

## A TECHNICAL DETAILS

### A.1 PROOF OF PROPOSITION 1

Note that  $\mathcal{L}^e(f \circ \phi)$  is defined on a set of fixed examples in  $E_e$ . Since  $f_e \in \arg \min_{f' \in \mathcal{F}} \mathcal{L}^e(f' \circ \phi)$  and  $f_e, f_{-e}$  are in the same parametric family  $\mathcal{F}$ , we have  $\mathcal{R}^e(\phi) = \mathcal{L}^e(f_{-e} \circ \phi) - \mathcal{L}^e(f_e \circ \phi) \geq 0$ .

### A.2 PROOF OF PROPOSITION 2

*Proof.* Consider any representation  $\phi^* \in \Phi_{\text{RGM}}$ . When there are only two environments  $\{E_1, E_2\}$ , we have  $F_{-2}(\phi^*) = F_1(\phi^*)$  and  $F_{-1}(\phi^*) = F_2(\phi^*)$  by definition. Thus the RGM constraint implies

$$F_2(\phi^*) = F_{-1}(\phi^*) \subseteq F_1(\phi^*) \quad F_1(\phi^*) = F_{-2}(\phi^*) \subseteq F_2(\phi^*)$$

Therefore  $F_1(\phi^*) = F_2(\phi^*)$ . Since the loss function is non-negative and  $\mathcal{F}$  is bounded and closed,  $F_1(\phi^*) \neq \emptyset$ . Thus,  $\cap_e F_e(\phi^*) = F_1(\phi^*) \neq \emptyset$ . Now consider any  $f \in \cap_e F_e(\phi^*)$ . By definition,

$$\forall e : \mathcal{L}^e(f \circ \phi^*) \leq \min_{h \in \mathcal{F}} \mathcal{L}^e(h \circ \phi^*)$$

By summing the above inequality over all environments, we have

$$\sum_e \mathcal{L}^e(f \circ \phi^*) \leq \sum_e \min_{h \in \mathcal{F}} \mathcal{L}^e(h \circ \phi^*) \leq \min_{h \in \mathcal{F}} \sum_e \mathcal{L}^e(h \circ \phi^*)$$

Since  $\sum_e \mathcal{L}^e(f \circ \phi^*) = \mathcal{L}(f \circ \phi^*)$ , the above inequality implies

$$\mathcal{L}(f \circ \phi^*) \leq \min_{h \in \mathcal{F}} \mathcal{L}(h \circ \phi^*) = \mathcal{L}_{\text{RGM}}^* = \mathcal{L}_{\text{IRM}}^*$$

Thus,  $f \circ \phi^*$  is an optimal solution under IRM and  $\phi^* \in \Phi_{\text{IRM}}$ .  $\square$

### A.3 PROOF OF PROPOSITION 3

*Proof.* Let us recall our assumption of the data generation process:

$$p(x, y, e) = p(e)p(x|e)p(y|x, e); \quad p(y|x, e) = p(y|x, e(x))$$

Under this assumption, we can rephrase the IRM objective as

$$\min_{f, \phi} \mathbb{E}_e \mathbb{E}_{x|e} \mathbb{E}_{y|x, e} \ell(y, f(\phi(x))) \tag{17}$$

$$\text{s.t. } \mathbb{E}_{x|e} \mathbb{E}_{y|x, e} \ell(y, f(\phi(x))) \leq \min_{f_e} \mathbb{E}_{x|e} \mathbb{E}_{y|x, e} \ell(y, f_e(\phi(x))) \quad \forall e \tag{18}$$

Given any label-preserving representation  $\phi(x)$ , its ERM optimal predictor is

$$f^*(\phi(x)) = \arg \min_f \mathbb{E}_{y|\phi(x)} \ell(y, f(\phi(x))) \tag{19}$$

To see that  $f^*$  is ERM optimal, consider

$$\min_f \mathbb{E}_e \mathbb{E}_{x|e} \mathbb{E}_{y|x, e} \ell(y, f(\phi(x))) \geq \mathbb{E}_e \mathbb{E}_{x|e} \min_f \mathbb{E}_{y|x, e} \ell(y, f(\phi(x))) \tag{20}$$

$$= \mathbb{E}_e \mathbb{E}_{x|e} \min_f \mathbb{E}_{y|\phi(x)} \ell(y, f(\phi(x))) \tag{21}$$

$$= \mathbb{E}_e \mathbb{E}_{x|e} \mathbb{E}_{y|\phi(x)} \ell(y, f^*(\phi(x))) \tag{22}$$

