uCAP: An Unsupervised Prompting Method for Vision-Language Models

A. Tuan Nguyen[1⋆], Kai Sheng Tai[1], Bor-Chun Chen[1], Satya Narayan Shukla[1], Hanchao Yu[1], Philip Torr[2], Tai-Peng Tian[1], and Ser-Nam Lim[3]

[1] Meta
[2] University of Oxford
[3] University of Central Florida
a.tuan.nguyen@outlook.com,philip.torr@eng.ox.ac.uk,sernam@ucf.edu

Abstract. This paper addresses a significant limitation that prevents Contrastive Language-Image Pretrained Models (CLIP) from achieving optimal performance on downstream image classification tasks. The key problem with CLIP-style zero-shot classification is that it requires domain-specific context in the form of prompts to better align the class descriptions to the downstream data distribution. In particular, prompts for vision-language models are domain-level texts (e.g., "a centered satellite image of ...") which, together with the class names, are fed into the text encoder to provide more context for the downstream dataset. These prompts are typically manually tuned, which is time consuming and often sub-optimal. To overcome this bottleneck, this paper proposes uCAP, a method to automatically learn domain-specific prompts/contexts using only unlabeled in-domain images. We achieve this by modeling the generation of images given the class names and a domain-specific prompt with an unsupervised likelihood distribution, and then performing inference of the prompts. We validate the proposed method across various models and datasets, showing that uCAP consistently outperforms manually tuned prompts and related baselines on the evaluated datasets: ImageNet, CIFAR-10, CIFAR-100, OxfordPets (up to 2%), SUN397 (up to 5%), and Caltech101 (up to 3%).

1 Introduction

Contrastive Language-Image Pretrained Models are trained to align images and natural language representations. Models trained for representation alignment across modalities (e.g., CLIP [19], CLIPA [12], CoCa [30]) excel at tasks such as zero-shot classification and information retrieval. In zero-shot classification, for example, one first collects a class description for each class, and then, for each image at test time, the predicted class is the nearest class description in the representation space. Typically, to retrieve these class descriptions with high fidelity, practitioners would often need to combine the class name with a hand-designed text prompt template. These prompt templates contain the domain-specific context, which helps to better align the class names and the image

⋆ Work done when Tuan was at the University of Oxford and doing an internship at Meta

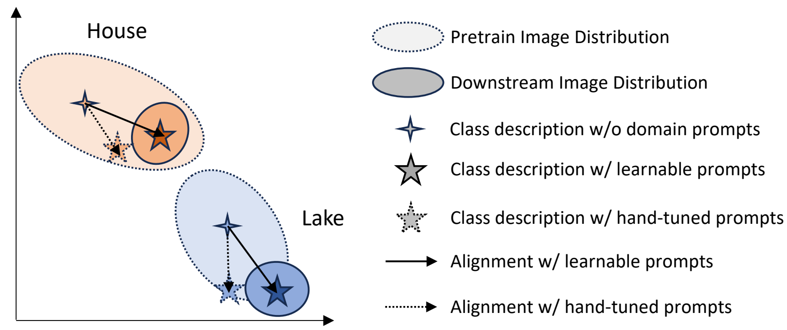

Fig. 1: Class Description Alignment: The downstream image distribution might be shifted from the pretraining distribution (e.g., "satellite images of houses" is a subdomain of "house"), thus the class description without any domain-specific context might be misaligned. This necessitates alignment through the use of domain-specific contexts. However, alignment with hand-tuned prompts (dotted line) is suboptimal. Our method, uCAP, automatically learns prompts in an unsupervised manner, leading to better alignment (solid line).

distribution (Fig. 1). For example, for a satellite image dataset, existing zero-shot classification methods use the prompt "a centered satellite image of {}", whereas, for ImageNet, they use prompts such as "a bad photo of {}, a graffiti of {}". We argue that these prompts are domain-specific information that is obtained by either prior domain knowledge or by inspecting the image dataset.

However, there are several problems with this approach of hand-designing domain-specific prompt templates: they are **ad-hoc and potentially dataset-specific**; they are **time- and labor-intensive** to tune for each dataset; and the process might **not be truly zero-shot** (as humans may use a small labeled validation set to tune the prompts).

Learning prompts automatically without human intervention is crucial, and we have seen some early attempts at this [1, 13, 31, 32]. However, these methods require labeled images, which goes against the spirit of zero-shot learning. Our method, on the other hand, offers a true zero-shot approach that does not rely on labeled data. While our method can be used in conjunction with few-shot prompt learning methods for optimal performance, we choose to isolate the two settings in this paper to evaluate our method's effectiveness independently.

We propose a method to learn the domain-specific prompts/contexts in an unsupervised manner using unlabeled target images. Here, we treat each downstream task as a domain; and introduce a graphical model of the data generation process, which assumes that a latent domain-specific context and the given class names generated the image dataset. To define the likelihood of an image being generated by this process, we leverage the distance (which is often the cosine distance in practice) learned by these pretrained vision-language models. Specifically, we use the above distance between an image and a class description to form a scalar energy function, which subsequently defines an energy-based likelihood distribution. The intuition behind this is as follows: if an image was generated by this generative process, it should be close to at least one class description in the representation space. Finally, we perform posterior inference for the prompts, which are then used for image classification. We name our method **uCAP** (**u**nsupervised **C**ontrastive **A**uto **P**rompting).

