Supplementary Material for Domain Generalisation via Risk Distribution Matching

Toan Nguyen, Kien Do, Bao Duong, Thin Nguyen
Applied Artificial Intelligence Institute, Deakin University, Australia
{s222165627, k.do, duongng, thin.nguyen}@deakin.edu.au

In this supplementary material, we first provide a detailed proof for our theorem on distributional variance, as outlined in Section 1. Next, in Section 2, we detail more about our experimental settings, covering both the ColoredMNIST synthetic dataset [1] and the extensive benchmarks from the DomainBed suite [5] in the main text. Additional ablation studies and discussions on our proposed method are given in Section 3. Finally, Section 4 provides domain-specific out-of-domain accuracies for each dataset within the DomainBed suite.

1. Theoretical Results

We provide the proof for the theorem on distributional variance discussed in the main paper. We revisit the concept of kernel mean embedding [11] to express the risk distribution $\mathcal{T}_e$ of domain e. Particularly, we represent $\mathcal{T}_e$ through its embedding, $\mu_{\mathcal{T}_e}$, in a reproducing kernel Hilbert space (RKHS) denoted as $\mathcal{H}$. This is achieved by using a feature map $\phi : \mathbb{R} \to \mathcal{H}$ below:

$$\mu_{\mathcal{T}_e} \coloneqq \mathbb{E}_{R_e \sim \mathcal{T}_e} \left[\phi \left(R_e \right) \right] \tag{1}$$

$$= \mathbb{E}_{R_e \sim \mathcal{T}_e} \left[k \left(R_e, \cdot \right) \right] \tag{2}$$

where a kernel function $k \left(\cdot, \cdot \right) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is introduced to bypass the explicit specification of ϕ.

Theorem. *[7] Denote $\mathcal{T} = \frac{1}{m} \sum_{e=1}^{m} \mathcal{T}_e$ the probability distribution over the risks of all samples in the entire training set, or equivalently, the set of all m domains. Given the distributional variance $\mathbb{V}_{\mathcal{H}} \left(\{ \mathcal{T}_1, ..., \mathcal{T}_m \} \right)$ is calculated with a characteristic kernel k, $\mathbb{V}_{\mathcal{H}} \left(\{ \mathcal{T}_1, ..., \mathcal{T}_m \} \right) = 0$ if and only if $\mathcal{T}_1 = ... = \mathcal{T}_m \left(= \mathcal{T} \right)$.*

Proof. In our methodology, we employ the RBF kernel, which is *characteristic* in nature. As a result, the term $\| \mu_{\mathcal{T}_e} - \mu_{\mathcal{T}} \|_{\mathcal{H}}^2$ acts as a metric within the Hilbert space $\mathcal{H}$ [7]. Importantly, this metric reaches zero if and only if $\left(\mathcal{T}_e = \mathcal{T} \right)$ [12]. Let's consider the distributional variance, $\mathbb{V}_{\mathcal{H}} \left(\{ \mathcal{T}_1, ..., \mathcal{T}_m \} \right)$, which is defined below:

$$\mathbb{V}_{\mathcal{H}} = \frac{1}{m} \sum_{e=1}^{m} \| \mu_{\mathcal{T}_e} - \mu_{\mathcal{T}} \|_{\mathcal{H}}^2 \tag{3}$$

This variance becomes zero if and only if $\| \mu_{\mathcal{T}_e} - \mu_{\mathcal{T}} \|_{\mathcal{H}}^2 = 0$ for each e. This logically implies that $\left(\mathcal{T}_e = \mathcal{T} \right)$ for all e, leading to $\left(\mathcal{T}_1 = \mathcal{T}_2 = ... = \mathcal{T}_m \right)$.

Conversely, we assume that $\left(\mathcal{T}_1 = \mathcal{T}_2 = ... = \mathcal{T}_m \right)$. Given this condition, for any e, it follows that:

$$\mu_{\mathcal{T}} = \frac{1}{m} \sum_{e=1}^{m} \mu_{\mathcal{T}_e} = \mu_{\mathcal{T}_e} \tag{4}$$

which implies

$$\| \mu_{\mathcal{T}_e} - \mu_{\mathcal{T}} \|_{\mathcal{H}}^2 = 0. \tag{5}$$

Consequently, by the given definition of distributional variance, we have: $\mathbb{V}_{\mathcal{H}} \left(\{ \mathcal{T}_1, ..., \mathcal{T}_m \} \right) = \frac{1}{m} \sum_{e=1}^{m} \| \mu_{\mathcal{T}_e} - \mu_{\mathcal{T}} \|_{\mathcal{H}}^2 = 0$. This completes the proof. □