where Eq. (21) holds because  $\phi(x)$  is label-preserving. Note that  $f^*$  satisfies the IRM constraint because it is simultaneously optimal across all environments:

$$\forall e : \min_{f_e} \mathbb{E}_{x|e} \mathbb{E}_{y|x, e} \ell(y, f_e(\phi(x))) \geq \mathbb{E}_{x|e} \min_{f_e} \mathbb{E}_{y|x, e} \ell(y, f_e(\phi(x))) \tag{23}$$

$$= \mathbb{E}_{x|e} \min_f \mathbb{E}_{y|\phi(x)} \ell(y, f(\phi(x))) \tag{24}$$

$$= \mathbb{E}_{x|e} \mathbb{E}_{y|\phi(x)} \ell(y, f^*(\phi(x))) \tag{25}$$

Moreover, if  $\phi \in \Phi_{\text{IRM}}$  is an optimal representation,  $f^* \circ \phi$  is an optimal solution of IRM.  $\square$

Table 3: Dataset statistics

	QM9	HIV	Tox21	BBBP	Homology	Stability
Training	4K	25243	6427	1580	12.3K	54K
Validation	18K	6352	568	206	736	2.4K
Testing	113K	3959	839	256	718	13K

#### A.4 STRUCTURED RGM UPDATE RULE

Since  $\tilde{f}_e$  and  $\phi$  optimizes  $\mathcal{L}(\tilde{f}_e \circ \phi, \tilde{E}_e)$  in different directions, we also introduce a gradient reversal layer between  $\phi$  and  $\tilde{f}_e$ . The SRGM update rule is the following:

$$\begin{aligned}
 \phi &\leftarrow \phi - \eta \nabla_{\phi} \mathcal{L}(f \circ \phi) - \eta \lambda_g \nabla_{\phi} \mathcal{L}_g(g \circ \phi) - \eta \lambda \sum_e \sum_{\psi \in \{0, \delta\}} \nabla_{\phi} \mathcal{R}^e(\phi + \psi) \\
 f &\leftarrow f - \eta \nabla_f \mathcal{L}(f \circ \phi) & g &\leftarrow g - \eta \nabla_g \mathcal{L}_g(g \circ \phi) \\
 \tilde{f}_e &\leftarrow \tilde{f}_e - \eta \nabla \mathcal{L}^e(\tilde{f}_e \circ \phi) & \tilde{f}_e &\leftarrow \tilde{f}_e - \eta \nabla \mathcal{L}(\tilde{f}_e \circ (\phi + \delta)) \quad \forall e \\
 f_{-e} &\leftarrow f_{-e} - \eta \nabla \mathcal{L}^{-e}(f_{-e} \circ \phi) \quad \forall e
 \end{aligned}$$

## B EXPERIMENTAL DETAILS

### B.1 MOLECULAR PROPERTY PREDICTION

**Data** The four property prediction datasets are provided in the supplementary material, along with the training/validation/test splits. The size of each training environment, validation and test set are listed in Table 3. The QM9, Tox21 and BBBP dataset are downloaded from Wu et al. (2018). The HIV dataset is downloaded from the original source with EC50 measurements.<sup>2</sup> The positive class is defined as molecules with EC50 less than 1  $\mu$ M.

For the QM9 ablation study, we consider three training sets  $\mathcal{D}_8, \mathcal{D}_7, \mathcal{D}_6$ : molecules with no more than 8, 7 and 6 atoms (increasing domain shift). When training on  $\mathcal{D}_8$ , we sample 20K compounds from those with 9 atoms as our validation set and the rest for testing. This is less ideal for domain generalization evaluation since we want the validation and test set to come from different domains.

**Model Hyperparameters** For the feature extractor  $\phi$ , we adopt the GCN implementation from Yang et al. (2019). We use their default hyperparameters across all the datasets and baselines. Specifically, the GCN contains three convolution layers with hidden dimension 300. The predictor  $f$  is a two-layer MLP with hidden dimension 300 and ReLU activation. The model is trained with Adam optimizer for 30 epochs with batch size 50 and learning rate  $\eta$  linearly annealed from  $10^{-3}$  to  $10^{-4}$ .