Main Contributions and Insights:

- We propose a novel unsupervised prompt learning method for CLIP-styled zero-shot classification. Our method is motivated by a natural and intuitive data generation process, in which a domain-specific text prompt, together with the class names, generate the downstream image data. This formulation allows us to perform unsupervised learning of the prompts, unlike most existing prompt learning works [1, 13, 31, 32] that require labels for optimization. To the best of our knowledge, uCAP is the first unsupervised prompt learning method that explicitly leverages the learned inductive bias of vision-language models (in the form of our energy-based likelihood).
- Through extensive evaluations with a wide range of models, datasets, and settings, we find that our method consistently outperforms prompt hand-tuning and other unsupervised learning methods. Thus, uCAP is a robust method that requires little hyperparameter tuning and generalizes well across a large number of settings.
- The prompts learned with our method exhibit strong transferability across settings. In particular, the prompts can be transferred across different distributions of the same task, and even to completely new tasks.

2 Related Work

Contrastive Language-Image Pretraining: One of the first, and perhaps most influential, works for cross-modality contrastive pretraining is CLIP [19]. CLIP jointly embeds text and images, which subsequently enables a wide range of tasks such as text-to-image generation [21], image captioning [15], and general visual question answering [11]. Since then, there have been several follow-up works that either expand to other modalities [3] or scale up model and training data [2, 12]. Our unsupervised prompt learning method can be applied straightforwardly to all of these multi-modal contrastive pretrained models [2, 3, 12].

Prompting Contrastive Language-Image Pretrained Models: Prompt learning has gained popularity alongside the rise of large and/or multimodal language models, as it offers an effective and efficient way to alter the behavior of such models. In the context of vision-language models, prompt learning helps to better align the representations of class descriptions and images, leading to better predictive performance. There are some early attempts to learn the prompt for CLIP [1, 9, 13, 31, 32] using labeled images of a downstream task. The drawback of these methods is that they need to collect some labeled data for training, and thus are not truly zero-shot. Their results are thus not directly comparable to ours. However, we will present results comparing our method against those that utilize few-shot transfer learning (that does not use labels from the target set).

There also exist methods [14, 18] that use large language models to generate the prompts for CLIP's zero-shot classification. However, these methods rely on much larger generative language models to generate the class descriptions, and still require partial human tuning. Additionally, our method can be used on top of these class descriptions to further improve the performance (as illustrated in the Supplementary Material).

Unsupervised domain adaptation and its application to vision-language models: Our problem setting can also be viewed as unsupervised domain adaptation, since we adapt the pretrained vision-language models during test time to the downstream classification tasks. Existing test time adaptation methods such as Pseudo Labels, Entropy Minimization [25], and Invariance Enforcing [16] typically fine-tune the backbone network (which would be the vision encoder in our case) with a surrogate loss. However, modifying the vision encoder would break the alignment learned by the two encoders, degrading the model's performance. The astute reader might wonder if such unsupervised surrogate objectives can be used to learn the prompts. In fact, [6] has tried this idea (with Pseudo Labels) for prompting Vision-Language models. To address any concerns here, we present results showing that our method is superior to these unsupervised training objectives in Subsection 4.2.

3 Approach: uCAP

Assume that the set of class names $Y = \{y_1, y_2, ..., y_K\}$ together with the domain-specific context w generated our dataset $X = \{x_1, x_2, ..., x_N\}$ (N image datapoints) according to the generative model in Figure 2. In particular, w is a text prompt which shows the domain context for the dataset generation. For example, if the class names are "plant" and "animal", two prompts "in water" and "on land" will yield completely different image datasets. Note that our formulation only requires knowing the set of class names $Y = \{y_1, y_2, ..., y_K\}$, not the specific class of each image (thus our method is unsupervised). We use x_i as the image representation, which is the embedding from the pretrained vision encoder for each image, for simplicity and efficiency (e.g., we only have to collect and store the image representations for the optimization). For exposition (noting that uCAP is unsupervised), assume that the class names Y and the dataset X are given, and that our goal is to infer the domain-specific context w. Here, and throughout this paper, we use "domain" to refer to a downstream task, e.g., ImageNet classification.

Note that throughout this section, we refer to the datapoints as images. However, our method can be easily extended to other modalities, such as video and audio. In the Supplementary Material, we present an experiment with a video dataset (UCF101 [23]) to further support this.

To define the joint distribution for the inference of w, we need to define the prior of w and the likelihood of the data given the domain-specific context w. Following standard practice [1,32] in the prompt learning literature, we perform

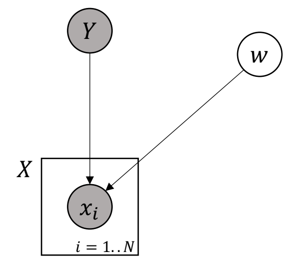

Fig. 2: Graphical Model: We assume a generative model $p(x|Y, w)$ that goes from language to images as follows: The class labels Y together with the domain-specific context w generate the image representations. For example, if the class labels are "animal" and "plant", two different domain contexts "in water" and "on land" will lead to two completely different datasets.

"soft prompt" learning, meaning that we learn the prompt in the continuous token embedding space.

