MODEL DIFFUSION FOR CERTIFIABLE FEW-SHOT TRANSFER LEARNING

Fady Rezk[1, 2], **Royson Lee**[2], **Henry Gouk**[1], **Timothy Hospedales**[1,2], **Minyoung Kim**[2]
[1] School of Informatics, University of Edinburgh, UK
[2] Samsung AI Research Center, UK

ABSTRACT

In modern large-scale deep learning, a prevalent and effective workflow for solving low-data problems is adapting powerful pre-trained foundation models (FMs) to new tasks via parameter-efficient fine-tuning (PEFT). However, while empirically effective, the resulting solutions lack generalisation guarantees to certify their accuracy - which may be required for ethical or legal reasons prior to deployment in high-importance applications. In this paper we develop a novel transfer learning approach that is designed to facilitate non-vacuous learning theoretic generalisation guarantees for downstream tasks, even in the low-shot regime. Specifically, we first use upstream tasks to train a *distribution over PEFT parameters*. We then learn the downstream task by a *sample-and-evaluate* procedure – sampling plausible PEFTs from the trained diffusion model and selecting the one with the highest likelihood on the downstream data. Crucially, this confines our model hypothesis to a *finite* set of PEFT samples. In contrast to learning in the typical continuous hypothesis spaces of neural network weights, this facilitates tighter risk certificates. We instantiate our bound and show non-trivial generalization guarantees compared to existing learning approaches which lead to vacuous bounds in the low-shot regime.

1 INTRODUCTION

Generalisation certificates are crucial for high-importance applications where accuracy should be guaranteed for legal or ethical reasons. Guarantees should certify the minimum testing accuracy expected on unseen data drawn from the training distribution. However, it is hard to establish non-trivial guarantees for large neural networks, since large learning capacity tends to produce looser guarantees. As such, there have only been a few successful demonstrations of non-vacuous guarantees for contemporary neural networks, even in the large-data regime (Dziugaite & Roy, 2017; Perez-Ortiz et al., 2021; Lotfi et al., 2024).

What about learning with sparse rather than large data? The problem of low-data learning is highly topical, due to the plethora of important limited-data applications Wang et al. (2020), but challenging due to the difficulty of learning a large number neural network parameters without overfitting. This need has inspired several lines of research that make use of different forms of knowledge transfer, including meta-learning Hospedales et al. (2021) and parameter-efficient transfer learning (PEFT) Hu et al. (2021) from foundation models. While PEFT methods have recently been more empirically effective, neither family of approach has produced methods that can provide low-shot generalisation guarantees, to our knowledge. From a learning theoretic perspective this is because existing algorithms still search a hypothesis space (e.g., all neural network weights $\theta \in \mathcal{R}^N$) large enough to make known bounds vacuous when instantiated.

This paper introduces a novel approach to knowledge transfer that ultimately learns downstream tasks by *picking from a finite set of hypothesis*, where the set of hypothesis is fit to the upstream tasks. Our method, STEEL (Sample ThEn Evaluate Learner), facilitates using classic finite-hypothesis bounds, which are simple and tight, but not typically used in contemporary machine learning – which focuses on learning continuous value neural network parameters.

More specifically, in the upstream phase, we fit PEFT modules to available source tasks, and then train a parameter diffusion model to generate PEFTs according to this task distribution. In the

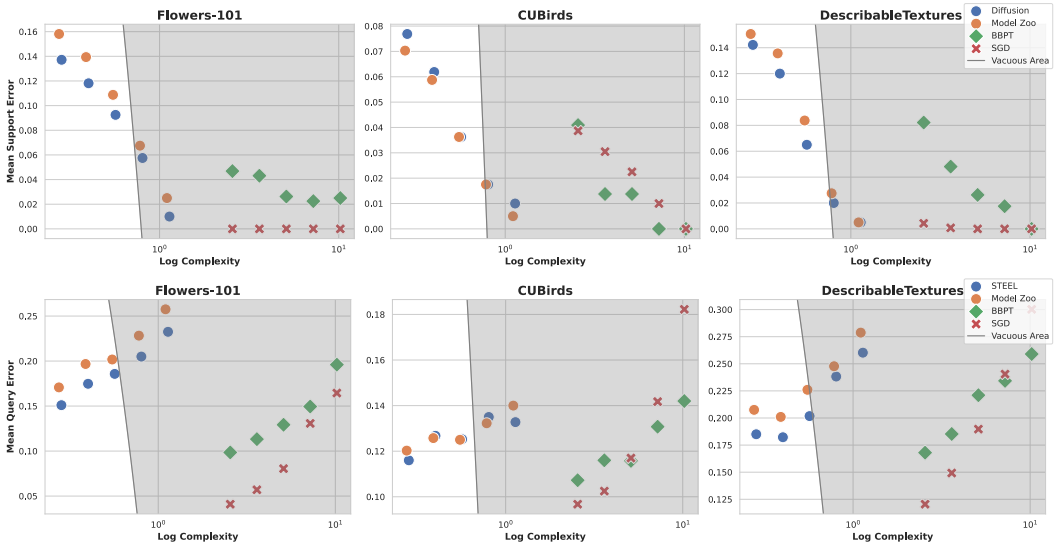

Figure 1: Generalisation bounds for adapting CLIP to novel tasks (5-way classification with from 1 to 16 examples per class). Plots show error (y-axis) vs log-complexity term of the bound of Equation 3 (x-axis). Top/Bottom: The mean support/query (train/test) error for new tasks. The unshaded area corresponds to non-vacuous guarantees, while in the shaded area the total sum of both axes is vacuous. Other approaches are not even close to providing non-vacuous guarantees, while our approach can do so without sacrificing too much fit quality (top), or empirical test accuracy (bottom).

downstream phase, we learn by sample-then-evaluate instead of traditional gradient descent. PEFT modules, unconditionally generated by the diffusion model, are scored using the target task training set, and learning is to choose the highest scoring module. Compared to the original set of upstream models, the diffusion model can be more compact, and can interpolate between the original model set to achieve higher accuracy. This procedure is gradient-free, which has some scalability benefits Malladi et al. (2023); Rezk et al. (2024), but more importantly it facilitates the use of PAC-Bayes finite-hypothesis bounds to provide non-vacuous guarantees, all while maintaining similar empirical accuracy to mainstream few-shot learning approaches. Figure 1 shows some illustrative results, demonstrating our learner's ability to maintain practical efficacy while being constrained to low enough complexity to provide non-vacuous guarantees (white zone).