2. More implementation details

For our experiments, we leveraged the PyTorch DomainBed toolbox [3, 5] and utilised an Ubuntu 20.4 server outfitted with a 36-core CPU, 767GB RAM, and NVIDIA V100 32GB GPUs. The software stack included Python 3.11.2, PyTorch 1.7.1, Torchvision 0.8.2, and Cuda 12.0. Additional implementation details, beyond the hyperparameters discussed in the main text, are elaborated below.

2.1. ColoredMNIST

In alignment with [3], we performed experiments on the ColoredMNIST dataset, the results of which are detailed in Table 1 in the main paper. We partitioned the original MNIST training dataset into distinct training and validation sets of 25,000 and 5,000 samples for each of two training domains, respectively. The original MNIST test set was adapted to function as our test set. Particularly, we synthesised this test set to introduce a distribution shift: red digits

have only a 10% probability of being classified as "zero", compared to 80% and 90% in the training sets for different domains. Besides the hyper-parameters highlighted in the main paper, we also leveraged a cosine annealing scheduler to further optimise the training process like other baselines.

For our RDM method, we constrained the alignment to focus only on *the first two empirical moments (mean and variance)* of $\mathcal{T}_w$ and $\mathcal{T}$. We experimented with five different penalty weight values for λ in the range of $\{500, 1000, 2500, 5000, 10000\}$, running each experiment ten times and varying λ. The reported results are the average accuracies and their standard deviations over these 10 runs, all measured on a test-domain test set. We adhered to test-domain validation for model selection across all methods, as recommended by [5]. We reference results for other methods from [3].

2.2. DomainBed

2.2.1 Description of benchmarks

For our evaluations, we leveraged five large-scale benchmark datasets from the DomainBed suite [5], comprising:

- VLCS [4]: The dataset encompasses four photographic domains: Caltech101, LabelMe, SUN09, VOC2007. It contains 10,729 examples, each with dimensions $(3, 224, 224)$, and spans five distinct classes.
- PACS [6]: The dataset includes 9,991 images from four different domains: Photo (P), Art-painting (A), Cartoon (C), and Sketch (S). These domains each have their own unique style, making this dataset particularly challenging for out-of-distribution (OOD) generalisation. Each domain has seven classes.
- OfficeHome [13]: The dataset features 15,500 images of objects commonly found in office and home settings, categorised into 65 classes. These images are sourced from four distinct domains: Art (A), Clipart (C), Product (P), and Real-world (R).
- TerraIncognita [2]: The dataset includes 24,788 camera-trap photographs of wild animals captured at locations $\{\text{L100}, \text{L38}, \text{L43}, \text{L46}\}$. Each image has dimensions $(3, 224, 224)$ and falls into one of 10 distinct classes.
- DomainNet [8]: The largest dataset in DomainBed, DomainNet, contains 586,575 examples in dimensions $(3, 224, 224)$, spread across six domains $\{\text{clipart}, \text{infograph}, \text{painting}, \text{quickdraw}, \text{real}, \text{sketch}\}$ and encompassing 345 classes.

2.2.2 Our implementation details

To ensure rigorous evaluation and a fair comparison with existing baselines [3, 9], we conducted experiments on five datasets from the DomainBed suite, the results of which are elaborated in Table 2 of the main text. In alignment with standard practices, we optimised hyper-parameters for each domain through a randomised search across 20 trials on the validation set, utilising a joint distribution as specified in Table 1. The dataset from each domain was partitioned into an 80% split for training and testing, and a 20% split for hyper-parameter validation. A comprehensive discussion on the hyper-parameters used in our experiments is provided below. For each domain, we performed our experiments ten times, employing varied seed values and hyper-parameters within the specified range, and reported the averaged results with their standard deviations. We reference results for other methods from [3, 9]. We kindly refer readers to our given source code for more detail.

In our methodology, we employed the MMD distance for aligning risk distributions $\mathcal{T}_e$ and $\mathcal{T}$, as described in Section 4 of the main text. Utilising the RBF kernel, we compute the average MMD distance across an expansive bandwidth spectrum $\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000\}$, bypassing the need for tuning this parameter.

