Collaborative Geometry-Aware Multi-Solution Optimizer for Efficient Model Fine-Tuning

Van-Anh Nguyen
Department of Data Science and AI
Monash University, Australia
van-anh.nguyen@monash.edu

Trung Le
Department of Data Science and AI
Monash University, Australia
trunglm@monash.edu

Mehrtash Harandi
Department of Electrical and Computer Systems Engineering
Monash University, Australia
mehrtash.harandi@monash.edu

Ehsan Abbasnejad
Department of Data Science and AI
Monash University, Australia
ehsan.abbasnejad@monash.edu

Thanh-Toan Do
Department of Data Science and AI
Monash University, Australia
toan.do@monash.edu

Dinh Phung
Department of Data Science and AI
Monash University, Australia
dinh.phung@monash.edu

Abstract

We propose a framework grounded in gradient flow theory and informed by geometric structure that provides multiple diverse solutions for a given task, ensuring collaborative results that enhance performance and adaptability across different tasks. This framework enables flexibility, allowing for efficient task-specific fine-tuning while preserving the knowledge of the pre-trained foundation models. Extensive experiments across transfer learning, few-shot learning, and domain generalization show that our proposed approach consistently outperforms existing Bayesian methods, delivering strong performance with affordable computational overhead and offering a practical solution by updating only a small subset of parameters. The code for our method is at https://github.com/anh-ntv/GAC-MSO

1 Introduction

The rapid growth of foundation models, particularly Transformers [41] based architecture, has fundamentally transformed the field of artificial intelligence. Pre-trained on massive datasets, these models have demonstrated extraordinary capabilities to learn rich, contextualized representations has led to state-of-the-art performance across a wide range of applications from natural language processing [14, 6, 43, 50] to computer vision [15, 39, 29, 7, 56].

While these models are powerful, adapting them to specific downstream tasks remains challenging due to their enormous size and computational demands. Moreover, in many real-world scenarios, generating multiple diverse solutions for a task can improve robustness, adaptability, and ensemble performance. However, directly optimizing multiple instances of such large models is computationally

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

infeasible. A practical and efficient way to overcome this problem is to optimize a set of compact auxiliary modules that integrate into the foundation model, while keeping most of the original pretrained parameters unchanged and shared across all solutions.

Multiple compact auxiliary modules have been proposed for parameter-efficient fine-tuning (PEFT) frameworks to effectively adapt foundation models to new tasks by tuning only a small fraction of parameters. Initially, full fine-tuning of these models, which often contain millions or billions of parameters, is computationally expensive, memory-intensive, and impractical when multiple tasks are involved. Additionally, limited task-specific data might require careful regularization to prevent overfitting. To address these challenges, pioneering PEFT techniques such as Adapter [20], prompt tuning [22], LoRA [21], SCT [55], and BitFit [52] have demonstrated promising results by maintaining most pre-trained parameters fixed, thus enhancing both computational efficiency and performance.

Building on the efficiency of PEFT, we propose a framework that generates multiple diverse solutions by optimizing lightweight modules while reusing a shared backbone. This design enables collaborative, robust predictions with minimal overhead, preserving the generalization capability of the pre-trained model. We initiate this process by guiding the posterior toward the target distribution using gradient-based updates. A common approach is Stein Variational Gradient Descent (SVGD) [27], which balances high predictive performance with solution diversity through repulsive interactions in parameter space. However, SVGD ignores the geometric relationships among solutions—specifically, how they align in output space or interact within the loss landscape.

To overcome these limitations, we propose the Geometry-Aware Collaborative Multi-Solution Optimizer (GAC-MSO), a theoretically grounded and tractable framework for generating diverse, high-collaborative solutions. Unlike SVGD, which promotes diversity solely in parameter space via kernel-based repulsion, GAC-MSO also integrates geometric structure [3] and enforces output-space diversity through a divergence term. This leads to efficient solution space exploration while maintaining strong predictive accuracy and calibration, even with limited computational resources.

To summarize, our key contributions in this paper include:

- We propose the Geometry-Aware Collaborative Multi-Solution Optimizer (GAC-MSO), a framework grounded in gradient flow theory [3] over the probability space of models. This formulation enables the incorporation of geometric structure [1, 2] and promotes the generation of diverse yet collaborative solutions.
- We conduct extensive experiments on PEFT across various settings, including model fine-tuning for transfer learning, few-shot learning, and domain generalization. The results demonstrate that our GAC-MSO consistently outperforms baseline methods by a significant margin, highlighting the effectiveness of incorporating geometric structure and promoting diverse yet collaborative solutions.

2 Related works

Parameter Efficiency Parameter Tuning. Parameter-efficient fine-tuning (PEFT) has gained attention for adapting large pre-trained models to downstream tasks by minimizing computational costs. Several methods have been developed to achieve this.

– Adapter tuning. These methods adapt Transformer-based models by inserting lightweight neural modules into each layer and fine-tuning only these modules, while keeping the core model parameters frozen. These adapters typically adopt a bottleneck architecture comprising two small fully connected (FC) layers [20] and an activation function [9]. In the context of vision tasks, certain methods [51] extend the adapter design by incorporating convolutional layers or Normalizing Flows [46] to better capture spatial patterns and complex feature distributions.

– Prompt Tuning. These methods adapt models to new tasks by introducing additional learnable visual prompts, which are either inserted into the backbone or applied as perturbations to existing weights. VPT-Shallow [22] inserts prompts only before the first encoder layer, while VPT-Deep [22] places prompts at each encoder layer for deeper integration.

– LoRA Tuning. These methods introduce additional parameters during training or fine-tune specific subsets of the model, while ensuring that these modifications are efficiently integrated into the

backbone architecture to minimize inference overhead. A pioneering approach in this category is LoRA [21], which inserts low-rank decomposition matrices into attention layers and merges them with the original weights at inference time, maintaining both computational efficiency and strong performance. Building on LoRA, several variants have been proposed to further enhance its flexibility and effectiveness. AdaLoRA [54] dynamically adjusts the rank of LoRA modules during training based on their importance, improving parameter efficiency. LoRA-Drop [57] introduces dropout within LoRA modules to regularize training and prevent overfitting, particularly in low-resource scenarios.

