Understanding prompt engineering does not require rethinking generalization

Victor Akinwande[1]* Yiding Jiang[1]* Dylan Sam[1] J Zico Kolter[1,2]

[1]Carnegie Mellon University [2]Bosch Center for AI

Abstract

Zero-shot learning in prompted vision-language models, the practice of crafting prompts to build classifiers without an explicit training process, has achieved performance in many settings. This success presents a seemingly surprising observation: these methods suffer relatively little from overfitting, i.e., when a prompt is manually engineered to achieve low error on a given training set (thus rendering the method no longer actually zero-shot), the approach still performs well on held-out test data. In this paper, we show that we can explain such performance remarkably well via recourse to classical PAC-Bayes bounds. Specifically, we show that the discrete nature of prompts, combined with a PAC-Bayes prior given by a language model, results in generalization bounds that are *remarkably* tight by the standards of the literature: for instance, the generalization bound of an ImageNet classifier is often within a few percentage points of the true test error. We can even *greedily* search over the prompt space, improving upon training performance while retaining the same bound. Furthermore, the resulting bound is remarkably suitable for model selection: the models with the best bound typically also have the best test performance. This work thus provides a substantial justification for the widespread practice of "prompt engineering," even if it seems that such methods could potentially overfit the training data.

1 Introduction

Generalization bounds provide statistical guarantees on the average-case performance of the output of a learning algorithm. However, Zhang et al. [2021] highlighted that classical approaches for deriving generalization bounds are insufficient for explaining the generalization ability of deep learning, spurring a flurry of new approaches for deriving tighter generalization bounds for deep neural networks [Bartlett et al., 2017, Dziugaite and Roy, 2017, Neyshabur et al., 2017b]. In the recent literature on generalization bounds for neural networks, a large focus has been on developing *data-dependent bounds*, or bounds that take into consideration of the data distribution in addition to the hypothesis space. Some of the best data-dependent bounds are based on the PAC-Bayes framework [McAllester, 1999] and are derived by bounding the KL divergence between a prior over the hypothesis space and the posterior yielded by the learning algorithm. PAC-Bayes bounds have led to the first non-vacuous generalization bounds for deep learning [Dziugaite and Roy, 2017], but they are still too loose to be practically useful. In fact, as Lotfi et al. [2022] have recently argued, many PAC-Bayes bounds with data-dependent priors, while non-vacuous, can be best described as validation bounds — i.e., the use of data-dependent priors effectively leverages held-out data in a manner similar to cross-validation, which undermines their ability to *explain* generalization.

Nonetheless, despite the lack of a clear theoretical basis, modern machine learning models are becoming increasingly large [Kaplan et al., 2020, Dosovitskiy et al., 2020]. One prevailing paradigm

*Equal contribution.

ICML 2023 Workshop: Sampling and Optimization in Discrete Space (SODS 2023).

Table 1: Comparison with existing state-of-the-art generalization bounds for test error on different datasets. We report both data-independent and data-dependent bounds ($\star$ indicates data-dependent prior and $-$ indicates that the bounds are not available). Our bounds are significantly tighter than the existing PAC-Bayes bounds in the literature, often only within a few percents of the actual test error.

Dataset	Zhou et al. [2019]	Dziugaite et al. [2021]	Lotfi et al. [2022]	Ours
CIFAR-10	–	0.230*	0.582 / 0.166*	**0.059**
CIFAR-100	–	–	0.946 / 0.444*	**0.251**
ImageNet	0.965	–	0.930 / 0.409*	**0.312**

is to use pretrained foundation models such as CLIP [Radford et al., 2021] or ALIGN [Jia et al., 2021] as feature extractors and provide weak supervision for a downstream target task via *prompts*, which are text descriptions of the desired tasks that are often significantly easier to obtain compared to full model weights or even a generic linear classifier over the last layer. The versatility and performance of prompting pretrained models have led to the rise of *prompt engineering*, an emergent paradigm in machine learning where the users carefully design the task specification in text or even learn the prompts in a data-driven fashion [Lester et al., 2021]. Despite its empirical success, little is understood of how and why prompting these pretrained models work, and in particular why the method seems to suffer little from overfitting: manually tuning or even greedily optimizing prompts on a given training set often performs nearly as well on the corresponding test set.

In this paper, we demonstrate that rather simple analysis tools in fact capture this behavior surprisingly well. In particular, we show that traditional PAC-Bayes bounds [McAllester, 1999], when applied to the discrete hypothesis class defined by prompts (and specifically with a prior given by a large language model), are often *remarkably* tight, even for large-scale domains: for example, we achieve an generalization bound of 31% error for a full ImageNet classifier, which is within *6%* of the actual observed test error. This represents a *vast* improvement over traditional deep-learning-based bounds, where achieving any non-vacuous bound on domains like ImageNet typically requires a great deal of effort; see, for instance, Table 1 for a comparison of our bounds with other approaches, especially the variants that do not use data-dependent priors (as our prompt-based bounds do not). To summarize, we find that, unlike conventional deep learning models, prompting pretrained models does not suffer from the issues surrounding generalization bounds [Zhang et al., 2021], and one can readily derive a strong theoretical guarantee for using prompts via well-studied techniques. Overall, these findings suggest that, despite a large amount of automatic or manual tuning, prompt engineering is potentially a principled approach for using these pretrained models that does not suffer the same lack of theoretical grounding as conventional deep learning models.

2 Related Works

Prompt Engineering. With the advent of large pretrained models, prompting developed as a surprising yet effective method to harness the abilities of these large models with limited labeled data [Brown et al., 2020, Le Scao and Rush, 2021, Liu et al., 2023]. The flexibility of prompting has enabled a wide range of new capabilities unavailable to previous machine learning models, leading to a significant effort to document successful prompting methods [Bach et al., 2022] in both classification and text-to-image generation. One downside of prompting is that the performance varies greatly depending on how the prompt is phrased. To address this issue, methods have been proposed to learn “optimal" prompts given labeled data, which empirically performs well and is parameter efficient [Lester et al., 2021, Li and Liang, 2021, Gao et al., 2021, Zhou et al., 2022a,b]. One drawback to these data-driven approaches is that they learn “soft" prompts or embedding vectors that are not interpretable and can overfit. As such, another class of methods proposes gradient-based methods to learn interpretable prompts [Wen et al., 2023]. This work studies the theoretical guarantees of the practice of prompt engineering, that is, why these methods seem to work without any overfitting.