Inspired by recent insights [3], we incorporated an initial pre-training-with-ERM phase to further improve the OOD performance. DomainNet, given its scale, requires longer ERM pre-training; specific parameters for all datasets are provided in Table 1. Our initial learning rate lies within [1e-4.5, 1e-4], which adapts to [8e-6, 2e-5] post-ERM pre-training. Incorporating additional variance regularisation on $\mathcal{T}_e$ and $\mathcal{T}$ proves beneficial for the PACS and VLCS datasets. This approach constrains the induced risks to fall within narrower, more optimal value ranges, facilitating more effective risk distribution alignment. Optimal regularisation coefficients for this strategy are detailed in Table 1.

We maintain minimal dropout and weight decay, reserving our focus for risk distribution alignment. Optimal batch sizes differ: $[70, 100]$ for VLCS and OfficeHome, and $[30, 60]$ for TerraIncognita and DomainNet. Despite computational constraints limiting our ability to test larger batch sizes, the selected ranges yield robust performance across datasets.

Regarding the matching coefficient λ in our objective, most datasets work well within $[0.1, 10.0]$, but DomainNet prefers a narrower $[0.1, 1.0]$ range. This fine-tuning is key, especially for large-scale datasets, to balance risk reduction and cross-domain alignment in the early training stages.

3. More ablation studies and analyses

3.1. Efficacy of DG matching methods

Table 2 compares the efficiency and effectiveness of various methods: Fish, CORAL, RDM with $\mathcal{L}_{\text{RDM}}$, and RDM with $\hat{\mathcal{L}}_{\text{RDM}}$ across several benchmarks - PACS, VLCS, Of-

Parameter	Dataset	Default value	Random distribution
steps	All	5,000	5,000
learning rate	All	5e-5	$10^{\text{Uniform}(-4.5,\,-4)}$
dropout	All	0	$\text{RandomChoice}([0, 0.003, 0.03])$
weight decay	All	0	$10^{\text{Uniform}(-8,\,-5)}$
batch size	PACS / VLCS / OfficeHome	88	$\text{Uniform}(70, 100)$
	TerraIncognita / DomainNet	40	$\text{Uniform}(30, 60)$
matching coefficient λ	All except DomainNet	5.0	$\text{Uniform}(0.1, 10.0)$
	DomainNet	0.5	$\text{Uniform}(0.1, 1.0)$
pre-trained iterations	All except DomainNet	1500	$\text{Uniform}(800, 2700)$
	DomainNet	2400	$\text{Uniform}(1500, 3000)$
learning rate after pre-training	All	1.5e-5	Uniform(8e-6, 2e-5)
variance regularisation coefficient	PACS / VLCS	0.004	$\text{Uniform}(0.001, 0.007)$
	OfficeHome / TerraIncognita / DomainNet	0	0

Table 1. Hyper-parameters, along with their default values and distributions, are optimised through random search across the five benchmark datasets.

Algorithm	Training (s)	Mem (GiB)	Acc (%)
Fish	7,566	**7.97**	85.5
CORAL	4,485	21.81	86.2
RDM with $\mathcal{L}_{\text{RDM}}$	4,783	21.87	86.6
RDM with $\hat{\mathcal{L}}_{\text{RDM}}$	**4,214**	21.71	**87.2**

(a) PACS

Algorithm	Training (s)	Mem (GiB)	Acc (%)
Fish	13,493	**7.97**	77.8
CORAL	6,329	21.81	**78.8**
RDM with $\mathcal{L}_{\text{RDM}}$	9,441	21.87	77.8
RDM with $\hat{\mathcal{L}}_{\text{RDM}}$	**6,151**	21.71	78.4

(b) VLCS

Algorithm	Training (s)	Mem (GiB)	Acc (%)
Fish	9,035	**7.97**	68.6
CORAL	4,762	21.81	**68.7**
RDM with $\mathcal{L}_{\text{RDM}}$	5,467	21.87	67.0
RDM with $\hat{\mathcal{L}}_{\text{RDM}}$	**4,588**	21.71	67.3

(c) OfficeHome

Algorithm	Training (s)	Mem (GiB)	Acc (%)
Fish	6,019	**4.08**	45.1
CORAL	2,973	10.21	**47.6**
RDM with $\mathcal{L}_{\text{RDM}}$	4,040	10.17	47.1
RDM with $\hat{\mathcal{L}}_{\text{RDM}}$	**2,697**	10.11	47.5

