Extracting information from fine-tuned weights

Nan Chen
Johns Hopkins University
nchen38@jh.edu

Hayden Helm
Helivan
hayden@helivan.io

Youngser Park
Johns Hopkins University
youngser@jhu.edu

Carey Priebe
Johns Hopkins University
cep@jhu.edu

Soledad Villar
Johns Hopkins University
soledad.villar@jhu.edu

Abstract

Pre-trained and fine-tuned language models have become a pervasive tool. While the behavior of a model is highly dependent on its training data, the data itself is often not available. However it is possible to infer properties of it via training artifacts such as the model's weights, internal representations, or its responses. As such, model weights are an increasingly popular data modality for predicting model-level and data-level covariates. These objects are in the non-Euclidean space of all possible weights modulo their symmetries (identifying different sets of weights that define the same function). Working in this space, we empirically observe that the fine-tuned weights capture information about the fine-tuning sets, consistent with recent investigations in the field. We argue that this may enable the prediction of fine-tuned weights from partially fine-tuned models. Our experimental findings indicate that, in certain cases, predicting fine-tuned weights is feasible. The results presented here are part of an ongoing research effort.

1 Introduction

The widespread deployment of fine-tuned language models has created a new data modality: the models themselves. Beyond their role as text processors, these models—and in particular their task-specific weight updates—offer a lens on the process that produced them. Treating weights as geometric objects allows analyses that complement performance-centric evaluation and may help us reason about how models change during fine-tuning.

Low-Rank Adaptation (LoRA) [14] has emerged as a dominant parameter-efficient fine-tuning approach, representing weight updates as products of two low-rank matrices. Critically, this representation exhibits General Linear (GL) group symmetries: different matrix factorizations can implement the same functional change. Specifically, for any invertible matrix M, the LoRA factorizations (A, B) and (AM, BM^{-T}) produce equivalent weight updates $\Delta W = AB^{\top}$. This fundamental symmetry defines a non-Euclidean geometry in the space of fine-tuning weights, where meaningful analysis requires GL-equivariant architectures that respect these transformations, as previously observed in [25].

Motivated by this perspective, we ask two questions: (i) to what extent do fine-tuning weights encode recoverable information about the datasets used for adaptation, and (ii) given access to partially fine-tuned model weights, is it possible to predict the remaining weight updates needed for complete adaptation? These questions examine whether weight updates during fine-tuning follow systematic patterns that can be learned and generalized across different fine-tuning scenarios. Answering them could enable more efficient fine-tuning strategies that leverage partial adaptation progress and provide

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Non-Euclidean Foundation Models: Advancing AI Beyond Euclidean Frameworks.

deeper insights into how models incrementally acquire task-specific knowledge through geometric changes in their weight spaces.

To address these questions, we develop a GL-equivariant neural architecture that processes LoRA weights while preserving their geometric structure. Our backbone combines equivariant linear transformations with a novel gated nonlinearity mechanism that maintains GL-equivariance while enabling expressive feature learning. We further introduce systematic scaling strategies, including a Mixture-of-Experts extension, to handle more demanding prediction tasks while respecting the underlying symmetries.