Prompt engineering has been extended to computer vision through CLIP (Contrastive Language-Image Pretraining) [Radford et al., 2021]. CLIP employs a neural network architecture combining natural language processing and computer vision to comprehend images and their corresponding text descriptions. It undergoes pretraining using a vast dataset of image and text pairs with a contrastive learning objective, enabling the model to differentiate between diverse image and text combinations.

This broad training allows CLIP to grasp various concepts and relationships between images and language. Unlike traditional computer vision models that rely on fixed labels for image classification, CLIP can perform multiple tasks based on natural language instructions. Examples include object recognition, image caption generation [Tewel et al., 2021], and zero-shot image classification using textual descriptions even for unseen labels.

Generalization bounds. Generalization bounds are upper bounds on the *generalization gap* of a model. Deriving such bounds for deep learning has been difficult, and most are usually vacuous [Zhang et al., 2021, Jiang et al., 2019, Dziugaite et al., 2020]. They also may suffer from fundamental limitations [Nagarajan and Kolter, 2019b]. The core component of a generalization bound is a *complexity measure*, a quantity that relates to some aspect of generalization. A complexity measure may depend on the properties of the trained model, optimizer, and possibly training data, as long as it does not have access to a validation set. The most classic bounds such as VC-dimension [Vapnik, 1971] are often related to some form of parameter counting which is often too pessimistic for deep neural networks. Norm-based bounds usually rely on the margin and some norms of the model weights [Langford and Caruana, 2001, Bartlett et al., 2017, Neyshabur et al., 2015, 2017b], but these bounds have been ineffective at studying generalization of deep learning [Nagarajan and Kolter, 2019a]. Another main class is the PAC-Bayes bounds McAllester [1999] which have been much more successful in deep learning due to the flexibility of prior [Neyshabur et al., 2017a, Dziugaite and Roy, 2017, Zhou et al., 2019, Lotfi et al., 2022] although these bounds are still much looser than the actual generalization error. Our approach also belongs to the PAC-Bayes family, but we apply the PAC-Bayes bounds to the distribution of discrete tokens rather than the parameter of the neural networks with a language model as the prior. This allows us to derive significantly tighter bounds compared to applying the PAC-Bayes bounds with less informative priors.

3 Preliminaries

Notations. Let $\mathcal{X} \in \mathbb{R}^d$ be a set of inputs and $\mathcal{Y} = [K]$ be a label set, and there exists a probability distribution D on $(\mathcal{X} \times \mathcal{Y})$ which is unknown. Let our data $(X_1, Y_1), \ldots, (X_n, Y_n)$ be drawn i.i.d from D, and consider a predictor $f : \mathcal{X} \to \mathcal{Y}$ and a fixed set of predictors indexed by the parameter set Θ. We use f_θ to denote the classifier indexed by θ. We consider the 0–1 loss given by $\ell(y', y) = \mathbb{1}\{y \neq y'\}$. The generalization error (risk) of a predictor is defined as $R(\theta) = \mathbb{E}_{(X,Y)\sim P}[\ell(f_\theta(X), Y)]$ and the empirical risk $r(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f_\theta(X_i), Y_i)$ satisfies $\mathbb{E}_{\mathcal{S}}[r(\theta)] = R(\theta)$ for a sample $\mathcal{S} = [(X_1, Y_1), \ldots, (X_n, Y_n)]$. An estimator is a function $\hat{\theta} : \bigcup_{n=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^n \to \Theta$.

Vision-language models. CLIP consists of two encoders g_{img} and g_{txt}. Given an image $X \in \mathcal{X}$, the image encoder $g_{\text{img}} : \mathcal{X} \to \mathbb{R}^d$ maps an image X to a d-dimension real-valued embedding. Given a text $T \in \mathcal{T}$, the image encoder $g_{\text{txt}} : \mathcal{T} \to \mathbb{R}^d$ maps a sentence Z to a d-dimension real-valued embedding. Given a batch of images $\{X_i\}_{i=1}^{B}$ and their corresponding texts $\{T_i\}_{i=1}^{B}$, the training objective maximizes the cosine similarity of the embeddings of the matching image and text pair and minimize the cosine similarity of image and text pairs that do not correspond to each other. The primary task we consider in this work is image classification via pretrained vision-language models. Here, we employ natural language by finding a *class prompt*, θ^k, for each class. For a K-class classification problem with $\theta = (\theta^1, \theta^2, \ldots, \theta^K) \in \Theta = \mathcal{T}^K$, the zero-shot classifier using class prompts and CLIP is:

$$f_\theta(X) = \underset{k \in [K]}{\arg\max} \ \left\langle g_{\text{txt}}(\theta^k),\, g_{\text{img}}(X) \right\rangle \tag{1}$$

PAC-Bayes framework. We often are interested in the generalization ability of a predictor and we can quantify this by upper bounds on the population loss (true risk). In general, we are concerned with the generalization gap which is the difference between the true risk and the empirical risk. Various generalization bounds can depend on the implicit bias of the learning algorithm [Neyshabur et al., 2014], the training data $\mathcal{S}$, or the data-generating distribution D. Classic VC bounds [Vapnik, 1971] do not depend on either. Distribution-dependent bounds are expressed in terms of quantities related to the data-generating distribution while data-dependent bounds are expressed in terms of empirical quantities that can be computed directly from data and can be used for self-certification [Pérez-Ortiz et al., 2021]. The PAC-Bayes framework defines a hierarchy over hypotheses in our hypothesis class

Θ that takes the form of a prior distribution P over Θ. That is, we assign a probability $P(\theta) \geq 0$ for each $\theta \in \Theta$ and refer to $P(\theta)$ as the prior score of θ. The learning process defines a posterior probability over Θ, which we denote by Q. In the context of supervised learning, we can think of Q as defining a prediction rule as follows: Given an instance X, we randomly pick a hypothesis θ according to Q and predict $f_\theta(X)$. Remarkably, it was shown that the generalization gap can be upper bounded by the KL-divergence between P and Q [McAllester, 1999].

