Transferring Pre-trained Multimodal Representations with Cross-modal Similarity Matching

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite surprising performance on zero-shot transfer, pre-training a large-scale multimodal model is often prohibitive as it requires a huge amount of data and computing resources. In this paper, we propose a method (**BeamCLIP**) that can effectively transfer the representations of a large pre-trained multimodal model (CLIP ViT-B/16) into a small target model (ResNet-50). For unsupervised transfer, we introduce *cross-modal similarity matching* (CSM) that enables a student model to learn the representations of a teacher model by matching the relative similarity distribution across text prompt embeddings. To better encode the text prompts, we design *context-based prompt augmentation* (CPA) that can alleviate the lexical ambiguity of input text prompts. Our experiments show that unsupervised representation transfer of a large pre-trained multimodal model can provide good ImageNet linear probe accuracy (74.8%), while outperforming the existing self-supervised learning methods (SimCLR: 65.6%) and closing the gap with supervised learning (76.2%).

1 Introduction

Learning transferable representations is crucial for successful downstream applications. Contrastive learning such as SimCLR and MoCo-v2 have shown notable success by forcing features of individual classes to be clustered and sufficiently scattered [38]. But their downstream performances are still far behind the supervised training as shown in Figure 1. Recently, large-scale vision and language pre-trained (VLP) models provide highly transferable visual representations via language supervision. However, learning VLP models from the scratch is prohibitive as it requires large amounts of training data and computing resource. For example, training CLIP [31] requires 400M paired image-text data and several hundreds of GPUs. ALIGN [21] further scales up to leverage alternative texts specified for descriptions of web images. While these models are often based on large Transformers [37], small ConvNets such as ResNet-50 [17] and MobileNet [20] are still widely used in practice [1] and even more crucial for low-resource environments. We reformulate represen-

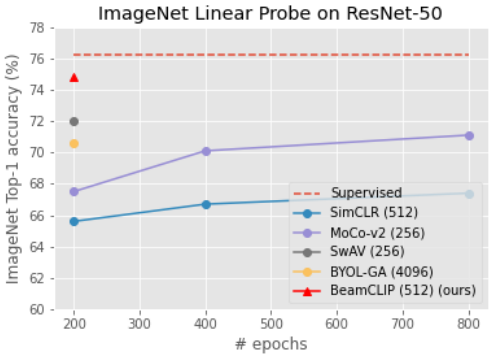

Figure 1: ImageNet top-1 accuracy of linear probe on ResNet-50. By transferring the representations of CLIP ViT-B/16 [31], the BeamCLIP achieves 74.8%. It outperforms SimCLR [4] (65.6%) and reduces the gap to the supervised learning (76.2%). The numbers in parentheses denote the batch sizes.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

tation learning in terms of knowledge transfer from a large pre-trained model to a small practical model.

Large pre-trained models exhibit strong alignments between different modalities. CLIP learns visual concepts from natural language supervision, mapping image and text into the same vector space. As their training data is not only huge but inaccessible, however, conventional knowledge distillation based on instance similarity is no longer a viable option. Instead, we propose *cross-modal similarity matching* (CSM). Imagine your goal is to learn a high quality representation for an input dog image in CIFAR-10 as in Figure 2. Since CLIP was trained on numerous image-caption pairs, angular distances from the dog image embedding to the caption embeddings of other anchor prompt texts such as "a photo of cat" or "a photo of horse" must comprehensively preserve their visual differences. By training the student to preserve angular relations witnessed from the teacher, our model achieves near benchmark performance without accessing the original data for training CLIP.

To measure the effects of different amounts of supervisions, we design *context-based prompt augmentation* (CPA) that can alleviate the lexical ambiguity of the input text prompts. We find that lexical ambiguity in prompt texts can lead to semantically incorrect text embeddings. This may result in unexpected discrepancies of image-text alignment in the teacher's embedding space. Also, it is known that the zero-shot performance of CLIP can be improved by designing task-specific prompt texts. Inspired by this, we design CPA that extends the basic prompt of CLIP to better encode prototypical anchor representations.

Our experimental result shows that our BeamCLIP achieves the strongest and near benchmark performance on ImageNet-1K Top-1 linear probe accuracy when using most popular ResNet-50 as the student network. We also compare the effectiveness of BeamCLIP against zero-shot transfer learning. Further, we provide ablation study results to show how much each component contribute to the performance.