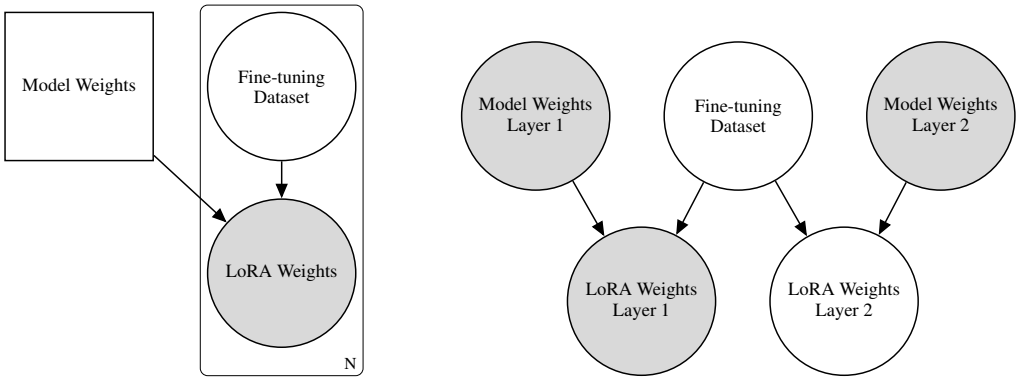


Figure 1: Probabilistic graphical models describing the learning tasks considered in this paper. The square nodes correspond to constants, the gray nodes correspond to observed variables, and the unshaded variables are latent and unobserved. (a) Given N LoRA weights for the same constant (unknown) model, fine-tuned with N different fine-tuning sets, we consider the task of inferring properties of the fine-tuning datasets given observations of the LoRA weights. (b) Given partial observations from early layers, we predict the unobserved fine-tuned weights in later layers by leveraging the joint dependency structure.

Using this framework, we report two empirical observations. First, LoRA weights contain recoverable information about the finetuning datasets, as illustrated by the graphical model in Figure 1a. Second, the weights from partially fine-tuned models contain sufficient information to enable prediction (to a certain extent) of subsequent LoRA updates for unseen layers as shown in Figure 1b. These results are preliminary and context-dependent, but they suggest that the LoRA updates leak information about the fine-tuning dataset and they induce structured relationships in the weight space that can be learned with symmetry-aware models. Our contributions can be summarized as follows:

- We empirically show that the fine-tuning weights encode recoverable information about the fine-tuning datasets. We formulate the problem as learning properties of the fine-tuning set from three different possible inputs:
  (i) the low rank factors $(A, B) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$ of the LoRA updates in Euclidean space,
  (ii) the non-Euclidean space of low rank factors $(A, B) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}/GL$ of weights modulo the GL symmetries, and
  (iii) the Euclidean space of Lora weights $\Delta W \in \mathbb{R}^{m\times n}$ (which naturally removes the need for the GL-symmetry).

  In our experiments, the optimal performance at learning information about the fine-tuning dataset from LoRA was obtained by a model defined on the space (ii) of low-rank updates modulo the symmetries. We believe this is because the effective dimension of the input is significantly reduced over the alternatives, and therefore the sample complexity of the problem is reduced accordingly. Some theoretical evidence for this claim is discussed in [28].
- We formulate machine learning models on the LoRA weights in the non-Euclidean space described above, defined modulo GL symmetries. To do so, we use a GL-equivariant backbone similar to [25] with a slightly more general gated nonlinearity mechanism.
- We present preliminary results showing that partially fine-tuned models contain sufficient structural information to enable prediction of subsequent weight updates.

We view this work as an exploratory step toward learning directly on weight updates with appropriate geometric symmetries. The findings have implications for developing more efficient fine-tuning strategies and gaining deeper insights into how models acquire task-specific knowledge through structured weight evolution. The broader scope—how far such methods generalize across different architectures, datasets, and training regimes—remains an open question for future investigation.

2 Related work

Parameter-Efficient Fine-Tuning Parameter-efficient fine-tuning (PEFT) has emerged as a crucial technique for adapting large language models to downstream tasks while minimizing computational and storage costs. PEFT approaches fall into three main categories [7, 16]: additive fine-tuning methods that introduce small trainable modules either within model architectures [26, 13, 23, 10] or as soft prompt engineering [20, 19, 24]; selective fine-tuning approaches that identify and update specific existing modules during adaptation [6, 21]; and reparameterized fine-tuning methods that decompose weight updates into low-dimensional representations. Low-Rank Adaptation (LoRA) [14] exemplifies the reparameterization approach, approximating task-specific weight updates through trainable low-rank matrices. This technique has spawned numerous variants targeting improved efficiency (DyLoRA[30], AdaLoRA [33], PLoP [9]) and enhanced stability (RSLoRA [18], LoRA+ [8]).

While existing work focuses on optimizing LoRA's performance and efficiency, we take a different perspective. We treat LoRA weight updates as geometric objects with inherent symmetries, investigating what information these weights encode and how this information can be leveraged for model adaptation.