Prior (for a prompt of length L):

$$p(w) = \prod_{l=1}^{L} p(w_l) = \prod_{l=1}^{L} \mathcal{N}(w_l; \mu, \Sigma), \tag{1}$$

where μ is the mean and Σ is the covariance matrix. These quantities can be hard-coded (e.g., $\mu = 0, \Sigma = \alpha I$), or can be computed from the empirical token embeddings of the model. In practice, we set μ and Σ to be the mean and covariance of the token embeddings across the vocabulary. This prior ensures that the soft prompt stays within the range of the natural token embeddings during training. We model the prior as a Gaussian distribution because of its simplicity, and that it works well in all of our experiments.

Likelihood: We use the inductive bias (i.e., the representation alignment between texts and images of similar concepts) learned by the fixed pretrained model to define the likelihood, in the form of an energy-based distribution. Specifically:

$$p(X|Y, w) = \prod_{i=1}^{N} p(x_i|Y, w) = \prod_{i=1}^{N} \frac{e^{-E(x_i, Y, w)}}{Z_Y(w)},$$

where $Z_Y(w)$ is the normalizing constant:

$$Z_Y(w) = \int_{\mathbb{R}^k} e^{-E(x, Y, w)} d^k x,$$

where k is the dimension of x and $d^k x$ is the k-dimensional volume differential.

The scalar energy function is defined as:

$$E(x, Y, w) = \min_m D\left[x, f_\phi(w \oplus y_m)\right], \tag{2}$$

where f_ϕ is the pretrained language encoder and D is the distance of choice (the CLIP models use the cosine distance). With some abuse of notation, $\oplus$ denotes the concatenation in the token embedding space (i.e., the class name y_m is first converted to the continuous token embedding, then concatenated to w).

Intuitively, this simple energy function means that, in order for an image to be created by Y and w, it should be close (in representation) to one domain-specific class description $w \oplus y_m$ of a class m. In other words, the domain-specific context w needs to bring the class descriptions closer to the image dataset.

Given this joint distribution, our goal is to infer the posterior $p(w|X,Y)$ and use it for image classification as:

$$\hat{p}(y^*|x^*) = \mathbb{E}_{p(w|X,Y)}\left[\hat{p}(y^*|x^*,w)\right], \tag{3}$$

where $\hat{p}(y^*|x^*,w)$ is the typical CLIP-styled predictive distribution given an image representation x^* and the prompt w.

Given the prior and the likelihood, there are several options for posterior inference: approximate inference (e.g., variational inference), exact sampling (e.g., stochastic gradient Langevin dynamics), or Maximum a Posteriori (MAP) point estimations then ensemble. In practice, we use the last option (MAP point estimations and then ensemble) as approximate Bayesian inference due to its simplicity and effectiveness (although this interpretation might be controversial and contested [28]). Simply put, we use gradient descent to find several points w (from different initializations) that maximize the MAP objective (Eq. 4) and ensemble them for the final prediction (Eq. 3).

Note that the above likelihood function is fully unsupervised. This allows us to do unsupervised inference of w. Specifically, the MAP objective is:

$$\begin{aligned}
&\max_w \left[\frac{1}{N}\log p(w) + \frac{1}{N}\sum_{i=1}^{N}\log p(x_i|Y,w)\right] \\
=&\max_w \left[\frac{1}{N}\log p(w) - \frac{1}{N}\sum_{i=1}^{N}E(x_i,Y,w) - \log Z_Y(w)\right].
\end{aligned} \tag{4}$$

While $\log p(w)$ (Eq. 1) and $E(x_i,Y,w)$ (Eq. 2) are easy to compute, $\log Z_Y(w)$ is intractable. Fortunately, we can compute the gradient of $\log Z_Y(w)$ w.r.t. w as below (the detailed derivation is provided in the Supplementary Material):

$$\nabla_w \log Z_Y(w) = -\mathbb{E}_{p(x|Y,w)}\left[\nabla_w E(x,Y,w)\right].$$

Therefore, our gradient estimator of Eq. 4 is:

$$\begin{aligned}
&\frac{1}{N}\nabla_w \log p(w) - \frac{1}{N}\sum_{i=1}^{N}\nabla_w E(x_i,Y,w) + \mathbb{E}_{p(x|Y,w)}\left[\nabla_w E(x,Y,w)\right] \\
\approx&\frac{1}{N}\nabla_w \log p(w) - \frac{1}{N}\sum_{i=1}^{N}\nabla_w E(x_i,Y,w) + \frac{1}{M}\sum_{i=1}^{M}\nabla_w E(\tilde{x}_i,Y,w),
\end{aligned} \tag{5}$$

where $\tilde{x}_i \sim p(x|Y, w), i \in \{1, 2, ..., M\}$.

Note that $\{\tilde{x}_i\}_{i=1}^M$ are the datapoints sampled from the current distribution $p(x|Y, w)$, not our real observed datapoints $\{x_i\}_{i=1}^N$. To sample from the distribution $p(x|Y, w)$ (knowing only its energy function), we use Stochastic Gradient Langevin Dynamics [27]:

$$\begin{aligned}
&\tilde{x}^{(0)} \sim \pi(x), \\
&\tilde{x}^{(t+1)} = \tilde{x}^{(t)} - \epsilon \nabla_x E(x, Y, w) + \sqrt{2\epsilon} z^{(t)},
\end{aligned} \tag{6}$$

where $z^{(t)} \sim \mathcal{N}(0, I)$ and $\pi(x)$ is a sampling prior distribution (often set to standard Gaussian). We set $\epsilon = 1e - 6$ and perform 200 (very light-weight) iterations of Eq. 6 for the sampling.