In summary, our contributions are: (1) Introducing a novel learning paradigm for gradient-free transfer learning designed to facilitate accuracy guarantees for downstream tasks, even in the low-shot regime. (2) The first practical demonstration of non-vacuous generalization bounds for low-shot learning in large language and vision architectures.

2 Risk Certificates for Deep Models

Certifying model generalization performance is fundamental in theoretical machine learning (Vapnik, 1995; Shalev-Shwartz & Ben-David, 2014; Mohri et al., 2018). VC, Rademacher, and PAC-Bayes bounds connect empirical risk (computable) to generalization risk (impossible to compute) through inequalities. Here we discuss the core concepts of risk certificates and why they're challenging to apply to deep models. Let $h \in \mathcal{H}$ be a hypothesis (prediction function $y = h(x)$) and $\mathcal{H}$ the hypothesis space. In deep learning, h corresponds to a model with parameters θ, and Θ ($\ni \theta$) represents all possible parameter values, serving as the hypothesis space. A learning algorithm selects θ from Θ given empirical data $S = (x_i, y_i)_{i=1}^n$ sampled i.i.d. from distribution T. The goal is minimizing generalization risk $R(\theta) = \mathbb{E}_{(x,y)\sim T}[l(\theta; x, y)]$, where $l(\theta; x, y)$ is the instance risk. Since computing $R(\theta)$ is impossible without access to T, we minimize empirical risk $r(\theta) = \frac{1}{n}\sum_{(x,y)\in S} l(\theta; x, y)$.

Theoretical bounds relate R and r as follows. For any $\theta \in \Theta$:

$$R(\theta) \leq r(\theta) + \text{ComplexityTerm}(\dim(\Theta), n) \tag{1}$$

where the complexity term depends on data size n and hypothesis space complexity $\dim(\Theta)$, decreasing as n increases and $\dim(\Theta)$ decreases[1]. The right side of (1) is the risk certificate, guaranteeing an upper bound on generalization risk. Certificates above 1 (with $l \in [0, 1]$) are vacuous, while those below 1 are non-vacuous (or sometimes less than the random guess risk in the classification setting). Traditional models like linear SVM achieve non-vacuous bounds through proper regularization (Vapnik, 1995; Shalev-Shwartz & Ben-David, 2014). However, conventional deep neural networks trained by gradient descent lack non-vacuous bounds unless n is extremely large, due to high $\dim(\Theta)$ from numerous parameters. Even sparse adapter methods (e.g., LoRA (Hu et al., 2021), LoRA-XS (Balazy et al., 2024)) face challenges from the continuous nature of Θ. No existing deep learning approach achieves meaningful generalization bounds in low-shot settings.

3 Proposed Approach

3.1 Our Approach at High Level

We propose a novel approach using a *finite* hypothesis space and gradient-free learning, departing from traditional continuous methods. In multi-task transfer learning, we learn a task distribution via a diffusion generative model, creating a *finite* hypothesis space Θ from model-generated samples. Our simple learning algorithm selects the $\theta \in \Theta$ with minimal empirical risk, using heuristic search for efficiency with large Θ to reduce the forward-pass overhead (as detailed in Sec. 3).

This approach, combined with finite-hypothesis PAC-Bayes bounds, yields tight non-vacuous risk certificates on large-scale LLM/vision benchmarks using FLAN-T5/CLIP models. Importantly, test performance remains comparable to standard learning algorithms. The *evaluate-then-select* strategy optimizes PAC-Bayes bounds by keeping the complexity term constant. We present the formal problem setup before detailing our approach in Sec. 3.3.

Before we jump into our main approach in greater detail in Sec. 3.3, we will formally define the problem setup.

3.2 Problem Setup and Notation

We describe the *low-shot cross-task transfer learning* problem that we aim to solve in the paper. Consider that we have a training pool of *related* tasks $T_1, \ldots, T_N$, i.i.d. sampled from some unknown but true task distribution $p_{true}(T)$. At test time, we are given a new test task T^* sampled from the same $p_{true}(T)$, but only in the form of low-shot data samples $S^* = \{(x_i, x_y)\}_{i=1}^n$ (aka support data) from T^*. By low shot, we mean the number of support samples n is small. Our goal is to find a way to ensure tight generalization bounds for the underlying deep models. As discussed in Sec. 2, a main challenge here is that we have low-shot data S^* and a large number of model parameters, where the latter immediately translates into high hypothesis space complexity. Hence, applying the traditional learning theories directly to this problem leads to vacuous error bounds. We come up with a new method that exploits the training tasks to transfer the knowledge to unseen tasks so that it can offer tight non-vacuous risk certificates.

3.3 Details of Our Approach

Our first observation is that gradient-based model adaptation to low-shot samples S^* must be avoided to reduce hypothesis space complexity (Sec. 2). Our key intuition is that we can learn the task distribution $p_{true}(T)$ from training tasks $\{T_i\}_{i=1}^N$, but doing so introduces a strong inductive bias or regularization. Let θ_i be the learned neural network parameters for task T_i. (Throughout, we treat PEFT adapter parameters as θ, keeping the pre-trained backbone fixed.) We view θ_i as the best description of T_i and collect task-wise parameters $\{\theta_i\}_{i=1}^N$. Learning $p_{true}(T)$ thus reduces to a *density estimation problem*, i.e., estimating $p(\theta)$ from i.i.d. samples $\{\theta_i\}_{i=1}^N$, treating $p(\theta)$ as a surrogate for $p_{true}(T)$.

[1] In the PAC-Bayes bounds, the risks are measured as expected risks over some (posterior) distribution over θ rather than point estimates. Also the complexity term is expressed in terms of divergence from a prior distribution. But if we confine the posterior to be sharply concentrated at a single point θ and use a flat prior, this roughly follows the form of (1). Also certain PAC-Bayes bounds have nonlinear relation between R and r, however, they can be approximated as (1), where this simplification does not affect the reasoning in our paper.

We learn a diffusion model $p(\theta)$ with $\{\theta_i\}_{i=1}^N$ as training data following (Ho et al., 2020). At test time, the estimated $p(\theta)$ serves as a proxy for $p_{true}(T)$. We generate plausible candidate samples θ (i.e., tasks T) and select the one closest to T^*. Since only S^* of T^* is available, we choose the sample with the least discrepancy from S^* via the minimum loss rule: $\arg\min_{\theta\in\Theta} \sum_{(x,y)\in S^*} l(\theta; x, y)$, where Θ is the set of diffusion model candidates. Crucially, this corresponds to empirical risk minimization where Θ acts as the *hypothesis space*.

