
Supplementary material for GAC-MSO: 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

1 Experimental Settings

PEFT module and Pre-trained backbone. We use the ViT-B/16 model pre-trained on ImageNet-21K as the backbone and initialization for all experiments, following standard practices in the visual PEFT literature. For parameter-efficient fine-tuning, we consistently apply the LoRA method [4] with a rank of 8 as the compact auxiliary module in all settings. The code for our method is at <https://github.com/anh-ntv/GAC-MSO>

Particle setting Each LoRA module introduces two trainable low-rank matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$, with $r \ll d$, to efficiently adapt pre-trained weights by approximating the updates of corresponding full-rank weight matrices to a new task. In the ViT-LoRA setup, two LoRA modules are injected into each self-attention block of the Transformer, modifying either the query W_q or value W_v projection matrices in the Attention layers. To integrate our GAC-MSO approach, each particle includes a unique set of LoRA modules, enabling the learning of diverse fine-tuning trajectories while keeping the shared backbone parameters fixed. This design facilitates efficient, scalable adaptation for the multi-solution approach without duplicating the full model. The final output is then generated by averaging the predictions from all particles.

Formally, let the ViT backbone consist of L Transformer blocks. Each block includes an Attention module, an MLP module, and optionally a parameter-efficient fine-tuning component. In the ViT-LoRA configuration, low-rank adaptation is applied to the query and value projections within the Attention module.

The weight set for the Attention module is defined as:

$$W_{\text{Att}} = \{W_q, W_v, A_q, B_q, A_v, B_v, W\},$$

where W_q and W_v are the pre-trained projection matrices for the query and value, A_q and B_q are the low-rank LoRA parameters for the query path, A_v and B_v for the value path, and W denotes the remaining attention parameters (e.g., key and output projections, biases, etc.).

In ViT-LoRA, given the input token representation x , the query Q and value V are computed as:

$$Q = xW_q + (xA_q)B_q \quad V = xW_v + (xA_v)B_v,$$

In the multi-particle setting, each particle m maintains its own set of LoRA parameters $\{A_q^m, B_q^m, A_v^m, B_v^m\}$, while sharing the base model parameters $\{W_q, W_v, W\}$. The query and value projections for particle m are thus:

$$Q^m = xW_q + (xA_q^m)B_q^m \quad V^m = xW_v + (xA_v^m)B_v^m.$$

The full set of weights for the Attention module in the m -th particle becomes:

$$W_{\text{Att}}^m = \{W_q, W_v, A_q^m, B_q^m, A_v^m, B_v^m, W\}.$$

This architecture allows each particle to follow a unique fine-tuning trajectory through its own LoRA parameters while executing in parallel via the shared Transformer backbone, ensuring efficient and scalable multi-solution training.

Kernel function Our proposed GAC-MSO method is compatible with any positive semi-definite kernel $K(\theta, \theta') : \Theta \times \Theta \rightarrow \mathbb{R}$. Similar to SVGD [6], we employ the RBF kernel $k(\theta, \theta') = \exp(-\frac{1}{2\sigma^2} \|\theta - \theta'\|_2^2)$ and we set the kernel width $\sigma = 1$ for all experiments.

Trade-off β and α and momentum γ In the practical implementation of our method, we introduce a trade-off coefficient β for the energy function and α for the divergence loss. For simplicity, we fix $\beta = 1$ across all experiments and set $\alpha = 0.2$. The effectiveness of the divergence term is further evaluated through ablation studies, as discussed in Section 3. Additionally, we approximate $H(\tilde{\theta})^{-1}$ using the diagonal elements, updated via an exponential moving average with momentum $\gamma \in [0, 1]$. We consistently set this momentum, as well as the momentum used in baseline methods, to 0.9.

2 Additional experiments

2.1 Image Classification with FGVC dataset

FGVC dataset. The FGVC benchmark comprises five fine-grained datasets for visual classification tasks: CUB-100-2011 [9], NABirds [8], Oxford Flowers [7], Stanford Dogs [1], and Stanford Cars [2]. Number of training samples in these datasets is from 1,000 to 21,000 images, significantly more than the VTAB-1K dataset. We evaluate our method in settings where data is not abundant.

We follow the same experimental setup as used with the VTAB-1K dataset, which trains all baselines in the multi-solution setting with four particles. The results, presented in Tables 1 and 2, 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.