Comparison to Prompt Learning with Pseudo Label [6]: As this, to the best of our knowledge, is the only prior work in the literature of unsupervised soft prompt learning for vision-language models, we discuss the differences between their method and ours here. At first glance, the two methods might look similar since our energy function in Eq. 2 also seems to encourage the representation of each image to be close to the current most likely predicted class. However, there are some subtle but important differences. First of all, [6] pre-computes the pseudo labels for the image dataset before training, and use them for finetuning with the cross-entropy loss. This does not allow for flexible rearrangement of the classes/representations. Our method, on the other hand, only encourages each image representation to be close to one class description, thus allowing flexible rearrangement of the classes and representations. Note that the version of [6] which recomputes the pseudo labels after every epochs (thus allowing class rearrangement) does not work in practice, as it often collapses like entropy minimization [25]. More importantly, our method outperforms [6] in all settings.

On the complexity of using an ensemble of prompts: As aforementioned, we find several MAP estimations of w and ensemble them for the final prediction (Eq. 3). Similarly, CLIP also ensembles a number of hand-tuned prompts. It should be noted that the inference cost (after the prompts are learned) with multiple prompts is almost unchanged, since the class embeddings are precomputed for each prompt, and we only have to run the vision encoder once for each image. Only the training cost will scale (linearly) with the number of prompts. This is often not a problem in practice.

Implementation: In practice, we replace the coefficient $\frac{1}{N}$ of $\nabla_w \log p(w)$ by a tunable hyperparameter α to make it independent of the datasets – this is also a standard practice in Bayesian inference. Additionally, we set $M = N$, similar to other works on energy-based models. We also observe that using pseudo labels help stabilize the training, so we use the cross-entropy loss with pseudo labels as a auxiliary objective with a small coefficient of `0.1`. Note also that we initialize the prompt w randomly for each MAP estimation. We do not use the dataset-specific hand-tuned prompts at any stage during optimization (some existing works use these prompts for either prompt initialization or pseudo labels generation). We emphasize that we only perform *very light hyper-parameter tuning for our method* – this illustrates the robustness of our method and shows that it requires minimal human tuning in practice. In particular, we use the same sets of hyper-parameters for all datasets; and only use two configurations of hyper-parameters (one for all CLIP variants and one for OpenCLIP and CLIPA, due to the differences in model sizes and their training datasets):

- CLIP variants: Adam [7] optimizer for 40 epochs, `lr=0.001`, α=`0.0001` (coefficient of the log prior), prompt w of length 10.
- OpenCLIP and CLIPA: Adam optimizer for 40 epochs, `lr=0.01`, α=`0.001` (coefficient of the log prior), prompt w of length 5.

4 Experiments

In this section, we extensively validate our method with different models, datasets, and settings. Our main results (Subsection 4.1 and 4.2) illustrate that our method is superior to prompt hand-tuning and other unsupervised methods, and the additional experiments in Subsection 4.3 show that uCAP is more favorable when compared to few-shot prompt learning methods, that our method works well in the online learning setting, and that the learned prompts can be transferred in various settings.

4.1 Experiment Settings

In this subsection, we describe the setting of our main experiments, which include a wide range of models and datasets.

Models: We consider popular and state-of-the-art contrastive language-image pretrained models, namely:

- All variants of CLIP [19]: RN50, RN101, ViT-B/32, ViT-B/16, and ViT-L/14@336px.
- The best performing variant of Open-CLIP [2]: ViT-G/14.
- The best performing variant of CLIPAv2 [12], ViT-H/14, which is also state-of-the-art for zero-shot classification among the open-sourced models.

Datasets: We benchmark several popular image classification datasets, including ImageNet [22], CIFAR-10 [8], CIFAR-100 [8], OxfordPets [17], SUN397 [29], and Caltech101 [10]. Our evaluation setting is identical to that of CLIP [19].

Baselines:

- **Class names only**: The baseline where we only use the class names as the class descriptions, without any prompts.

- **Class names + HTPs (Hand-tuned prompts)**: This baseline uses the hand-tuned prompts reported in the CLIP, CLIPA, and Open-CLIP papers. Note that the number of hand-tuned prompts varies among datasets (presumably due to lack of human labor to excessively tune all datasets): ImageNet has the most number of hand-tuned prompts (80). We re-evaluate all the models on the chosen datasets (our re-evaluated numbers are very close to the reported ones).
- **Pseudo Labels** (for Prompt Learning): We also apply some of the most common unsupervised domain adaptation methods, namely Entropy Minimization and Pseudo Labels, for the prompt learning setting. However, Entropy Minimization often collapses in practice (this agrees with the observations in [16,25]). Therefore, we only report the results for the Pseudo Labels baseline. Note that [6] proposed the same unsupervised prompt learning method with Pseudo Labels. However, they only report results for CLIP-RN50, and some numbers are different from existing works (e.g., largely different than the numbers reported in CLIP [19]). Therefore, we re-evaluate the method for all the models and datasets considered in this paper, with our standardized setting.
- **uCAP (ours)**: Our proposed method. We test the variants with 1 prompt and 10 prompts.