Contributions of this paper can be summarized as follows:

- We propose a method (**BeamCLIP**) that can effectively transfer the representations of a large pre-trained multimodal model (CLIP ViT-B/16) into a small target model (ResNet-50). To achieve this, we introduce *cross-modal similarity matching* (CSM) and *context-based prompt augmentation* (CPA). (Figure 2).
- We empirically show that BeamCLIP can provide favorably good ImageNet linear probe accuracy (74.8%), while outperforming the state-of-the-art self-supervised methods (SimCLR: 65.6%) and closing the gap with supervised learning (76.2%). (Figure 1 and Table 2).
- We also show that BeamCLIP (from CLIP-ViT to ResNet50) can significantly improve zero-shot transfer learning accuracy, especially when the zero-shot accuracy (CLIP-RN50) is relatively week. (Table 3 and Table 4).

2 Related Work

Vision and language pre-trainig. Vision and language pre-training (VLP) aims to jointly learn vision and language representations that can be transferred to the downstream tasks such as visual question answering (VQA), image captioning, and vision and language navigation (VLN). There are BERT-based vision and language models such as VLBERT [32], ViLBERT [25], and UNITER [7]. Also, there are contrastive learning-based models such as CLIP [31] and ALIGN [21]. These models use contrastive loss [28] to learn aligned vision and language representations by performing a task of matching a large-scale image and text pairs. The BeamCLIP aims to transfer the rich representations of large-scale vision and language pre-trained models such as CLIP and ALIGN to a small target model.

Self-supervised learning. Self-supervised learning (SSL) aims to learn highly transferable representations by using unlabeled data. In computer vision, at the early stage, task-specific self-supervised methods were introduced. These include Context Prediction [11], Rotation Prediction [14], and Colorization [40]. More recently, contrastive learning-based methods were introduced as a task-agnostic approach. These include SimCLR [4] and MoCo-v2 [18]. However, since contrastive self-supervised methods require a large batch size, non-contrastive methods have been introduced. These include SwAV [3], BYOL [16], and SimSiam [5]. In this paper, we empirically show that the BeamCLIP can

provide better visual representations than the state-of-the-art SSL methods by leveraging a large-scale pre-trained multimodal model.

Knowledge distillation. Knowledge distillation (KD) [19] aims to transfer rich knowledge from a strong teacher model to a target student model. In a conventional setting, it encourages the student model to mimic the task-specific prediction of the teacher model. As the student model is trained to predict the same probability distribution over pre-defined classes as the teacher model's, using Kullback-Leibler (KL) divergence is a natural metric to measure the error between the two models. For a classification task, the loss function can be formulated as follows:

$$\mathcal{L}_{\text{KD}} = \sum_i H(p_i, q_i^S) + \sum_i KL(p_i^T || p_i^S). \tag{1}$$

The first term indicates the supervised loss, where p_i denotes the one-hot labels and $H(p, q)$ denotes cross-entropy. The second term is the distillation loss, where p_i^T and p_i^S are the softmax predictions of the teacher and student models, respectively.

Similarity-based knowledge distillation. Recently, similarity-based knowledge distillation such as SEED [13], OSS [8], and ISD [34] was introduced in the context of self-supervised learning (SSL). SEED [13] showed that the linear probe accuracy of a small student (ResNet-18) can be improved by transferring the representations of a larger teacher (ResNet-50) pre-trained by SSL methods such as MoCo-v2 [6]. Unlike this, OSS [8] aims to transfer representations of an evolving teacher (ResNet-50) into a smaller student (ResNet-18) on the fly. Unlike SEED and OSS, ISD [34] considered the same size student and teacher network (ResNet-18), and showed a student can learn visual representations by iterativly distilling the similarity of teacher's representations. These works are closely related to our work. Unlike these works, the BeamCLIP aims to transfer rich vision and language representations of large-scale pre-trained models such as CLIP-ViT/16 [31] into a smaller network such as ResNet-50.

Prompt engineering. Recently, researchers showed that prompt engineering [2] is surprisingly effective at improving the performance of large-scale language models (LLMs) on downstream tasks without fine-tuning. Prompts are input texts of language models that usually consists of a task description or several examples. To further simplify prompt engineering, prompt tuning [23] proposed to add k learnable tokens to the input texts, while having language models frozen. Similar to GPT-3, it is known that the zero-shot performance of CLIP [31] can be improved by designing the prompt texts to each task. For example, on satellite image classification datasets, `"A satellite photo of a {label}"` provides better performance than the default `"A photo of a {label}"`. Inspired by this, we propose context-based prompt augmentation that extends the basic prompt of CLIP to better encode prototypical text anchor representations by alleviating the lexical ambiguity of class label texts.