For RGM, we explore  $\lambda \in \{0.01, 0.1\}$  for each dataset. For SRGM, we explore  $\lambda_g \in \{0.1, 1\}$  for the classification datasets while  $\lambda_g \in \{0.01, 0.1\}$  for the QM9 dataset as  $\lambda_g = 1$  causes gradient explosion. For DANN and CDAN, its hyperparameter is the weight of the adversarial domain classifier  $\lambda_d \in \{0.1, 1\}$ . For MLDG, its hyperparameter is  $\beta \in \{0.1, 1\}$ . Its inner optimization also uses Adam optimizer with default learning rate  $\alpha = 10^{-3}$ . For IRM, its hyperparameter is the weight of its gradient penalty term  $\gamma \in \{0.1, 1\}$ . The suggested value  $\gamma = 100$  results in severe gradient explosion problem. For CrossGrad, its hyperparameter is the weight of the domain classifier  $\lambda_d \in \{0.1, 1\}$ .

**Scaffold Classification** The scaffold classifier is trained by negative sampling since scaffolds are structured objects. Specifically, for each molecule  $x_i$  in a minibatch  $B$ , the negative samples are the scaffolds  $\{s_k\}$  of other molecules in the minibatch. The probability that  $x_i$  is mapped to its correct scaffold  $s_i$  is then defined as

$$p(s_i | x_i, B) = \frac{\exp\{g(\phi(x_i))^{\top} g(\phi(s_i))\}}{\sum_{k \in B} \exp\{g(\phi(x_i))^{\top} g(\phi(s_k))\}} \quad (26)$$

<sup>2</sup>[https://wiki.nci.nih.gov/download/attachments/158204006/aids\\_ec50\\_may04.txt?version=1&modificationDate=1378736563000&api=v2](https://wiki.nci.nih.gov/download/attachments/158204006/aids_ec50_may04.txt?version=1&modificationDate=1378736563000&api=v2)

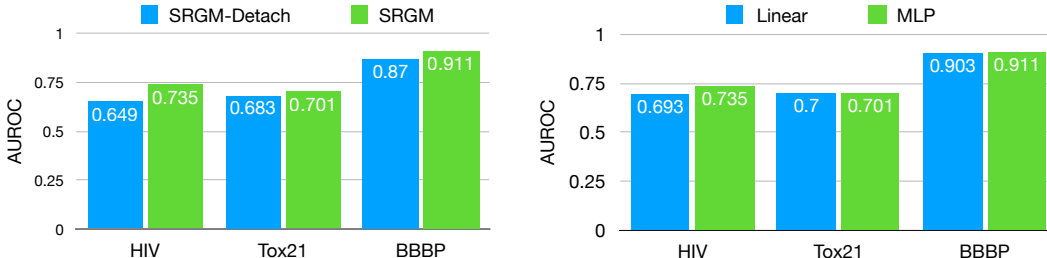


Figure 5: Ablation study of SRGM. *Left*: SRGM performs better than SRGM-detach which does not update  $\phi$  to optimize the scaffold classification loss  $\mathcal{L}_g$ . *Right*: SRGM performs better than when the scaffold classifier is a MLP instead of a linear layer.

The scaffold classification loss is  $-\sum_i \log p(s_i | x_i, B)$  for a minibatch  $B$ . We choose the classifier  $g$  to be a two-layer MLP with hidden dimension 300 and ReLU activation. As shown in Figure 5, the two-layer MLP performs better than a simple linear function across multiple tasks.

## B.2 PROTEIN MODELING

**Data** The homology and stability dataset are downloaded from Rao et al. (2019). The size of each training environment, validation and test set are listed in Table 3.

**Model hyperparameters** For both tasks, our protein encoder is a pre-trained BERT (Rao et al., 2019). The predictor is a linear layer and the superfamily/topology classifier is a two-layer MLP whose hidden layer dimension is 768. The model is fine-tuned with an Adam optimizer with learning rate  $10^{-4}$  and linear warm up schedule. The batch size is 16 and 20 for the homology and stability task. For RGM and SRGM, we explore  $\lambda \in \{0.01, 0.1\}$  and  $\lambda_g \in \{0.1, 1\}$  respectively. For the other baselines, please refer to section B.1.

## B.3 ADDITIONAL ABLATION STUDY

In section 3.2, we mentioned that the feature extractor  $\phi$  is updated to optimize the scaffold classification loss  $\mathcal{L}_g$ . To study the effect of this design choice, we experiment with a variant of SRGM called SRGM-detach, in which  $\phi$  is not updated to optimize the scaffold classification loss. As shown in Figure 5, the performance of SRGM-detach is worse than SRGM in general. This is because the scaffold classifier performs much better in SRGM and the gradient  $\delta(x)$  clearly corresponds to the change of scaffold information.