4.2 Main Results

Table 1 shows the results of uCAP as well as the baselines for the seven models and six datasets. Overall, using more prompts leads to better performance (similar to the hand-tuned prompts case), and our method outperforms other baselines in virtually all settings. This indicates that the hand-tuned prompts are often not optimal (while being time- and labor- intensive). For datasets where the hand-selected prompts are relatively under-tuned (e.g., SUN397), uCAP offers up to 5% improvement over the human-selected prompts. Note that Pseudo Labels (a classic unsupervised domain adaptation method) does offer some improvement over the Class Names Only baseline. However, using that method alone is not sufficient (compared to hand-tuned prompts and our method), since it does not directly leverage the inductive bias (representation alignment) captured by the encoders.

We also investigate the effect of the number of prompts on the final classification performance. Figure 3 illustrates this relationship for both natural (hand-tuned) prompts and the prompts learned by uCAP. In both cases, more prompts generally translate to better performance. Interestingly, the performance of our method scales much faster with the number of prompts. We recommend using 5-10 prompts in practice.

4.3 Additional Results and Discussion

Comparison with (few-shot) supervised prompt learning methods: There are also methods [1, 13, 31, 32] that learn the prompts with labeled data points.

Table 1: Main Result across 7 models and 6 datasets. Reported numbers are from 5 runs.

Model	Method	ImageNet	CIFAR-10	CIFAR-100	OxfordPets	SUN397	Caltech101
CLIP RN50	Classnames + HTPs	59.6	71.5	41.9	85.8	59.5	83.1
	Classnames Only	54.8	66.3	36.2	79.4	53.1	75.4
	Pseudo Labels	57.6	67.6	36.3	80.4	55.9	77.1
	uCAP (1)	60.5	70.2	41.8	86.6	62.9	85.0
	uCAP (10)	**61.9**	**72.4**	**43.3**	**87.5**	**64.6**	**86.2**
CLIP RN101	Classnames + HTPs	62.2	80.7	48.9	**86.8**	57.7	84.4
	Classnames Only	58.4	77.3	43.9	80.4	52.8	81.1
	Pseudo Labels	59.2	79.2	45.1	81.8	54.5	84.1
	uCAP (1)	62.2	82.6	50.3	86.0	62.7	88.0
	uCAP (10)	**63.7**	**83.5**	**51.7**	**86.8**	**64.0**	**88.5**
CLIP ViT-B/32	Classnames + HTPs	63.2	89.9	65.1	**87.5**	62.1	87.1
	Classnames Only	58.5	85.9	60.3	80.0	58.5	84.8
	Pseudo Labels	61.1	87.5	61.4	81.0	62.1	86.1
	uCAP (1)	62.8	89.9	65.4	85.8	65.7	88.0
	uCAP (10)	**64.7**	**90.9**	**66.5**	87.2	**67.0**	**88.6**
CLIP ViT-B/16	Classnames + HTPs	68.6	90.8	68.3	89.1	64.1	88.0
	Classnames Only	63.5	86.5	62.0	81.6	59.6	85.0
	Pseudo Labels	66.1	89.3	63.8	82.7	62.8	87.4
	uCAP (1)	68.1	91.6	68.1	88.5	67.9	89.3
	uCAP (10)	**69.7**	**91.9**	**69.2**	**89.9**	**69.8**	**89.9**
CLIP ViT-L/14 @336px	Classnames + HTPs	76.2	94.9	77.0	93.8	68.2	**91.3**
	Classnames Only	72.2	89.0	72.3	87.7	63.7	86.4
	Pseudo Labels	73.9	92.1	74.3	89.4	66.3	87.9
	uCAP (1)	75.6	95.7	77.5	94.8	72.1	91.0
	uCAP (10)	**77.2**	**96.0**	**78.7**	**95.5**	**73.8**	**91.3**
OpenCLIP ViT-G/14	Classnames + HTPs	80.1	98.2	87.6	95.2	74.3	**92.8**
	Classnames Only	76.7	86.6	77.8	90.9	69.4	90.0
	Pseudo Labels	78.3	88.7	81.8	91.3	72.8	92.3
	uCAP (1)	79.3	98.6	87.5	94.9	76.4	92.6
	uCAP (10)	**80.3**	**98.7**	**88.1**	**95.3**	**77.5**	**92.8**
CLIPA ViT-H/14	Classnames + HTPs	**81.0**	98.8	89.1	95.3	73.2	92.9
	Classnames Only	79.4	98.6	87.0	93.2	70.8	91.4
	Pseudo Labels	80.0	98.9	88.1	93.3	72.0	92.4
	uCAP (1)	80.3	99.0	89.5	95.1	75.0	92.5
	uCAP (10)	**81.0**	**99.1**	**90.0**	**95.6**	**75.3**	**93.0**

Since these methods require labels for the prompt optimization, they need to

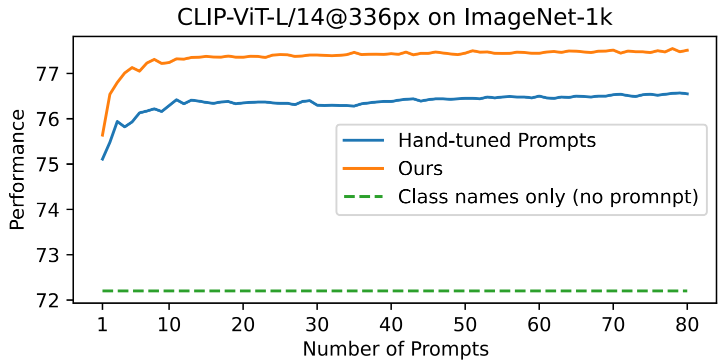

Fig. 3: Effect of the number of prompts on zero-shot classification performance. For both hand-tuned and learned prompts, more prompts generally lead to better performance. Interestingly, our performance scales much faster with the number of prompts when compared to hand-tuned prompts.