3 Method

Problem formulation. Formally, our problem is to transfer aligned cross-modal representations of a strong teacher model $f_\phi^T(\cdot)$ into a target student model $f_\theta^S(\cdot)$ with unlabeled data $\mathcal{D}_u = \{x_i\}_{i=1}^N$. Given each unlabeled image x_i, we formulate representation transfer as a regression task that matches teacher representations $f_\phi^T(x_i)$ to a student's $f_\theta^S(x_i)$. As the student network is parameterized by θ, the learning objective is

$$\arg\min_\theta \sum_i^N \|f_\theta^S(x_i) - f_\phi^T(x_i)\|_2^2. \tag{2}$$

Normalizing the representations via l_2-normalization (*i.e.,* $q_i = f_\theta^S(x_i)/\|f_\theta^S(x_i)\|_2$ and $k_i = f_\phi^T(x_i)/\|f_\phi^T(x_i)\|_2$) leads to the following simplification:

$$\arg\min_\theta \sum_i^N \|q_i - k_i\|_2^2 = \arg\min_\theta \sum_i^N (2 - 2q_i \cdot k_i). \tag{3}$$

The problem now involves maximizing the cosine similarity between l_2-normalized representations from teacher and student models.

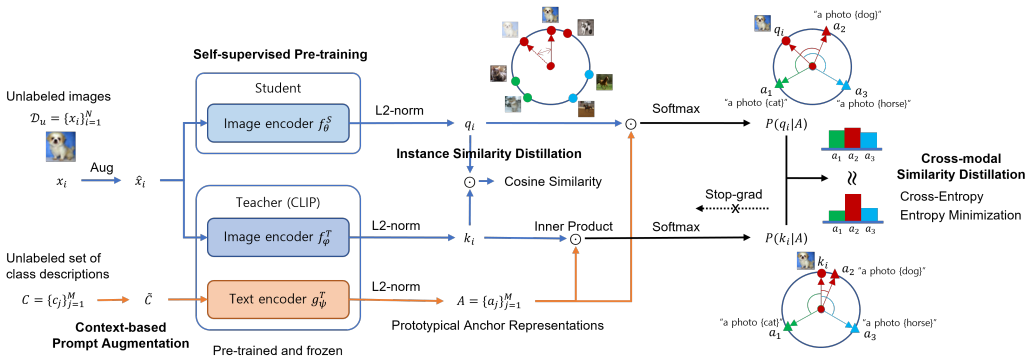

Figure 2: **Overview of the BeamCLIP.** Representation transfer can be viewed as a task in which, given a query input, a student model learns to regress a vector representation of a teacher model. The BeamCLIP first measures the normalized cross-modal similarity of the query image compared to anchor text representations in the teacher's embedding space. Then, it encourages the student to mimic the same cross-modal similarity in the student's embedding space. To better align image representations, our method uses self-supervised pre-training of the student model. Finally, to avoid text ambiguity, we uses context-based prompt augmentation.

Method overview. The overview of **BeamCLIP** is shown in Figure 2. The teacher model of the BeamCLIP consists of an image encoder $f_\phi^T(\cdot)$ and a text encoder $g_\psi^T(\cdot)$. These encoders are pre-trained under a simple task of matching images to texts with large-scale corpora. Image representations $f_\phi^T(x_i)$ and text representations $g_\psi^T(t_i)$ are thus well-aligned within a cross-modal embedding space. We provide the details of the BeamCLIP in the following sections. More specifically, we describe how to extend the basic problem setting by leveraging the unique features of CLIP where vision and language representations are precisely aligned. Throughout the paper, we use the notation CLIP-ViT/16 to denote the CLIP [31] model that uses Vision Transformer (ViT) [12] with the patch size of 16x16 as the image encoder. Similar to this, CLIP-RN50 denotes the CLIP model with ResNet-50 [17] as the image encoder.

3.1 Similarity-based cross-modal representation transfer

To effectively distill cross-modal representations, we use similarity-based matching as described above. Our similarity-based representation transfer utilizes two carefully designed loss functions: (1) instance similarity matching (ISM) loss and (2) cross-modal similarity matching (CSM) loss.

Instance similarity matching. This objective is directly derived from Eq. 3. Given a query image x_i, it encourages the student image encoder $f_\theta^S(\cdot)$ to regress the representation of the teacher image encoder $f_\phi^T(\cdot)$. We apply conventional image augmentations (see Appendix B) on a query image x_i, and the same augmented image $\hat{x}_i$ is fed to both the teacher and student image encoders. Given unlabeled query images $\mathcal{D}_u = \{x_i\}_{i=1}^N$, it is formulated as follows:

$$\mathcal{L}_{\text{ISM}} = -\sum_{i=1}^{N}\left(\frac{f_\theta^S(\hat{x}_i)}{\|f_\theta^S(\hat{x}_i)\|_2} \cdot \frac{f_\phi^T(\hat{x}_i)}{\|f_\phi^T(\hat{x}_i)\|_2}\right) = -\sum_{i=1}^{N}(q_i \cdot k_i). \tag{4}$$

However, the similarity signal from a single instance is not enough to constraint the student representations. For example, the topological ambiguity may occur in image encoding, since two symmetric representations have the same cosine similarity compared to a single teacher representation. We conjecture that this can be mitigated by incorporating multiple anchor points to the query points. Based on this idea, we introduce cross-modal similarity matching loss.