4 Prompt Search

A key observation we make is that designing a prompt is analogous to finding a set of weights in regular machine learning models, where the hypothesis space is the space of texts/tokens. The goal is to find class prompts that maximize the training accuracy without finetuning the parameters of the model. This process can be formulated as a discrete optimization over the space of tokens, $\mathcal{V}$. Suppose that we are looking for class prompts of length L, then we will be searching for $K \cdot L$ tokens over the space of $|\mathcal{V}|^{K \cdot L}$. This search is exponential in the length of the prompt, which can be intractable for even small token spaces. To circumvent this problem, we generate the prompts in a sequential manner; that is, we increment the prompts by selecting the token that maximizes the *search criterion*, $\mathcal{J}$, on the training dataset from a set of *candidate tokens*, $\widehat{\mathcal{V}}$. The search criterion is the objective being optimized, and candidate tokens are permissible tokens that can be used to extend the current class prompts. At every step of the search, we keep the class prompts fixed except for all but one class. The prompt for each class k is a sequence of $l < L$ tokens $v \in \mathcal{V}$, $\theta^k_{\leq l} = (v_1, v_2, \ldots, v_l)$, and the next token for θ^k is obtained via:

$$v_{l+1} = \operatorname*{arg\,max}_{v \in \widehat{\mathcal{V}}(\theta)} \; \mathcal{J}\big(v, \theta^k_{\leq l}, \theta^{\neg k}\big). \tag{2}$$

$\theta^{\neg k}$ denotes the class prompts for all classes except for the k^{th} class. The pseudocode for this process is outlined in detail in Algorithm 1. Using $\oplus$ to denote concatenation, the simplest instantiation of search is a *greedy search*, where we use:

$$\widehat{\mathcal{V}}_{\text{greedy}}(\theta) = \mathcal{V}, \quad \mathcal{J}_{\text{greedy}}\left(v, \theta^k_{\leq l}, \theta^{\neg k}\right) = -r\big((\ldots, \theta^{k-1}, \theta^k_{\leq l} \oplus v, \theta^{k+1}, \ldots)\big). \tag{3}$$

In other words, we always search over all the tokens in the inner loop to maximize the training accuracy. This simplest greedy search can be seen as an instantiation of *empirical risk minimization* [Vapnik, 1991, ERM] since its only objective is to minimize the training error.

There are several drawbacks to this simple algorithm, the chief of which is that we need to search over $\mathcal{V}$ exhaustively in the inner loop (line 6). This can be expensive since it consists of all the tokens the vision-language model uses (e.g., CLIP has about 50000 tokens). Instead, we could search over only a subset of $\mathcal{V}$. To reduce this search space, we use a language model (LM) to induce a distribution over the next tokens conditioned on θ^k and only evaluate the tokens with high probabilities:

$$p_{\text{next}}(v_{l+1} \mid \theta^k_{\leq l}) = p_{\text{LM}}\left(v_{l+1} \mid \theta^k_{\leq l} = [v_0, v_1, \ldots, v_l]\right). \tag{4}$$

Since CLIP was trained with natural language supervision, it is likely that a reasonable next token can be captured by an autoregressive LM, which is trained to model the probability of the next token. We then take the top N candidates and only evaluate the accuracy of these candidates. Conveniently, this can be seen as constraining the complexity of the prompt as the language model provides a structured prior over the set of tokens. We observe that using a language model to propose likely tokens incurs minimal performance loss, suggesting that language models indeed are good prior for searching for class prompts on image classification tasks. Furthermore, we can use predefined strings to further constrain the space of hypothesis by starting with an *initial prompt* such as “This is an image of” instead of using an empty string. These provide additional structure to the generated prompts.

This procedure can be further augmented to optimize the PAC-Bayes bound via *structural risk minimization* [Vapnik and Chervonenkis, 1974, SRM] similar to the approach of Dziugaite and Roy [2017], namely, we will take the hypothesis complexity (e.g., KL-divergence) into account as we search for the next token for each prompt. We use the KL-divergence directly in the objective optimization without sacrificing the quality of the solution. Once again, we do this optimization in a sequential manner via `Greedy`:

$$\widehat{\mathcal{V}}_{\text{LM}}(\theta) = \texttt{TopN}\big(p_{\text{next}}(v_{l+1} \mid \theta^k_{\leq l})\big), \tag{5}$$

$$\mathcal{J}_{\text{LM}}(v, \theta^k_{\leq l}, \theta^{\neg k}) = -r\big((\ldots, \theta^{k-1}, \theta^k_{\leq l} \oplus v, \theta^{k+1}, \ldots)\big) + \beta \; p_{\text{next}}(v \mid \theta^k_{\leq l}), \tag{6}$$

Algorithm 1 Sequential Prompt Search (Greedy)

```
$\theta \leftarrow$ [initial_prompt]$\times K$
for $l = 0$ to $L - 1$ do
    class_order $\leftarrow$ randomly sampled order of class indices
    for $k$ in class_order do
        criteria $\leftarrow -\infty$
        for $v$ in $\widehat{\mathcal{V}}(\theta)$ do                ▷ This step is vectorized in practice.
            score $\leftarrow \mathcal{J}(v, \theta^k_{\leq l}, \theta^{\neg k})$                ▷ Evaluate the score of $v$.
            if score > criteria then                ▷ Keep the prompt with best performance.
                criteria $\leftarrow$ score                ▷ Update the current best score.
                $\theta^k_{l+1} \leftarrow v$                ▷ Update $\theta^k$ with the better token.
            end if
        end for
    end for
end for
Return $\theta$
```

where β is a hyperparameter that controls the strength of the regularization and $\texttt{TopN}(\cdot)$ is the set of tokens with N highest values of $p_{\text{next}}(v_{l+1} \mid \theta^k_{\leq l})$. The number of tokens to search over is also a hyperparameter that can be adjusted according to the computational constraints. We refer to this version of search as *regularized greedy*.