do transfer learning from a labeled source dataset when the target dataset is unlabeled. We compare uCAP to two such approaches, namely CoOp [32] and Plot [1], since they **1)** use the same domain-level (not instance-level) prompts as ours and **2)** conduct the transfer learning experiment. Specifically:

– CoOp [32], Plot [1]: perform transfer learning from a labeled source dataset to an unlabeled target dataset. In their experiment, CoOp and Plot perform transfer learning from ImageNet to ImageNetV2 [20], ImageNet-Sketch [26], ImageNet-R [4], ImageNet-A [5]. We use the reported performance from their paper. Note that Plot only reports results for CLIP-RN50, while CoOp reports results for CLIP-RN50, RN101, ViT-B/32, and ViT-B/16. Both Plot and CoOp use 16000 labels.
– uCAP (ours): performs **unsupervised learning** on the unlabeled dataset directly. In this sense, our method tackles a harder problem than CoOp and Plot, since it does not have/need access to a labeled source dataset.

We focus on CLIP variants RN50, RN101, ViT-B/32, and ViT-B/16, as these are the models investigated in the CoOp paper. Our findings, presented in Table 2, show that our method uCAP surpasses CoOp by approximately 1.5% on average. Notably, uCAP outperforms CoOp and Plot in most datasets, including ImageNet-Sketch, ImageNet-A, and ImageNet-R, demonstrating its effectiveness and versatility. This is remarkable, given that uCAP does not rely on labeled source data from ImageNet, unlike CoOp and Plot. However, CoOp and Plot exhibit better performance in ImageNetV2, and we hypothesize that this is likely due to the relatively small distribution shift between ImageNet and ImageNetV2, which enables labeled data from ImageNet to provide substantial information for ImageNetV2.

Table 2: Comparison with CoOp. CoOp performs transfer learning from labeled ImageNet to other target datasets, while our method performs unsupervised prompt learning directly on each dataset (thus does not require a labeled source dataset). Reported numbers are from 5 runs

Model	Method	ImageNetV2	ImageNet-Sketch	ImageNet-A	ImageNet-R	Average
CLIP RN50	Classnames + HTPs	51.84	32.89	23.95	56.73	41.35
	Classnames Only	48.64	30.27	20.83	56.87	39.15
	Plot (16000 labels)	55.11	33.00	21.86	55.61	41.39
	CoOp (16000 labels)	**55.40**	34.67	23.06	56.60	42.43
	uCAP	54.70	**36.21**	**24.07**	**61.80**	**44.19**
CLIP RN101	Classnames + HTPs	55.90	39.74	30.95	65.96	48.14
	Classnames Only	51.85	35.80	27.56	64.43	44.91
	CoOp (16000 labels)	**58.60**	40.40	29.60	64.98	48.39
	uCAP	57.47	**41.51**	**31.02**	**69.09**	**49.77**
CLIP ViT-B/32	Classnames + HTPs	55.40	40.92	32.13	66.66	48.79
	Classnames Only	51.50	37.37	29.19	65.93	46.00
	CoOp (16000 labels)	**58.24**	41.48	31.34	65.78	49.21
	uCAP	57.74	**42.33**	**32.90**	**70.39**	**50.84**
CLIP ViT-B/16	Classnames + HTPs	61.49	46.96	50.28	75.36	58.22
	Classnames Only	58.04	43.08	45.39	74.61	55.28
	CoOp (16000 labels)	**64.56**	47.89	49.93	75.14	59.38
	uCAP	63.38	**48.84**	**51.76**	**78.40**	**60.60**

Do the prompts have semantic meaning? A natural question is if the learned prompts have any meaning. Recall that we optimize in the token embedding space (soft prompt). Therefore, the best way we can check for semantic meaning is to map the soft prompts back to actual words by algorithms such as nearest neighbour. Similar to the observations from other recent soft prompt learning works [32], we also find that the prompts do not have any semantic meaning (nor are even legible English).

Transferability of the prompts: Another natural question is whether the learned prompts can be transferred, i.e., learn the prompts in one setting and test them in another setting. We first re-emphasize that uCAP is easily applicable to virtually any inference setting (just use the unlabeled test data for prompt learning), and thus transfer learning is often not needed. However, it is still interesting to see how the unsupervised-learned prompts perform in a transfer learning setting, in case of a sudden change in the data distribution (e.g., the camera sensor gets replaced, continual task change, etc.). Note that this result

Table 3: Transfer learning. Unsupervised prompt learning on ImageNet and inference on other target datasets. We use 10 prompts for uCAP. Reported numbers are from 5 runs.