Cross-modal similarity matching. To better align a student representation q_i with the teacher representation k_i, we introduce cross-modal similarity matching (CSM) loss. We use multiple anchor points to cope with the ambiguity problem mentioned above. Further, we use text representations as

Table 1: Examples of context-based prompt augmentation for ambiguous class labels on Flowers102.

Label Index	Label Name	Text Prompt
7	bird of paradise	"A photo of {bird of paradise}. {Strelitzia is a genus of five species of perennial plants, native to South Africa. It belongs to the plant family Strelitziaceae}."
10	snapdragon	"A photo of {snapdragon}. {Antirrhinum is a genus of plants commonly known as dragon flowers or snapdragons because of the flowers' fancied resemblance to the face of a dragon that opens and closes its mouth when laterally squeezed}."

anchor points, since we can easily generate prototypical anchor points by using text prompts and class labels. Since image and text representations are precisely aligned in CLIP, we can effectively apply this approach. More specifically, the BeamCLIP first measures the normalized image-text similarity of the query image compared to prototypical text points in the teacher's embedding space. Then, it encourages the student to mimic the same image-text similarity in the student's embedding space.

More formally, we generate multiple anchor representations $A = \{a_j\}_{j=1}^{M}$ by encoding class texts $C = \{c_j\}_{j=1}^{M}$ with the teacher text encoder $g_\psi^T(\cdot)$ (in other words, $a_j = g_\psi^T(c_j)$). To measure the similarity regarding to multiple anchor representations A, we define the normalized cross-modal similarity as follows:

$$s_j(k_i, A) = \frac{\exp\left((k_i \cdot a_j)/\tau\right)}{\sum_{m=1}^{M} \exp\left((k_i \cdot a_m)/\tau\right)} \tag{5}$$

where τ is a temperature hyperparameter that is set to 0.01 in our experiments.

Then, we evaluate the cross-modal similarity distribution by using a set of normalized cross-modal similarities:

$$P(k_i|A) = [s_1(k_i, A), ..., s_M(k_i, A)]. \tag{6}$$

Then, the student model is optimized to mimic the normalized cross-modal similarity of the teacher's embedding space by minimizing the cross entropy, *i.e.,* $H(P(k_i|A), P(q_i|A))$.

We further minimize the entropy of normalized cross-modal similarities in the student embedding space *i.e.,* $H(P(q_i|A))$. This minimization helps the student provide query representations q_i that are more attracted to anchor representations $A = \{a_j\}_{j=1}^{M}$. This entropy minimization is also known to be effective in other domains such as semi-supervised learning [15].

Altogether, the CSM loss is formulated as follows:

$$\mathcal{L}_{\text{CSM}} = \sum_{i=1}^{N} H(P(k_i|A), P(q_i|A)) + \sum_{i=1}^{N} H(P(q_i|A)). \tag{7}$$

Final Loss. The final loss of the BeamCLIP is formulated as follows:

$$\mathcal{L}_{\text{BeamCLIP}} = \mathcal{L}_{\text{CSM}} + \lambda_{\text{ISM}} \mathcal{L}_{\text{ISM}} \tag{8}$$

where λ_{ISM} is the scale hyperparameter that is set to 10 in our experiments.

3.2 Context-based prompt augmentation

We found that lexical ambiguity in prompt texts can lead to semantically incorrect text embeddings. This may result in an unexpected discrepancy of image-text alignment in the teacher's embedding space. For example, Flowers102 [27] dataset has some classes with unusual and ambiguous flower names, such as "snapdragon", "bird of paradise", and "colt's foot".[1] Therefore, incorrect prototypical anchor points might be compared with a query image. To address this issue of semantic ambiguities in the text, we introduce context-based prompt augmentation (CPA), a data-driven approach that augments basic prompts with contextual text such as Wikipedia descriptions or hierarchical labels.

[1]Examples can be found at: 102 Category Flower Dataset.