Weight Space Learning on LoRA Recent works have begun treating LoRA weights as a data modality for analysis and processing. Dravid et al. [3] study the weight space of LoRA fine-tuned text-to-image diffusion models by applying PCA to flattened LoRA weights. Their constructed weight space enables operations, such as sampling and editing, to generate new LoRA weights for customized models. DSiRe[27] takes a Euclidean approach, extracting representations of LoRA weights by computing matrix norms and spectral properties to recover dataset size. In contrast, LoL [25] studies the GL-equivariance inherent in LoRA weight matrices and proposes GL-Net for deriving non-Euclidean representations that respect these symmetries. Our work builds upon the GL-equivariant perspective introduced by LoL [25], but makes several key advances. First, we develop a slightly more expressive GL-equivariant backbone incorporating an equivariant gating nonlinearity module. Second, we propose systematic strategies for extending this backbone to handle more complex tasks. Most importantly, while DSiRe and LoL focus on predicting hyperparameters or downstream performance from LoRA weights, we tackle a fundamentally different challenge: extracting fine-tuning information from partially trained models and leveraging this information to guide further adaptation.

Learning representations of models At the core of our methodology is treating a collection of LoRA weights as geometric objects. This perspective is part of a growing literature related to learning representations of language models across different model accessibility regimes. For example, [11] introduces DKPS – a method that maps a collection of responses to a low-dimensional Euclidean subspace that can be used to track the dynamics of a population of interacting language models. More recent work has shown that DKPS has desirable statistical properties [1] and can be applied to model-level inference problems such as predicting properties of a model's training data [12]. In settings where internal representations for inputs are available, [15] treats models across data modalities as geometric objects and argues that the models' representations are converging. In the same setting, [4] proposes representing models as an input-dependent nearest neighbor graph and shows that the geometry of the representations encodes meaningful information about each model's training data.

3 The space of finetuning weights and their symmetries

3.1 GL-Equivariance in LoRA Weight Space

LoRA [14] approximates weight updates through a low-rank factorization: for a weight matrix $W \in \mathbb{R}^{m\times n}$, the update is parameterized as $\Delta W = AB^\top$ where $A \in \mathbb{R}^{m\times r}$, $B \in \mathbb{R}^{n\times r}$, with $r \ll \min(m, n)$. There is an underlying symmetry to this parameterization [25]: for any invertible matrix $M \in \mathrm{GL}(r)$ we denote its inverse transposed as $M^{-\top}$, the factorizations (A, B) and $(AM, BM^{-\top})$ yield identical updates, since:

$$\Delta W = AB^\top = (AM)(BM^{-\top})^\top$$

This GL-invariance implies that any function f processing LoRA weights must be GL-equivariant. Namely, we define the GL action on pairs of matrices as:

$$M(A, B) = (AM, BM^{-\top}), \tag{1}$$

then $f : \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r} \to \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r}$ is GL-equivariant if for all $M \in \mathrm{GL}(r)$ we have

$$f(M(A, B)) = Mf(A, B). \tag{2}$$

Similar GL-symmetries also apply to other network parameters, appearing in query-key pairs and value-output pairs within self-attention mechanisms [29].

3.2 Symmetry Preservation in Weight Updates

We now examine the symmetric relationships between base weights and their corresponding LoRA updates, focusing on query-key projection matrices as a canonical example. Let $Q, K \in \mathbb{R}^{m\times n}$ denote the query and key matrices with LoRA updates (A_q, B_q) and (A_k, B_k) respectively. Following [29], the weight pairs (Q, K) and $(QM, KM^{-\top})$ are functionally equivalent for any $M \in \mathrm{GL}(n)$. If LoRA adaptation preserves this functional equivalence, the mapping g from weights to updates

$$g(Q, K) = (A_q, B_q^\top, A_k, B_k^\top) \tag{3}$$

must satisfy the following symmetry constraint:

$$g(QM, KM^{-\top}) = (A_q, B_q^\top M, A_k, B_k^\top M^{-\top}) \tag{4}$$

This reveals that the A_* matrices remain invariant under GL-transformations of the base weights, while the $B_*^\top$ matrices transform equivariantly—a structural property we exploit in our architecture design.