(d) TerraIncognita

Table 2. Comparison between Fish, CORAL, and two variants of our method in terms of the training time (seconds), memory usage per iteration (GiB) and accuracy (%) on PACS, VLCS, OfficeHome and TerraIncognita.

ficeHome, and TerraIncognita. Notably, the approximate variant, denoted as RDM with $\hat{\mathcal{L}}_{\text{RDM}}$, stands out for its exceptional performance. This version emphasises the alignment of risk distribution for the worst-case domain and exhibits both faster training times and improved accuracy over its counterpart that optimises distributional variance, RDM with $\mathcal{L}_{\text{RDM}}$. For instance, on the VLCS dataset, this variant is trained in under an hour while achieving a 0.6% accuracy boost.

When compared to the gradient-matching Fish method, our approach demonstrates similar advantages but requires additional memory to store MMD distance values. The memory constraint is not unique to our method; CORAL also encounters this limitation. However, RDM outper-

forms CORAL in both training time and memory usage, especially evident on large-scale datasets like DomainNet. This efficiency gain is noteworthy, given that CORAL's increased computational requirements arise from its handling of high-dimensional representation vectors.

In terms of the accuracy, as confirmed by our main text, RDM outperforms CORAL substantially on both PACS and DomainNet, while maintaining competitiveness on TerraIncognita and VLCS. On OfficeHome, although RDM lags behind CORAL, we provide an in-depth explanation for this behavior both in the main text and in the subsequent section.

3.2. Decreased Performance on OfficeHome

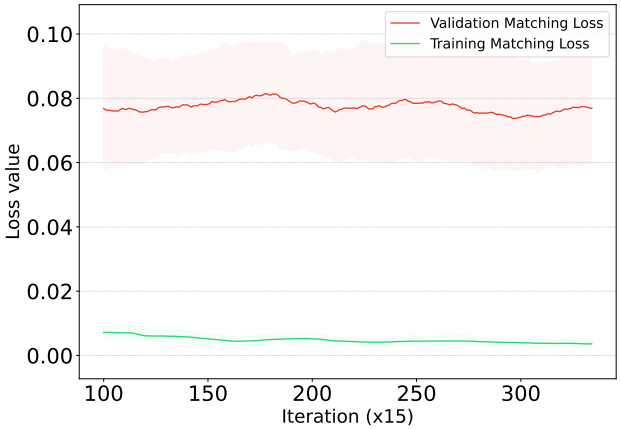


Figure 1. The notable gap between training and validation matching loss on the OfficeHome dataset, excluding the OOD Art domain. Analysis begins after RDM completes 1,500 pre-training iterations via ERM. Metrics recorded at every 15-iteration interval.

In our evaluation, RDM generally surpasses competing matching methods in OOD settings but faces challenges in specific datasets like OfficeHome. The dataset's limitations are noteworthy: with an average of only 240 samples per class, OfficeHome has significantly fewer instances per class than other datasets, which usually have at least 1,400. This limited sample size may constrain the model from learning sufficiently class-semantic features or diverse risk distributions, leading to overfitting on the training set. To shed light on this issue, we present a visual analysis in Figure 1. Starting from the 100th iteration, when we perform the task of matching risk distributions, we note that the training matching loss is already minimal, forming a clear divergence with the validation matching loss. While the training loss continues to converge to minimal values, the validation loss remains inconsistent throughout the training phase. This inconsistency showcases that the limited diversity in OfficeHome's risk distributions may induce the model's overfitting on training samples, reducing its generalisation capabilities. Despite these constraints, our method still outperforms other well-known baselines, such as MLDG, VREx, and ERM, on OfficeHome.

3.3. Impact of batch size and matching coefficient

In our analysis, we closely examine how batch size and the matching coefficient λ affect RDM's performance across four benchmark datasets: VLCS, OfficeHome, TerraIncognita, and DomainNet. Consistent with our main text findings on PACS, Figure 2 shows that using larger batch sizes enhances the model's generalisation by facilitating accurate risk distribution matching. Similarly, Figure 3 highlights the importance of λ in improving OOD performance; as λ increases, OOD performance generally improves.

