A. Implementation Details

Hyperparameter search strategy. Similar to [8], the hyperparameter tuning strategy differs depending on the types of pre-trained models. In the experiments of this work, we use three different pre-trained models: ResNet-50 [15] pre-trained on ImageNet [9] (RN50), ViT-B/16 [10] with CLIP [37] (CLIP), and RegNetY-16GF [38] with SWAG [42] (SWAG).

Table 7: Hyperparameters used for RN50 in the experiments.

Hyperparameter	PACS	VLCS	OfficeHome	TerraInc	DomainNet
λ	0.01	0.05	0.01	0.01	0.01
Learning rate	5e-5	5e-5	5e-5	5e-5	5e-5
Weight decay	0.0	1e-4	1e-6	0.0	1e-4
Dropout	0.0	0.5	0.5	0.0	0.1

For RN50, we apply a two-stage hyperparameter search strategy. The batch size and the moving average coefficient (m) are fixed as 32 and 0.999 in the entire search procedure, respectively. In the first stage, we search the gradient scale factor λ from $\{0.01, 0.05, 0.1, 0.5\}$ with the fixed learning rate of 5e-5. In this stage, we do not apply weight decay and dropout, *i.e.,* weight decay and dropout rate are equal to 0. In the second stage, we search the learning rate from $\{$1e-5, 3e-5, 5e-5$\}$, weight decay rate from $\{$0, 1e-6, 1e-4$\}$, and dropout rate from $\{0, 0.1, 0.5\}$ with the fix λ searched in the first stage. We summarize the optimal set of hyperparameters for RN50 in Table 7.

Table 8: The gradient scale factor λ used for CLIP and SWAG in the experiments.

Pre-trained Model	PACS	VLCS	OfficeHome	TerraInc	DomainNet
CLIP	0.05	0.1	0.05	0.05	0.05
SWAG	0.05	0.1	0.05	0.05	0.05

Unlike the experiments with RN50, we apply a single-stage hyperparameter search strategy to CLIP and SWAG due to the size of the larger-scale pre-trained models. We only search the gradient scale factor λ from $\{0.01, 0.05, 0.1, 0.5\}$. In particular, we use the same learning rate, weight decay, and dropout rate used in the hyperparameter search of RN50. For the batch size of CLIP, we fix it as 32 except for the one case on DomainNet [34] where the batch size is set as 24. For SWAG, we fix the batch size as 16 for all experiments. In Table 8, we summarize the searched λ for CLIP and SWAG.

Similar to [8], we set the total number of iterations as 15,000 for DomainNet and 5,000 for the others regardless of types of pre-trained models throughout the entire experiments.

B. Additional Results

B.1. Main results

In § 3.2, we only compare baselines superior to ERM [45] with GESTUR for simplicity. Here, we provide the entire results of the main experiment in Table 9.

Baselines. In the main experiment, we compare GESTUR against a number of baselines: MMD [24], MixStyle [58], GroupDRO [40], IRM [1], ARM [57], VREx [20], CDANN [25], DANN [13], RSC [16], MTL [5], Mixup [48, 50, 51], MLDG [22], Fish [41], Fishr [39], ERM [45], SagNet [32], SelfReg [18], CORAL [43], mDSDI [6], GVRT [30], MIRO [8], SMA [2], and SWAD [7].

B.2. Relationship between λ and the types of the pre-trained model

In § 3.4, we analyze the relationship between λ and the size of the pre-trained model. However, we only present the results from PACS [23] in Table 3 for simplicity. Here, we provide the additional results from VLCS [12], OfficeHome [46], and TerraIncognita [4] in Table 11, Table 12, and Table 13, respectively. We also provide the additional results on PACS again containing the standard error which is omitted in main manuscript due to the page limit (Table 10).