Variational Gradient Descent Approach. This strategy enables sampling multiple models from the posterior distribution, a central technique in neural network inference often realized through Hamiltonian Monte Carlo (HMC) [31]. Although HMC is effective, it is computationally intensive due to its reliance on full gradient evaluations. To improve scalability, Stochastic Gradient HMC (SGHMC) [10] uses noisy gradient estimates, facilitating efficient exploration of the solution space. Similarly, Stochastic Gradient Langevin Dynamics (SGLD) [47] incorporates Langevin dynamics into a stochastic gradient framework. In contrast, Stein Variational Gradient Descent (SVGD) [27] approximates the posterior using a set of interacting particles. Building on this, [44] enhances SVGD with nonlinear transformations that encourage greater particle diversity, addressing SVGD's tendency to collapse in multimodal settings and improving its ability to learn complex mixture models. Complementarily, [11] proposes a repulsive mechanism for deep ensembles that fosters diverse yet plausible members, resulting in a more faithful approximation of the Bayesian posterior.

3 Background

3.1 Gradient Flow in Probability Space

Problem Setting. We start with the problem setting used throughout this paper. Consider the *target distribution* $p(\boldsymbol{\theta}) \propto \exp\{-\beta\Psi(\boldsymbol{\theta})\}$ over $\mathbb{R}^d$, where $\Psi(\cdot)$ is the energy function, we need to find efficient ways of sampling from this target distribution. It should be noted that this setting can be directly applied to Bayesian inference where the energy function is the empirical loss $\mathcal{L}_S(\boldsymbol{\theta})$ over a training set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, which is defined as

$$\mathcal{L}_S(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} l\left(f\left(\boldsymbol{x_i};\boldsymbol{\theta}\right), y_i\right),$$

where $f(\boldsymbol{x_i};\boldsymbol{\theta})$ is the prediction output of the model with the model parameter $\boldsymbol{\theta}$ and l is a loss function.

It is evident that $p(\cdot)$ is the solution of the following optimization problem:

$$\min_{\rho\in\mathcal{P}(\mathbb{R}^d)}\left\{\mathcal{F}(\rho) := \beta\int\Psi d\rho + \int\log\rho d\rho\right\}, \tag{1}$$

where $\mathcal{P}(\mathbb{R}^d)$ is the space of distributions over $\mathbb{R}^d$ with the Wasserstein distance [3].

The gradient flow of $\mathcal{F}$ in the Wasserstein space [3] is described by:

$$\partial_s\rho_s + \operatorname{div}\left(\rho_s\nabla\frac{\partial\mathcal{F}}{\partial\rho_s}\right) = 0,$$

where $\frac{\partial\mathcal{F}}{\partial\rho_s}$ is the first variation (functional derivative) of $\mathcal{F}$ and div is the divergence operator.

3.2 Stein Variational Gradient Descent

Given the current distribution ρ_t as the time step t, our aim is to find the velocity field $v_t = id + \eta u_t$ using the steepest descent direction:

$$u_t = \operatorname{argmin}_u \frac{d}{d\eta}\mathcal{F}\left((id+\eta u)\#\rho_t\right)|_{\eta=0}, \tag{2}$$

where $\#$ is the transport operator, id is the identity function, and $\eta > 0$ is the step size.

The next distribution solution $\rho_{t+1} = v_t \# \rho_t$ where $v_t = id + \eta u_t$. Moreover, by restricting the velocity $u \in \mathcal{H}_K^d$, where $\mathcal{H}_K$ is the Reproducing Kernel Hilbert Space (RKHS) corresponding to the positive semi-definite kernel $K(\boldsymbol{\theta}, \boldsymbol{\theta}') : \Theta \times \Theta \to \mathbb{R}$, Stein Variational Gradient Descent (SVGD) [27] reaches the closed-form solution for the optimal velocity v_t as

$$v_t\left(\tilde{\boldsymbol{\theta}}\right) = \tilde{\boldsymbol{\theta}} + \eta \mathbb{E}_{\boldsymbol{\theta} \sim \rho_t}\left[-K\left(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}\right) \nabla_{\boldsymbol{\theta}} \Psi\left(\boldsymbol{\theta}\right) + \nabla_{\boldsymbol{\theta}} K\left(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}\right)\right].$$

4 Collaborative Multi-Solution Optimizers

Given the current solution ρ_t, to find the next solution ρ_{t+1}, we use the proximal operator as follows:

$$\min_{\rho \in \mathcal{P}(\mathbb{R}^d)} \{\mathcal{F}\left(\rho\right) + d\left(\rho, \rho_t\right)\}, \tag{3}$$

where $d\left(\rho, \rho_t\right)$ is a proximal operator, which is defined below.

4.1 Proximal Operator

We define the divergence $d\left(\rho, \rho_t\right)$ as

$$d\left(\rho, \rho_t\right) = \mathbb{E}_{\boldsymbol{\theta}' \sim \rho, \boldsymbol{\theta} \sim \rho_t}\left[\mathbb{E}_{\boldsymbol{x}}\left[KL\left(f\left(\boldsymbol{x}; \boldsymbol{\theta}'\right), f\left(\boldsymbol{x}; \boldsymbol{\theta}\right)\right) + KL\left(f\left(\boldsymbol{x}; \boldsymbol{\theta}\right), f\left(\boldsymbol{x}; \boldsymbol{\theta}'\right)\right)\right]\right], \tag{4}$$

where KL is the Kullback-Leibler (KL) divergence.

In the following lemma, we approximate $d(\rho, \rho_t)$, exposing the geometry around the current solution ρ_t. All proof in our theory development can be found in Appendix B in the supplementary material.

Lemma 4.1. *The divergence $d(\rho, \rho_t)$ in Eq. (4) can be approximated as*

$$d\left(\rho, \rho_t\right) \approx \mathbb{E}_{\boldsymbol{\theta}' \sim \rho, \boldsymbol{\theta} \sim \rho_t}\left[\left(\boldsymbol{\theta}' - \boldsymbol{\theta}\right)^\top H\left(\boldsymbol{\theta}\right)\left(\boldsymbol{\theta}' - \boldsymbol{\theta}\right)\right],$$

where $H\left(\boldsymbol{\theta}\right) = \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_y\left[\nabla_{\boldsymbol{\theta}} \log f_y\left(\boldsymbol{x}; \boldsymbol{\theta}\right) \nabla_{\boldsymbol{\theta}} \log f_y\left(\boldsymbol{x}; \boldsymbol{\theta}\right)^\top\right]\right]$ and $f_y\left(\boldsymbol{x}; \boldsymbol{\theta}\right)$ is the y-th prediction output in the prediction probability vector $f\left(\boldsymbol{x}; \boldsymbol{\theta}\right)$.