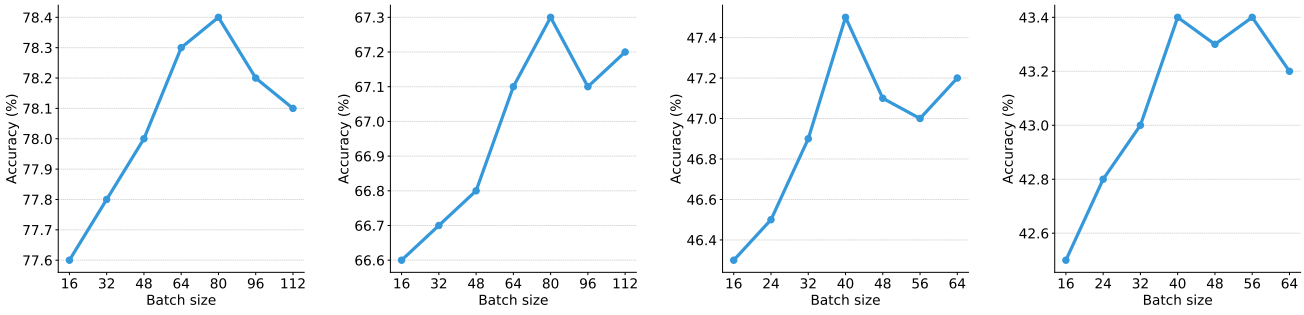
We find optimal batch size ranges for each dataset: VLCS and OfficeHome perform best with sizes between $[70, 100]$, while the larger datasets of TerraIncognita and DomainNet benefit from a more limited range of $[30, 60]$. Even with computational limitations, these batch sizes lead to strong performance. For most datasets, a λ value between $[0.1, 10.0]$ is effective. In the case of DomainNet, a smaller λ range of $[0.1, 1.0]$ works well, balancing the reduction of training risks and the alignment of risk distributions across domains. This is particularly important for large-scale datasets where reducing training risks is crucial for learning predictive features, especially during the initial phases of training.

3.4. Risk distributions

We present visualisations of risk distribution histograms accompanied by their KDE curves for two datasets, PACS and DomainNet, in Figures 4 and 5, respectively. These visualisations compare the risk distributions of ERM and our proposed RDM method on the validation sets. Both figures confirm our hypothesis that variations in training domains lead to distinct risk distributions, making them valuable indicators of *domain differences*.

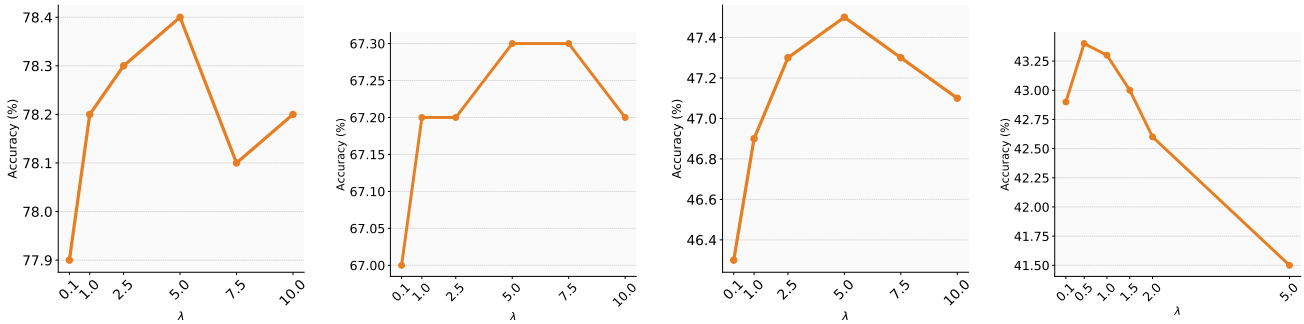
On PACS, we observe that ERM tends to capture domain-specific features, resulting in low risks within the training domains. However, ERM's substantial deviation of the average risk for the test domain from that for the training domains suggests sub-optimal OOD generalisation. In contrast, our RDM approach prioritises stable, domain-invariant features, yielding more *consistent risk distributions* and enhanced generalisation. This trend holds across both two datasets, as our approach consistently aligns risk distributions across domains better than ERM. This alignment effectively narrows the gap between test and training domains, especially *reducing risks for test domains*.

These findings underscore the efficacy of our RDM method in mitigating domain variations by aligning risk distributions, ultimately leading to enhanced generalisation.



(a) VLCS (b) OfficeHome (c) TerraIncognita (d) DomainNet

Figure 2. The influence of batch size in our method on VLCS, OfficeHome, TerraIncognita and DomainNet.