For prompt tuning with Wikipedia descriptions, we use the template `"A photo of a {label}. {Wikipedia description}"`. We use this template for Flowers102 and Pets37 in our experiments. We provide some examples from the Flowers102 dataset in Table 1. For prompt tuning with hierarchical labels, we use the template `"A photo of a {fine label}, categorized as {coarse label}"`. We use this template for CIFAR100 and ImageNet in our experiments. Analogous examples from CIFAR100 can be found in Table 8 of Appendix C.

3.3 Other details

Self-supervised pre-training of student. To help the student mimic the teacher's cross-modal embedding space better, we pre-train the student image encoder with a self-supervised method. Since self-supervised pre-training such as SimCLR [4] and MoCo-V2 [18] provides a weakly clustered embedding space based on similarities, it can be used as a better initial state for the student to mimic the teacher's embedding space.

For self-supervised pre-training, we adopt SimCLR, since it is simple and effective. SimCLR learns transferable visual representations by using InfoNCE loss [29, 36] which encourages agreement between multiple views of the same image. More specifically, InfoNCE maximizes the similarity between multiple views of the same image (*i.e.,* positive samples) and minimizes the similarity to multiple views of all other images in a training batch (*i.e.,* negative samples). InfoNCE loss of SimCLR can be formulated as follows:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp\left((h_i^S \cdot h_{i'}^S)/\tau\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} \exp\left((h_i^S \cdot h_k^S)/\tau\right)} \tag{9}$$

where $h_i^S \in \mathbb{R}^{128}$ is a projection of a student representation $q_i \in \mathbb{R}^{512}$, τ is a temperature hyperparameter that is set to 0.1, $\mathbb{1}_{[k\neq i]}$ is an indicator function whose value is 1 if $k \neq i$, and B is a batch size. Here, h_i and $h_{i'}$ are projections of multiple views of the same input images x_i.

Optimization. For optimization we use SGD with cosine annealing schedule (SGDR) [24]. To stabilize training, we use a momentum encoder that updates its weights via exponential moving average (EMA) [18, 16]. The momentum encoder of a student $\theta_{\hat{S}}$ is updated using the following rule:

$$\theta_{\hat{S}} \leftarrow m\theta_{\hat{S}} + (1-m)\theta_S \tag{10}$$

where θ_S is the image encoder of a student model and m is a momentum hyperparameter that is set 0.99 in our experiments.

4 Experiments

Downstream datasets. We evaluate the BeamCLIP on six standard benchmark datasets: CIFAR10 [22], CIFAR100 [22], STL10 [9], Flowers102 [27], Pets37 [30], and ImageNet-1K [10]. Following convention, we split the datasets into train, validation, and test sets. Then, we use train set for transfer, and test set for evaluation. For ImageNet, we use the validation set as a test set, since its test set does not provide labels. More details on the datasets can be found in Appendix A.

4.1 Representation transfer

ImageNet linear probe accuracy. We compare the BeamCLIP with various self-supervised methods in terms of linear probe accuracy on ImageNet-1K. Following the conventional protocol, we use ResNet-50 [17] as the base encoder and evaluate the learned representations by using logistic regression. We use LBFGS algorithm [41] for logistic regression. Its hyperparameter C is determined through coarse-grained hyperparameter search on the validation split. And, the accuracy is evaluated in the test split. We found that it provides the best linear probe accuracy when C is set to 100. We perform our experiments on 8 NVIDA A100 GPUs and it takes about 30 hours for 200 epoch training.

Table 2 shows the comparison of the BeamCLIP with vision-only self-supervised methods such as SimCLR [4], MoCo-v2 [6], SwAV [3], BYOL [16], and SimSiam [5]. The BeamCLIP provides better visual representation by achieving 74.8% top-1 linear probe accuracy on ImageNet-1K [10]. While self-supervised methods take long training epochs to achieve comparable accuracy, BeamCLIP-RN50

Table 2: **Comparison of linear probe accuracy on ImageNet-1K.** We compare the BeamCLIP with vision-only self-supervised methods in terms of linear probe accuracy on ImageNet-1K. The BeamCLIP representations provide higher linear probe accuracy than self-supervised methods. This means better transferability. We denote each row with gray color in case the number of epochs is greater than 200.