2.2 Image Classification with ViT-B Adapter

Adapter is one of the pioneering approaches in parameter-efficient fine-tuning (PEFT), which introduces a bottleneck module into each Transformer layer—typically placed after the Multi-Head

Method	CUB-200 -2011	NABirds	Oxford Flowers	Stanford Dogs	Stanford Cars	Mean
Single solution setting						
AdamW [4]	85.6	79.8	98.9	87.6	72.0	84.78
Multi-solution setting						
Deep-ensemble [5]	79.2	46.7	96.7	90.6	24.7	67.57
SGLD [10]	85.6	74.1	97.4	91.0	48.4	79.50
SVGD [6]	87.8	78.8	99.1	91.0	81.2	87.58
GAC-MSO (Ours)	87.5	81.8	99.4	91.1	83.2	88.60

Table 1: Image classification experiments on FGCV benchmarks with ViT-B/16 pre-trained models

Method	CUB-200 -2011	NABirds	Oxford Flowers	Stanford Dogs	Stanford Cars	Mean
Multi-solution setting						
Deep-ensemble [5]	0.62	0.45	0.70	0.45	0.21	0.48
SGLD [10]	0.26	0.18	0.49	0.06	0.29	0.26
SVGD [6]	0.11	0.26	0.07	0.03	0.25	0.14
GAC-MSO (Ours)	0.04	0.07	0.00	0.03	0.12	0.05

Table 2: Image classification experiments on FGCV benchmarks with ViT-B/16 pre-trained models

Self-Attention (MHSA) or MLP modules. ViT-Adapter enables fine-tuning of a small subset of parameters, significantly reducing the total number of trainable parameters. We evaluate our GAC-MSO approach on the ViT-Adapter by inserting M Adapter modules after each Transformer block, where M denotes the number of particles. The experimental setup follows the same configuration as used with ViT-LoRA, and the bottleneck rank for each Adapter is set to 8.

The results on the VTAB-1K dataset are summarized in Table 3. Our method outperforms baseline models in both the single-solution setting ($M = 1$) and the multi-solution setting ($M = 4$), demonstrating that our MSO framework is flexible and can be effectively integrated with different PEFT modules.

Table 3: VTAB-1K results evaluated on Top-1 accuracy. All methods are applied to fine-tune the ViT-B/16 Adapter with pre-training on ImageNet-21K dataset.

Method	Natural						Specialized				Structured								AVG	
	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
Accuracy																				
Single-solution setting																				
AdamW [3]	69.2	90.1	68.0	98.8	89.9	82.8	54.3	84.0	94.9	81.9	75.5	80.9	65.3	48.6	78.3	74.8	48.5	29.9	41.6	71.44
Multi-solution setting																				
Deep-Ensemble [5]	76.4	91.8	73.6	99.2	91.9	87.8	57.4	86.1	95.6	86.1	74.4	81.4	63.2	51.8	79.3	79.7	56.2	30.2	40.8	73.83
SGLD [10]	57.0	84.2	67.2	98.3	91.4	60.2	32.2	82.6	92.3	74.8	74.1	58.6	37.2	40.8	72.3	75.9	29.7	13.6	27.0	61.50
SVGD [6]	75.7	91.4	74.0	99.1	91.9	86.8	54.7	85.8	95.7	86.1	74.3	76.8	62.7	50.5	79.6	76.2	56.6	29.2	41.2	73.06
GAC-MSO (Ours)	77.7	93.3	74.3	99.1	91.7	85.5	58.4	87.8	96.3	87.3	74.1	83.4	66.2	51.0	79.3	82.1	59.4	31.9	46.0	74.98
ECE score																				
Multi-solution setting																				
Deep-Ensemble [5]	0.05	0.06	0.04	0.05	0.01	0.05	0.11	0.11	0.01	0.02	0.18	0.09	0.25	0.30	0.15	0.11	0.21	0.27	0.31	0.125
SGLD [10]	0.37	0.28	0.27	0.33	0.11	0.15	0.22	0.03	0.06	0.30	0.01	0.11	0.03	0.03	0.09	0.15	0.03	0.01	0.03	0.137
SVGD [6]	0.08	0.07	0.03	0.05	0.01	0.05	0.13	0.11	0.01	0.03	0.17	0.06	0.21	0.28	0.15	0.09	0.15	0.21	0.25	0.112
GAC-MSO (Ours)	0.07	0.02	0.12	0.00	0.05	0.09	0.09	0.10	0.02	0.05	0.12	0.10	0.25	0.33	0.07	0.09	0.21	0.30	0.32	0.126