4.2 Theory Development

We denote the gradient flow of the optimization problem (OP) in (3) as $\left(\rho_s\right)_{s \geq t}$ that satisfies the continuity equation [38] (Chapter 4, Page 110):

$$\partial_s \rho_s + \operatorname{div}\left(\rho_s v_s\right) = 0, \forall s \geq t,$$

where div is the divergence operator and v_s is the velocity field.

Let $f \in T_{\rho_t}\mathcal{M}$ with $\mathcal{M} = \mathcal{P}\left(\mathbb{R}^d\right)$ be the perturbation function (*i.e.*, describing how ρ_t changes over time) on the tangent space $T_{\rho_t}\mathcal{M}$ such that

$$f + \operatorname{div}\left(\rho_t v_t\right) = 0, \tag{5}$$

which further implies $f = \partial_t \rho_t$.

Thus, for a small step size $\eta > 0$, it follows $f \approx \frac{\rho_{t+\eta} - \rho_t}{\eta}$, which leads to $\rho_{t+\eta} \approx \rho_t + \eta f$. We further derive

$$\mathcal{F}\left(\rho_{t+\eta}\right) - \mathcal{F}\left(\rho_t\right) \approx \mathcal{F}\left(\rho_t + \eta f\right) - \mathcal{F}\left(\rho_t\right) \approx \left\langle \eta f, \frac{\partial \mathcal{F}\left(\rho_t\right)}{\partial \rho_t} \right\rangle \tag{6}$$

$$= \eta \int \frac{\partial \mathcal{F}\left(\rho_t\right)}{\partial \rho_t}\left(\boldsymbol{\theta}\right) f\left(\boldsymbol{\theta}\right) d\boldsymbol{\theta} \overset{(1)}{=} -\eta \int \frac{\partial \mathcal{F}\left(\rho_t\right)}{\partial \rho_t}\left(\boldsymbol{\theta}\right) \operatorname{div}\left(\rho_t\left(\boldsymbol{\theta}\right) v_t\left(\boldsymbol{\theta}\right)\right) d\boldsymbol{\theta} \tag{7}$$

$$\overset{(2)}{=} \eta \int \left\langle \nabla_{\boldsymbol{\theta}} \frac{\partial \mathcal{F}\left(\rho_t\right)}{\partial \rho_t}\left(\boldsymbol{\theta}\right), v_t\left(\boldsymbol{\theta}\right) \right\rangle \rho_t\left(\boldsymbol{\theta}\right) d\boldsymbol{\theta}, \tag{8}$$

where $\overset{(1)}{=}$ is due to Eq. (5) and $\overset{(2)}{=}$ is due to the integral by part.

Noting that $v_t \# \rho_t \approx \rho_{t+\eta}$ by the definition of the velocity field, and relating this to the proximal operator in (3), we arrive at

$$\min_{v_t} \left\{ \mathcal{F}(v_t \# \rho_t) - \mathcal{F}\left(\rho_t\right) + d\left(\rho, \rho_t\right) \right\},$$

which can be reformulated into due to (8) and Lemma 4.1

$$\min_{v_t} \left\{ \eta \int \left\langle \nabla \frac{\partial \mathcal{F}\left(\rho_t\right)}{\partial \rho_t}\left(\boldsymbol{\theta}\right), v_t\left(\boldsymbol{\theta}\right) \right\rangle \rho_t\left(\boldsymbol{\theta}\right) d\boldsymbol{\theta} + \mathbb{E}_{\boldsymbol{\theta} \sim \rho_t}\left[\Delta \boldsymbol{\theta}^\top H\left(\boldsymbol{\theta}\right) \Delta \boldsymbol{\theta} \right] \right\}. \tag{9}$$

where $\Delta \boldsymbol{\theta} = v_t(\boldsymbol{\theta}) - \boldsymbol{\theta}$.

Moreover, Theorem 4.2 characterizes the optimal solution of OP in (9), which involves the geometry of the particles sampled from ρ_t.

Theorem 4.2. *The OP in (9) receives the following optimal solution*

$$v_t^*\left(\tilde{\boldsymbol{\theta}}\right) = \tilde{\boldsymbol{\theta}} - \eta H\left(\tilde{\boldsymbol{\theta}}\right)^{-1} \nabla \frac{\partial \mathcal{F}\left(\rho_t\right)}{\partial \rho_t}\left(\tilde{\boldsymbol{\theta}}\right), \tag{10}$$

where $H\left(\tilde{\boldsymbol{\theta}}\right) = \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_y\left[\nabla_{\boldsymbol{\theta}} \log f_y\left(\boldsymbol{x}; \tilde{\boldsymbol{\theta}}\right) \nabla_{\boldsymbol{\theta}} \log f_y\left(\boldsymbol{x}; \tilde{\boldsymbol{\theta}}\right)^\top \right] \right]$.

It should be noted that although the update formula in Theorem 4.2 has a closed form, it is intractable because $\nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}\left(\tilde{\boldsymbol{\theta}}\right) = \beta \nabla \Psi\left(\tilde{\boldsymbol{\theta}}\right) + \nabla \log \rho_t\left(\tilde{\boldsymbol{\theta}}\right)$ is *intractable* due to the term $\nabla \log \rho_t\left(\tilde{\boldsymbol{\theta}}\right)$. In what follows, we present how to estimate this term to obtain a tractable solution.