5 Generalization Guarantees for Prompts

As alluded to earlier, deriving generalization bound is closely connected to assigning hypotheses prior probabilities (that is, before seeing the training data) of them being good hypothesis [Shalev-Shwartz and Ben-David, 2014]. The most naive approach is to assign a uniform probability, $\frac{1}{|\Theta|}$, to each hypothesis. With a uniform prior, we recover the well-known uniform convergence bound:

Theorem 5.1 (Shalev-Shwartz and Ben-David [2014])**.** *For every $\delta > 0$, with probability $1 - \delta$ over the training set of size n, for any hypothesis $\theta \in \Theta$, the following holds*

$$R(\theta) \leq r(\theta) + \sqrt{\frac{\log|\Theta| + \log(\frac{1}{\delta})}{2n}}. \tag{7}$$

Since the space of all prompts is discrete, for a single hypothesis $\hat{\theta}$, we have the following uniform convergence bound for prompts that depend on the prompt length, the number of classes, and the number of tokens in the vocabulary:

$$R(\hat{\theta}) \leq r(\hat{\theta}) + \sqrt{\frac{L\,K\ \log|\mathcal{V}| + \log(\frac{1}{\delta})}{2n}}. \tag{8}$$

However, not all prompts are equally likely to be correct. To obtain a tighter generalization guarantee on the learned $\hat{\theta}$, we will leverage a classical PAC-Bayes bound to derive an upper bound on the generalization error of the learned prompts:

Theorem 5.2 (McAllester [1999])**.** *For every $\delta > 0$, prior P over Θ, with probability $1 - \delta$ over the training set of size n, for any posterior Q over Θ, the following holds*

$$\mathbb{E}_{\theta\sim Q}[R(\theta)] \leq \mathbb{E}_{\theta\sim Q}[r(\theta)] + \sqrt{\frac{D_{KL}(Q \parallel P) + \log(\frac{n}{\delta}) + 2}{2n - 1}}. \tag{9}$$

In conventional deep learning, P and Q are often chosen to be isotropic Gaussian distributions [Langford and Caruana, 2001] so the KL-divergence between the prior and posterior can be easily computed. We use a language model as the prior over K independent prompts,

$$P(\theta) = \prod_{i=1}^{K}\prod_{j=1}^{L} p_{\text{LM}}(\theta^i_j \mid \theta^i_{\leq j}).$$

Table 2: Performance and generalization bounds for prompts produced by `Greedy` and for handcrafted prompts on different datasets and using different CLIP architectures. UC represents the uniform convergence bound. Handcrafted prompts are taken from CLIP and Wise-FT [Wortsman et al., 2022].

Dataset	**Model**	**Method**	Train Err	Test Err	UC	PAC-Bayes
CIFAR-10	B-16	`Greedy`	0.050	0.060	0.201	0.086
	L-14	`Greedy`	0.023	0.028	0.175	0.059
	L-14	`handcrafted`	0.040	0.040	0.192	0.077
CIFAR-100	B-16	`Greedy`	0.208	0.255	0.688	0.317
	L-14	`Greedy`	0.142	0.180	0.621	0.251
	L-14	`handcrafted`	0.221	0.221	0.699	0.329
fMoW	B-16	`Greedy`	0.598	0.621	0.902	0.667
	L-14	`Greedy`	0.514	0.547	0.819	0.584
	L-14	`handcrafted`	0.725	0.402	0.877	0.795
OfficeHome	B-16	`Greedy`	0.104	0.150	0.879	0.281
	L-14	`Greedy`	0.070	0.115	0.845	0.248
	L-14	`handcrafted`	0.926	0.928	1.700	1.104
ImageNet	L-14	`handcrafted`	0.243	0.256	0.543	0.312

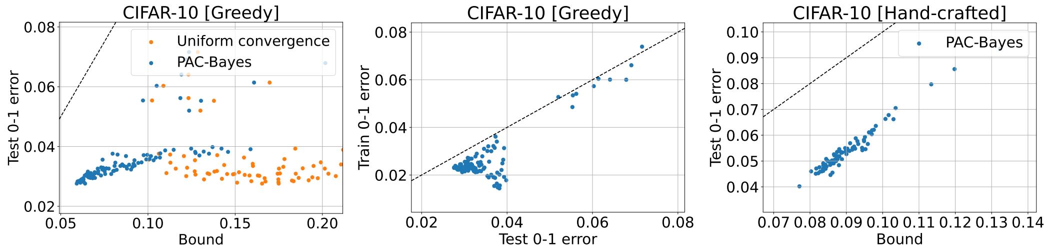

Figure 1: Test error vs generalization bound on CIFAR-10. We report the uniform convergence bound (left) and PAC-Bayes bound (middle, right), when evaluated on prompts produced by `Greedy` (left, middle) and handcrafted prompts (right). The dashed line is $y = x$.

Further, we treat the prompts $\hat{\theta}$ found through search or through prompt engineering as a point mass posterior, $Q(\theta) = \mathbb{1}\{\theta = \hat{\theta}\} = \prod_{i=1}^{K} \prod_{j=1}^{L} \mathbb{1}\{\theta_j^i = \hat{\theta}_j^i\}$. In this case, the KL-divergence is conveniently equal to the negative log-likelihood of $\hat{\theta}$ under the language model because the posterior is zero everywhere except for at $\hat{\theta}$:

$$D_{\mathrm{KL}}(Q \,\|\, P) = \sum_{\theta \in \Theta} Q(\theta) \log \frac{Q(\theta)}{P(\theta)} = \log \frac{1}{P(\hat{\theta})} = -\sum_{i=1}^{K} \sum_{j=1}^{L} \log p_{\mathrm{LM}} \left(\hat{\theta}_j^i \mid \hat{\theta}_{\leq j}^i \right). \tag{10}$$

This bound gives an intuitive interpretation, which is that the generalizing prompts are the ones that achieve good training performance and are likely under the language model. Having a point-mass posterior over discrete space also means that we can *derandomize* the PAC-Bayes bound for free [Viallard et al., 2021]. Combining these observations, we have the following deterministic upper bound on the generalization error:

$$R(\hat{\theta}) \leq r(\hat{\theta}) + \sqrt{\frac{-\sum_{i=1}^{K} \sum_{j=1}^{L} \log p_{\mathrm{LM}} \left(\hat{\theta}_j^i \mid \hat{\theta}_{\leq j}^i \right) + \log(\frac{n}{\delta}) + 2}{2n - 1}}. \tag{11}$$

In the next section, we will observe that this bound is *surprisingly* tight even for complex datasets such as ImageNet.