3.3 GL-Equivariant Backbone Architecture

The analysis in Sections 3.1 and 3.2 argues the GL-equivariance as a fundamental symmetry governing LoRA weight spaces and their relationships to base model parameters. To preserve the geometric information encoded in these weights, neural architectures must respect this structural constraint. In our experiments we will employ a GL-equivariant backbone. Following standard practice in equivariant architecture design, our model consists of stacked GL-equivariant layers followed by task-specific output heads. Each equivariant layer combines a GL-equivariant linear module with a gated nonlinearity mechanism that maintains symmetries while enabling expressive transformation. The output head produces either GL-equivariant or GL-invariant representations depending on the downstream task—invariant for dataset property prediction, equivariant for weight prediction tasks.

GL-equivariant linear module Given input pairs (A, B) we consider the GL-equivariant linear transformation [25]:

$$F_{\text{linear}}(A, B) = (\Phi \cdot A, \Psi \cdot B) = (A', B') \tag{5}$$

where $\Phi \in \mathbb{R}^{m'\times m}$ and $\Psi \in \mathbb{R}^{n'\times n}$ are learnable weight matrices.

GL-equivariant gated nonlinearity We design an equivariant non-linearity using a gating mechanism, where GL-invariant features control the interpolation between the original and transformed representations—a common approach in equivariant architecture designs [31, 2, 25]. Our gating mechanism applies row-wise interpolation as follows:

$$\tilde{A}_{i,:} = \lambda_i \cdot A'_{i,:} + (1 - \lambda_i) \cdot A_{i,:}\,, \quad i = 1, \ldots, m' \tag{6}$$

where λ_i is a GL-invariant scalar function of (A, B) that serves as a gating weight for row i. The gating weights are derived from the GL-invariant product $P = A'B'^{\top} \in \mathbb{R}^{m' \times n'}$. We first normalize P using root mean square normalization [32] for stability:

$$\hat{P}_{ij} = \frac{P_{ij}}{\text{RMS}(P)} \cdot \gamma_{ij} \tag{7}$$

where $\text{RMS}(P) = \sqrt{\frac{1}{m'n'} \sum_{ij} P_{ij}^2}$ and γ_{ij} are learnable scales. We employ separate MLPs for computing gating weights for A and B. For A, we process each row of $\hat{P}$:

$$\lambda_i = 0.1 + 0.8 \cdot \phi_A(\hat{P}_{i,:}) \tag{8}$$

where $\phi_A : \mathbb{R}^{n'} \to [0, 1]$ is a two-layer MLP with sigmoid activation; we rescale its output to $[0.1, 0.9]$ to avoid saturation and preserve gradient flow. Similarly, for B, we process each column of $\hat{P}$:

$$\mu_j = 0.1 + 0.8 \cdot \phi_B(\hat{P}_{:,j}) \tag{9}$$

$$\tilde{B}_{j,:} = \mu_j \cdot B'_{j,:} + (1 - \mu_j) \cdot B_{j,:} \tag{10}$$

where $\phi_B : \mathbb{R}^{m'} \to [0, 1]$ is a separate MLP with the same architecture as ϕ_A.

Output heads We employ task-specific output heads that either respect the GL-equivariance or project to invariant representations. For tasks requiring equivariant outputs (e.g., weight prediction), we apply the GL-equivariant linear module from Equation 5. For tasks requiring invariant outputs (e.g., dataset property prediction), we extract invariant features and process them through an MLP:

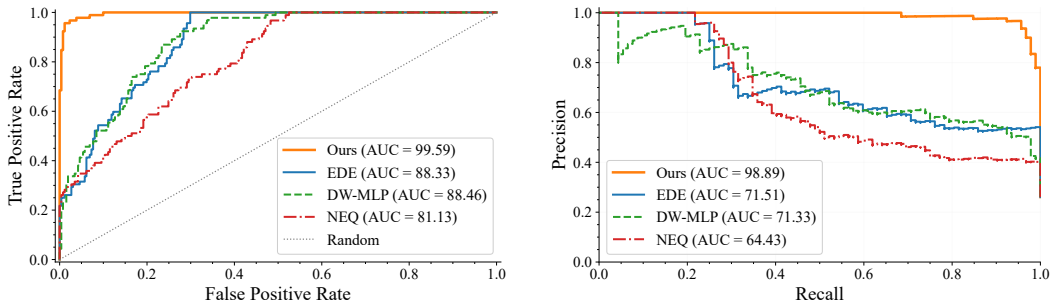
$$F_{\text{inv}}(A, B) = \phi_{\text{out}}(\text{vec}(AB^{\top})) \tag{11}$$

where $\phi_{\text{out}} : \mathbb{R}^{mn} \to \mathbb{R}^d$ denotes a two-layer MLP, $\text{vec}(\cdot)$ vectorization, and d the output dimension.

Model scaling Our experiments show that standard scaling approaches—stacking additional GL-equivariant layers or increasing hidden dimensions—yield marginal performance improvements. To achieve more effective scaling, we adopt a Mixture-of-Experts (MoE) mechanism. Each layer contains multiple experts, where each expert is instantiated as a GL-equivariant layer. We partition experts into two categories: shared experts that process all inputs, and routed experts that are selectively activated. The routing mechanism employs a GL-invariant gating network similar to the invariant output head (Eq. 11), which assigns each input to a subset of routed experts. This ensures routing decisions respect the underlying symmetries while enabling specialized processing paths. We maintain a fixed number of activated experts and employ load balancing to ensure diverse expert utilization.

4 Learning properties of finetuning data sets from LoRA weights

Experimental setup We investigate whether fine-tuning weights encode recoverable information about their training datasets. We utilize the Lots-of-LoRAs repository [5] containing 1,173 *Mistral-7B-Instruct-v0.2* models [17] fine-tuned on diverse natural instruction tasks. Each input sample consists of LoRA updates across 32 transformer layers, with three rank-16 matrix pairs (A, B) per layer corresponding to query, key, and value projections. The query projections have $A, B \in \mathbb{R}^{4096 \times 16}$, while key and value projections have $A \in \mathbb{R}^{4096 \times 16}$ and $B \in \mathbb{R}^{1024 \times 16}$. Our task is binary classification: predicting whether the word "translation" appears in a dataset's description given only its LoRA weight updates (306 positive samples out of 1,173 total). This is implemented by a GL-invariant function $f : (A, B) \mapsto \{0, 1\}$ depicted in Figure 1a. This tests whether weight geometry captures semantic properties of training data. We employ $60/10/30$ stratified train-validation-test splits and report average ROC-AUC and PR-AUC scores with standard deviations across 10 random splits.



(a) Receiver Operating Characteristic (ROC) curves

(b) Precision-Recall (PR) curves

Figure 2: ROC and PR curves for predicting dataset property from LoRA weights across all methods.

Table 1: Performance comparison on predicting dataset characteristics from LoRA weights. Model size reports trainable parameters in millions. Results show the mean and standard deviation across 10 random splits. Best results are highlighted in bold.

Method	**Size**	**ROC-AUC**	**PR-AUC**
Euclidean Distance Embedding (EDE)	-	88.86$\pm$1.77	72.91$\pm$4.19
Direct Weight MLP (DW-MLP)	202M	86.87$\pm$2.28	71.05$\pm$4.52
Non-Equivariant Network (NEQ)	0.08M	81.61$\pm$1.93	66.53$\pm$3.29
Ours	0.16M	**99.11**$\pm$0.36	**97.60**$\pm$1.06

Model and Baselines Our model consists of two GL-equivariant layers followed by a GL-invariant output head, without MoE scaling for this experiment. We compare against three baselines that progressively relax symmetry constraints:

- **Euclidean Distance Embedding (EDE)**: We compute the pairwise Frobenius distances between weight updates $\Delta W = AB^T$, apply classical multidimensional scaling to obtain low-dimensional embeddings, then we use a two-layer MLP classifier. This approach attempts to capture the weight relationships through explicit Euclidean distance rather than learned representations.
- **MLP on the weights (DW-MLP)**: We use an MLP classifier on the weight updates $\text{vec}(\Delta W) = \text{vec}(AB^T)$. This method treats the weights as standard Euclidean vectors in $\mathbb{R}^{mn}$. The symmetry is not needed in this space, but the dimension of the space of weights is significantly larger than the space of its low rank factors (A, B).
- **Non-Equivariant Network (NEQ)**: We consider the LoRA factorization structure (A, B) but replace our GL-equivariant gating mechanism with standard element-wise ReLU activation, ignoring the weight-space symmetries.