(a) VLCS (b) OfficeHome (c) TerraIncognita (d) DomainNet

Figure 3. The influence of matching coefficient λ in our method on VLCS, OfficeHome, TerraIncognita and DomainNet.

4. More experimental results

We provide domain-specific out-of-domain accuracies for each dataset within the DomainBed suite in Tables 3, 4, 5, 6, 7. In each table, the accuracy listed in each column represents the out-of-domain performance when that specific domain is excluded from the training set and used solely for testing within the respective dataset. We note that the per-domain results for Fish [10] are not available.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, pages 456–473, 2018. 2

[3] Cian Eastwood, Alexander Robey, Shashank Singh, Julius Von Kügelgen, Hamed Hassani, George J Pappas, and Bernhard Schölkopf. Probable domain generalization via quantile risk minimization. *NeurIPS*, 35:17340–17358, 2022. 1, 2, 7, 8

[4] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *ICCV*, pages 1657–1664, 2013. 2

[5] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021. 1, 2

[6] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pages 5542–5550, 2017. 2

[7] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, pages 10–18, 2013. 1

[8] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 2

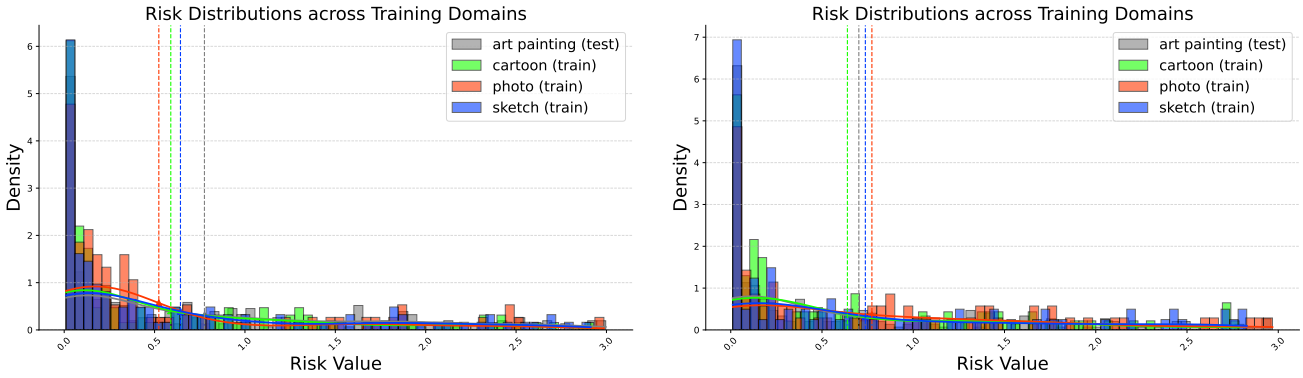
[9] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *ICML*, pages 18347–18377, 2022. 2

[10] Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *ICLR*, 2022. 5, 7, 8

[11] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International conference on algorithmic learning theory*, pages 13–31, 2007. 1

[12] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *JMLR*, 11:1517–1561, 2010. 1

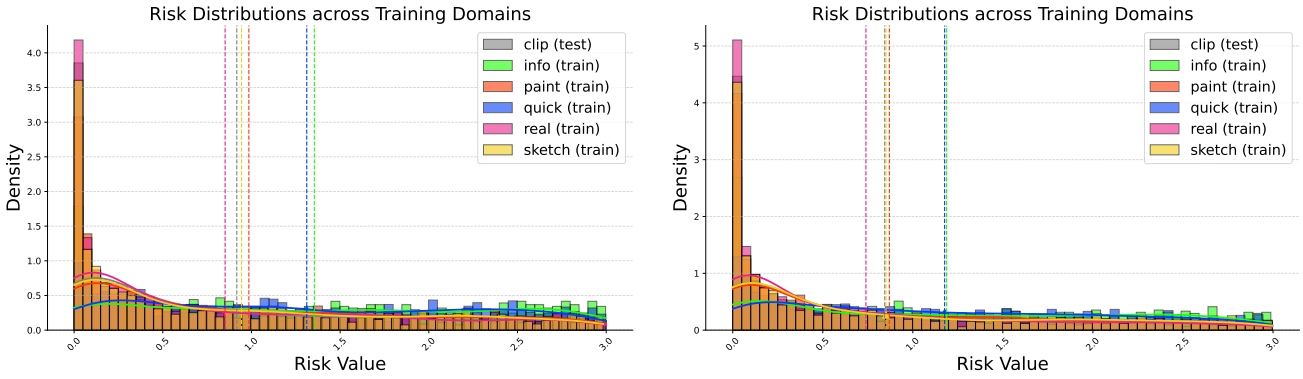
[13] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for



(a) ERM's histogram

(b) RDM's histogram

Figure 4. Histograms with their KDE curves depicting the risk distributions of ERM and our RDM method across four domains on PACS. Vertical ticks denote the mean values of all distributions.