This strategy amounts to *selection from a finite hypothesis space*, rather than searching a continuous space via gradient-based fine-tuning. The choice of a finite hypothesis set (diffusion model samples) before learning enables a strong regularizing inductive bias learned from upstream tasks.

3.4 Risk Certificate with Finite Hypothesis Space

Suppose we have a well-trained zoo of (PEFT adapter) parameters $\overline{\Theta} = \{\theta_i\}_{i=1}^N$, where each θ_i is the optimal parameters for the i-th training task ($i = 1, \ldots, N$). Without the diffusion training with $\overline{\Theta}$, nothing can prevent us from using the the model zoo $\overline{\Theta}$ itself as our hypothesis space, i.e., $\Theta = \overline{\Theta}$. This becomes more reasonable as N goes larger since $\overline{\Theta}$ would be richer and closer to the true $p_{true}(T)$. We will call this strategy of $\Theta = \overline{\Theta}$ the *model zoo* strategy. In contrast, another proposal of ours is to train a diffusion model $p(\theta)$ with $\overline{\Theta}$, and build Θ using the samples from it, which we call the STEEL[2]. Whereas both are our proposals, the diffusion is our main strategy as it is more attractive in the following aspects: i) the model zoo strategy requires a large amount of space to store the N models while the diffusion strategy is more scalable, only needing to store the diffusion model itself; ii) The diffusion model is widely known to have strong interpolation capability to approximate the target density better, which in practice can provide improved test accuracy

Few-shot adaptation is done by evaluate-then-select:

$$\theta^* = \arg\min_{\theta\in\Theta} r(\theta) = \frac{1}{n} \sum_{(x,y)\in S^*} l(\theta; x, y) \tag{2}$$

We expect that Θ is rich enough to represent the true task distribution $p_{true}(T)$ faithfully, and the adapted ("selected") θ^* will generalize well on unseen samples from T^*.

A crucial benefit of our test-time adaptation strategy (2) is that we have a tight provable generalization error bound that can serve as a risk certificate for its test-time prediction quality. This mainly originates from the *finite* hypothesis space Θ. More specifically, using the PAC-Bayes theorems (e.g., Sec. 2.1.3 in (Alquier, 2021)), we can show that with probability at least $1 - \epsilon$,

$$R(\theta) \le r(\theta) + C \cdot \sqrt{\frac{\log \frac{|\Theta|}{\epsilon}}{2n}} \quad \text{for any } \theta \in \Theta \tag{3}$$

where $R(\theta) = \mathbb{E}_{(x,y)\sim T^*}[l(\theta; x, y)]$ is the generalization error of θ, and C is the maximal loss value (i.e., $0 \le l \le C$). The bound immediately comes from the PAC-Bayes theorem with the (data-independent) uniform prior over Θ and the Dirac's delta posterior choice. Since the size of the hypothesis space $|\Theta|$ only appears in the $\log$ term, a massively large Θ is allowable while retaining a tight bound. Furthermore, the bound can be minimized with the smallest empirical error $r(\theta)$, i.e., $\theta = \theta^*$, which justifies our evaluate-then-select strategy (2).

However, the computational question naturally arises: *How do we solve (2) efficiently?* We consider two solutions:.

- **Exhaustive search.** We go through every θ in Θ, evaluate the loss $r(\theta)$, and choose the minimum one. Even though we guarantee to find θ^* always with the minimal empirical $r(\theta)$, this is computationally very expensive (often intractable for the LLM cases due to prohibitive $|\Theta|$ forward passes or text generation).
- **Hierarchical search.** This is the well-known tree search strategy to find an approximate solution. For instance, we can do hierarchical clustering of θs in Θ, evaluate the losses of

[2]Alternatively, we can union model zoo and diffusion samples together to build a larger hypothesis space. This is practically effective, but in this paper we focus only on model zoo and diffusion strategies so as to contrast their behaviors more carefully.

the top level clusters (either cluster centroids or medoids), and choose the best cluster. Then we focus only on the θs that belong to the selected cluster, discarding the rest, and go on recursively. This may find a good approximate solution θ close to θ^* in $O(\log |\Theta|)$ time. However, we may possibly end up with suboptimal (underfit) empirical error $r(\theta)$.

We emphasize again that in all these three strategies, the generalization error bound (3) holds true, but with possibly differently/suboptimally selected θs in the latter case, which may imply (slight) increase in the empirical loss $r(\theta)$, and hence a slight loosening of the obtained certificate.

4 Related Work

Risk certificates. While traditionally applied to simple models, recent work extends risk certificates to deep learning. Notable approaches include using data-dependent priors (Perez-Ortiz et al., 2021) and parameter quantization of PEFT adapters (Lotfi et al., 2024), though these require large training sets. In multi-task settings, Zakerinia et al. (2024) proposed meta-learning PAC-Bayes bounds but only demonstrated on toy problems and easily incurs out-of-memory computational issue due to the nested gradient computations.

Model diffusion methods. Several works explore diffusion for generating model parameters: NNDiffusion (Wang et al., 2024) for BN modules, ProtoDiff (Du et al., 2023) for ProtoNet-based few-shot learning, and MetaDiff (Zhang et al., 2024) and D2NWG (Soro et al., 2025) for gradient-free meta-learning. Scaling properties of Diffusion based learning were also explored (Schürholt et al., 2024). However, none provide risk certificates for the generated models.

Sparse adapter (PEFT) methods. Sparse adapters reduce learnable parameters, crucial for our diffusion-based sampling. While LoRA (Hu et al., 2021) uses trainable low-rank matrices and VeRA (Kopiczko et al., 2024) uses fixed matrices with trainable diagonals, we adopt LoRA-XS (Balazy et al., 2024), which uses SVD with a trainable full matrix for the singular values. Such adapters have facilitated large-data guarantees Lotfi et al. (2024), but in our framework they will facilitate low-shot guarantees.