3 Ablation study

Effectiveness of trade-off α of the divergence term We propose incorporating a divergence term to encourage diversity in the predictions of each solution in the output space, which enhances the overall combined prediction. However, the effectiveness of this term can vary depending on the number of classes in a dataset. For instance, CIFAR-100, with 100 categories, tends to naturally yield a smaller divergence loss l_{div} compared to a dataset like sNORB-Ele, which contains only 9 categories.

To evaluate the impact of the divergence term, we conduct an ablation study on the VTAB-1K dataset by varying the trade-off coefficient in the loss function. Results are presented in Table 4. In most cases, applying the divergence term (*i.e.*, $\alpha > 0$) leads to improved performance in both accuracy and ECE compared to omitting it ($\alpha = 0$). On average, setting yields the best accuracy, while higher values of α may further improve model calibration, as reflected in lower ECE scores.

Table 4: Effectiveness of the divergence term with different trade-off α

α	Natural							Specialized				Structured								AVG
	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	
0.0	Results evaluated on Top-1 accuracy																			
0.0	74.0	94.3	73.0	99.3	91.8	84.7	57.8	86.2	96	86.5	74.2	77.8	63.1	51.2	77.5	82.2	57.4	33.2	45.3	73.97
0.1	73.6	94.9	72.9	99.3	92.1	85.9	58.1	86.2	96.1	86.8	73.1	79.0	63.8	51.8	79.2	84.2	58.0	33.8	45.8	74.45
0.2	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	74.52
0.3	73.5	94.8	73.1	99.4	91.7	86.0	58.3	86.2	95.9	87.2	73.8	79.5	63.9	50.7	74.3	84.3	58.2	32.8	46.4	74.21
0.0	Results evaluated on ECE score																			
0.0	0.15	0.03	0.17	0.01	0.06	0.12	0.16	0.12	0.03	0.08	0.22	0.17	0.30	0.39	0.2	0.11	0.26	0.43	0.39	0.18
0.1	0.14	0.03	0.17	0.00	0.06	0.11	0.16	0.12	0.03	0.08	0.20	0.16	0.29	0.39	0.13	0.09	0.26	0.41	0.39	0.17
0.2	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
0.3	0.14	0.03	0.16	0.00	0.06	0.10	0.15	0.12	0.03	0.07	0.19	0.16	0.29	0.39	0.11	0.09	0.25	0.41	0.38	0.16

Experiment of number of particles We conducted experiments on six datasets from VTAB-1K, from the Natural, Specialized, and Structured categories, to investigate how the number of particles affects final performance. In these experiments, the number of particles was varied from 2 to 5, while all other settings remained consistent with the VTAB-1K setup described in Section 4. As shown in Table 5, increasing the number of particles generally improves performance. However, the performance gaps from 3 to 4 particles and from 4 to 5 particles are less than the gap from 2 to 3 particles.

Table 5: Experiments on different numbers of particles

# particle	Natural		Specialized		Structured		Average
	SVHN	Sun397	Resisc45	Retinopathy	sNORB-Azi	sNORB-Ele	
2	84.10	57.48	86.21	72.89	31.89	45.15	62.953
3	85.55	58.00	86.49	73.47	33.56	46.21	63.880
4	85.41	57.81	87.13	73.84	33.75	46.13	64.011
5	85.52	58.27	87.02	73.60	33.88	46.28	64.095

4 All Proof

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

$$d(\rho, \rho_t) = \mathbb{E}_{\theta' \sim \rho, \theta \sim \rho_t} [\mathbb{E}_{\mathbf{x}} [KL(f(\mathbf{x}; \theta'), f(\mathbf{x}; \theta)) + KL(f(\mathbf{x}; \theta), f(\mathbf{x}; \theta'))]], \quad (1)$$

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

Lemma 4.1. *The divergence $d(\rho, \rho_t)$ in Eq. (2) (which is Eq. 4 in the main paper) can be approximated as*

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

$$\text{where } H(\theta) = \mathbb{E}_{\mathbf{x}} [\mathbb{E}_{\mathbf{y}} [\nabla_{\theta} \log f_{\mathbf{y}}(\mathbf{x}; \theta) \nabla_{\theta} \log f_{\mathbf{y}}(\mathbf{x}; \theta)^\top]],$$

where $f_{\mathbf{y}}(\mathbf{x}; \theta)$ is the y -th prediction output in the prediction probability vector $f(\mathbf{x}; \theta)$.