C. Further Analysis

C.1. Comparison with CLIP-based baselines

Setup. CLIP [37] is pre-trained on the huge web-crawled image-caption pair dataset and has been widely adopted in various computer vision tasks due to its generalization ability. CLIP-based methods could be strong baselines in domain generalization because the text content they used in pre-training could act as a robust anchor to the domain shift of images. Therefore, we conduct additional experiments using CLIP-based methods, CLIP Zero-shot and WiSE-FT [49]. The CLIP-based methods require text-based queries to output text-based representations of target classes. Following the previous study, we obtain the 80 text-based queries from the official repository[1] of CLIP and compute the final text-based representation of each target class by averaging the text-based representations of the queries. Finally, the model predictions are computed as the dot product of the text-based representations and the representations of input images. For WiSE-FT, an ensemble

[1] https://github.com/openai/CLIP/blob/main/notebooks/Prompt_Engineering_for_ImageNet.ipynb

Table 9: Domain generalization accuracy (%) on the five domain generalization benchmark datasets with the three different pre-trained models. We mark $*$, †, and ‡ for the results from [14], [7] and [8] respectively. We use the reported numbers from each paper for Fish, Fishr, SelfReg, mDSDI, GVRT, and SMA.

Method	PACS	VLCS	OfficeHome	TerraInc	DomainNet	Avg.
Using ResNet-50 pre-trained on ImageNet.						
MMD*	84.7 ±0.5	77.5 ±0.9	66.3 ±0.1	42.2 ±1.6	23.4 ±9.5	58.8
MixStyle†	85.2 ±0.3	77.9 ±0.5	60.4 ±0.3	44.0 ±0.7	34.0 ±0.1	60.3
GroupDRO*	84.4 ±0.8	76.7 ±0.6	66.0 ±0.7	43.2 ±1.1	33.3 ±0.2	60.7
IRM*	83.5 ±0.8	78.5 ±0.5	64.3 ±2.2	47.6 ±0.8	33.9 ±2.8	61.6
ARM*	85.1 ±0.4	77.6 ±0.3	64.8 ±0.3	45.5 ±0.3	35.5 ±0.2	61.7
VREx*	84.9 ±0.6	78.3 ±0.2	66.4 ±0.6	46.4 ±0.6	33.6 ±2.9	61.9
CDANN*	82.6 ±0.9	77.5 ±0.1	65.8 ±1.3	45.8 ±1.6	38.3 ±0.3	62.0
DANN*	83.6 ±0.4	78.6 ±0.4	65.9 ±0.6	46.7 ±0.5	38.3 ±0.1	62.6
RSC*	85.2 ±0.9	77.1 ±0.5	65.5 ±0.9	46.6 ±1.0	38.9 ±0.5	62.7
MTL*	84.6 ±0.5	77.2 ±0.4	66.4 ±0.5	45.6 ±1.2	40.6 ±0.1	62.9
Mixup*	84.6 ±0.6	77.4 ±0.6	68.1 ±0.3	47.9 ±0.8	39.2 ±0.1	63.4
MLDG*	84.9 ±1.0	77.2 ±0.4	66.8 ±0.6	47.7 ±0.9	41.2 ±0.1	63.6
Fish	85.5 ±0.3	77.8 ±0.3	68.6 ±0.4	45.1 ±1.3	42.7 ±0.2	63.9
Fishr	85.5 ±0.4	77.8 ±0.1	67.8 ±0.1	47.4 ±1.6	41.7 ±0.0	64.0
ERM†	84.2 ±0.1	77.3 ±0.1	67.6 ±0.2	47.8 ±0.6	44.0 ±0.1	64.2
SagNet*	86.3 ±0.2	77.8 ±0.5	68.1 ±0.1	48.6 ±1.0	40.3 ±0.1	64.2
SelfReg	85.6 ±0.4	77.8 ±0.9	67.9 ±0.7	47.0 ±0.3	42.8 ±0.0	64.2
CORAL*	86.2 ±0.3	78.8 ±0.6	68.7 ±0.3	47.6 ±1.0	41.5 ±0.1	64.5
mDSDI	86.2 ±0.2	79.0 ±0.3	69.2 ±0.4	48.1 ±1.4	42.8 ±0.1	65.1
GVRT	85.1 ±0.3	79.0 ±0.2	70.1 ±0.1	48.0 ±1.4	44.1 ±0.1	65.2
MIRO‡	85.4 ±0.4	79.0 ±0.0	70.5 ±0.4	50.4 ±1.1	44.3 ±0.2	65.9
SMA	87.5 ±0.2	78.2 ±0.2	70.6 ±0.1	50.3 ±0.5	46.0 ±0.1	66.5
SWAD†	**88.1** ±0.1	79.1 ±0.1	70.6 ±0.2	50.0 ±0.3	**46.5** ±0.1	66.9
GESTUR	88.0 ±0.2	**80.1** ±0.2	**71.1** ±0.0	**51.3** ±0.2	46.3 ±0.1	**67.4**
Using ViT-B/16 with CLIP.						
ERM‡	83.4 ±0.5	75.9 ±1.3	66.4 ±0.5	35.3 ±0.8	44.4 ±0.6	61.1
SWAD	91.3 ±0.1	79.4 ±0.4	76.9 ±0.1	45.4 ±0.5	51.7 ±0.8	68.9
SMA	92.1 ±0.2	79.7 ±0.2	78.1 ±0.1	48.3 ±0.7	55.9 ±0.2	70.8
MIRO‡	95.6 ±0.8	82.2 ±0.3	82.5 ±0.1	54.3 ±0.4	54.0 ±0.3	73.7
GESTUR	**96.0** ±0.0	**82.8** ±0.1	**84.2** ±0.1	**55.7** ±0.2	**58.9** ±0.1	**75.5**
Using RegNetY-16GF with SWAG.						
ERM‡	89.6 ±0.4	78.6 ±0.3	71.9 ±0.6	51.4 ±1.8	48.5 ±0.6	68.0
SWAD‡	94.7 ±0.2	79.7 ±0.2	80.0 ±0.1	57.9 ±0.7	53.6 ±0.6	73.2
MIRO‡	**97.4** ±0.2	79.9 ±0.6	80.4 ±0.2	58.9 ±1.3	53.8 ±0.1	74.1
SMA	95.5 ±0.0	80.7 ±0.1	82.0 ±0.0	59.7 ±0.0	60.0 ±0.0	75.6
GESTUR	96.9 ±0.1	**83.5** ±0.1	**83.1** ±0.0	**61.1** ±0.4	**60.1** ±0.0	**76.9**