5 Experiments

5.1 Guarantees for few-shot LLM adaptation

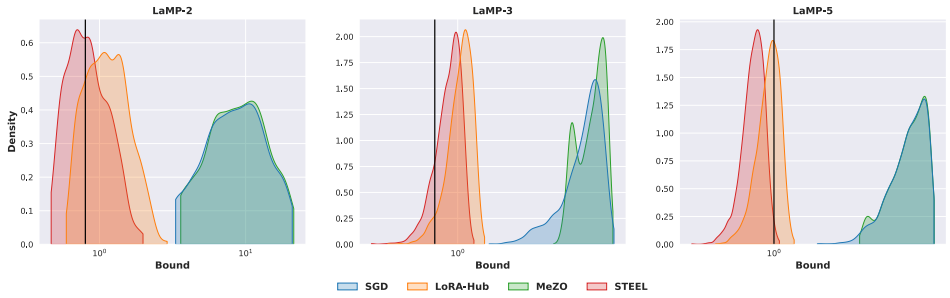

Figure 2: Distribution of generalisation guarantees (x-axis, log scale) obtained over few-shot LLM adaptation episodes. Vertical lines indicate the vacuous bound threshold. Our STEEL learner provides a dramatically better distribution of outcomes in terms of provable generalisation compared to alternatives.

Datasets: For low-shot LLM adaptation, we use the LaMP personalization benchmark Salemi et al. (2024). LaMP contains a fixed number of users per dataset that is split over training (seen) and evaluation (unseen) clients. Training and evaluation clients are mutually exclusive. Each client has their own support data and query data. We choose three datasets from the benchmark, namely LaMP-2 (Personalized Movie Tagging), LaMP-3 (Personalized Product Rating), and LaMP-5 (Personalized Scholarly Title Generation). These make nominal classification, ordinal classification, and text generation tasks respectively.

Figure 3: Results on the LaMP LLM adaptation benchmark.

		SGD	LoRA-Hub	MeZO	STEEL
LaMP-2	% Non-Vacuous Tasks	0.00%	32.13%	0.00%	**65.12%**
	Median Gap	8.12	0.68	8.45	**0.43**
	Min Bound	3.32	0.59	3.60	**0.47**
	Median Bound	8.52	1.12	8.85	**0.80**
	Max Bound	20.86	2.90	21.36	**1.99**
	Accuracy↑	63.25%	57.51%	63.30%	**63.74%**
	F1↑	56.15%	50.84%	57.03%	**55.69%**
LaMP-3	% Non-Vacuous Tasks	0.00	5.00	0.00	**15.48**
	Median Gap	3.09	0.23	3.10	**0.14**
	Min Bound	1.35	0.51	2.54	**0.43**
	Median Bound	3.56	1.04	3.76	**0.93**
	Max Bound	4.75	1.30	4.53	**1.17**
	MAE↓	0.217	**0.230**	0.242	0.231
	RMSE↓	0.511	0.526	0.531	**0.524**
	Cross-Entropy↓	0.479	0.739	**0.626**	0.693
LaMP-5	% Non-Vacuous Tasks	0.00	63.84	0.00	**99.16**
	Median Gap	4.04	0.41	4.04	**0.26**
	Min Bound	1.61	0.52	2.58	**0.40**
	Median Bound	4.57	0.96	4.57	**0.81**
	Max Bound	5.86	1.25	5.88	**1.06**
	ROUGE-1↑	47.04%	47.05%	47.03%	**47.22%**
	ROUGE-L↑	42.79%	42.75%	42.73%	**42.89%**

Setup: Following recent work on LaMP Tan et al. (2024); Salemi et al. (2024), we first build a base model with task-specific capabilities by end-to-end fine-tuning of Flan-T5 base Chung et al. (2024) on the user support data seen. Subsequently, to personalize the base model for a user, we train one LoRA-XS module on the support data of each training client Bałazy et al. (2024). LoRA-XS rank of 6 and an alpha of 16 was used, producing a total of 2592 tunable parameters. We build the model zoo by collecting the LoRA-XS modules trained using seen users support data; same data that was aggregated to finetune the base model.

Baselines: We compare our proposal of using the model zoo directly, and our diffusion samples against two gradient-free methods, namely LoRA-Hub Huang et al. (2024), MeZO Malladi et al. (2023), and include SGD baseline for completeness. Please note that LoRA-Hub randomly samples from the model zoo and learns a new adapter as a linear combination of the sampled adapters. MeZO and SGD do not use the zoo. We refer the reader to Appendix 9.2 for detailed hyperparameters. STEEL uses the bound described in Equation 3. Meanwhile, MeZO, SGD and LoRA-Hub use the quantization bound from Equation 5. Finally, for computational efficiency, we use Hierarchical search as described in Section 3.4.

Results: Our main contribution relates to the ability to provably certify the generalisation of low-shot learning. In terms of low-shot LLM adaptation, Figure 2 visualises the distribution of certification outcomes over a large number of episodes for the three LAMP benchmarks. Taking note of the log-scale on the x-axis for generalisation guarantee strength, we can see that our STEEL learner provides dramatically better guarantees than conventional continuous-parameter learner alternatives, thanks to its discrete hypothesis space. The vertical lines indicate the threshold for vacuous bounds. Standard learners such as SGD and MeZO have no mass left of the threshold, while a substantial number of STEEL learning episodes are non-vacuously guaranteed. Table 3 provides more detailed quantitative results in terms of various metrics. To assess provable generalisation, the data visualised in Figure 2 is summarised as the % non-vacuous metric (the fraction of episodes which have guarantees with a strength above chance-level), and the median guarantee strength across episodes. To asses practical performance we report relevant metrics for each task such as Accuracy, RMSE, and ROGUE score.

From these results we can see that: (1) Standard supervised learning approaches such as (gradient-based) SGD and (gradient-free) MeZO have no non-vacuous episodes - no few-shot learning task can be guaranteed. (2) Our STEEL model has the most non-vacuous episodes for each benchmark, with almost every few-shot learning episode being guaranteed in the LAMP-5 benchmarks. And the median STEEL episode has a substantially non-vacuous guarantee for all three benchmarks. (3) Interesingly, LoraHUB combined with Lotfi et al. (2024)'s discretization bound also has some non-vacuous episodes, but less than STEEL. (3) STEEL has comparable or better empirical test accuracy compared to existing approaches such as SGD and MeZO, while providing a huge improvement in certifiability.