Tractable Solution. To develop a tractable solution, we first notice that $\tilde{v}_t^*\left(\tilde{\boldsymbol{\theta}}\right) = \tilde{\boldsymbol{\theta}} - \eta \nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}\left(\tilde{\boldsymbol{\theta}}\right) = \tilde{\boldsymbol{\theta}} + \eta \tilde{u}_t^*(\tilde{\boldsymbol{\theta}})$ is the velocity so that $\rho_{t+\eta} = \tilde{v}_t^* \# \rho_t$ minimizes $\mathcal{F}(\rho) - \mathcal{F}(\rho_t)$ in a vicinity of ρ_t. To find the *optimal increment* $\tilde{u}_t(\tilde{\boldsymbol{\theta}}) = -\nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}\left(\tilde{\boldsymbol{\theta}}\right)$, we seek the steepest descent direction as in Eq. (2). To this end, we strengthen $\mathcal{F}(\rho)$ by adding the divergence term and then replacing $\int \log \rho d\rho$ by a similar term using the convolution operation inspired by [8].

Inspired by [8], to make smooth the entropy function, we redefine $\mathcal{F}(\rho)$ as

$$\mathcal{F}\left(\rho\right) = \beta \int \Psi d\rho + \int \log\left(K * \rho\right) d\rho, \tag{11}$$

where $K * \rho\left(\boldsymbol{\theta}\right) = \int K\left(\boldsymbol{\theta}, \boldsymbol{\theta'}\right) \rho\left(\boldsymbol{\theta'}\right) d\boldsymbol{\theta'}$ is the convolution operation using the kernel K, which aims to make the entropy function smooth.

Moreover, given the current solution ρ_t and the velocity $\tilde{v}_t = id + \eta \tilde{u}_t$, we define the *divergence term* as

$$\begin{aligned} \mathcal{L}_{div}\left(\tilde{u}_t, \eta\right) &= \int l_{\text{div}}\left(\boldsymbol{\theta}_{1:M}^{[\tilde{v}_t]}\right) \prod_{m=1}^{M} d\rho^{[\tilde{v}_t]}\left(\boldsymbol{\theta}_m^{[\tilde{v}_t]}\right) \\ &= \int l_{\text{div}}\left(\left[\boldsymbol{\theta}_m + \eta \tilde{u}_t\left(\boldsymbol{\theta}_m\right)\right]_{m=1}^{M}\right) \prod_{m=1}^{M} \rho_t\left(\boldsymbol{\theta_m}\right) d\boldsymbol{\theta}_{1:M}, \end{aligned} \tag{12}$$

where $l_{\text{div}}(\boldsymbol{\theta}_{1:M})$ is the loss that encourages the particles $\boldsymbol{\theta}_{1:M}$ more diverge and $\rho^{[\tilde{v}_t]} = \tilde{v}_t \# \rho_t$.

Conceptually, by minimizing $\mathcal{L}_{div}\left(\tilde{u}_t, \eta\right)$ in (12), we aim to learn a velocity $\tilde{v}_t = id + \eta \tilde{u}_t$ in such a way that $\boldsymbol{\theta}_{1:M}^{[\tilde{v}_t]} \sim \rho^{[\tilde{v}_t]} = \tilde{v}_t \# \rho_t$ are encouraged to diverge. Eventually, given the current solution ρ_t, we learn the velocity $\tilde{v}_t = id + \eta \tilde{u}_t$ to minimize

$$\mathcal{G}\left(\tilde{u}_t, \eta\right) = \mathcal{F}\left(\rho^{[\tilde{v}_t]}\right) + \alpha \mathcal{L}_{div}\left(\tilde{u}_t, \eta\right), \tag{13}$$

where $\alpha > 0$ is a trade-off parameter and $\rho^{[\tilde{v}_t]} = \tilde{v}_t \# \rho_t$.

In particular, we aim to find an optimal velocity $\tilde{v}_t$ that simultaneously minimizes $\mathcal{F}(\rho)$ and pushes the diverge particles. In the following theorem, we characterize the steepest descent direction.

Theorem 4.3. *The steepest descent direction has the following form:* $\nabla_\eta \mathcal{G}\left(\tilde{u}_t, \eta\right) \mid_{\eta=0} = \langle h, \tilde{u}_t \rangle$, *where* $\langle ., . \rangle$ *is the dot product on* $\mathcal{H}_K^d$ *and*

$$h\left(\cdot\right) = \mathbb{E}_{\boldsymbol{\theta}\sim\rho_t}\left[\beta\nabla\Psi\left(\boldsymbol{\theta}\right)K\left(\boldsymbol{\theta}, .\right) - \frac{\mathbb{E}_{\boldsymbol{\theta'}\sim\rho_t}\left[K\left(\boldsymbol{\theta}, \boldsymbol{\theta'}\right)\nabla K\left(\boldsymbol{\theta}, .\right)\right]}{\mathbb{E}_{\boldsymbol{\theta'}\sim\rho_t}\left[K\left(\boldsymbol{\theta}, \boldsymbol{\theta'}\right)\right]}\right] + \alpha\mathbb{E}_{\boldsymbol{\theta}_{1:M}\sim\rho_t}\left[\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta}_m} l_{div}\left(\boldsymbol{\theta}_{1:M}\right)K\left(\boldsymbol{\theta}_m, .\right)\right].$$

Furthermore, using the first-order Taylor expansion, we obtain

$$\mathcal{G}\left(\tilde{u}_t, \eta\right) = \mathcal{G}\left(\tilde{u}_t, 0\right) + \eta\nabla_\eta\mathcal{G}\left(\tilde{u}_t, \eta\right)\mid_{\eta=0} + O(\eta^2).$$

Therefore, by restricting $\tilde{u}_t$ in the ball of radius $\langle h, h\rangle$ inside $\mathcal{H}_K^d$, we yield $\tilde{u}_t^* = -h$, hence $\tilde{v}_t^* = id + \eta\tilde{u}_t^* = id - \eta h$. Finally, referring to (10), we reach

$$\begin{aligned} v_t^*\left(\tilde{\boldsymbol{\theta}}\right) &= \tilde{\boldsymbol{\theta}} - \eta H\left(\tilde{\boldsymbol{\theta}}\right)^{-1}\nabla\frac{\partial\mathcal{F}\left(\rho_t\right)}{\partial\rho_t}\left(\tilde{\boldsymbol{\theta}}\right) = \tilde{\boldsymbol{\theta}} + \eta H\left(\tilde{\boldsymbol{\theta}}\right)^{-1}\tilde{u}_t^*\left(\tilde{\boldsymbol{\theta}}\right) \\ &= \tilde{\boldsymbol{\theta}} - \eta H\left(\tilde{\boldsymbol{\theta}}\right)^{-1} h\left(\tilde{\boldsymbol{\theta}}\right). \end{aligned}$$

