Appendix

The following manuscript provides the supplementary materials of the main paper: Regularizing Neural Networks with Meta-Learning Generative Models.

A Algorithm of Meta Generative Regularization

Algorithm 1 Meta Generative Regularization

Require: Training dataset $\mathcal{D}$, validation dataset $\mathcal{D}_{\text{val}}$ main model f_θ, generator G, finder F_ϕ, training batchsize B, pseudo batchsize B_{p}, validation batchsize B_{val}, step size η and ξ, hyperparameter λ and λ_{KL}

Ensure: Trained main model f_θ

while not converged **do**
  $\{(x^i, y^i)\}_{i=1}^{B} \sim \mathcal{D}$
  $\{z^i\}_{i=1}^{B_{\text{p}}} \sim \mathcal{N}(0, I)$
  // Updating ϕ for MPS
  $\{(x^i_{\text{val}}, y^i_{\text{val}})\}_{i=1}^{B_{\text{val}}} \sim \mathcal{D}$
  $\{x^i_{\text{p}}\}_{i=1}^{B_{\text{p}}} = \{G_\Phi(F_\phi(z^i), y^i_{\text{p}})\}_{i=1}^{B_{\text{p}}}$
  $\theta' \leftarrow \theta - \eta\nabla_\theta(\frac{1}{B}\ell(f_\theta(x^i), y^i) + \frac{\lambda}{B_{\text{p}}}\ell_{\text{PCR}}(x^i_{\text{p}}; \psi))$
  $\phi \leftarrow \phi - \xi\nabla_\phi(\frac{1}{B_{\text{val}}}\ell(f_{\theta'}(x_{\text{val}}), y_{\text{val}}) + \lambda_{\text{KL}}(D_{\text{KL}}(p_\phi(z)\|p(z))))$
  // Updating θ with PCR
  $\{x^i_{\text{p}}\}_{i=1}^{B_{\text{p}}} = \{G_\Phi(F_\phi(z^i), y^i_{\text{p}})\}_{i=1}^{B_{\text{p}}}$
  $\theta \leftarrow \theta - \eta\nabla_\theta(\frac{1}{B}\ell(f_\theta(x^i), y^i) + \frac{\lambda}{B_{\text{p}}}\ell_{\text{PCR}}(x^i_{\text{p}}; \psi))$
end while

B Additional Experiments

B.1 Evaluation of Gradient Approximation

Here, we evaluate the gradient approximation by Eq. (9). As shown in Table 1, the 1st-order approximation by Eq. (9) well approximated the second-order gradients in speeding up over 10% with 0.08 of the accuracy drop.

Table 1: Performance comparison between MPS with 2nd-order gradients and 1st-order approximated gradients (ResNet-18, Cars).

Method	Top-1 Acc. (%)	Wall Clock Time (hours)
2nd-Order	$87.30^{\pm.39}$	6.55
1st-Order Approx.	$87.22^{\pm.15}$	5.79

B.2 Ablation study of F_ϕ

In Section 3.2, we introduce F_ϕ for meta-optimized parameters and the residual architectures with MLP defined by Eq. (10). We performed an ablation study of MPS with respect to the meta-optimized parameters and the architectures of F_ϕ. We compared MPS with a variant of MPS optimizing G_Φ instead of F_ϕ. We also attempted other architectures for F_ϕ including **Linear**: $W_\phi(z) + b$, **MLP**: $\text{MLP}_\phi(z)$, and **Residual+Shallow**: $z + \tanh(W_\phi(z) + b)$. The results of these variations are shown in Table 2. We observed that MPS with G_Φ caused failures of training f_θ and degrades the accuracy. On the other hand, all variants of MPS with F_ϕ succeeded in boosting the models without MPS. Thus, restricting the number of optimized parameters is important, and determining an optimal z with the finder F_ϕ is effective on the optimization problems of MPS. For the variants of MPS with F_ϕ, we observed that the residual architectures and regularization by $D_{\text{KL}}(p_\phi(z)\|p(z))$ contributed to the successes. Interestingly, MPS with Linear F_ϕ outperformed MPS with MLP F_ϕ, i.e., significantly transforming the input $z \sim p(z)$ by complex functions results in low accuracy. These results suggest

that better latent vectors in $\mathcal{Z}$ to train f_θ can exist near the uniformly sampled input z. Thus, limiting the search range by $\tanh$ in the residual architectures can help in finding better latent vectors.

Table 2: Ablation study of MPS (ResNet-18, Cars).1

Method	Top-1 Acc. (%)
Without MPS (PCR)	$86.32^{\pm.07}$
MPS	$\mathbf{87.22}^{\pm\mathbf{.15}}$
MPS with G_Φ	$84.47^{\pm.05}$
MPS with Linear F_ϕ	$86.51^{\pm.09}$
MPS with MLP F_ϕ	$86.35^{\pm.13}$
MPS with Residual+Shallow F_ϕ	$86.88^{\pm.16}$
MPS w/o $D_{\mathrm{KL}}(p_\phi(z)\|p(z))$	$86.92^{\pm.22}$

B.3 MGR with Diffusion Models

We tested our method on EDM [1], a recent diffusion model. Due to the computation cost, we used a 10% reduced CIFAR-10 as the dataset. We optimized F_ϕ to search the first step noise of the diffusion process. Table 3 shows that our method with EDM improves Base Model. However, the overhead of incorporating diffusion models was significant; it takes more than ten times longer training than GANs. In future work, we will investigate lighter-weight methods using the diffusion model.

Table 3: Performance studies on Diffusion Model (ResNet-18 on Cars)

Method	Top-1 Acc. (%)
Base Model	$86.49^{\pm.48}$
GDA (EDM)	$85.80^{\pm.30}$
MGR	$\mathbf{88.49}^{\pm\mathbf{.12}}$

B.4 Updating F_ϕ without Meta-optimization

MPS consists of meta-learning on validation losses requiring bi-level optimization, which is a relatively heavy computation. One can consider if F_ϕ could be trained without meta-optimization. Here, we try alternative methods other than meta-learning to update F_ϕ. Instead of meta-learning, we used a strategy of choosing hard examples via optimizing F_ϕ. That is, we optimize F_ϕ by maximizing the training cross-entropy (CE) loss and the PCR loss on synthetic samples. Note that, in both cases, we used the PCR loss for synthetic samples when training classifiers. Table 4 shows the results. Optimizing F_ϕ with CE and PCR slightly improved the baselines but significantly underperformed our method (MGR). This result can justify using meta-optimizing F_ϕ to generate useful samples for classifiers. Nevertheless, this idea could inspire a sampling method that does not require bi-level optimization in future work.

Table 4: Performance comparison of updating strategies for F_ϕ (ResNet-18 on Cars)

Method	Top-1 Acc. (%)
Base Model	$85.50^{\pm.10}$
PCR	$86.36^{\pm.08}$
Optimizing F_ϕ w/ CE	$86.52^{\pm.21}$
Optimizing F_ϕ w/ PCR	$86.44^{\pm.68}$
MGR	$\mathbf{87.22}^{\pm\mathbf{.15}}$

References

[1] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.