Results We present the results in Table 1 and Figure 2. The results show that the fine-tuning weights encode recoverable dataset characteristics, with our GL-equivariant model achieving near-perfect classification performance. The results reveal a clear performance hierarchy corresponding to how each method handles the weight space structure. Methods operating on expanded weights $\Delta W = AB^T$ (EDE and DW-MLP) achieve reasonable but suboptimal performance. Although these Euclidean representations remain invariant to GL transformations, the dimensionality of the input is significantly higher dimension than the LoRA factors (A, B), which could make the learning problem harder since it doesn't really exploit the low-rank structure of ΔW. To avoid this issue [25] uses an SVD on ΔW for different tasks. NEQ, which maintains the LoRA factorization but employs standard element-wise nonlinearities, shows the largest performance degradation. This indicates that preserving factorization alone is insufficient -—the architecture must also respect the GL-equivariance inherent in the weight space to effectively extract encoded information. These results suggest that the weight geometry contains rich, recoverable information about fine-tuning data when processed with appropriate architectural inductive biases. The substantial performance gap between GL-equivariant and non-equivariant processing underscores the importance of incorporating proper symmetries when analyzing fine-tuning weights.

Table 2: Test ROUGE-L on three Lots-of-LoRAs tasks with *Mistral-7B-v0.1*. Higher is better.

Method	Task-342	Task-889	Task-1503
Base	0.05	0.03	0.08
Oracle	0.99	0.82	0.78
Predicted-16	0.62	0.14	0.21
Predicted-32	0.68	0.15	0.28

5 Predicting LoRA weights from partially finetuned models

Experimental setup We investigate whether partial fine-tuning encodes sufficient information to predict complete adaptation patterns. Specifically, we test if we can learn from a model fine-tuned on early layers and generalize to predict appropriate updates for later layers, potentially learning the underlying adaptation structure. We fine-tune *Mistral-7B-v0.1* model [17] on three small-scale datasets from Lots-of-LoRAs, updating only the query and key projection matrices using LoRA with rank 16. Our GL-equivariant model learns the mapping from model weights to LoRA updates defined in Equation 3. We train on weight-update pairs from the first 16 transformer layers using a reconstruction objective, then evaluate the model's ability to predict updates for all 32 layers—both seen and unseen layers during training.

Evaluation protocol We compare four model configurations:

- **Base**: *Mistral-7B-v0.1* without fine-tuning
- **Oracle**: Base model with ground-truth LoRA updates on all 32 layers.
- **Predicted-16**: Base model with predicted LoRA updates for the first 16 layers seen during training.
- **Predicted-32**: Base model with predicted LoRA updates for all 32 layers.

For each dataset, we evaluate all configurations on the task's test split and report ROUGE-L [22] as the evaluation metric. The comparison between Base and Predicted-16 evaluates reconstruction quality on training layers. The gap between Predicted-16 and Predicted-32 reveals generalization to unseen layers, while Oracle provides an upper bound on achievable performance.