Practical Method. In what follows, we present the practical implementation of our method. Similar to SVGD [27], we maintain a set of M particle models, denoted by $\boldsymbol{\theta}_{1:M}$. For implementation convenience, we use the same number of particles as in Eq. (12). To estimate $H(\tilde{\boldsymbol{\theta}})^{-1}$, we approximate it using only its diagonal elements. Furthermore, we compute this estimate using a moving average of $H(\tilde{\boldsymbol{\theta}})$ accumulated from past to current iterations.

$$m_t\left(\tilde{\boldsymbol{\theta}}\right) = \gamma\text{diag}\left(\mathbb{E}_{(\boldsymbol{x},y)\sim\boldsymbol{B}_t}\left[\nabla_{\tilde{\boldsymbol{\theta}}}\log f_y\left(\boldsymbol{x};\tilde{\boldsymbol{\theta}}\right)\nabla_{\tilde{\boldsymbol{\theta}}}\log f_y\left(\boldsymbol{x};\tilde{\boldsymbol{\theta}}\right)^\top\right]\right) + \left(1-\gamma\right)m_{t-1}\left(\tilde{\boldsymbol{\theta}}\right),$$

where $\gamma \in [0; 1]$ is a momentum decay and $\boldsymbol{B}_t$ is the current mini-batch of $(\boldsymbol{x}, y)$ at the current iteration.

Moreover, each particle $\boldsymbol{\theta}^t$ (i.e., $\boldsymbol{\theta}_{1:M}^t$) is updated as follows:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \frac{\eta}{\left[m_t\left(\boldsymbol{\theta}^t\right) + \epsilon\right]M}\sum_{m=1}^{M}\Big[-\beta\nabla\Psi\left(\boldsymbol{\theta}_m^t\right)K\left(\boldsymbol{\theta}_m^t, \boldsymbol{\theta}^t\right) + \frac{\sum_{m'}\left[K\left(\boldsymbol{\theta_m^t}, \boldsymbol{\theta_{m'}^t}\right)\nabla K\left(\boldsymbol{\theta}_m^t, \boldsymbol{\theta}^t\right)\right]}{\sum_{m'}\left[K\left(\boldsymbol{\theta_m^t}, \boldsymbol{\theta_{m'}^t}\right)\right]} - \frac{\alpha}{M}\sum_{m=1}^{M}\nabla_{\boldsymbol{\theta}_m} l_{\text{div}}\left(\boldsymbol{\theta}_{1:M}^t\right)K\left(\boldsymbol{\theta}_m^t, \boldsymbol{\theta}^t\right)\Big], \tag{14}$$

where $\boldsymbol{\theta}^t$ represents a particle model at the t-th iteration. The training algorithm is presented in Supplementary.

Divergence Term. Our framework facilitates the integration of diverse computational terms to promote both collaboration and diversity among particle models. To illustrate its effectiveness, we introduce a specific term designed to encourage the particle models to generate diverse output predictions, ultimately enhancing the performance of the final ensemble.

We now describe the formulation of the divergence loss $l_{\text{div}}(\boldsymbol{\theta}_{1:M}; \boldsymbol{x}, y)$. For a given data point $(\boldsymbol{x}, y) \in S$, let $f(\boldsymbol{x}; \boldsymbol{\theta}_i)$ denote the predicted probability distribution produced by the particle model $\boldsymbol{\theta}_i$. Define $f_{-y}(\boldsymbol{x}; \boldsymbol{\theta}_i)$ (abbreviated as f_{-y}^i when context permits) as the *non-maximal* prediction vector obtained by removing the ground-truth class y from the prediction. Following the approach in [35], we encourage the non-maximal predictions f_{-y}^i (for $i = 1, \ldots, C$, where C is the number of classes) to be mutually dissimilar, while simultaneously promoting the confidence in the ground-truth predictions f_y^i. Drawing inspiration from Determinantal Point Processes (DPP) theory [25], we define the ensemble diversity as:

$$l_{\text{div}}\Big(\boldsymbol{\theta}_{1:M}; \boldsymbol{x}, y\Big) = -\log\left(\det\Big(\Big[\tilde{f}_{-y}^i\Big]_{i\in[C]}^\top\Big[\tilde{f}_{-y}^i\Big]_{i\in[C]}\Big)\right),$$

where $\tilde{f}^i_{-y} = \frac{f^i_{-y}}{\|f^i_{-y}\|}$ and $\Big[\tilde{f}^i_{-y}\Big]_{i\in[C]} \in \mathbb{R}^{(C-1)\times K}$ where $\Big[C\Big] = \Big\{1, \ldots, C\Big\}$.

Moreover, according to the matrix theory [4],

$$\det\Big(\Big[\tilde{f}^i_{-y}\Big]^\top_{i\in[C\}}\Big[\tilde{f}^i_{-y}\Big]_{i\in[C]}\Big) = \mathrm{Vol}^2\Big(\Big[\tilde{f}^i_{-y}\Big]_{i\in[C]}\Big),$$

where $\mathrm{Vol}\Big(\Big[\tilde{f}^i_{-y}\Big]_{i\in[C]}\Big)$ specifies the volume spanned the vectors in $\Big[\tilde{f}^i_{-y}\Big]_{i\in[C]}$, indicating that we aim to maximize the diversity of the non-maximal predictions by maximally increasing their spanned volume.

Model Fine-tuning with Parameter Efficiency. We focus on the fine-tuning problem, where a pre-trained model, denoted as Φ, is provided, and the goal is to identify the optimal parameters $\boldsymbol{\theta} = \Phi + \Delta$, with Δ representing an additional component. Various parameter-efficient fine-tuning (PEFT) methods, such as LoRA [21], Adapters [20], or prompt-tuning [22] have been developed to achieve this objective and have demonstrated remarkable performance compared to the conventional full fine-tuning. Since Δ is typically a much smaller component than the complete model in PEFT methods, we can conveniently maintain and learn an empirical distribution over several light-weight components Δ, making our approach feasible.