Type	Method	Img. Enc.	Teacher	Text	Batch	Epochs	Top-1
	Supervised	RN50	✗	✗	256	100	76.2
Self-Supervised Learning	SimCLR [4]	RN50	✗	✗	512	200	65.6
	SimCLR	RN50	✗	✗	512	400	66.7
	SimCLR	RN50	✗	✗	512	800	67.4
	SimCLR	RN50	✗	✗	4096	200	66.8
	SimCLR	RN50	✗	✗	4096	800	69.1
	MoCo-v2 [6]	RN50	✗	✗	256	200	67.5
	MoCo-v2	RN50	✗	✗	256	400	70.1
	MoCo-v2	RN50	✗	✗	256	800	71.1
	SwAV [3]	RN50	✗	✗	256	200	72.0
	SwAV	RN50	✗	✗	256	400	74.3
	BYOL-GA [16]	RN50	✗	✗	4,096	200	70.6
	BYOL-GA	RN50	✗	✗	4,096	1,000	74.3
Vision-Language Pre-training	CLIP [31]	RN50	✗	✓	32,768	32	73.3
	CLIP	ViT-B/16	✗	✓	32,768	32	80.2
Unsupervised Representation Transfer	**BeamCLIP** (ours)	RN50	CLIP	✗	512	200	**74.8**
	Δ SimCLR	-	-	-	-	-	+9.2
	Δ CLIP-RN50	-	-	-	-	-	+1.5

Table 3: **Comparison of teacher and student accuracy on various datasets.** Conventional knowledge distillation with the KL divergence is not effective in transferring CLIP-ViT representations. In contrast, the BeamCLIP effectively transfers CLIP-ViT/16 representations into ResNet-50 by using unlabeled query data. Compared to zero-shot CLIP-RN50, the BeamCLIP improves about 6.57% accuracy on ImageNet-1K. We denoted with $*$ in cases our student model outperforms the teacher model.

Method	Model Type	Img. Enc.	Param. Size	CIFAR10	CIFAR100	STL10	Flowers102	Pets37	ImageNet-1K
CLIP [31] (zero-shot)	T	ViT-B/16	76M	91.6	**68.7**	**98.2**	70.4	**88.9**	**68.6**
CLIP [31] (zero-shot)	-	RN50	24M	75.6	41.6	94.3	65.9	85.4	59.6
CRD [35]	S	RN50	24M	76.78	38.83	81.41	26.15	62.22	30.90
KD [19]	S	RN50	24M	90.89	58.02	93.28	49.01	77.51	56.38
BeamCLIP (ours)	S	RN50	24M	**92.10***	**67.35**	**97.45**	**75.86***	**86.94**	**66.17**
Δ CLIP-RN50	-	-	-	+16.4	+25.78	+3.15	+9.96	+1.54	+6.57
Δ Teacher (%)	-	-	-	100.5	98.0	99.2	107.7	97.8	96.5

achieves better accuracy with less training epochs. Also, note that BeamCLIP-RN50's representations provide better accuracy than CLIP-RN50's representations.

4.2 Transfer learning

Setting. We evaluate how effectively the BeamCLIP transfers CLIP-ViT representations into a student model by evaluating classification accuracy on various datasets (see Table 3). We choose ResNet-50 [17] as the student model to compare the distilled target model with CLIP-RN50. Also,

ResNet-50 is conventionally used in evaluating linear probe accuracy on ImageNet-1K. However, the BeamCLIP can adopt any architecture, not just ResNet-50.

As baselines, we choose two representative distillation methods among many methods: (1) conventional knowledge distillation (KD) [19] and (2) contrastive representation distillation (CRD) [35]. Since conventional KD aims to mimic the task-specific predictions of the teacher model unlike the BeamCLIP , we apply the KL divergence on minimizing the cross-modal similarity distribution (*i.e.,* $P(q_i|A)$ and $P(k_i|A)$), instead of the Cross-entropy (CE). CRD proposes a variant of InfoNCE loss for representation distillation, which we apply on normalized representations (*i.e.,* q_i and k_i). The details of each method can be found in the related work section 2.

Results. Table 3 shows a comparison of teacher and student accuracy on various datasets. We empirically demonstrate that the BeamCLIP can effectively transfer vision and language representations of a large teacher model (CLIP-ViT/16) into a small student model (ResNet-50), while achieving comparable or better classification accuracy. We find that the KL divergence used in conventional knowledge distillation (KD) is not effective in transferring CLIP-ViT representations. Also, the contrastive learning-based approach is not effective. Unlike this, the BeamCLIP can effectively transfer CLIP-ViT/16 representations into ResNet-50, achieving very high accuracy that is comparable or better than the teacher accuracy. For example, for ImageNet-1K, when compared to zero-shot CLIP-RN50, BeamCLIP-RN50 provides about 10.5% better accuracy on the same size model. When compared to CLIP-ViT/16, BeamCLIP-RN50 provides about 1.8% better accuracy on the much smaller model. KD simply minimizes the error between single instances. We conjecture that the cross-modal similarity to multiple anchor points introduced in the BeamCLIP helps the student preserve the topology of the teacher's embedding space.