Proof. We first define the divergence $d(\rho, \rho_t)$ as

$$d(\rho, \rho_t) = \mathbb{E}_{\theta' \sim \rho, \theta \sim \rho_t} [\mathbb{E}_{\mathbf{x}} [KL(f(\mathbf{x}; \theta'), f(\mathbf{x}; \theta)) + KL(f(\mathbf{x}; \theta), f(\mathbf{x}; \theta'))]]. \quad (2)$$

We derive as

$$KL(f(\mathbf{x}; \boldsymbol{\theta}'), f(\mathbf{x}; \boldsymbol{\theta})) + KL(f(\mathbf{x}; \boldsymbol{\theta}), f(\mathbf{x}; \boldsymbol{\theta}')) = \sum_y [f_y(\mathbf{x}; \boldsymbol{\theta}') - f_y(\mathbf{x}; \boldsymbol{\theta})] [\log f_y(\mathbf{x}; \boldsymbol{\theta}') - \log f_y(\mathbf{x}; \boldsymbol{\theta})]$$

$$\begin{aligned} f_y(\mathbf{x}; \boldsymbol{\theta}') - f_y(\mathbf{x}; \boldsymbol{\theta}) &= f_y(\mathbf{x}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log f_y(\mathbf{x}; \boldsymbol{\theta})^T d\boldsymbol{\theta}, \\ \log f_y(\mathbf{x}; \boldsymbol{\theta}') - \log f_y(\mathbf{x}; \boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \log f_y(\mathbf{x}; \boldsymbol{\theta})^T d\boldsymbol{\theta}, \end{aligned}$$

where $d\boldsymbol{\theta} = \boldsymbol{\theta}' - \boldsymbol{\theta}$.

Therefore, we reach

$$\sum_y [f_y(\mathbf{x}; \boldsymbol{\theta}') - f_y(\mathbf{x}; \boldsymbol{\theta})] [\log f_y(\mathbf{x}; \boldsymbol{\theta}') - \log f_y(\mathbf{x}; \boldsymbol{\theta})] \approx \sum_y f_y(\mathbf{x}; \boldsymbol{\theta}) d\boldsymbol{\theta}^T \nabla_{\boldsymbol{\theta}} \log f_y(\mathbf{x}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log f_y(\mathbf{x}; \boldsymbol{\theta})^T d\boldsymbol{\theta}.$$

$$\begin{aligned} d(\rho, \rho_t) &\approx \mathbb{E}_{\boldsymbol{\theta}' \sim \rho, \boldsymbol{\theta} \sim \rho_t} \left[\mathbb{E}_{\mathbf{x}} \left[\mathbb{E}_y \left[d\boldsymbol{\theta}^T \nabla_{\boldsymbol{\theta}} \log f_y(\mathbf{x}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log f_y(\mathbf{x}; \boldsymbol{\theta})^T d\boldsymbol{\theta} \right] \right] \right] \\ &= \mathbb{E}_{\boldsymbol{\theta}' \sim \rho, \boldsymbol{\theta} \sim \rho_t} \left[d\boldsymbol{\theta}^T H(\boldsymbol{\theta}) d\boldsymbol{\theta} \right], \end{aligned}$$

$$d(\rho, \rho_t) \approx \mathbb{E}_{\boldsymbol{\theta}' \sim \rho, \boldsymbol{\theta} \sim \rho_t} \left[(\boldsymbol{\theta}' - \boldsymbol{\theta})^T H(\boldsymbol{\theta}) (\boldsymbol{\theta}' - \boldsymbol{\theta}) \right]$$

$$\text{where } H(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}} \left[\mathbb{E}_y \left[\nabla_{\boldsymbol{\theta}} \log f_y(\mathbf{x}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log f_y(\mathbf{x}; \boldsymbol{\theta})^T \right] \right]. \quad \square$$

$$\min_{v_t} \left\{ \eta \int \left\langle \nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}(\boldsymbol{\theta}), v_t(\boldsymbol{\theta}) \right\rangle \rho_t(\boldsymbol{\theta}) d\boldsymbol{\theta} + \mathbb{E}_{\boldsymbol{\theta} \sim \rho_t} \left[\Delta \boldsymbol{\theta}^T H(\boldsymbol{\theta}) \Delta \boldsymbol{\theta} \right] \right\}. \quad (3)$$

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

Moreover, Theorem 4.2 characterizes the optimal solution of OP in (3) (which is OP 9 in the main paper), which involves the geometry of the particles sampled from ρ_t .