5 Experiments

In this section, we conduct extensive experiments across various settings to validate the effectiveness of our proposed method: Image classification Benchmark, Domain generalization setting, and Few-shot Learning. Each experiment is repeated with three random seeds, and the mean accuracy is reported.

Detail of the experimental setting is presented in Appendix A, which includes the backbone, how to set up multiple particles, the kernel function, and trade-off parameters.

5.1 Image classification

VTAB-1k dataset [53] consists of 19 distinct datasets, which are grouped into three categories: Natural, Specialized, and Structured. Each dataset contains only 1,000 images for training, making the task challenging due to the limited amount of data. Additionally, the images show significant variation in data distribution across the datasets, further complicating the learning process.

We conduct experiments using four particles for our GAC-MSO and all baselines, except full fine-tuning (FFT), AdamW, and SAM, for which we use a single particle consistent with standard LoRA-based fine-tuning of foundation models. Each particle is randomly initialized at the start.

It's important to note that using four particles increases the total number of trainable parameters by a factor of four compared to single-particle methods. However, more parameters do not necessarily lead to improved performance and can sometimes even degrade it. As shown in Table 1, most baseline methods with multi-solution settings perform worse compared to the single-solution setting methods. This may be due to inefficient model scaling or the trade-off between learning diverse solutions and optimizing individual performance. Despite these challenges, our GAC-MSO method outperforms all baselines, achieving the highest average accuracy with a notable 2.3% improvement.

Additionally, we evaluate all methods using the Expected Calibration Error (ECE), which measures how well the predicted probabilities align with actual outcomes. The results, presented in Table 2, show that GAC-MSO achieves a comparable ECE score to SAM under single-particle settings and outperforms other SAM-based methods, such as SADA-JEM [49] and SA-BNN [32], in the multi-particle scenario. SAM-based approaches are known for producing solutions that lie in flatter regions of the loss landscape, correlating with better generalization. Additionally, our method achieves better ECE performance compared to SVGD [28], which records the best ECE score among the other multi-solution baselines.

FGVC dataset. The FGVC benchmark comprises five fine-grained datasets for visual classification tasks: CUB-100-2011 [42], NABirds [40], Oxford Flowers [34], Stanford Dogs [12], and Stanford Cars [17]. Each dataset contains between 1,000 and 21,000 images for training, offering a diverse

Table 1: VTAB-1K results evaluated on Top-1 accuracy. All methods are applied to finetune the same set of LoRA parameters on ViT-B/16 pre-trained with ImageNet-21K dataset.

	Natural							Specialized				Structured								
Method	CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI	dSpr-Loc	dSpr-Ori	sNORB-Azi	sNORB-Ele	**AVG**
Single solution setting																				
FFT [22]	68.9	87.7	64.3	97.2	86.9	87.4	38.8	79.7	95.7	84.2	73.9	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	65.6
AdamW [21]	67.1	90.7	68.9	98.1	90.1	84.5	54.2	84.1	94.9	84.4	73.6	**82.9**	69.2	49.8	78.5	75.7	47.1	31.0	44.0	72.0
SAM [16]	72.7	90.3	71.4	99.0	90.2	84.4	52.4	82.0	92.6	84.1	74.0	76.7	68.3	47.9	74.3	71.6	43.4	26.9	39.1	70.5
Multi-solution setting																				
DeepEns [26]	69.1	88.9	67.7	98.9	90.7	85.1	54.5	82.6	94.8	82.7	75.3	46.6	47.1	47.4	68.2	71.1	36.6	30.1	35.6	67.0
BayesTune [23]	67.2	91.7	69.5	99.0	90.7	86.4	54.7	84.9	95.3	84.1	75.1	82.8	68.9	49.7	79.3	74.3	46.6	30.3	42.8	72.2
SGLD [48]	68.7	91.0	67.0	98.6	89.3	83.0	51.6	81.2	93.7	83.2	76.4	80.0	70.1	48.2	76.2	71.1	39.3	31.2	38.4	70.4
SADA-JEM [49]	70.3	91.9	70.2	98.2	91.2	85.6	54.7	84.3	94.1	83.4	**77.0**	79.9	**72.1**	**51.6**	79.4	70.7	45.3	29.6	40.1	72.1
SA-BNN [32]	65.1	91.5	71.0	98.9	89.4	**89.3**	55.2	83.2	94.5	86.4	75.2	61.4	63.2	40.0	71.3	64.5	34.5	27.2	31.2	68.1
SVGD [28]	71.3	90.2	71.0	98.7	90.2	84.3	52.7	83.4	93.2	86.7	75.1	75.8	70.7	49.6	79.9	69.1	41.2	30.6	33.1	70.9
GAC-MSO (Ours)	**73.7**	**94.9**	**72.6**	**99.4**	**91.6**	85.8	**58.3**	**86.2**	**96.2**	**86.9**	74.0	79.0	63.8	51.0	**79.9**	**84.4**	**58.3**	**33.4**	**46.4**	**74.5**

Table 2: VTAB-1K results evaluated on the Expected Calibration Error (ECE) metric. All methods are applied to fine-tune the same set of LoRA parameters on ViT-B/16 pre-trained with ImageNet-21K dataset.