6 Experiments

In this section, we evaluate `Greedy` on CIFAR-10, CIFAR-100, as well as domain generalization datasets fMoW and OfficeHome. We also evaluate existing well-performing handcrafted prompts

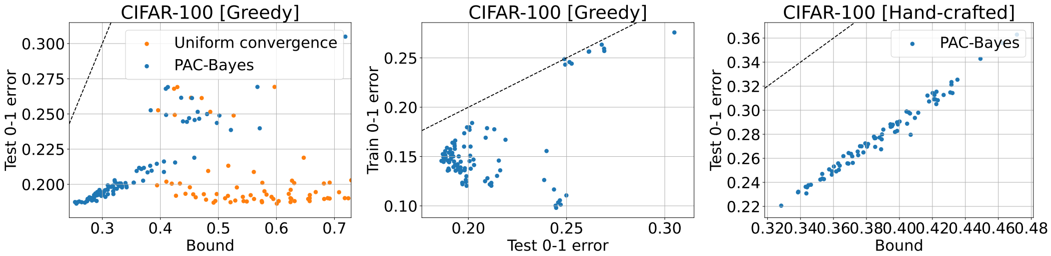

Figure 2: Test error vs generalization bounds on CIFAR-100. We report the uniform convergence bound and PAC-Bayes bound, when evaluated on prompts produced by `Greedy` (left). We plot its train vs. test error (middle). We also report the performance of handcrafted prompts and their corresponding PAC-Bayes bound (right). The dashed line is $y = x$.

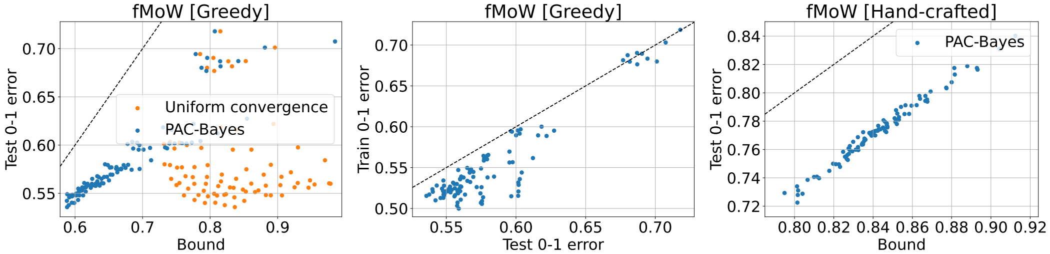

Figure 3: Test error vs generalization bounds on fMoW. We report the uniform convergence bound and PAC-Bayes bound when evaluated on prompts produced by `Greedy` (left). We plot its train vs. test error (middle). We also report the performance of handcrafted prompts and their corresponding PAC-Bayes bound (right). The dashed line is $y = x$.

taken from CLIP and Wise-FT [Wortsman et al., 2022]. Given these prompts, we compute generalization bounds via PAC-Bayes (`PAC-Bayes`) and via uniform convergence (`UC`). The PAC-Bayes bounds are computed using LLaMA-7B [Touvron et al., 2023] as the prior. For `Greedy`, We search using the CLIP vocabulary of $49\,408$ tokens and measure the generalization bounds for 100 realizations of `Greedy` with each corresponding to a fixed prompt length $l \in \{1, \ldots, 10\}$ and split portion of the dataset $s \in \{0.1, \ldots, 1.0\}$. We extract features from the ViT-B/16 and ViT-L/14 CLIP models and normalize them to have a unit norm. While generalization bounds on CIFAR-10 and CIFAR-100 have been well studied in the literature, much less is known about datasets closer to what is obtainable in practice like fMoW [Christie et al., 2018] and OfficeHome [Venkateswara et al., 2017].

Baselines We compare our generalization bounds against the state-of-the-art generalization bounds for deep learning CIFAR-10, CIFAR-100, and ImageNet. In particular, we compare to the works of Lotfi et al. [2022] and Zhou et al. [2019] which represent the latest progress in PAC-Bayes bounds for deep learning. As shown in Table 1, we achieve much tighter bounds than the existing state-of-the-art across all reported datasets. We remark that our approach is also data-independent, while still achieving a tighter bound than the data-dependent results from the work of Lotfi et al. [2022].

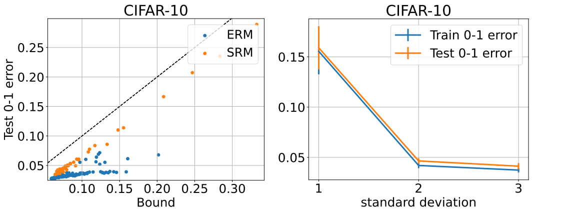

Figure 5: Test error vs the PAC-Bayes generalization bound on CIFAR-10 when using SRM (i.e., directly penalizing the PAC-Bayes bound). (left). We also report the train and test performance when the CLIP vocabulary is pruned using the language model. (right). These yield prompts with tighter bound at the cost of slightly higher error.

Structural risk minimization with the PAC-Bayes bound PAC-Bayes is related to SRM [Vapnik and Chervonenkis, 1974], where one tries to optimize both the goodness of fit and complexity of the model. We consider using structural

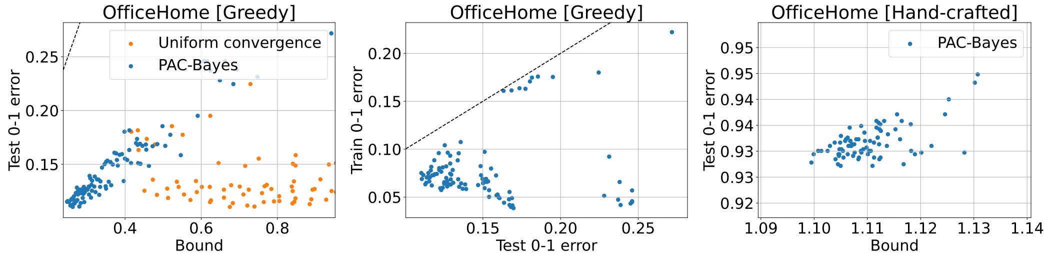

Figure 4: Test error vs generalization bounds on OfficeHome. We report the uniform convergence bound and PAC-Bayes bound when evaluated on prompts produced by Greedy (left). We plot its train vs. test error (middle). We also report the performance of handcrafted prompts and their corresponding PAC-Bayes bound (right). The handcrafted prompts perform poorly for this task so the resulting bound for them is vacuous. The dashed line is $y = x$.