5.2 Guarantees for few-shot visual recognition

Datasets: For vision, we use fine-grained classification datasets that have readily available training (seen) and novel (unseen) classes split. We choose CUBirds Wah et al. (2011), FGVC-Aircraft Maji et al. (2013), Describable Textures Cimpoi et al. (2014) and Flowers-101 Nilsback & Zisserman (2008). We use the split offered by learn2learn for the first three datasets Arnold et al. (2020), and the Flowers-101 split from Meta-Dataset Triantafillou et al. (2020).

Setup: We sample random few-shot n-shot k-way learning tasks as per meta-learning literature Triantafillou et al. (2020). We randomly select 5 classes (5-way) from a given split and for each class we randomly sample n-shots. We evaluate 1, 2, 4, 8 and 16-shots. The model zoo is built by sampling

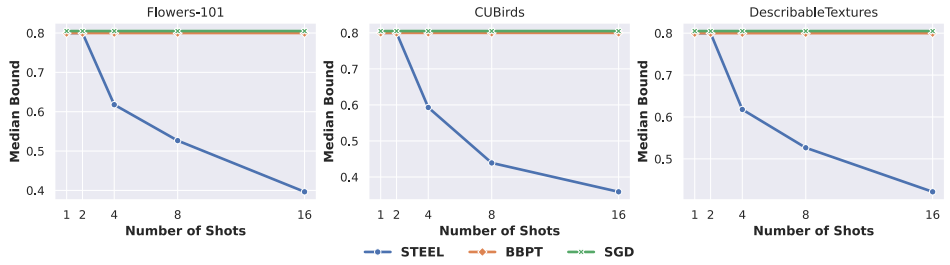

Figure 4: Dependence of generalisation guarantee on training set size. Our finite-hypothesis class learner STEEL achieves non-vacuous guarantees from 4-shot onward. Standard approaches provide no guarantees anywhere in this low-shot range.

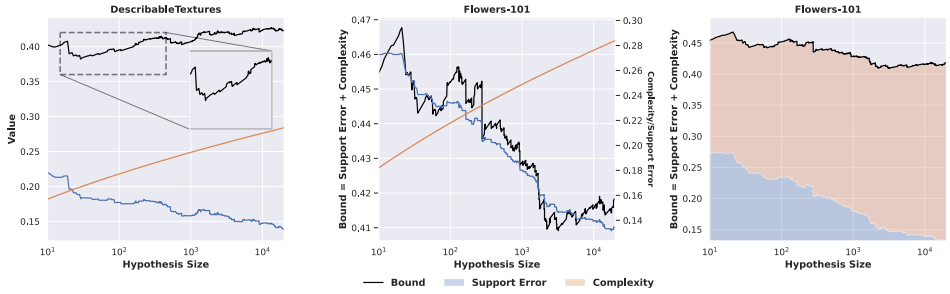

Figure 5: "Learning curves" illustrating empirical and certified learning dynamics of STEEL with respect to samples/iterations, which is equivalent to hypothesis space size. More samples improves the training (support) error, while increasing the complexity penalty. The sum of these two terms instantiates the generalisation guarantee (Eq. 3) achieved for a given number of samples.

from the training classes of the aforementioned datasets. Meanwhile, we evaluate on unseen classes and samples. Please note that training and evaluation classes are mutually exclusive. For a given task, we use CoOp for adaptation Zhou et al. (2022) which is simply prompt tuning for CLIP. We finetune a 2-token prompt appended in front of the class name for every task to build a model zoo. This results in total tunable parameters of length 1024. For vision experiments, we found that the forward passes were fast enough to conduct exhaustive search over sampled prompts from the diffusion model.

Baselines: We compare our proposal of using the model zoo directly, and our diffusion samples against Black-Box Prompt Tuning (BBPT) Yu et al. (2023) which is a gradient-free version of CoOp, optimizing the prompt using evolutionary optimization. We also include SGD based CoOp for completeness. For details on methods hyperparameters, we refer the reader to Appendix 9.3. STEELuses the bound described in Equation 3. Meanwhile, BBPT and SGD use the quantization bound from Equation 5.

Results: The results in terms of mean training and testing error versus complexity are summarised for three datasets in Figure 1. The dots for each learner reflect the training set sizes of 1, 2, 4, 8, and 16-examples per-class, and the white/grey zone separation delineates the space of non-vacuous vs vacuous bound outcomes. The main message is that only our finite hypothesis class approaches achieve any non-vacuous guarantees across this whole range of training set sizes. Every result for the standard SGD and BBPT approaches is vaccuous and cannot be guaranteed.

For the 16-shot case, these experiments are quantified in more detail in Table 6. Similarly to the results for LLM adaptation, we can see the dramatic difference in % of non-vacuous episodes, and dramatic improvement in the min, median and max bound obtained over episodes. Compared to the LLM case, our STEEL pays a slighly higher price here in terms of empirical test accuracy compared to SGD for some benchmarks, however this is still small compared to the stark difference in certification performance.

Figure 4 highlights the evolution of the median generalisation bound as a function of the training set size. For STEEL it becomes non-vacuous from 4-shots onward, and the standard approaches never become non-vacuous[3]

Further Analysis: We finally discuss and provide some insight into the learning process of our discrete hypothesis class learner. Standard gradient-descent takes repeated update steps to find a model that better fits a training set. By analogy, our STEEL gradient-free learner would draw more samples as it attempts to iteratively sample a model that better fits the training set. Our main experiments use a fixed number of 20,000 samples on all vision datasets, but Figure 5 illustrates our learner's behaviour by showing the equivalent of a learning curve for our model. The x-axis is the number of samples drawn, and equivalently the learning theoretic hypothesis space size. Unlike SGD, this means that there is a direct dependence of hypothesis class complexity ($|\Theta|$ in Eq. 3) and the number of iterations/samples. This is reflected in the steadily increasing red complexity curve in Figure 5(left, middle). We can also see that the training/support error goes down consistently over iterations/samples as the sampler progressively discovers better models. The generalisation bound (black line) is given by the sum of the training error and complexity. The figure illustrates one case (Flowers, middle, right) where the bound continues to improve up to a large number of samples/hypothesis size, because the continued improvement in training error outweighs the complexity gain. It also illustrates a case (DTD, left) where the training error improvement is slower and quite rapidly outweighed by the complexity gain, so that the best bound is actually achieved after quite a small number of samples.