Also, note that context-based prompt augmentation helps achieve better accuracy after distillation. Since Flowers102 has many ambiguous labels, our experiment shows that text prompt augmentation significantly increases the student's accuracy compared to the teacher's accuracy.

4.3 Ablation study

Table 4 shows the ablation study results of the BeamCLIP . Our empirical findings are as follows: (1) instance similarity distillation is not enough by itself to preserve the topology of the teacher's embedding space. (2) cross-modal similarity distillation compared to multiple anchor points helps the student mimic the teacher's embedding space. (3) self-supervised pre-training of the student significantly helps the student mimic the teacher's embedding space. (4) Entropy minimization (EntMin) helps to improve the accuracy. (5) Context-based prompt augmentation helps measure the similarity more precisely.

4.4 Qualitative results

Figure 3 shows qualitative results via top-5 text-image retrieval. A red rectangle denotes an incorrect result. CLIP-RN50 provides many incorrect results since it shows relatively low zero-shot accuracy. In contrast, BeamCLIP-RN50 provides much improved results, since it is distilled from CLIP-ViT/16 which shows very high zero-shot accuracy. More interestingly, BeamCLIP-RN50 provides surprisingly good text-image retrieval results, even though unseen text prompts such as `"a photo of {pink rose}"` or `"a photo of {yellow rose}"` are given.

5 Conclusion

Our BeamCLIP provides an effective and accessible framework that distills large-scale multimodal representations of CLIP. Combining our similarity-based distillation and context-based prompt augmentation with self-supervised training, we empirically demonstrate that the BeamCLIP can effectively transfer the vision and language representations of a large teacher model (CLIP-ViT/16) into a smaller student model (ResNet-50) while achieving comparable or better classification accuracy. We also show that the BeamCLIP provides better visual representations by achieving 74.8% top-1 linear probe accuracy on ImageNet-1K when compared to vision-only self-supervised methods (SimCLR: 65.6%, supervised: 76.2%).

Table 4: **Ablation study results.** The acronym denotes the sub-methods introduces in the BeamCLIP : (1-1) ISM means instance similarity matching loss, (1-2) CSM means cross-modal similarity matching loss, (2) CPA means context-based prompt augmentation, and (3) SSL PT means self-supervised pre-training of student, . All the technical components contribute to the improvements of transfer learning accuracy at the student. Note that we only perform CPA on datasets with more than 100 of class labels.

Method	Type	Img. Enc.	(1-1) ISM	(1-2) CSM	(2) CPA	(3) SSL PT	CIFAR10	CIFAR100	Flowers102	ImageNet-1K
CLIP [31] (zero-shot)	T	ViT-B/16	-	-	-	-	91.6	**68.7**	70.4	**68.6**
Representation Transfer (**BeamCLIP**)	S	RN50	Cosine	✗	✗	✗	39.07	6.04	-	-
	S	RN50	Cosine	✗	✗	SimCLR	87.00	48.26	3.59	51.83
	S	RN50	✗	CE	✗	✗	91.28	58.90	12.94	63.30
	S	RN50	✗	CE	✗	SimCLR	91.53	65.64	62.18	65.45
	S	RN50	✗	CE+EntMin	✗	SimCLR	91.71	66.14	63.96	**66.23**
	S	RN50	Cosine	CE+EntMin	✗	SimCLR	**92.10***	66.18	64.14	65.76
	S	RN50	Cosine	CE+EntMin	✓	SimCLR	-	**67.35**	**75.86***	66.17

(a) CLIP-RN50 (b) BeamCLIP-RN50

Figure 3: **Comparison of CLIP-RN50 and BeamCLIP-RN50.** This figure shows the top-5 text-image retrieval results. A red rectangle denotes an incorrect result. CLIP-RN50 provides many incorrect results, since it shows relatively low zero-shot accuracy. In contrast, BeamCLIP-RN50 provides much improved results, since it is distilled from CLIP-ViT/16 that shows very high zero-shot accuracy.

While we mostly experiment with ResNet-50 for its popularity, we expect our the BeamCLIP is versatile enough to plug and play with various lightweight student models, which will be interesting for future studies.

Broader impact

This research aims to advance transfer learning by leveraging large-scale pre-trained models such as CLIP. Since large-scale pre-training is expensive, it is highly required to effectively reuse representative pre-trained models.

References

[1] L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. *arXiv preprint arXiv:2106.05237*, 2021.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[5] X. Chen and K. He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, 2021.

[6] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision (ECCV)*, 2020.