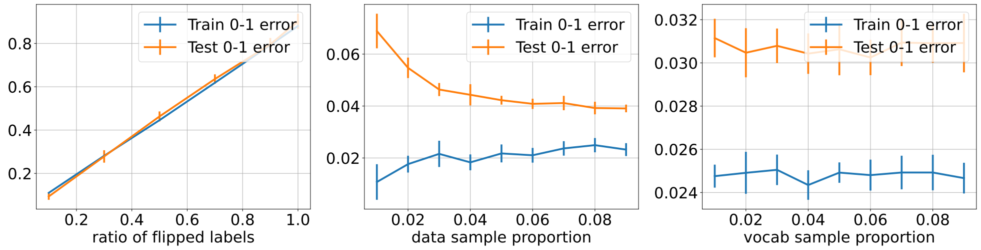

Figure 7: We show the generalization of Greedy with randomly labeled data on CIFAR-10(left). We also report the performance of Greedy when search is done with 1% - 9% of the labeled data (middle), and when search is done with 1% - 9% of the tokens in the CLIP vocabulary (right). We fix the prompt length to be 5.

risk minimization, where our complexity term is exactly the KL divergence term in Equation 10. As such, our Greedy search now jointly maximizes train accuracy and minimizes this KL divergence term when adding new tokens to each class prompt. We observe that this naturally leads to slightly tighter bounds for prompts yielded by Greedy on CIFAR-10 (Figure 5), while slightly degrading the accuracy of the prompt.

Effects of prompt length Another key quantity of the hypothesis class determined by prompt engineering is the prompt length. We also analyze how the length of class prompts impacts the performance of Greedy (Figure 6). We note that at a certain length, the train accuracy plateaus, which denotes that a relatively small prompt length suffices for strong performance.

Fitting random labels Since the prompts form a much smaller hypothesis, we hypothesize that the learned prompts are more robust to noise in the data. Zhang et al. [2021] showed that conventional deep neural networks can fit both *random labels* and *random data*, arguing that these models have much higher capacity than what traditional statistical

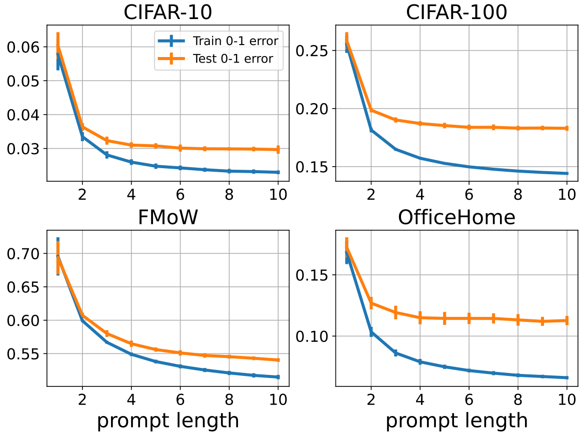

Figure 6: The train and test accuracy on CIFAR-10 and CIFAR-100, FmoW and OfficeHome with different prompt lengths for greedy search. Although the generalization gap does increase with prompt length, it can be seen that there is very little overfitting even at the longest lengths.

learning theory can deal with. To demonstrate that prompt engineering does not overfit, we experiment with running `Greedy` on training data with a certain proportion of randomly flipped training labels. We observe that training accuracy significantly drops as we flip these training labels (Figure 7), supporting that the hypothesis space of prompts does not overfit to random noise in the training dataset. As such, this supports that prompt engineering defines a hypothesis class with rather low complexity that is able to achieve high performance on complex tasks for which the pretrained vision-language model is well-suited.

Fitting with small data We also run experiments to evaluate the effectiveness of `Greedy` in limited labeled data settings. Again, the hypothesis class defined by discrete prompts is rather simple and should have a relatively small sample complexity. As such, we believe that this would yield comparatively stronger performance than more complex models with few labeled examples per class. In Figure 7, we report the train and test accuracy of `Greedy` as we vary the amount of training data (between 1%–10% of the full data) we use in computing the search objective. We observe less than 2% increase in error with 1% of the vocabulary. We include plots for even smaller data sizes in the Appendix A.

Pruning the hypothesis space The runtime of `Greedy` is dominated by inference time of the CLIP text encoder, as it extracts the embedding for each new candidate token. We can reduce the runtime by pruning the search space and restricting it to a smaller vocabulary. To understand the sensitivity of `Greedy` to the vocabulary size, we randomly sample a small subset of the vocabulary and run `Greedy` on this subset. We observe that the performance of `Greedy` is not sensitive to the vocabulary size (Figure 7). In addition to regularizing the search objective with the KL term directly, another way to constrain the hypothesis space is to prune the vocabulary using the language model. We experiment with conditioning the language model on the class names and then selecting tokens from the language model's vocabulary with the lowest log-likelihood. In Figure 5 we report the performance and generalization of `Greedy` when the set of tokens in search is restricted to within k standard-deviation from the minimum log-likelihood. While the vocabulary size of LLaMA-7b is 32 000 tokens, the number of tokens within 3, 2, 1 standard deviations from the minimum log-likelihood token are 6894, 1361, 185 respectively. We observe this implicitly prunes the hypotheses to contain those with good generalization at small cost to the train and test error while also reducing the runtime of the algorithm. One may wish to use a vocabulary that encodes prior knowledge about the data or domain or has desirable properties. Further results using a vocabulary of English words in Appendix A show that we can learn somewhat interpretable prompts.

7 Conclusion and Limitations

In this paper, we study the generalization properties of engineered prompts on image recognition tasks. We observe the surprising fact: prompt engineering does not seem to overfit, performing well on the test distribution. We provide a principled approach to analyze this generalization behavior by framing discrete prompts as a relatively small hypothesis class, onto which we can naturally apply classical PAC-Bayes bounds using a LLM prior. This results in the *tightest* bounds yet observed across multiple complex datasets, including CIFAR-10, CIFAR-100, and ImageNet. As a whole, this supports the use of prompt-engineering or simple greedy searches over potential class prompts as a high-performing and well-generalizing classifier.