		Same Task (shifted dist.)				Across Tasks				
Model	Method	ImageNetV2	ImageNet-Sketch	ImageNet-A	ImageNet-R	CIFAR10	CIFAR100	OxfordPets	SUN397	Caltech101
CLIP RN50	Classnames + HTPs	51.8	32.9	**24.0**	56.7	71.5	41.9	85.8	59.5	83.1
	Classnames Only	48.6	30.3	20.8	56.9	66.3	36.2	79.4	53.1	75.4
	uCAP	**54.4**	**35.1**	**24.0**	**58.9**	**72.0**	**44.1**	**86.1**	**62.8**	**85.9**
CLIP RN101	Classnames + HTPs	55.9	39.7	**31.0**	66.0	80.7	48.9	**86.8**	57.7	84.4
	Classnames Only	51.9	35.8	27.6	64.4	77.3	43.9	80.4	52.8	81.1
	uCAP	**56.9**	**40.9**	30.5	**66.7**	**82.8**	**51.7**	**86.8**	**63.6**	**88.5**
CLIP ViT-B/32	Classnames + HTPs	55.4	40.9	32.1	66.7	89.9	65.1	**87.5**	62.1	87.1
	Classnames Only	51.5	37.4	29.2	65.9	85.9	60.3	80.0	58.5	84.8
	uCAP	**57.1**	**42.1**	**32.3**	**68.0**	**90.0**	**66.5**	87.3	**65.6**	**89.0**
CLIP ViT-B/16	Classnames + HTPs	61.5	47.0	50.3	75.4	90.8	68.3	89.1	64.1	88.0
	Classnames Only	58.0	43.1	45.4	74.6	86.5	62.0	81.6	59.6	85.0
	uCAP	**62.7**	**48.7**	**50.8**	**77.4**	**91.1**	**69.7**	**89.8**	**68.0**	**90.9**
CLIP ViT-L/14 @336px	Classnames + HTPs	70.3	60.3	77.5	88.7	94.9	77.0	**93.8**	68.2	**91.3**
	Classnames Only	65.7	56.6	71.0	86.0	89.0	72.3	87.7	63.7	86.4
	uCAP	**71.0**	**60.7**	**78.7**	**89.2**	**95.3**	**78.6**	92.6	**72.0**	91.0

should not be compared to the transfer learning setting of CoOp, since CoOp uses labeled data from the (ImageNet) source dataset. Our findings are as follows:

- Transferability across models: This is not feasible because each model has a different token embedding look-up table (the token embedding dimension might even change across models).
- Transferability across different domains of the same task: For this experiment, we perform transfer learning from ImageNet to ImageNetV2, ImageNet-Sketch, ImageNet-A, and Imagenet-R. Note that we only use the unlabeled image from the source dataset (ImageNet). Table 3 (first half) shows the result of this experiment, indicating that the prompts can be transferred across different domains and withstand some degree of distribution shift.
- Transferability across tasks: This is not guaranteed, since the prompts are learned in a task-specific manner. However, in practice, we observe a strong capability of transferring the prompts across tasks, especially from a more complex task (ImageNet) to other tasks. We report the performance of prompts learned by ImageNet on other tasks (CIFAR-10, CIFAR-100, Ox-

Table 4: Different variants of our method (using 10 learnable prompts) on ImageNet.

uCAP variant	CLIP Model				
	RN50	RN101	ViT-B/32	ViT-B/16	ViT-L/14
uCAP - Online Learning	61.4	63.3	64.5	69.4	76.9
uCAP (offline)	61.9	63.7	64.7	69.7	77.2
Classnames + HTPs	59.6	62.2	63.2	68.6	76.2

fordPets, SUN397, and Caltech101) in the second half of Table 3. Overall, the learned prompts offer a large performance gain compared to the Class Names Only baseline, and are even better than the hand-tuned prompts in most cases.

Online learning setting: To enable true zero-shot learning, we also consider an online learning setting. This can be viewed as test time adaptation [24,25], as the model does not receive any training/adaptation data before deployment, and must simultaneously give prediction to unlabeled test data points and use them for adaptation. Our method can be applied straightforwardly to this setting; specifically, for each test data mini-batch, we use the gradient in Eq. 5 to update the prompts, and use the prompts to make prediction for the test data points. We perform this process continually without resetting the prompts at each mini-batch (similar with test time adaptation). Note that prior works [1, 6, 32] do not consider this setting. Table 4 shows that our method performs well in this scenario (using a test batch size of 64), with the results being just a bit worse than the offline setting (which is attributable to the initial warm-up period of several mini-batches when the prompts are not yet learned optimally).

5 Conclusion

To conclude, we propose uCAP, an unsupervised method to automatically learn the prompts for zero-shot prediction in vision-language models. In particular, we model a data generation process that includes a domain-specific context (which, together with the class names, generates the images), and use an unsupervised energy-based distribution as the likelihood. This allows us to infer back the domain contexts, and use them for the classification. Our method shows superior performance when compared to hand-tuned prompts and relevant baselines. Future research directions include: to generalize the energy-based formulation to other prompt learning settings, and to consider the instance-level prompts for zero-shot classification (although our preliminary experiments suggest that the performance gain is not worth the added complexity of additional networks).

