arXiv: 1903.03825.
- [147] K. Han, Sylvestre-Alvise Rebuffi, Sébastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Autonovel: Automatically discovering and learning novel visual categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:6767–6781, 2022.

- [148] Enrico Fini, E. Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9264–9272, 2021.
- [149] Yixin Fei, Zhongkai Zhao, Si Xiao Yang, and Bingchen Zhao. Xcon: Learning with experts for fine-grained category discovery. *ArXiv*, abs/2208.01898, 2022.
- [150] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.
- [151] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.
- [152] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [153] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.
- [154] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *ArXiv*, abs/1306.5151, 2013.
- [155] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52, 2010.
- [156] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049, 2020.
- [157] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*, 2021.
- [158] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.
- [159] Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lin Liang, Zhe Gan, Lijuan Wang, Yezhou Yang, and Zicheng Liu. Injecting semantic concepts into end-to-end image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18009–18019, 2022.
- [160] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.
- [161] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. *arXiv preprint arXiv:1906.05963*, 2019.
- [162] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: a survey on deep learning-based image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [163] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.

- [164] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
- [165] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
- [166] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [167] Jisoo Jeong, Seungeui Lee, Jeessoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. 2019.
- [168] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.
- [169] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. Vancouver, Canada, December 2019.
- [170] Priyadarshini Kumari, Ritesh Goru, Siddhartha Chaudhuri, and Subhasis Chaudhuri. Batch decorrelation for active metric learning. Online, January 2020.
- [171] Michael Wilber, Iljung Kwak, and Serge Belongie. Cost-effective hits for relative similarity comparisons. In *Proc. AAAI Conf. on Human Computation and Crowdsourcing (HCOMP)*, Pittsburgh, PA, November 2014.
- [172] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. Vancouver, Canada, December 2006.

Appendix A - Publications and Presentations

Publications

- [1] Namrata Nadagouda, Austin Xu, and Mark A Davenport. Active metric learning and classification using similarity queries. In *Uncertainty in Artificial Intelligence*. PMLR, 2023.
- [2] Marissa Connor, Kion Fallah, and Christopher Rozell. Learning identity-preserving transformations on data manifolds. In *Transactions on Machine Learning Research*, 2023.
- [3] Chia-Wen Kuo and Zsolt Kira. Haav: Hierarchical aggregation of augmented views for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11039–11049, 2023.
- [4] Kion Fallah and Christopher J Rozell. Variational sparse coding with learned thresholding. *ICML*, 2022.
- [5] Chia-Wen Kuo, Chih-Yao Ma, Judy Hoffman, and Zsolt Kira. Structure-encoding auxiliary tasks for improved visual representation in vision-and-language navigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1104–1113, 2023.
- [6] Chia-Wen Kuo and Zsolt Kira. Beyond a pre-trained object detector: Cross-modal textual and visual context for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17969–17979, 2022.
- [7] Zhuoran Yu, Yen-Cheng Liu, Chih-Yao Ma, and Zsolt Kira. Cross-match: Improving semi-supervised object detection via multi-scale consistency. 2021.
- [8] Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In *International Conference on Artificial Intelligence and Statistics*, pages 2359–2367. PMLR, 2021.
- [9] Marissa Connor and Christopher Rozell. Representing closed transformation paths in encoded network latent space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3666–3675, 2020.
- [10] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Feat-match: Feature-based augmentation for semi-supervised learning. In *European Conference on Computer Vision*, pages 479–495. Springer, 2020.
- [11] John Lee, Max Dabagia, Eva Dyer, and Christopher Rozell. Hierarchical optimal transport for multimodal distribution alignment. *Advances in neural information processing systems*, 32, 2019.
- [12] Gregory Canal, Andy Massimino, Mark Davenport, and Christopher Rozell. Active embedding search via noisy paired comparisons. In *International Conference on Machine Learning*, pages 902–911. PMLR, 2019.
- Zsolt Kira, **University of Texas at San Antonio**, April 2022, *Handling Distribution Shift in Visual Learning. Improving Label Efficiency and Generalization for Vision and Language*.

Presentations

- Zsolt Kira, **IRIM Symposium**, October 2021, *Adaptive and Label-Efficient Robot Learning*.
- Zsolt Kira, **SURE Robotics**, July 2021, *Leveraging the Low-Labeled Revolution for Real-World Robotics*.
- Zsolt Kira, **Georgia Tech/Google Robotics Workshop**, May 2021, *Leveraging the Low-Labeled Revolution for Real- World Robotics*.
- Zsolt Kira, **2nd workshop on Unsupervised Learning for Automated Driving**, October 2020, *Learning using Limited Labels and Robust Multi-Modal Fusion*.
 - Zsolt Kira, **Google**, July 2020, *Improving Label Efficiency and Generalization for Vision and Language*.
- Zsolt Kira, **CVPR 2020 Workshop on Visual Learning with Limited Labels**, June 2020, *Improving Label Efficiency Through Complex Data Augmentation, and Beyond*.
 - Zsolt Kira, **HRL Laboratories**, June 2019, *Beyond Supervised Learning: Learning with Less Data and More Goals*.
 - Zsolt Kira, **Magic Leap**, February 2019, *Beyond Supervised Learning for Interactive ML Experiences*.
 - All accepted conference papers were presented as either poster or oral during the conference dates.

List of Symbols, Abbreviations, and Acronyms

CLIP - Contrastive Language-Image Pre-Training
CNN - Convolutional Neural Network
COCO - Common Objects in Context
DML - Deep Metric Learning
DNN - Deep Neural Network
GCD - Generalized Category Discovery
HAAV - Hierarchical Aggregation of Augmented Views
NCD - Novel Class Discovery
NN - Neural Network
LwLL – Learning with Less Labels
GCNN – Graph Convolutional Neural Network
TA – Technical Area (LwLL Program defined TA’s)
SSL – Self-Supervised Learning
CVPR – Computer Vision & Pattern Recognition
ECCV – European Conference on Computer Vision
NeurIPS – Neural Information Processing Systems
ICML – International Conference on Machine Learning
AAAI - Association for the Advancement of Artificial Intelligence
VAI
TMLR – Transactions on Machine Learning Research
RESNet – Residual Neural Networks
CIFAR - Canadian Institute for Advanced Research
VAE – Variational Autoencoder
CVAE – Convolutional Variational Autoencoder
VAELLS - Variational Autoencoder with Learned Latent Structure
ELBO - Evidence lower bound
AVI - Amortized variational inference