(a) ERM's histogram

(b) RDM's histogram

Figure 5. Histograms with their KDE curves depicting the risk distributions of ERM and our RDM method across six domains on DomainNet. Vertical ticks denote the mean values of all distributions.

unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017. 2

Algorithm	clip	info	paint	quick	real	sketch	Avg
ERM	58.1 ± 0.3	18.8 ± 0.3	46.7 ± 0.3	12.2 ± 0.4	59.6 ± 0.1	49.8 ± 0.4	40.9
Mixup	55.7 ± 0.3	18.5 ± 0.5	44.3 ± 0.5	12.5 ± 0.4	55.8 ± 0.3	48.2 ± 0.5	39.2
MLDG	59.1 ± 0.2	19.1 ± 0.3	45.8 ± 0.7	13.4 ± 0.3	59.6 ± 0.2	50.2 ± 0.4	41.2
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	33.8 ± 0.5	9.3 ± 0.3	51.6 ± 0.4	40.1 ± 0.6	33.3
IRM	48.5 ± 2.8	15.0 ± 1.5	38.3 ± 4.3	10.9 ± 0.5	48.2 ± 5.2	42.3 ± 3.1	33.9
VREx	47.3 ± 3.5	16.0 ± 1.5	35.8 ± 4.6	10.9 ± 0.3	49.6 ± 4.9	42.0 ± 3.0	33.6
EQRM	56.1 ± 1.3	19.6 ± 0.1	46.3 ± 1.5	12.9 ± 0.3	61.1 ± 0.0	50.3 ± 0.1	41.0
Fish	-	-	-	-	-	-	42.7
Fishr	58.2 ± 0.5	20.2 ± 0.2	47.7 ± 0.3	12.7 ± 0.2	60.3 ± 0.2	50.8 ± 0.1	41.7
CORAL	59.2 ± 0.1	19.7 ± 0.2	46.6 ± 0.3	13.4 ± 0.4	59.8 ± 0.2	50.1 ± 0.6	41.5
MMD	32.1 ± 13.3	11.0 ± 4.6	26.8 ± 11.3	8.7 ± 2.1	32.7 ± 13.8	28.9 ± 11.9	23.4
RDM (*ours*)	**62.1 ± 0.2**	**20.7 ± 0.1**	**49.2 ± 0.4**	**14.1 ± 0.4**	**63.0 ± 1.3**	**51.4 ± 0.1**	**43.4**

Table 3. Domain-specific out-of-domain accuracy on DomainNet where the best results are marked as bold. Results of other methods are referenced from [3, 10].

Algorithm	A	C	P	S	Avg
ERM	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
Mixup	86.1 ± 0.5	78.9 ± 0.8	**97.6 ± 0.1**	75.8 ± 1.8	84.6
MLDG	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9
GroupDRO	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
VREx	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
EQRM	86.5 ± 0.4	**82.1 ± 0.7**	96.6 ± 0.2	80.8 ± 0.2	86.5
Fish	-	-	-	-	85.5
Fishr	**88.4 ± 0.2**	78.7± 0.7	97.0 ± 0.1	77.8 ± 2.0	85.5
CORAL	88.3 ± 0.2	80.0 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2
MMD	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6
RDM (*ours*)	**88.4 ± 0.2**	81.3 ± 1.6	97.1 ± 0.1	**81.8 ± 1.1**	**87.2**

Table 4. Domain-specific out-of-domain accuracy on PACS where the best results are marked as bold. Results of other methods are referenced from [3, 10].