[8] H. M. Choi, H. Kang, and D. Oh. Unsupervised representation transfer for small networks: I believe i can distill on-the-fly. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[9] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[11] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[13] Z. Fang, J. Wang, L. Wang, L. Zhang, Y. Yang, and Z. Liu. Seed: Self-supervised distillation for visual representation. In *International Conference on Learning Representations (ICLR)*, 2021.

[14] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

[15] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2005.

[16] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[21] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[22] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[24] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

[25] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[26] R. Müller, S. Kornblith, and G. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[27] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.

[28] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[29] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[30] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[31] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[32] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations (ICLR)*, 2019.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] A. Tejankar, S. A. Koohpayegani, V. Pillai, P. Favaro, and H. Pirsiavash. Isd: Self-supervised learning by iterative similarity distillation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[35] Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. In *International Conference on Learning Representations (ICLR)*, 2020.

[36] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[37] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision (ECCV)*, 2020.

[38] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[39] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng. Revisiting knowledge distillation via label smoothing regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

[41] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Section 4 and Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [No] Our work does not include theoretical results.
   (b) Did you include complete proofs of all theoretical results? [No] Our work does not include theoretical results.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide the code, data, and instructions in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 2, Section 3, and Section 4.
   (b) Did you mention the license of the assets? [No] We only used public benchmark datasets and open-sourced software. See Section 4.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We did not use new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We only used public benchmark datasets. See Section 4.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We did not use any data containing personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No] We did not use crowdsourcing or conduct research with human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No] We did not use crowdsourcing or conduct research with human subjects.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No] We did not use crowdsourcing or conduct research with human subjects.

A Datasets Details

Table 5: Details of datasets used for the BeamCLIP evaluation.

Dataset	Image Size	Classes	Train Size	Val Size	Test Size
CIFAR10 [22]	32x32	10	40,000	10,000	10,000
CIFAR100 [22]	32x32	100	40,000	10,000	10,000
STL10 [9]	128x128	10	4,000	1,000	8,000
Flowers102 [27]	224x224	102	1,020	1,020	6,149
Pets37 [30]	224x224	37	2,944	736	3,669
ImageNet [10]	224x224	1,000	1,231,167	50,000	50,000

B Method Details

B.1 Image Augmentation

Table 6: A list of image augmentations used in the BeamCLIP .

Mode	Augmentation	Parameters
Train	RandomResizedCrop	-
	RandomHorizontalFlip	p=0.5
	RandomColorJitter	p=0.8
	GaussianBlur	p=0.5, min=0.1, miax=2.0
	Normalize	-
Val	Resize	input_size + 0.1 * input_size
	CenterCrop	input_size
	Normalize	-

B.2 Context-based prompt augmentation

Table 7: Coarse and fine labels for CIFAR100.

Coarse Label	Fine Label
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchid, poppy, rose, sunflower, tulip
food containers	bottle, bowl, can, cup, plate
household electrical devices	clock, keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple tree, oak tree, palm tree, pine tree, willow tree
vehicles 1	bicycle, bus, motocycle, pickup truck, train
vehicles 2	lawn mower, rocket, streetcar, tank, tractor

Table 8: Examples of prompt augmentation with hierarchical labels for CIFAR100.

Label Name	Text Prompt
baby	`"A photo of a {baby}, categorized as {people}."`
beaver	`"A photo of a {beaver}, categorized as {aquatic mammals}."`
bee	`"A photo of a {bee}, categorized as {insect}."`

B.3 Other Implementation Details

Similarity Smoothing. To improve the effectiveness of distillation, we apply Label Smoothing (LS) [33] to the cross-modal similarity distillation loss. Recent works [26, 39] show that Label Smoothing helps knowledge distillation. To apply Label Smoothing, we determine the most similar anchor representation as follows:

$$j^* = \arg\max_j s_j(k_i, A). \tag{11}$$

Then, we generate a modified cross-modal similarity distribution:

$$s_j(k_i, A)^{LS} = \mathbb{1}_{[j=j^*]}(1-\alpha) + \alpha/M \tag{12}$$

where $\mathbb{1}_{[j=j^*]}$ is the indicator function whose value is 1 if $j = j^*$, M is the number of anchors, and α is the smoothing hyperparameter that is set to 0.2 in our experiments.

B.4 Model Hyperparameters

Table 9: BeamCLIP hyperparameters.

Hyperparameter	Value
CSD loss temperature	0.01
ISD loss scale	{0.1, 1.0, 10.0}
similarity smoothing (LS) sacle	0.2
optimizer	SGDR [24]
initial learning rate	0.5
weight decay	1e-6
EMA momentum	0.99
batch size	{256, 512}
epochs	200