Method	SGD	BBPT	STEEL
CUBirds			
Non-Vacuous Ratio	0.00%	0.00%	**100.00%**
Average Gap	2.49	2.48	**0.24**
Min Bound	2.55	2.55	**0.30**
Median Bound	2.58	2.59	**0.36**
Max Bound	2.64	2.68	**0.47**
Average Accuracy	90.32	**89.27**	88.40
Describable Textures			
Non-Vacuous Ratio	0.00%	0.00%	**100.00%**
Average Gap	2.43	2.46	**0.24**
Min Bound	2.55	2.55	**0.36**
Median Bound	2.55	2.63	**0.42**
Max Bound	2.58	2.78	**0.53**
Average Accuracy	87.95	**83.20**	81.50
FGVCAircrafts			
Non-Vacuous Ratio	0.00%	0.00%	**97.50%**
Average Gap	2.31	2.45	**0.22**
Min Bound	2.58	2.64	**0.45**
Median Bound	2.64	2.80	**0.61**
Max Bound	2.78	3.01	**0.85**
Average Accuracy	65.57	**62.37**	61.37
Flowers-101			
Non-Vacuous Ratio	0.00%	0.00%	**100.00%**
Average Gap	2.51	2.50	**0.27**
Min Bound	2.55	2.55	**0.31**
Median Bound	2.55	2.59	**0.40**
Max Bound	2.55	2.70	**0.61**
Average Accuracy	95.90	**90.15**	84.90

Figure 6: CLIP+CoOp-based few-shot learning. Aggregates over multiple 16-Shots 5-way learning episodes.

Sampler vs Zoo: Our approach compresses the upstream set of pre-trained models into a learned model generator. Selecting among the upstream models using downstream task performance as a criterion provides an alternative approach to learning that also corresponds to a finite hypothesis space. Our generator approach was motivated by ensuring scalability with respect to a large number of upstream models, and also to improve accuracy by enabling interpolation between upstream models rather than solely being limited to selecting one of them. Figure 1 shows that STEEL's diffusion sampler tends to provide improved accuracy compared to its raw model zoo, especially for Flowers and DTD. On average across all datasets STEEL consistently outperforms Model Zoo. Detailed per-dataset per-shot performance is deferred to Appendix 10, Table 1.

6 Conclusion

We have introduced a novel Sample-Then-Evaluate approach to transfer learning. Our STEEL is designed to facilitate non-trivial performance guarantees, even in the low-shot regime, through use of a discrete hypothesis space. Our results instantiating the bound demonstrate that for both LLM and visual transfer learning STEEL performs comparably to alternatives it terms of test accuracy, while being dramatically better in terms of ability to provably guarantee this performance level.

[3]Note their bound is substantially worse than 0.8, but for simple visualisation, we plot it as chance-level for 5-way classification.

REFERENCES

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. *arXiv preprint arXiv:2110.11216*, 2021.

Sébastien M. R. Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for meta-learning research, 2020. URL https://arxiv.org/abs/2008.12284.

Klaudia Bałazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. LoRA-XS: Low-rank adaptation with extremely small number of parameters, 2024. URL https://openreview.net/forum?id=l80AgHoRaN.

Klaudia Balazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. LoRA-XS: Low-Rank Adaptation with Extremely Small Number of Parameters. *arXiv preprint arXiv:2405.17604*, 2024.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53, 2024. URL https://jmlr.org/papers/v25/23-0870.html.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Yingjun Du, Zehao Xiao, Shengcai Liao, and Cees Snoek. ProtoDiff: Learning to Learn Prototypical Networks by Task-Guided Diffusion. In *Advances in Neural Information Processing Systems*, 2023.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *UAI*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, 2020. In Advances in Neural Information Processing Systems.

Timothy M Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3079209.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TrloAXEJ2B.

Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based Random Matrix Adaptation. In *International Conference on Learning Representations*, 2024.

Sanae Lotfi, Marc Anton Finzi, Yilun Kuang, Tim G. J. Rudner, Micah Goldblum, and Andrew Gordon Wilson. Non-vacuous generalization bounds for large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=6Kg9p8URlj.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Vota6rFhBQ.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Maria Perez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba SzepesvÃ¡ri. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227):1–40, 2021. URL http://jmlr.org/papers/v22/20-879.html.

Fady Rezk, Antreas Antoniou, Henry Gouk, and Timothy Hospedales. Liouna: Biologically plausible learning for efficient pre-training of transferrable deep models. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ICML 2024)*, 2024. URL https://openreview.net/forum?id=bYwg5Awx6n.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization, 2024. URL https://arxiv.org/abs/2304.11406.

Konstantin Schürholt, Michael W. Mahoney, and Damian Borth. Towards scalable and versatile weight space learning, 2024. URL https://arxiv.org/abs/2406.09997.

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

Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018. URL https://arxiv.org/abs/1708.07120.

Bedionita Soro, Bruno Andreis, Hayeon Lee, Wonyong Jeong, Song Chong, Frank Hutter, and Sung Ju Hwang. Diffusion-based neural network weights generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j8WHjM9aMm.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. Personalized pieces: Efficient personalized large language models through collaborative efforts. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6459–6475, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.371. URL https://aclanthology.org/2024.emnlp-main.371/.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgAGAVKPr.

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Cubirds dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Kai Wang, Dongwen Tang, Boya Zeng, Yida Yin, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural Network Diffusion. *arXiv preprint arXiv:2402.13144*, 2024.

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), June 2020. doi: 10.1145/3386252.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes, 2020. URL https://arxiv.org/abs/1904.00962.

Lang Yu, Qin Chen, Jiaju Lin, and Liang He. Black-box prompt tuning for vision-language model as a service. In Edith Elkind (ed.), *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pp. 1686–1694. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/187. URL https://doi.org/10.24963/ijcai.2023/187. Main Track.

Hossein Zakerinia, Amin Behjati, and Christoph H. Lampert. More Flexible PAC-Bayesian Meta-Learning by Learning Learning Algorithms. 2024.

Baoquan Zhang, Chuyao Luo, Demin Yu, Huiwei Lin, Xutao Li, Yunming Ye, and Bowen Zhang. MetaDiff: Meta-Learning with Conditional Diffusion for Few-Shot Learning. 2024.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022.