Algorithm	C	L	S	V	Avg
ERM	97.7 ± 0.4	64.3 ± 0.9	73.4 ± 0.5	74.6 ± 1.3	77.5
Mixup	98.3 ± 0.6	64.8 ± 1.0	72.1 ± 0.5	74.3 ± 0.8	77.4
MLDG	97.4 ± 0.2	65.2 ± 0.7	71.0 ± 1.4	75.3 ± 1.0	77.2
GroupDRO	97.3 ± 0.3	63.4 ± 0.9	69.5 ± 0.8	76.7 ± 0.7	76.7
IRM	98.6 ± 0.1	64.9 ± 0.9	73.4 ± 0.6	77.3 ± 0.9	78.5
VREx	98.4 ± 0.3	64.4 ± 1.4	**74.1 ± 0.4**	76.2 ± 1.3	78.3
EQRM	98.3 ± 0.0	63.7 ± 0.8	72.6 ± 1.0	76.7 ± 1.1	77.8
Fish	-	-	-	-	77.8
Fishr	**98.9 ± 0.3**	64.0 ± 0.5	71.5 ± 0.2	76.8 ± 0.7	77.8
CORAL	98.3 ± 0.1	**66.1 ± 1.2**	73.4 ± 0.3	77.5 ± 1.2	**78.8**
MMD	97.7 ± 0.1	64.0 ± 1.1	72.8 ± 0.2	75.3 ± 3.3	77.5
RDM (*ours*)	98.1 ± 0.2	64.9 ± 0.7	72.6 ± 0.5	**77.9 ± 1.2**	78.4

Table 5. Domain-specific out-of-domain accuracy on VLCS where the best results are marked as bold. Results of other methods are referenced from [3, 10].

Algorithm	A	C	P	R	Avg
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
Mixup	62.4 ± 0.8	54.8 ± 0.6	**76.9 ± 0.3**	78.3 ± 0.2	68.1
MLDG	61.5 ± 0.9	53.2 ± 0.6	75.0 ± 1.2	77.5 ± 0.4	66.8
GroupDRO	60.4 ± 0.7	52.7 ± 1.0	75.0 ± 0.7	76.0 ± 0.7	66.0
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3
VREx	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4
EQRM	60.5 ± 0.1	**56.0 ± 0.2**	76.1 ± 0.4	77.4 ± 0.3	67.5
Fish	-	-	-	-	68.6
Fishr	62.4 ± 0.5	54.4 ± 0.4	76.2 ± 0.5	78.3 ± 0.1	67.8
CORAL	**65.3 ± 0.4**	54.4 ± 0.5	76.5 ± 0.1	**78.4 ± 0.5**	**68.7**
MMD	60.4 ± 0.2	53.3 ± 0.3	74.3 ± 0.1	77.4 ± 0.6	66.3
RDM (*ours*)	61.1 ± 0.4	55.1 ± 0.3	75.7 ± 0.5	77.3 ± 0.3	67.3

Table 6. Domain-specific out-of-domain accuracy on OfficeHome where the best results are marked as bold. Results of other methods are referenced from [3, 10].

Algorithm	L100	L38	L43	L46	Avg
ERM	49.8 ± 4.4	42.1 ± 1.4	56.9 ± 1.8	35.7 ± 3.9	46.1
Mixup	**59.6 ± 2.0**	42.2 ± 1.4	55.9 ± 0.8	33.9 ± 1.4	**47.9**
MLDG	54.2 ± 3.0	44.3 ± 1.1	55.6 ± 0.3	36.9 ± 2.2	47.7
GroupDRO	41.2 ± 0.7	38.6 ± 2.1	56.7 ± 0.9	36.4 ± 2.1	43.2
IRM	54.6 ± 1.3	39.8 ± 1.9	56.2 ± 1.8	39.6 ± 0.8	47.6
VREx	48.2 ± 4.3	41.7 ± 1.3	56.8 ± 0.8	38.7 ± 3.1	46.4
EQRM	47.9 ± 1.9	**45.2 ± 0.3**	**59.1 ± 0.3**	38.8 ± 0.6	47.8
Fish	-	-	-	-	45.1
Fishr	50.2 ± 3.9	43.9 ± 0.8	55.7 ± 2.2	**39.8 ± 1.0**	47.4
CORAL	51.6 ± 2.4	42.2 ± 1.0	57.0 ± 1.0	**39.8 ± 2.9**	47.6
MMD	41.9 ± 3.0	34.8 ± 1.0	57.0 ± 1.9	35.2 ± 1.8	42.2
RDM (*ours*)	52.9 ± 1.2	43.1 ± 1.0	58.1 ± 1.3	36.1 ± 2.9	47.5

Table 7. Domain-specific out-of-domain accuracy on TerraIncognita where the best results are marked as bold. Results of other methods are referenced from [3, 10].