Results Table 2 shows ROUGE-L scores across different configurations. The Predicted-16 configuration demonstrates that our model can reconstruct LoRA weights observed during training, though not perfectly. Despite incomplete reconstruction, these predicted weights yield meaningful performance gains over Base, confirming that our model captures structural information from the training layers. The improvement from Predicted-16 to Predicted-32 across all tasks indicates generalization to unseen layers. Our model extracts adaptation patterns from early layers and applies them to later layers, suggesting that fine-tuning weights exhibit learnable cross-layer structure. However, the substantial gap between Predicted-32 and Oracle reveals that current predictions capture only partial adaptation patterns. Task-specific variation is notable: Task-342 shows strong generalization while Task-889 exhibits minimal improvement from Predicted-16 to Predicted-32. This variability suggests that the feasibility of weight prediction depends on task characteristics—some fine-tuning patterns may be more structured and predictable than others. These preliminary results indicate that while weight prediction from partially fine-tuned models is possible, achieving oracle-level performance requires addressing task-dependent factors and improving cross-layer generalization mechanisms.

6 Discussion

Our preliminary experiments demonstrate that both of the questions we asked in Section 1 are answered in the affirmative: (i) there is information about the datasets used for adaptation (e.g., if a model is a translation model or not) that is encoded and recoverable from the weight matrices; and (ii) it is possible to use partially fine-tuned model weights to improve upon predicting the remaining weight updates. Although we can answer both questions, there are several additional experiments and analyses to consider.

For example, the performance-first classification task that we considered – predicting whether or not the weights corresponded to a translation task – appears to be saturated. There are interesting

extensions of this task that would enable more detailed analysis of the strengths of our proposed method. Namely, each translation task is trained on data from a (source, target) pair. For example, Task-1023 in Lots-of-LoRAs is translation from English to Hindi. It may be possible to use the information encoded in the LoRA weights to infer a combination of the source and target languages. The answer to this question will help us understand the level of classification precision that is possible with our proposed method.

Another potential additional investigation is the ability to include unsupervised signal from LoRA matrices that do not necessarily have a relevant supervised signal. For the two prediction tasks we considered herein, all models have the required supervised signal and we can, in theory, use models from all tasks to train our predictors. Some tasks definable on datasets like Lots-of-LoRAs, however, do not equip each model in the collection with a relevant label. For example, consider again the task of classifying a translation model based on a combination of the the source and target language. For this task only translation models have a relevant label associated with them. As currently constructed, our method cannot naturally take advantage of models trained on other tasks. We think that adapting our method to be able to take advantage of all models for all tasks in a semi-supervised fashion will improve our methods utility and efficiency.

Along these lines, additional experiments are required to understand our method's sensitivities to relevant problem hyperparameters – such as the number of models available, the rank of the LoRA matrices, the depth of our neural model, etc. In particular, in preliminary experiments the order of the performance of the methods is sensitive to the number of models in the training and validation sets – with simple, non-neural methods outperforming neural methods when the number of models available is small (< 100). Understanding the phase transition of the order of the methods' performances is critical, both in the fully supervised setting and in the semi-supervised setting described above. It is likely that the phase transition is dependent on the rank of the LoRA matrices.

The weight prediction experiment reveals both opportunities and challenges in learning fine-tuning patterns from partially adapted models. The performance gap between predicted and oracle configurations likely stems from insufficient training signal (only 16 layers) and task-dependent factors—Task-342 shows strong cross-layer generalization while Task-889 exhibits minimal improvement, suggesting that adaptation predictability varies with task characteristics. The current experimental design is limited to decoder-only transformers fine-tuned on small-scale Lots-of-LoRAs tasks; scaling to larger corpora presents additional challenges as extended training would increase the complexity of learned adaptation patterns, while investigating encoder-only and encoder-decoder architectures would determine whether such cross-layer adaptation structures emerge across different architectural paradigms. Extending to vision transformers would test whether visual and linguistic modalities exhibit similar adaptation patterns. Despite current limitations, the ability to extract meaningful signals from partial fine-tuning and generalize to unseen layers suggests that fine-tuning weights encode learnable structural patterns, opening avenues for more efficient adaptation strategies.

References

[1] Aranyak Acharyya, Michael W Trosset, Carey E Priebe, and Hayden S Helm. Consistent estimation of generative model representations in the data kernel perspective space. *arXiv preprint arXiv:2409.17308*, 2024.

[2] Ben Blum-Smith and Soledad Villar. Machine learning and invariant theory. *Notices of the American Mathematical Society*, 70(08):1–1, 2023.