7 APPENDIX

8 EXISTING RISK BOUNDS FOR DEEP MODELS

8.1 VANILLA PAC-BAYES BOUND

This is the vanilla, non-transfer learning bound. As a baseline, one can also contrast with the vanilla PAC-Bayes bound (i.e., non-meta learning bound). This essentially follows the Cantoni's bound, and can be written as follows. With probability at least $1-\epsilon$, the following holds:

$$\mathbb{E}_{Q(\theta)}[R(\theta)] \;\leq\; \mathbb{E}_{Q(\theta)}[r(\theta)] + \sqrt{\frac{\mathrm{KL}(Q||\pi) + \log(1/\epsilon)}{2n}} \tag{4}$$

Here n is the test support size. We can set $\pi = \mathcal{N}(0, \kappa^2 I)$ for some fixed κ_π and $Q(\theta) = \mathcal{N}(\mu, \Sigma^2)$ where the parameters (μ, Σ) can be learned by minimizing the right hand side. The sampled version $\theta^z = \mu + \sqrt{\Sigma} \cdot z$, $z \sim \mathcal{N}(0, I)$ can be used during the optimization. Once optimized, the minimum value of the right hand side serves as the error bound for the test task.

8.2 BOUND WITH PARAMETER-LEVEL QUANTIZATION

In (Lotfi et al., 2024), they proposed a non-vacuous bound for the LLM based on the model parameter quantization (e.g., fixed-length floating point machine representation). There are several differences to our approach:

1. The paper is about LLM pre-training setup with large training data, and the bound would be vacuous if training data size is not large enough (e.g., $\geq$ 10K).
2. They derive the same finite hypothesis space PAC-Bayes bound, but replace the $\log|H|$ term by $\log(1/p(h))$ where $p(h)$ is the prior likelihood, and $\log(1/p(h))$ is approximated and upper-bounded by $C(h)$ which is the number of bits for representing the hypothesis h.
3. The finite hypothesis space comes from the fixed-size floating point representation for real numbers (e.g., if there are d trainable parameters, then $C(h) = d \cdot 32$), but to reduce it further, they propose what is called the SubLoRA, which is a random subspace representation (i.e., $\theta = Pw$, $P =$ random subspace basis, w = coefficients) of the LoRA A/B matrices.
4. Also, instead of 32 bit for each of d params, they do some clustering to reduce it to shorter coding, more precisely the arithmetic coding.

The followings are some details of their bound derivation. With probability at least $1-\epsilon$,

$$R(\theta) \leq r(\theta) + C \cdot \sqrt{\frac{K(\theta) + 2\log K(\theta) + \log(1/\epsilon)}{2n}} \tag{5}$$

where $K(\theta)$ is the Kolmogorov complexity bound that can be estimated as:

$$K(\theta) = \sum_{i=1}^{d} (\#\text{ of bits in the arithmetic coding of } \theta_i) \tag{6}$$

where $d = \dim(\theta)$ for the PEFT parameters θ. The arithmetic coding requires clustering of parameters θ_is, thus being dependent on the particular θ used. However, we can consider the best (i.e., the tightest) bound possible. That is, even if we have 1 bit for every θ_i (the minimal code length possible), $K(\theta) = d$, and plugging this into (5) yields:

$$R(\theta) \leq r(\theta) + C \cdot \sqrt{\frac{d + 2\log d + \log(1/\epsilon)}{2n}} \tag{7}$$

which is the *best* scenario.

In (5), as before, $R(\theta) = \mathbb{E}_{z\sim T^*}[l(\theta; z)]$ is the generalization error of θ, $r(\theta) = \frac{1}{n}\sum_{z\in S^*} l(\theta; z)$ is the empirical error on the support data with size $n = |S^*|$, and C is the maximal loss value (i.e., $0 \leq l \leq C$).

9 ARCHITECTURE AND TRAINING RECIPES

9.1 DIFFUSION ARCHITECTURE AND TRAINING RECIPE

First, the diffusion model forwad encoder uses a 1000 timesteps with a linear scheduler over noise between 1e-4 to 2e-2. For the decoder, we use an MLP network for the diffusion model with 3 hidden-layers. The hidden layer dimension is $4\times$ the size of it's input. This is 10,240 for LaMP (divisible by 512 for parallelization concerns) and 4096 for vision. A layer conditioned time embedding of the diffusion step is added (summed with) the hidden layer's hidden representation.

The time embedding is generated from the diffusion timestep using sinsusoidal embeddings as per the original DDPM model Ho et al. (2020). The dimensionality of the sinusoidal embedding is equal to the Diffusion MLP hidden dimension. Subsequently, the embedding is transformed using a two-layer MLP with first layer expanding the dimension to $4\times$ the network's hidden dimension and the second layer downscaling again to original hidden dimension. For example, on the vision experiments, the time embedding network has hidden dimension of 4098×4. Finally, to condition the time embedding computed by the two-layer MLP time embedding network for each layer, we apply a different linear transformation per diffusion hidden layer.

We train the diffusion model for 30K epochs for all experiments with a batch size of 1,024. We use the LAMB optimizer You et al. (2020) with a learning rate of 0.01. For vision experiments, we found that we can continue improving performance if we continue training for a second stage of 10K more epochs. For the second stage, we use a one-cycle learning rate scheduler Smith & Topin (2018) with default hyperparameters (as found in pytorch). The maximum learning rate starts from 0.0004 reaching a maximum of 0.001 over a 1000 steps. We keep an exponential-moving average of the network weights throughout training with a decaying rate of 0.9999.

9.2 FLAN-T5 + LAMP HYPERPARAMETERS

For training the base model across all datasets, we use LaMP's original recipe Salemi et al. (2024). We use a batch size of 64, AdamW optimizer with a learning rate of 5e-5 and weight decay of size 0.0001. We use a linear warmup for the learning rate over 5% of the total number of training steps.

To build the model zoo, we found that we required to tune Adam optimizer learning rate and per-dataset epochs per-dataset. We optimizer the hyperparameters to improve performance on the training split (seen users) query data. We use learning rates of 0.01, 0.01, 0.0001 and 20, 10, 10 epochs for LaMP-2, LaMP-3 and LaMP-5 respectively. For all datasets, we used a linear warmup for the learning rate over 5% of the total number of training steps. We used the same recipe to train a per-user LoRA-XS model on the unseen users/novel tasks.

For LoRA-Hub, we used default hyperparameters as proposed by the original authors. First, we sample 20 random adapters from the model zoo. The weights of the linear combination is initialized with zeros and truncate min/max weights to -1.5/1.5. We do maximum inference steps of 40 with NeverGrad default hyperparameters.