	Natural							Specialized				Structured								
Method	CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI	dSpr-Loc	dSpr-Ori	sNORB-Azi	sNORB-Ele	**AVG**
Single solution setting																				
FFT [22]	0.29	0.23	0.20	0.13	0.27	0.19	0.45	0.21	0.13	0.18	0.17	0.41	0.44	0.42	0.22	0.14	0.23	0.24	0.40	0.26
AdamW [21]	0.38	0.19	0.18	0.05	0.09	0.10	**0.14**	0.11	0.09	0.12	0.11	**0.12**	0.19	0.34	0.18	0.14	0.21	0.18	0.31	0.17
SAM [16]	0.21	0.25	0.20	0.11	**0.12**	0.15	**0.14**	0.17	0.16	0.14	**0.09**	**0.12**	**0.17**	0.24	0.16	0.21	**0.19**	0.13	0.16	**0.16**
Multi-solution setting																				
DeepEns [26]	0.24	0.12	0.22	0.04	0.10	0.13	0.23	0.16	0.07	0.15	0.21	0.31	0.32	0.36	0.13	0.32	0.31	0.16	0.29	0.20
BayesTune [23]	0.32	0.08	0.20	0.03	0.85	0.12	0.22	0.13	0.07	0.13	0.22	**0.12**	0.23	0.30	0.24	0.28	0.28	0.31	0.26	0.23
SGLD [48]	0.26	0.20	0.17	0.05	0.18	0.14	0.23	0.18	0.09	0.12	0.32	0.26	0.29	0.21	0.26	0.42	0.39	**0.11**	0.24	0.22
SADA-JEM [49]	0.22	0.11	0.20	0.05	0.13	0.16	0.18	0.15	0.21	0.23	0.26	0.19	0.20	0.25	0.27	0.35	0.20	0.14	**0.13**	0.19
SA-BNN [32]	0.22	0.08	0.19	0.15	0.12	0.12	0.24	0.13	0.06	0.12	0.18	0.14	0.21	0.22	0.24	0.25	0.41	0.46	0.34	0.20
SVGD [28]	0.20	0.13	0.19	0.04	0.16	**0.09**	0.20	0.15	0.11	0.13	0.12	0.17	0.21	0.30	0.18	0.21	0.25	0.14	0.26	0.18
GAC-MSO (Ours)	**0.14**	**0.03**	**0.16**	**0.00**	**0.06**	0.11	0.15	**0.12**	**0.03**	**0.08**	0.18	0.16	0.29	0.38	**0.05**	**0.09**	0.25	0.41	0.38	**0.16**

range of challenges for fine-grained image recognition. Detailed results and experimental setup are placed in Appendix A.

The results demonstrate that our GAC-MSO method achieves notable improvements in both accuracy and ECE score. In particular, GAC-MSO significantly outperforms the SVGD approach on calibration, achieving an ECE score of 0.05 compared to 0.14 on SVGD. This highlights the effectiveness of our approach not only in enhancing predictive performance but also in improving model confidence and trustworthiness in fine-grained classification tasks.

5.2 Few-shot learning

In this section, we extend our analysis to a few-shot learning setting by varying the number of training samples (shots) per class across 1, 2, 4, 8, and 16. We evaluate performance on five fine-grained datasets: FGVC-Aircraft [30], Oxford-Pets [36], Food-101 [5], Stanford Cars [24], and Oxford-Flowers102 [33].

We adopt the same experimental setup as described in Section 2 for standard image classification, using four particles for GAC-MSO, SVGD [28], and Deep Ensemble methods, and a single particle for AdamW. The results for each dataset are shown in Figure 1, where our GAC-MSO achieves the highest accuracy across most of the shot settings compared to the baselines. The detailed results, including accuracy and ECE scores, are provided in Appendix A. Overall, GAC-MSO also demonstrates a notable improvement in calibration performance, achieving lower ECE scores than competing methods.

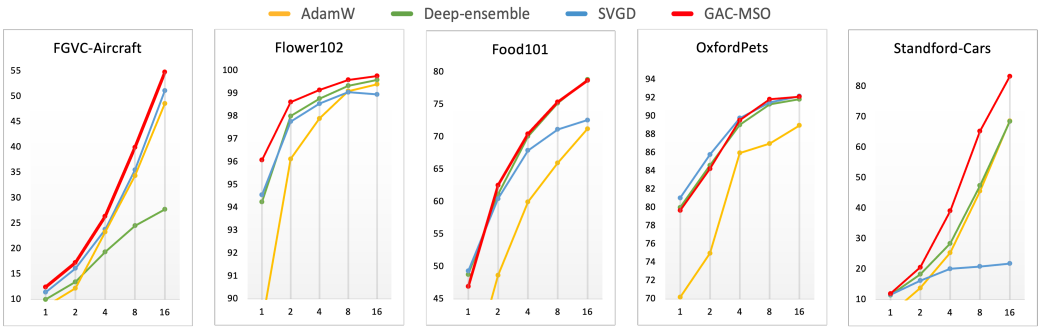

Figure 1: Accuracy on the few-shot benchmark FGVC. The x-axis represents the number of shots (training samples per class) in this setting. Evaluation of ECE scores is provided in Appendix.

Table 3: Top-1 accuracy on domain generalization experiments. All models are fine-tuned with a subset ImageNet-1K dataset and tested on five datasets.

Method	Accuracy					ECE				
	Source	Target				Source	Target			
	ImageNet	-Sketch	-V2	-A	-R	ImageNet	-Sketch	-V2	-A	-R
Single-solution setting										
AdamW [21]	70.8	20.0	59.3	6.9	23.3	-	-	-	-	-
Multi-solution setting										
Deep-ensemble [26]	79.4	36.2	68.9	17.6	33.9	0.069	**0.028**	**0.041**	0.130	**0.034**
SVGD [28]	77.4	36.6	67.3	17.1	35.1	0.500	0.253	0.435	**0.088**	0.232
GAC-MSO (Ours)	**79.6**	**37.3**	**69.1**	**19.6**	**35.8**	**0.058**	0.049	0.044	0.123	0.049

5.3 Domain generalization

We analyze the robustness of our method in practical scenarios where domain shift [58] is unavoidable. In this setting, the model is fine-tuned on a subset of the ImageNet-1K dataset [13], which includes 16 samples per class. After fine-tuning, we test the model on three widely used validation sets derived from ImageNet-1K: the original ImageNet-1K validation set, ImageNet-V2 [37], and ImageNet-Sketch [45]. Additionally, we also include two challenging benchmarks: ImageNet-A [19], which consists of naturally adversarial samples, and ImageNet-R [18], which contains artistic and abstract renditions of ImageNet classes. As shown in Table 3, our GAC-MSO method consistently achieves higher accuracy than all baseline methods across all test sets, including both mild (Sketch and V2) and extreme (Adversarial and Rendition) domain shifts. The improvement gap is notable on the more challenging ImageNet-A and ImageNet-R datasets. Additionally, GAC-MSO maintains a comparable or better ECE score, indicating that the predictions remain well-calibrated even under out-of-distribution conditions.