of the fine-tuned and zero-shot models, we set the balance factor α as 0.5 following its original paper since target unseen domains are inaccessible in the domain generalization setting.

Results. Table 14 shows the evaluation results where GESTUR achieves the best averaged performance. In detail, GESTUR outperforms CLIP Zero-shot on VLCS, OfficeHome, and TerraIncognita, and shows comparable performance on PACS. Likewise, GESTUR achieves better performance on PACS, OfficeHome, and TerraIncognita than WiSE-FT and comparable performance on VLCS.

Interestingly, the CLIP-based methods exhibit severe performance degradation on TerraIncognita. We conjecture that their performance is sensitive to pre-defined text-based queries. For example, the query "a sketch of a {}" is help-

Table 10: Evaluation results (%) on PACS with the three different pre-trained models varying λ.

Pre-trained model	Dataset (size)	λ = 0.01	0.05	0.1	0.5
RN50	ImageNet (1.3M)	**88.0** ±0.2	86.0 ±0.2	82.1 ±0.2	73.4 ±0.4
CLIP	CLIP (400M)	94.8 ±0.2	96.0 ±0.0	**96.2** ±0.1	96.0 ±0.0
SWAG	Instagram (3.6B)	96.3 ±0.2	96.9 ±0.1	97.6 ±0.1	**97.9** ±0.1

Table 11: Evaluation results (%) on VLCS with the three different pre-trained models varying λ.