References

1. Chen, G., Yao, W., Song, X., Li, X., Rao, Y., Zhang, K.: PLOT: Prompt Learning with Optimal Transport for Vision-Language Models. ICLR (2023)
2. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. CVPR (2023)
3. Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K.V., Joulin, A., Misra, I.: ImageBind: One Embedding Space To Bind Them All. CVPR (2023)
4. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. ICCV (2021)
5. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural Adversarial Examples. CVPR (2021)
6. Huang, T., Chu, J., Wei, F.: Unsupervised Prompt Learning for Vision-Language Models. arXiv (2022)
7. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. ICLR (2015)
8. Krizhevsky, A., Hinton, G., et al.: Learning Multiple Layers of Features from Tiny Images (2009)
9. Lee, D., Song, S., Suh, J., Choi, J., Lee, S., Kim, H.J.: Read-only Prompt Optimization for Vision-Language Few-shot Learning. ICCV (2023)
10. Li, F.F., Andreeto, M., Ranzato, M., Perona, P.: Caltech 101 (2022). `https://doi.org/10.22002/D1.20086`
11. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. ICML (2023)
12. Li, X., Wang, Z., Xie, C.: CLIPA-v2: Scaling CLIP Training with 81.1% Zero-shot ImageNet Accuracy within a $10, 000 Budget; An Extra $4, 000 Unlocks 81.8% Accuracy. arXiv (2023)
13. Lu, Y., Liu, J., Zhang, Y., Liu, Y., Tian, X.: Prompt Distribution Learning. CVPR (2022)
14. Menon, S., Vondrick, C.: Visual Classification via Description from Large Language Models. ICLR (2023)
15. Mokady, R., Hertz, A., Bermano, A.H.: ClipCap: CLIP Prefix for Image Captioning. arXiv (2021)
16. Nguyen, A.T., Nguyen-Tang, T., Lim, S.N., Torr, P.H.: TIPI: Test Time Adaptation with Transformation Invariance. CVPR (2023)
17. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and Dogs. CVPR (2012)
18. Pratt, S., Liu, R., Farhadi, A.: What does a platypus look like? Generating customized prompts for zero-shot image classification. ICCV (2022)
19. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning Transferable Visual Models From Natural Language Supervision. ICML (2021)
20. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do ImageNet Classifiers Generalize to ImageNet? ICML (2019)
21. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. CVPR (2022)
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV (2015)

23. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. arXiv (2012)
24. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. ICML (2020)
25. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully Test-Time Adaptation by Entropy Minimization. ICLR (2021)
26. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning Robust Global Representations by Penalizing Local Predictive Power. NeurIPS (2019)
27. Welling, M., Teh, Y.W.: Bayesian Learning via Stochastic Gradient Langevin Dynamics. ICML (2011)
28. Wilson, A.G., Izmailov, P.: Deep Ensembles as Approximate Bayesian Inference (2021)
29. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: SUN database: Large-scale scene recognition from abbey to zoo. CVPR (2010)
30. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: CoCa: Contrastive Captioners are Image-Text Foundation Models. arXiv (2022)
31. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional Prompt Learning for Vision-Language Models. CVPR (2022)
32. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to Prompt for Vision-Language Models. IJCV (2022)

A Gradient of $\log Z_Y(w)$

$$
\begin{aligned}
\nabla_w \log Z_Y(w) &= \frac{\nabla_w Z_Y(w)}{Z_Y(w)} = \frac{\int \nabla_w e^{-E(x,Y,w)} dx}{Z_Y(w)} \\
&= \frac{\int e^{-E(x,Y,w)} \nabla_w \log e^{-E(x,Y,w)} dx}{Z_Y(w)} \\
&= \int \frac{e^{-E(x,Y,w)}}{Z_Y(w)} \nabla_w \log e^{-E(x,Y,w)} dx \\
&= -\int p(x|Y,w) \nabla_w E(x,Y,w) dx \\
&= -\mathbb{E}_{p(x|Y,w)} \left[\nabla_w E(x,Y,w) \right] \\
&\approx -\frac{1}{M} \sum_{i=1}^{M} \nabla_w E(\tilde{x}_i, Y, w),
\end{aligned}
$$

where $\tilde{x}_i \sim p(x|Y,w), i \in \{1,2,...,M\}$, and the last equation is obtained from Monte Carlo sampling.

B Classification via Descriptions

To show that our method is truly versatile, in this section, we combine uCAP with [14] (which use class descriptions generated by a large language model instead of the default class names). We use the same class descriptions as released

Table 5: Comparison with "Classification via Descriptions".

Method	ViT-B/16 ImageNet	ViT-B/16 OxfordPets	ViT-B/16 CUB	ViT-L/14@336px ImageNet	ViT-L/14@336px OxfordPets	ViT-L/14@336px CUB
Classification via Description [14]	67.9	86.9	58.1	76.2	91.7	64.9
uCAP + Descriptions	**69.6**	**90.1**	**58.3**	**77.6**	**94.7**	**65.2**

Table 6: Results for UCF101 action recognition dataset with CLIP.

Method	RN50	RN101	ViT-B/32	ViT-B/16	ViT-L/14	ViT-L/14@336
Classnames Only	61.79	63.68	66.28	68.26	74.23	73.82
Classnames + HTPs	68.39	67.52	69.18	73.83	80.44	79.75
uCAP	**68.64**	**70.50**	**70.71**	**74.46**	**80.77**	**80.74**

in their official github page [4], and additionally learn domain-level prompts with uCAP. The results of this experiment is presented in Table 5, which shows that uCAP can further improve the performance of class descriptions generated by Large Language Models.

C Video Experiment

We also perform an experiment on the UCF101 video dataset [23]. This is an action recognition dataset which consists of 101 actions. For each video, we sample 60 frames randomly, and average the frames' representation to get the video representation. For the hand-tuned prompts baseline, we use the same prompts as proposed in the CLIP paper [19]. As illustrated in Table 6, our method uCAP can be straightforwardly extended to this modality and the learned prompts lead to significant improvement in accuracy.

[4] https://github.com/sachit-menon/classify_by_description_release