Theorem 4.2. *The OP in (3) (which is OP 9 in the main paper) receives the following optimal solution*

$$v_t^*(\tilde{\boldsymbol{\theta}}) = \tilde{\boldsymbol{\theta}} - \eta H(\tilde{\boldsymbol{\theta}})^{-1} \nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}(\tilde{\boldsymbol{\theta}}), \quad (4)$$

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

Proof. Given $\tilde{\boldsymbol{\theta}}$, taking derivative w.r.t. $v_t(\tilde{\boldsymbol{\theta}})$ and setting it to zero, we obtain

$$\eta \nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}(\tilde{\boldsymbol{\theta}}) \rho_t(\tilde{\boldsymbol{\theta}}) + H(\tilde{\boldsymbol{\theta}}) (v_t(\tilde{\boldsymbol{\theta}}) - \tilde{\boldsymbol{\theta}}) \rho_t(\tilde{\boldsymbol{\theta}}) = \mathbf{0}.$$

$$v_t(\tilde{\boldsymbol{\theta}}) = \tilde{\boldsymbol{\theta}} - \eta H(\tilde{\boldsymbol{\theta}})^{-1} \nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}(\tilde{\boldsymbol{\theta}}).$$

□

Tractable Solution. To develop a tractable solution, we first notice that $\tilde{v}_t^*(\tilde{\boldsymbol{\theta}}) = \tilde{\boldsymbol{\theta}} - \eta \nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}(\tilde{\boldsymbol{\theta}}) = \tilde{\boldsymbol{\theta}} + \eta \tilde{u}_t^*(\tilde{\boldsymbol{\theta}})$ 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* $\tilde{u}_t(\tilde{\boldsymbol{\theta}}) = -\nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}(\tilde{\boldsymbol{\theta}})$, we seek the steepest descent direction as in Eq. (2).

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

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

Proof. Given the current solution ρ_t , we learn the velocity $\tilde{v}_t = id + \eta \tilde{u}_t$ to minimize

$$\mathcal{G}(\tilde{u}_t, \eta) = \mathcal{F}(\rho^{[\tilde{v}_t]}) + \alpha \mathcal{L}_{div}(\tilde{u}_t, \eta), \quad (5)$$

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

We rewrite the above OP as

$$\min_{\eta, \tilde{u}_t} \left\{ \beta \int \Psi(\theta + \eta \tilde{u}_t(\theta)) d\rho_t(\theta) + \int \log(k * \rho^{[v_t]}(\theta + \eta \tilde{u}_t(\theta))) d\rho_t(\theta) + \alpha \mathcal{L}_{div}(\tilde{u}_t, \epsilon) \right\}.$$

Noting that

$$\rho_t(\theta) = \rho^{[v_t]}(\theta + \eta \tilde{u}_t(\theta)) |det(\nabla(\theta + \eta \tilde{u}_t(\theta)))|. \\ \beta \int \Psi(\theta + \eta \tilde{u}_t(\theta)) d\rho_t(\theta) + \int \log(K * \rho^{[v_t]}(\theta + \eta \tilde{u}_t(\theta))) d\rho_t(\theta) \\ = \beta \int \Psi(\theta + \eta \tilde{u}_t(\theta)) d\rho_t(\theta) + \int \log\left(\int K(\theta, \theta') \rho^{[v_t]}(\theta' + \eta \tilde{u}_t(\theta')) d\theta'\right) d\rho_t(\theta) \\ = \beta \int \Psi(\theta + \eta \tilde{u}_t(\theta)) d\rho_t(\theta) + \int \log\left(\int K(\theta, \theta') |det(\nabla(\theta + \eta \tilde{u}_t(\theta)))|^{-1} \rho_t(\theta') d\theta'\right) d\rho_t(\theta)$$