Pre-trained model	Dataset (size)	λ = 0.01	0.05	0.1	0.5
RN50	ImageNet (1.3M)	78.9 ±0.3	**80.1** ±0.2	80.0 ±0.1	77.6 ±0.1
CLIP	CLIP (400M)	81.3 ±0.4	82.7 ±0.1	**82.8** ±0.1	82.1 ±0.3
SWAG	Instagram (3.6B)	81.7 ±0.0	82.7 ±0.2	**83.5** ±0.1	82.4 ±0.2

Table 12: Evaluation results (%) on OfficeHome with the three different pre-trained models varying λ.

Pre-trained model	Dataset (size)	λ = 0.01	0.05	0.1	0.5
RN50	ImageNet (1.3M)	**71.1** ±0.0	**71.1** ±0.1	70.4 ±0.2	68.9 ±0.1
CLIP	CLIP (400M)	82.5 ±0.2	84.2 ±0.1	84.4 ±0.0	**84.7** ±0.0
SWAG	Instagram (3.6B)	81.5 ±0.2	83.1 ±0.0	**83.5** ±0.0	81.1 ±0.1

Table 13: Evaluation results (%) on TerraIncognita with the three different pre-trained models varying λ.

Pre-trained model	Dataset (size)	λ = 0.01	0.05	0.1	0.5
RN50	ImageNet (1.3M)	**51.3** ±0.2	50.0 ±0.4	45.5 ±0.2	31.2 ±0.1
CLIP	CLIP (400M)	51.3 ±0.2	**55.7** ±0.2	54.0 ±0.3	42.3 ±0.9
SWAG	Instagram (3.6B)	57.6 ±0.9	61.1 ±0.4	**62.1** ±0.3	54.9 ±0.1

Table 14: Evaluation results (%) on the four datasets with CLIP. We compare GESTUR with CLIP-based baselines, CILP Zero-shot and WiSE-FT [49] which require additional text-based queries. Our proposed GSETUR outperforms the CLIP-based baselines without requiring additional text-based queries.

Method	PACS	VLCS	OfficeHome	TerraInc	Avg.
CLIP Zero-shot	**96.8** ±0.0	81.7 ±0.3	83.0 ±0.3	31.3 ±0.2	73.2
WiSE-FT ($\alpha = 0.5$)	94.5 ±0.0	**83.9** ±0.3	83.9 ±0.2	47.5 ±1.2	77.5
GESTUR	96.0 ±0.0	82.8 ±0.1	**84.2** ±0.1	**55.7** ±0.2	**79.7**

ful for the "Sketch" domain of PACS. On the other hand, the queries "a plastic {}" and "a {} in a video game" are not helpful for TerraIncognita, which is composed of animal images taken from the wild. These observations indicate that the CLIP-based methods require hard prompt engineering for each target dataset. Moreover, the CLIP-based methods depend on language modality, which cannot be extended to other architecture or learning methods trained on only visual modality, such as RN50 and SWAG. Considering these, our GESTUR achieves a meaningful performance.

Table 15: Evaluation results (%) of combination of SWAD and GESTUR on the four datasets with the three different pre-trained models.

Method	PACS	VLCS	OfficeHome	TerraInc	Avg.
Using ResNet-50 pre-trained on ImageNet.					
GESTUR	88.0 ±0.2	80.1 ±0.2	71.1 ±0.0	51.3 ±0.2	72.6
GESTUR + SWAD	88.3 ±0.1	80.1 ±0.1	71.0 ±0.0	51.2 ±0.2	72.7
Using ViT-B/16 with CLIP.					
GESTUR	96.0 ±0.0	82.8 ±0.1	84.2 ±0.1	55.7 ±0.2	79.7
GESTUR + SWAD	95.9 ±0.0	82.8 ±0.1	84.3 ±0.0	55.3 ±0.6	79.6
Using RegNetY-16GF with SWAG.					
GESTUR	96.9 ±0.1	83.5 ±0.1	83.1 ±0.0	61.1 ±0.4	81.2
GESTUR + SWAD	96.8 ±0.0	83.0 ±0.1	83.4 ±0.1	60.6 ±0.8	81.0

C.2. Applicability of SWAD [7] to GESTUR

Setup. The recent studies [7, 8] have observed that SWAD [7] that seeks the flat minima is a good optimizer for domain generalization, improving the generalization performance of several baselines by applying it to the baselines as a optimizer. Motivated by this observation, we evaluate the performance of our GESTUR applied with SWAD as a optimizer to verify whether GESTUR and SWAD are orthogonal directions to each other.