From a broader perspective, it is worth emphasizing to what degree the PAC-Bayes generalization bounds here can really "explain" or allow us to "understand" prompt engineering. Obviously, despite the ability to produce highly non-vacuous bounds, the bounds rely on the fact that pretrained vision-language models already contain some hypothesis class that will perform well on the training set (for whatever the desired task is). This, in turn, naturally relies on the generalization performance of the underlying model itself, which our bounds of course do not address (naturally, as they depend only on the prompt, the bounds do not provide any linkage between e.g. CLIP's train/test performance). But what our bounds *do* address is the fact that when *given* these performant models, manual prompt engineering (even when "overfitting" to a test set) often exhibits *surprisingly* strong generalization behavior. Given the prevalence of prompt engineering in modern ML, we believe that this provides an important perspective on this widespread practice.

References

S. Bach, V. Sanh, Z. X. Yong, A. Webson, C. Raffel, N. V. Nayak, A. Sharma, T. Kim, M. S. Bari, T. Févry, et al. Promptsource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, 2022. 2

P. L. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. *ArXiv*, abs/1706.08498, 2017. 1, 3

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*, 33:1877–1901, 2020. 2

G. Christie, N. Fendley, J. Wilson, and R. Mukherjee. Functional map of the world. In *CVPR*, 2018. 7

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. *arXiv preprint arXiv:2010.11929*, 2020. 1

G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017. 1, 3, 4

G. K. Dziugaite, A. Drouin, B. Neal, N. Rajkumar, E. Caballero, L. Wang, I. Mitliagkas, and D. M. Roy. In search of robust measures of generalization. *Advances in Neural Information Processing Systems*, 33:11723–11733, 2020. 3

G. K. Dziugaite, K. Hsu, W. Gharbieh, G. Arpino, and D. Roy. On the role of data in pac-bayes bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 604–612. PMLR, 2021. 2

T. Gao, A. Fisch, and D. Chen. Making pre-trained language models better few-shot learners. In *Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL-IJCNLP 2021*, pages 3816–3830. Association for Computational Linguistics (ACL), 2021. 2

C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 2

Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019. 3

J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1

J. Langford and R. Caruana. (not) bounding the true error. In *NIPS*, 2001. 3, 5

T. Le Scao and A. M. Rush. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, 2021. 2

B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021. 2

X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021. 2

P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 2

S. Lotfi, M. Finzi, S. Kapoor, A. Potapczynski, M. Goldblum, and A. G. Wilson. Pac-bayes compression bounds so tight that they can explain generalization. *Advances in Neural Information Processing Systems*, 35:31459–31473, 2022. 1, 2, 3, 7

D. A. McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999. 1, 2, 3, 4, 5

V. Nagarajan and J. Z. Kolter. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019a. 3

V. Nagarajan and J. Z. Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019b. 3

B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014. 3

B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. *ArXiv*, abs/1503.00036, 2015. 3

B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017a. 3

B. Neyshabur, S. Bhojanapalli, D. A. McAllester, and N. Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *ArXiv*, abs/1707.09564, 2017b. 1, 3

M. Pérez-Ortiz, O. Rivasplata, J. Shawe-Taylor, and C. Szepesvári. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22, 2021. 3

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. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 5

Y. Tewel, Y. Shalev, I. Schwartz, and L. Wolf. Zero-shot image-to-text generation for visual-semantic arithmetic. *arXiv preprint arXiv:2111.14447*, 2021. 3

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 7

V. Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991. 4

V. Vapnik and A. Chervonenkis. Theory of pattern recognition, 1974. 4, 7

V. N. Vapnik. Chervonenkis: On the uniform convergence of relative frequencies of events to their probabilities. 1971. 3

H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 7

P. Viallard, P. Germain, A. Habrard, and E. Morvant. A general framework for the disintegration of pac-bayesian bounds. *arXiv preprint arXiv:2102.08649*, 2021. 6

Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *arXiv preprint arXiv:2302.03668*, 2023. 2

M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 6, 7

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 1, 2, 3, 8

K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022a. 2

W. Zhou, V. Veitch, M. Austern, R. P. Adams, and P. Orbanz. Non-vacuous generalization bounds at the imagenet scale: A pac-bayesian compression approach. In *7th International Conference on Learning Representations, ICLR 2019*, 2019. 2, 3, 7

Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022b. 2

Supplementary Material for
Understanding prompt engineering does not require rethinking generalization

A Additional Results

Pruning the hypothesis space In addition to the result on CIFAR-10 in Figure 7, we report the performance of greedy on CIFAR-100 when a random subset of the CLIP vocabulary is used in Figure 8. We observe less than 2% increase in error with 1% of the vocabulary. This provides further evidence of the robustness of `Greedy` to the vocabulary size. Random sampling, while easy to implement, prunes hypotheses that may have desirable properties. As such, we report the performance of greedy on CIFAR-100 when the vocabulary is pruned using the language model, and observe that `Greedy` is able to recover prompts with better generalization (See Figure 9).

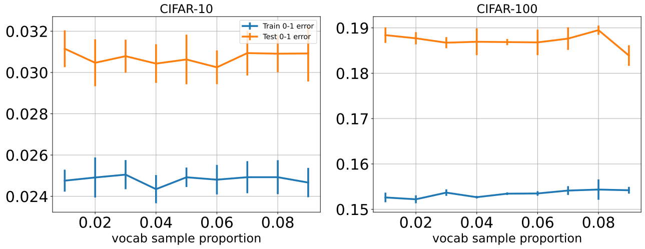

Figure 8: We show the generalization of `Greedy` when search is done with 1% - 9% of the tokens sampled randomly from the CLIP vocabulary on CIFAR-10 (left), and CIFAR-100 (right). We fix the prompt length to be 5.

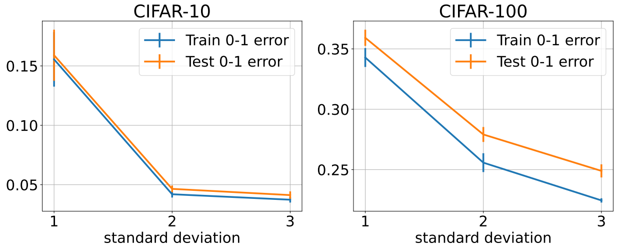

Figure 9: We show the generalization of `Greedy` when search is done with subsets of the tokens sampled from the language model as described in the text on CIFAR-10(left), and CIFAR-100(right). We fix the prompt length to be 5.