5.4 Effectiveness of geometry-aware and divergency term

In this section, we analyze the effectiveness of two key components: the geometry term, which encourages diversity in the model space by leveraging geometric structure, and the divergence term, which promotes diversity in predictions within the output space. Results are presented in Table 4. Incorporating the geometry term leads to a significant performance improvement compared to models without it, highlighting the benefit of modeling relationships in parameter space. Furthermore, adding the divergence term provides additional gains, demonstrating its effectiveness in enhancing ensemble diversity and improving predictive performance.

5.5 Additional experiments on trade-off α of divergence term and number of particles

Detailed experiments and results are presented in the Supplementary.

Table 4: Results on the VTAB-1K Dataset. We report the average performance across the three task groups: natural, specialized, and structured.

Geometry	Divergence	Natural	Specialized	Structured	Average
		79.77	84.60	56.25	73.54
x		82.13	85.73	60.96	76.27
x	x	**82.32**	**85.83**	**62.03**	**76.72**

6 Conclusion and Limitation

In this work, we have addressed the challenges of adapting large foundation models to downstream tasks, particularly when diverse solutions are needed for improved robustness and ensemble performance. We introduced the Geometry-Aware Collaborative Multi-Solution Optimizer (GAC-MSO), a novel framework that leverages parameter-efficient fine-tuning (PEFT) by optimizing lightweight modules while sharing a common backbone. Grounded in gradient flow theory and geometric structure, GAC-MSO promotes diversity not only in parameter space but also in output behavior. Our extensive experimental evaluation across transfer learning, few-shot learning, and domain generalization demonstrates that GAC-MSO significantly outperforms existing baseline methods, providing strong predictive performance with affordable computational cost. These results highlight the potential of GAC-MSO for efficient and effective adaptation of foundation models in resource-constrained settings.

Acknowledgment

Trung Le, Mehrtash Harandi, and Dinh Phung were supported by the ARC Discovery Project grants DP230101176 and DP250100262. Trung Le and Mehrtash Harandi were also supported by the Air Force Office of Scientific Research under award number FA9550-23-S-0001.

References

[1] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[2] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2000. Translated by Daishi Harada.

[3] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics ETH Zürich. Birkhäuser, 2. ed edition.

[4] Dennis S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*. Princeton University Press, Princeton, NJ, 2nd edition, 2009.

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[8] José Antonio Carrillo, Katy Craig, and Francesco S. Patacchini. A blob method for diffusion, 2019.

[9] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[10] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

[11] Francesco D'Angelo and Vincent Fortuin. Repulsive deep ensembles are bayesian. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[12] E Dataset. Novel datasets for fine-grained image categorization. In *First workshop on fine grained visual categorization, CVPR. Citeseer. Citeseer. Citeseer*, volume 5, page 2. Citeseer, 2011.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[17] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, and Li Fei-Fei. Fine-grained car detection for visual census estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[18] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[19] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021.

[20] 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.

[21] 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.

[22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022.

[23] Minyoung Kim and Timothy M Hospedales. Bayestune: Bayesian sparse deep model fine-tuning. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, volume 36, pages 65317–65365. Curran Associates Inc, December 2023. Thirty-Seventh Conference on Neural Information Processing Systems, NeurIPS 2023 ; Conference date: 10-12-2023 Through 16-12-2023.

[24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[25] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. In *Advances in Neural Information Processing Systems*, volume 24, 2012.

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[27] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[28] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[31] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996.

[32] Van-Anh Nguyen, Tung-Long Vuong, Hoang Phan, Thanh-Toan Do, Dinh Phung, and Trung Le. Flat seeking bayesian neural networks, 2023.

[33] M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1447–1454. IEEE, 2006.

[34] 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*, pages 722–729. IEEE, 2008.

[35] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979, 2019.

[36] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[37] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[38] Filippo Santambrogio. Optimal transport for applied mathematicians. calculus of variations, pdes and modeling. 2015.

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[40] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 595–604, 2015.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[43] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[44] Dilin Wang and Qiang Liu. Nonlinear stein variational gradient descent for learning diversified mixture models. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6576–6585. PMLR, 09–15 Jun 2019.

[45] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019.

[46] Yaoming Wang, Bowen Shi, Xiaopeng Zhang, Jin Li, Yuchen Liu, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Qi Tian. Adapting shortcut with normalizing flow: An efficient tuning framework for visual recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15965–15974. IEEE, 2023.

[47] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[48] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.

[49] Xiulong Yang, Qing Su, and Shihao Ji. Towards bridging the performance gaps of joint energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15732–15741, 2023.

[50] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[51] Dongshuo Yin, Leiyi Hu, Bin Li, and Youqun Zhang. Adapter is all you need for tuning visual tasks. *arXiv preprint arXiv:2311.15010*, 2023.

[52] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

[53] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

[54] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

[55] Henry Hengyuan Zhao, Pichao Wang, Yuyang Zhao, Hao Luo, Fan Wang, and Mike Zheng Shou. Sct: A simple baseline for parameter-efficient fine-tuning via salient channels. *International Journal of Computer Vision*, 132(3):731–749, 2024.

[56] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[57] Hongyun Zhou, Xiangyu Lu, Wang Xu, Conghui Zhu, Tiejun Zhao, and Muyun Yang. Lora-drop: Efficient lora parameter pruning based on output evaluation. *arXiv preprint arXiv:2402.07721*, 2024.

[58] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4396–4415, 2022.

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: our claims match theoretical and experimental results. The overall setting could apply to other PEFT methods.

   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: [NA]

   Justification: There is no limitation discussed.

   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: [Yes]

Justification: the proof of theories are presented in the Appendix and Suppermentary.

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: We provide details of the experimental setup and full code to reproduce the results.

   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: [Yes]

Justification: We provide the link and the code with detailed instructions for setting up the dataset and hyperparameters to reproduce the experiments.

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: We provide detailed settings for each experiment along with the code to reproduce.

   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: All experiments are repeated at least 3 times with different random seeds. We report the mean accuracy to save space.

   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: We do not include discussion of the resource.

   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: We reviewed and followed the Code of Ethics.

   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: [NA]

    Justification: There is no societal impact of the work performed

    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: Our research poses no risks for misuse.

    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: We cite every dataset and baseline we mentioned and used in this paper.

    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: Our 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: 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: Our 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: The core method development in this research involves deep neural networks in general.

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.