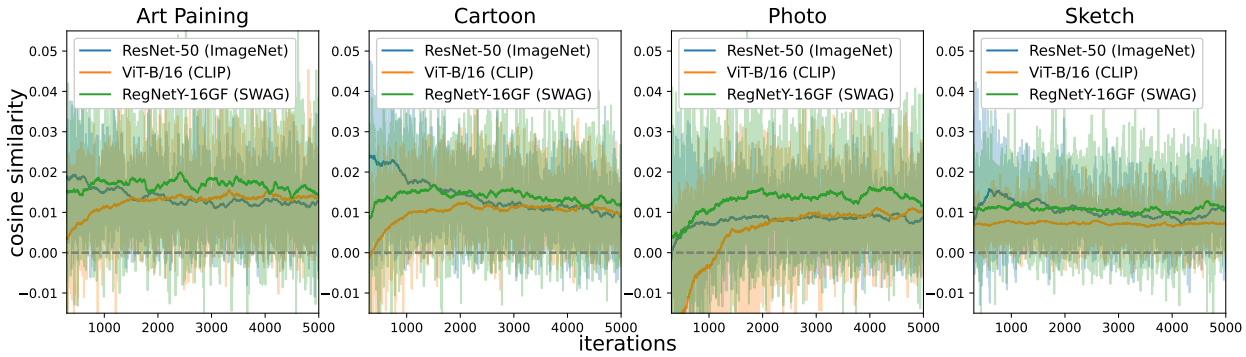
Results. Table 15 shows that SWAD does not improve the performance of GESTUR. We conjecture that it is because EMA used to transfer the knowledge of TE to GE has a similar effect as SWAD to find a flat minima by averaging the model's weights.

C.3. Similarity between true unobservable gradients $\mathbf{g}_u$ and estimated unobservable gradients $\tilde{\mathbf{g}}_u$ of GESTUR

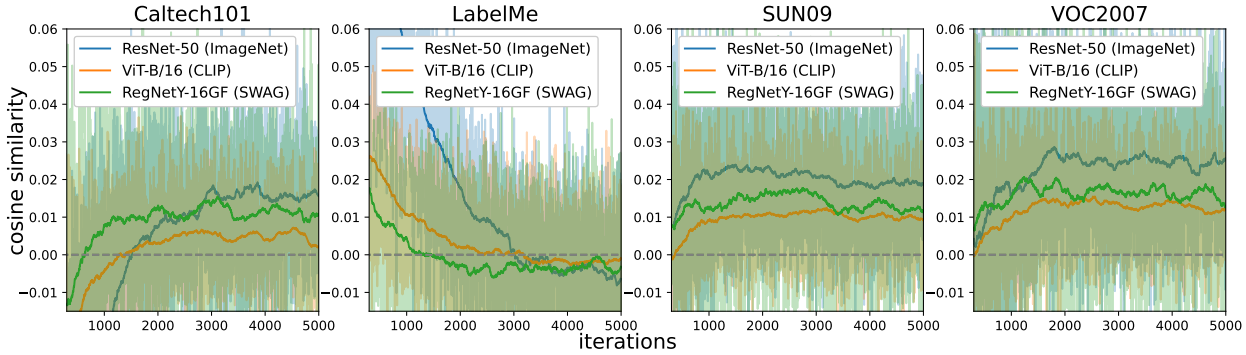
Setup. In this paper, we argue that gradient bias is a major culprit in degrading domain generalization performance (Figure 1) and our proposed method relieves the gradient bias by estimating unobservable gradients. To support this argument, we reported the number of iterations where gradient conflicts exist in Figure 2 and Table 2. To examine whether the estimated unobservable gradients $\tilde{\mathbf{g}}_u$ are similar to the true unobservable gradients $\mathbf{g}_u$, we add the analysis calculating the cosine similarity of the true and estimated

unobservable gradients. Note that the true unobservable gradients are computed by cross-entropy loss using true labels of unseen domain datasets $\mathcal{D}_u$. On the other hand, the estimated unobservable gradients are just computed as the parameter difference between GE and TE ($\theta_{\text{GE}} - \theta_{\text{TE}}$).