[3] Amil Dravid, Yossi Gandelsman, Kuan-Chieh Wang, Rameen Abdal, Gordon Wetzstein, Alexei Efros, and Kfir Aberman. Interpreting the weight space of customized diffusion models. *Advances in Neural Information Processing Systems*, 37:137334–137371, 2024.

[4] Brandon Duderstadt, Hayden S. Helm, and Carey E. Priebe. Comparing foundation models using data kernels, 2024. URL `https://arxiv.org/abs/2305.05126`.

[5] Rickard Brüel Gabrielsson, Jiacheng Zhu, Onkar Bhardwaj, Leshem Choshen, Kristjan Greenewald, Mikhail Yurochkin, and Justin Solomon. Compress then serve: Serving thousands of lora adapters with little overhead. In *Forty-second International Conference on Machine Learning*, 2024.

[6] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. 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 4884–4896, 2021.

[7] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[8] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+ efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 17783–17806, 2024.

[9] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Plop: Precise lora placement for efficient finetuning of large models. *arXiv preprint arXiv:2506.20629*, 2025.

[10] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022.

[11] Hayden Helm, Brandon Duderstadt, Youngser Park, and Carey Priebe. Tracking the perspectives of interacting language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1508–1519, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.90. URL https://aclanthology.org/2024.emnlp-main.90/.

[12] Hayden Helm, Aranyak Acharyya, Youngser Park, Brandon Duderstadt, and Carey Priebe. Statistical inference on black-box generative models in the data kernel perspective space. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3955–3970, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.204. URL https://aclanthology.org/2025.findings-acl.204/.

[13] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[15] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis, 2024. URL https://arxiv.org/abs/2405.07987.

[16] Wenlong Ji, Weizhe Yuan, Emily Getzen, Kyunghyun Cho, Michael I Jordan, Song Mei, Jason E Weston, Weijie J Su, Jing Xu, and Linjun Zhang. An overview of large language models for statisticians. *arXiv preprint arXiv:2502.17814*, 2025.

[17] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[18] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

[19] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[20] Xiang Lisa Li and Percy 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.

[21] Baohao Liao, Yan Meng, and Christof Monz. Parameter-efficient fine-tuning without introducing new latency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4260, 2023.

[22] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[23] Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 441–459, 2020.

[24] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 5:208–215, 2024.

[25] Theo Putterman, Derek Lim, Yoav Gelberg, Stefanie Jegelka, and Haggai Maron. Learning on loras: Gl-equivariant processing of low-rank weight spaces for large finetuned models. *arXiv preprint arXiv:2410.04207*, 2024.

[26] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017.

[27] Mohammad Salama, Jonathan Kahana, Eliahu Horwitz, and Yedid Hoshen. Dataset size recovery from lora weights. *CoRR*, 2024.

[28] Behrooz Tahmasebi and Stefanie Jegelka. The exact sample complexity gain from invariances for kernel regression. *Advances in Neural Information Processing Systems*, 36:55616–55646, 2023.

[29] Hoang V. Tran, Thieu Vo, An Nguyen The, Tho Tran Huu, Minh-Khoi Nguyen-Nhat, Thanh Tran, Duy-Tung Pham, and Tan Minh Nguyen. Equivariant neural functional networks for transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=uBai0ukstY`.

[30] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287, 2023.

[31] Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. *Advances in neural information processing systems*, 34:28848–28863, 2021.

[32] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

[33] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Section 3, Section 4, and Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theorem is proposed in this work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 4 and Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: This is ongoing research submitted to a workshop, with code and methods being frequently updated as we refine our approach. We are not releasing code at this stage to avoid confusion from rapidly changing implementations. Implementation details sufficient for understanding our approach are provided in the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 4 and Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Table 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Detailed compute resources were not tracked for these preliminary experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research complies with the NeurIPS Code of Ethics as it involves only computational experiments on publicly available models with no human subjects or privacy concerns.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This is preliminary research on the geometric properties of fine-tuning weights without direct societal applications or deployment considerations.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets are properly cited with references to their original papers and sources.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM is used only for writing, editing, or formatting purposes.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.