Fitting random labels In addition to the result on CIFAR-10 in Figure 7, we report results on fitting to randomly labelled data for CIFAR-100, FmoW, and OfficeHome in in Figure 10, and observe consistently that `Greedy` does not fit random labels. This provides evidence that contrasts the current literature on the ability for neural networks to easily fit random labels.

Fitting with small data In Figure 11 we report results on fitting to small sample sizes on both CIFAR-10 and CIFAR-100. We consider random subsets between 1% – 10% of the data and between 0.1% – 1%. We observe that `Greedy` is able learn even with small sample sizes with good generalization that degrades as the sizes decrease.

Learning with a different vocabulary The `Greedy` algorithm is agnostic to the set of tokens used in the search procedure. In practice, one may use a vocabulary that encodes prior knowledge about the data or domain. Additionally, certain properties like intepretability may be desired. We

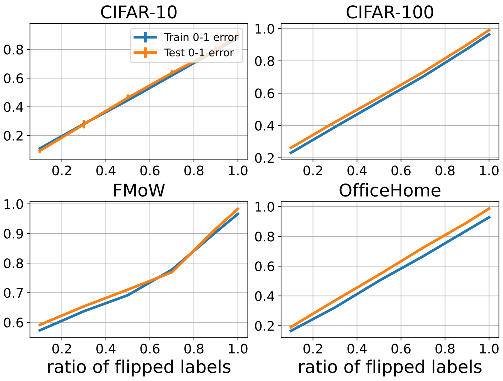

Figure 10: We show the generalization of `Greedy` with randomly labeled data on CIFAR-10, CIFAR-100, FmoW, and OfficeHome. We fix the prompt length to be 5.

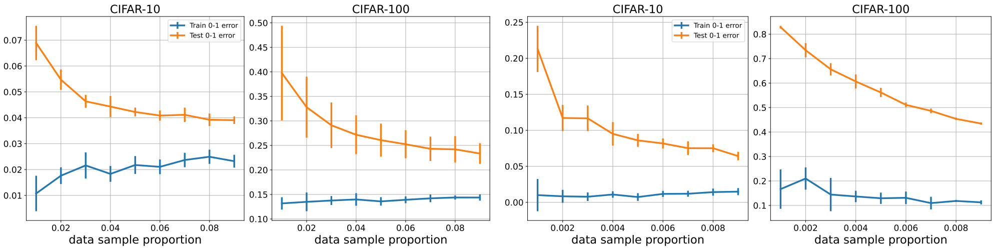

Figure 11: We show the generalization of `Greedy` when search is done with 1% - 9% of the data sampled randomly on CIFAR-10 (left), and CIFAR-100 (second-left). We also show the generalization of `Greedy` when search is done with 0.1% - 0.9% of the data sampled randomly on CIFAR-10 (second-right), and CIFAR-100 (right). We fix the prompt length to be 5.

report results on searching with the language model's vocabulary (See Figure 12. We do not observe significant degradation in performance. We also report results on penalizing the search criteria using the bound (i.e SRM) with different β values (See Figure 13). We observe that `Greedy` is able to recover prompts with better generalization as the penalty increases at a small cost to accuracy. We run `Greedy` on a vocabulary of English words obtained from the *english-words*[2] package. We show the prompts learned on CIFAR-10 in Figure 14.

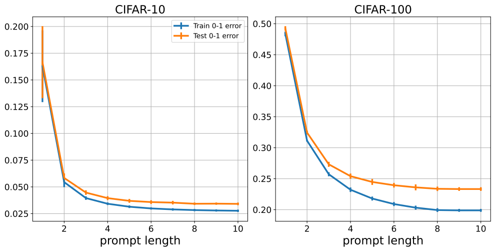

Figure 12: We show the generalization of `Greedy` with the Llama-7b vocabulary on CIFAR-10 (left) and CIFAR-100 (right).

B Experimental Details

Hyperparameters We report the hyper-parameters used in CLIP, LLaMA-7b, and the `Greedy` algorithm in Table 3.

[2] https://github.com/mwiens91/english-words-py

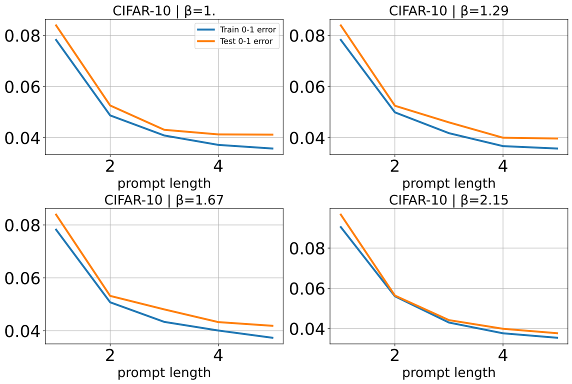

Figure 13: We show the generalization of `Greedy` with the LLaMA-7b vocabulary on CIFAR-10 with different values of penalty β with the SRM objective.

```
[airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck]
```

```
aviarist nonsonant confirmment establishmentism hemiteratics
nonmotoring known vaticinal allot nth
ornithophile slimsy renishly redivive muchness
wheencat compearant stintedness osiery thisness
stagnature unchawed lophobranch primariness primariness
dogrib babooism pneumococcic kaoliang bogusness
froggery phthalid auhuhu rippit hideousness
horsemonger fooyoung inordinary spreadingness forthbring
seafaring rumness tralatition babeship knocking
truckling phthartolatrae semantology waywarden decess
```

Figure 14: We show the learned prompts using a full-word vocabulary of English words on CIFAR-10. This achieves 3.3% test error with the L-14 base model.

Table 3: Hyperparameters used in CLIP, LLaMA-7b and `Greedy`.

Hyperparameter	Value
Batch size	100
CLIP Vocabulary size	49,408
LLaMA-7B Vocabulary size	32,000
Temperature	1.0
Bound δ	0.99
SRM β	1.0

Compute All experiments were run on cluster with 10 Quadro RTX 8000 GPUs with each experiment running on one GPU at a time.