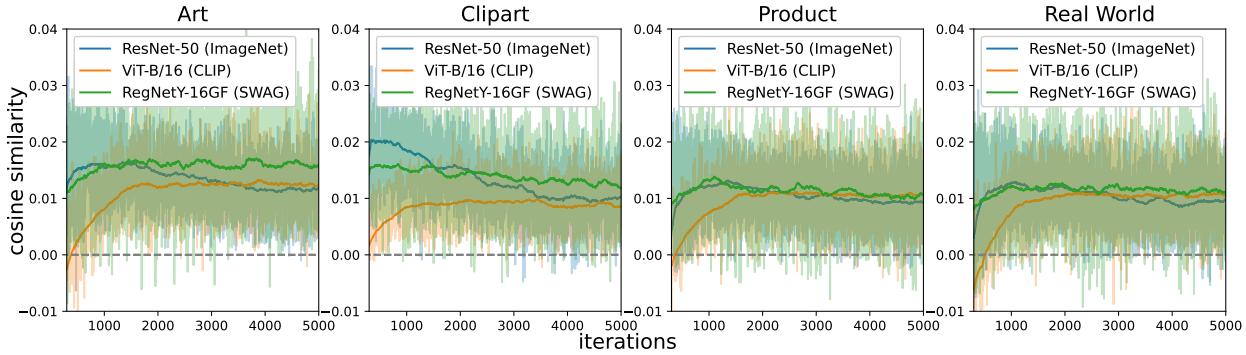
Results. Figure 3 shows that our estimated gradients display positive similarity scores with the true gradients. This trend demonstrates that the estimated gradients reduce the number of gradient conflicts, leading models to reduce the risks of unseen domains without accessing unseen domain data.



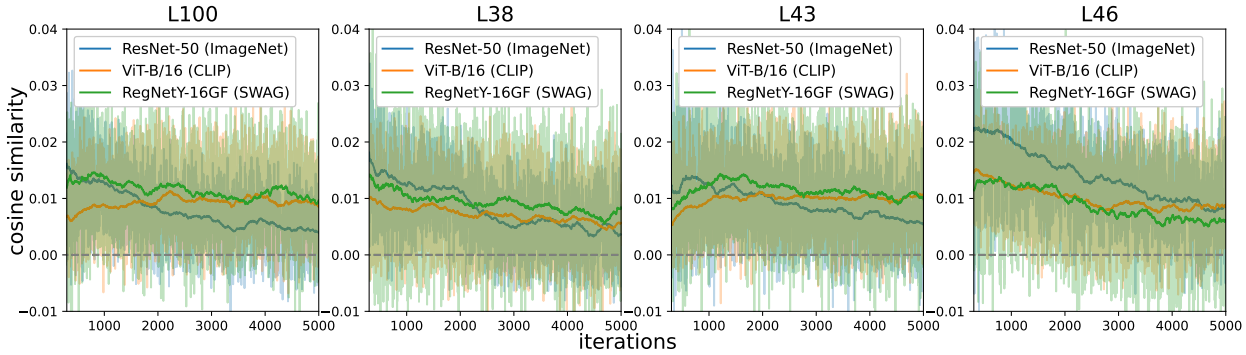
(a) PACS



(b) VLCS



(c) OfficeHome



(d) TerraIncognita

Figure 3: Cosine similarity between the true unobservable gradients $\mathbf{g}_u$ and the estimated unobservable gradients $\tilde{\mathbf{g}}_u$