Taking derivative w.r.t. η at $\eta = 0$, we obtain

$$\beta \int \nabla \Psi(\theta)^T \tilde{u}_t(\theta) d\rho_t(\theta) - \int \frac{\int K(\theta, \theta') tr(\nabla \tilde{u}_t(\theta)) \rho_t(\theta') d\theta'}{\int K(\theta, \theta') \rho_t(\theta') d\theta'} d\rho_t(\theta) \\ = \beta \mathbb{E}_{\rho_t} [\langle \nabla \Psi(\theta) K(\theta, \cdot), \tilde{u}_t \rangle] - \mathbb{E}_{\rho_t} \left[\frac{1}{\mathbb{E}_{\rho_t} [K(\theta, \theta')]} \mathbb{E}_{\rho_t} [K(\theta, \theta') \langle \tilde{u}_t, \nabla K(\theta, \cdot) \rangle] \right] \\ = \left\langle \mathbb{E}_{\rho_t} \left[\beta \nabla \Psi(\theta) K(\theta, \cdot) - \frac{\mathbb{E}_{\rho_t} [K(\theta, \theta') \nabla K(\theta, \cdot)]}{\mathbb{E}_{\rho_t} [K(\theta, \theta')]} \right], \tilde{u}_t \right\rangle \quad (6)$$

Furthermore, we have

$$\mathcal{L}_{div}(\tilde{u}_t, \eta) = \int l_{div}(\theta_{1:M}^{[\tilde{v}_t]}) \prod_{m=1}^M d\rho^{[\tilde{v}_t]}(\theta_m^{[\tilde{v}_t]}) \quad (7)$$

$$= \int l_{div}([\theta_m + \eta \tilde{u}_t(\theta_m)]_{m=1}^M) \prod_{m=1}^M \rho_t(\theta_m) d\theta_{1:M}.$$

$$\nabla_{\eta} \mathcal{L}_{div}(u_t, \eta) |_{\eta=0} = \left\langle \mathbb{E} \left[\sum_{m=1}^M \nabla_{\theta_m} l_{div}(\theta_{1:M}) K(\theta_m, \cdot) \right], \tilde{u}_t \right\rangle. \quad (8)$$

By combining Eqs. (6) and (8), we reach the conclusion. \square

Finally, we update as follows:

$$v_t^*(\tilde{\theta}) = \tilde{\theta} - \eta H(\tilde{\theta})^{-1} \nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}(\tilde{\theta}) = \tilde{\theta} + \eta H(\tilde{\theta})^{-1} \tilde{u}_t^*(\tilde{\theta}) \\ = \tilde{\theta} - \eta H(\tilde{\theta})^{-1} h(\tilde{\theta}).$$

The training algorithm is presented in 1.

Algorithm 1 GAC-MSO algorithm to update M particles using kernel method in model space

Initialize particles $\theta_1, \theta_2, \dots, \theta_M$, kernel $K(., .)$, momentum $\gamma = 0.9$, trade-off $\beta = 1$, $\alpha = 0.2$ and learning rate η

for each particle $i = 1, \dots, M$ **do**

 # Compute the empirical estimate of the score function:

$$h(\theta_i) = \sum_{m=1}^M \left[\beta \nabla_{\theta_m} \Psi(\theta_m) K(\theta_m, \theta_i) - \frac{\sum_{m=1}^M [K(\theta_m, \theta_i) \nabla_{\theta_m} K(\theta_m, \theta_i)]}{\sum_{m=1}^M K(\theta_m, \theta_i)} \right] \\ + \alpha \sum_{m=1}^M \nabla_{\theta_m} l_{\text{div}}(\theta_{1:M}) K(\theta_m, \theta_i).$$

$$H(\theta_i) = (\nabla_{\theta_m} \Psi(\theta_m) + \nabla_{\theta_m} l_{\text{div}}(\theta_{1:M}))^2$$

 # Using moving average with momentum:

$$m(\theta_i) = \gamma H(\theta_i) + (1 - \gamma)m(\theta_i)$$

 # Update the particle using the gradient descent step:

$$\theta_i = \theta_i - \frac{\eta}{M} m(\theta_i)^{-1} h(\theta_i)$$

end for

Return the updated particles $\theta_1, \theta_2, \dots, \theta_M$,

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [5] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [10] 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.