Finally, we transform the SGD number of epochs to MeZO. The authors used $32\times$ as many epochs as SGD in the original paper Malladi et al. (2023). This translates to 640, 320, 320 epochs for LaMP-2, LaMP-3, and LaMP-5. We tested the three learning rates proposed to search over by the authors. We fixed a learning rate of 1e-3 across all datasets because we consistently found 1e-4 to not learn and 1e-2 to be unstable.

For LaMP, we sample 10K LoRA-XS modules from the diffusion model. We use k-means clustering on the diffusion samples to produce N clusters where N is chosen as the minimum Silhouette score. We evaluate N between [2,150] inclusive. For each cluster, we find the medoid; the adapter closest to the centroid of the cluster. During evaluation, we choose a cluster and evaluate all adapters therein. From the cluster, we short-list the best 15 adapters using the Flan-T5 training loss. On the best 15 adapters, we use text generation to produce an answer with greedy sampling. Using the LaMP benchmark proposed model selection metric for each dataset, we select the "winning" adapter.

9.2.1 BOUND METRICS

For support errors, we use 1-Accuracy for LaMP-2, and ROUGE-1 for LaMP-5. For LaMP-3, both RMSE and MAE are not bounded. Therefore, we devise a cross-entropy like metric for the dataset. First, we convert the ordinal vectors to one-hot encodings. Subsequently, we calculate the absolute error between the labels one-hot encoding and Flan-T5 model logits, and divide by 2. This guarantees the error to be bounded in the [0-1] range inclusive. We use this metric as support error term.

LaMP-2 Intricacies: LaMP benchmark has one query sample per user. For LaMP-3 and LaMP-5, this suffices since the generated support error is continuous. Nevertheless, for LaMP-2, the accuracy term, which we use to derive the support error, becomes the 0/1 loss. Therefore, we split the support data in novel tasks to support and query data with ratio 80% and 20% respectively. If the split generates only 1 query samples, we move one sample from support to query to have a minimum of two-samples in query. We use the same split for all evaluations across SGD, MeZO, LoRA-Hub and our proposed methods. For reproducibility, all splits were done deterministically. Furthermore, we did not constrain the split to have same classes across both support and query. LaMP classification tasks are long-tailed. Therefore, for a novel task, a user might have classes X and Y in support but the query ends up with classes A and B making it a more challenging benchmark for all methods.

Finally, we truncate the support sizes of LaMP-3 and LaMP-5 to 256 samples across all methods. This is done deterministically for reproducibility. The reason for truncating the dataset is pure computational concerns.

9.2.2 MODEL ZOO SIZE

For LaMP, we build a model zoo by training one PEFT adapter per-task in the dataset. Each user is treated as one task. This yields 3820, 20,000, and 9,682 total adapters/tasks in LaMP-2, LaMP-3, and LaMP-5 respectively.

9.3 CLIP + COOP HYPERPARAMETERS

To build the model zoo, we used the authors original hyperparameters to train CoOP because we found them to work the best. This is SGD with a learning rate of 0.002. For Flowers-101, we train for 200 epochs. For DTD, FGVCAircraft and CUBirds, we trained for 300 epochs and found a One Cycle learning rate useful to stabilize training. These same hyperparameters were used to evaluate SGD on novel tasks. For BBPT, we use the authors default hyperparameters Yu et al. (2023). We found that the method converges within 8000 "API call". We attempted to run for a budget of 20K as our diffusion model offers but found that performance did not improve. Please note that the original authors reduce the dimensionality of the prompt using a small network because evolutionary optimization struggles in high-dimension. They reduce dimensionality to 512. Nevertheless, since we train only 2-tokens (dimensionality=1,024), then we do not use the small dimensionality reduction network.

For vision experiments, we found that exhaustive search was fast enough even though we sample 20K adapters from the diffusion model.

Bound Metrics: we use $1 -$ Accuracy as the support error term in our bound calculation for all vision experiments.

Model Zoo Size: We randomly sample 16-shot 5-way tasks for building the model zoo from each respective dataset. We train 10,000 total tasks per-dataset for the model zoo. The diffusion model is trained on this model zoo. Once trained, we sample the diffusion model once and fix the samples across all downstream evaluation for 1, 2, 4, 8 and 16 shots.

10 EXTRA VISION RESULTS

Dataset	Zero Shot	SGD	BBPT	Model Zoo	Diffusion
		1-Shots			
CUBirds	83.82%	81.77%	85.80%	86.00%	86.72%
DescribableTextures	67.29%	69.97%	74.10%	72.12%	73.97%
FGVCAircraft	47.44%	53.00%	55.05%	54.37%	55.40%
Flowers-101	81.14%	83.55%	80.40%	74.25%	76.75%
Avg	69.92%	72.07%	73.84%	71.69%	73.21%
		2-Shots			
CUBirds	83.82	85.82%	86.92%	86.77%	86.50%
DescribableTextures	67.29	75.95%	76.57%	75.22%	76.17%
FGVCAircrafts	47.44	52.30%	57.90%	55.82%	57.30%
Flowers-101	81.14	86.92%	85.05%	77.17%	79.50%
Avg	69.92	75.25%	76.61%	73.75%	74.87%
		4-Shots			
CUBirds	83.82	88.30%	88.42%	87.50%	87.47%
DescribableTextures	67.29	81.02%	77.90%	77.40%	79.82%
FGVCAircrafts	47.44	58.65%	60.55%	57.55%	60.10%
Flowers-101	81.14	91.95%	87.07%	79.82%	81.42%
Avg	69.92%	79.98%	78.49%	75.57%	77.21%
		8-Shots			
CUBirds	83.82	89.75%	88.40%	87.42%	87.32%
DescribableTextures	67.29	85.07%	81.47%	79.90%	81.77%
FGVCAircrafts	47.44	62.07%	61.55%	58.77%	61.72%
Flowers-101	81.14	94.30%	88.67%	80.32%	82.52%
Avg	69.92%	82.80%	80.02%	76.61%	78.34%
		16-Shots			
CUBirds	83.82	90.32%	89.27%	87.97%	88.40%
DescribableTextures	67.29	87.95%	83.20%	79.25%	81.50%
FGVCAircrafts	47.44	65.57%	62.37%	61.02%	61.37%
Flowers-101	81.14	95.90%	90.15%	82.92%	84.90%
Avg	69.92%	84.94%	81.25%	77.79%	79.04%

Table 1: CLIP+CoOp few-shot learning. Accuracies over different number of shots.