QT-DoG: Quantization-aware Training for Domain Generalization

Saqib Javed [1] **Hieu Le** [1] **Mathieu Salzmann** [1 2]

Abstract

A key challenge in Domain Generalization (DG) is preventing overfitting to source domains, which can be mitigated by finding flatter minima in the loss landscape. In this work, we propose Quantization-aware Training for Domain Generalization (QT-DoG) and demonstrate that weight quantization effectively leads to flatter minima in the loss landscape, thereby enhancing domain generalization. Unlike traditional quantization methods focused on model compression, QT-DoG exploits quantization as an implicit regularizer by inducing noise in model weights, guiding the optimization process toward flatter minima that are less sensitive to perturbations and overfitting. We provide both an analytical perspective and empirical evidence demonstrating that quantization inherently encourages flatter minima, leading to better generalization across domains. Moreover, with the benefit of reducing the model size through quantization, we demonstrate that an ensemble of multiple quantized models further yields superior accuracy than the state-of-the-art DG approaches with no computational or memory overheads. Code is released at: https://saqibjaved1.github.io/QT_DoG/.

1. Introduction

Many works have shown that deep neural networks trained under the assumption that the training and test samples are drawn from the same distribution fail to generalize in the presence of large training-testing discrepancies, such as texture (Geirhos et al., 2019; Bahng et al., 2020), background (Xiao et al., 2020), or day-to-night (Dai & Van Gool, 2018; Michaelis et al., 2019) shifts. Domain Generalization (DG) addresses this problem and aims to learn models that perform well not only in the training (source) domains but also in new, unseen (target) data distributions (Blanchard et al., 2011; Muandet et al., 2013; Zhou et al., 2022).

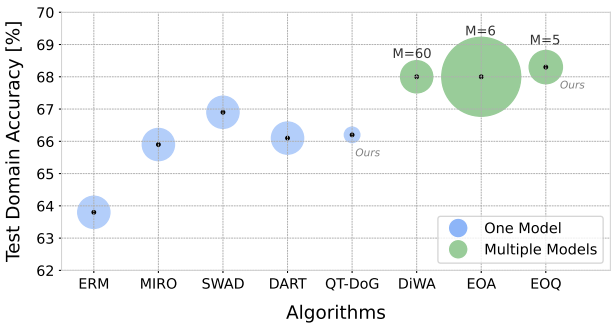

Figure 1. **Performance Comparison on the Domainbed Benchmark.** We show the average accuracy on 5 different datasets. *One Model* refers to methods training a single ResNet-50 model. *Multiple Models* refers to training M models for averaging or ensembling, which affects the training cost. We compare QT-DoG and EoQ to other state-of-the-art methods. The marker size is proportional to the memory footprint. EoQ shows superior performance despite being 4 times smaller than its full-precision counterpart. Additionally, QT-DoG demonstrates comparable performance to *One Model* methods, despite its significantly smaller size.

In the broader context of generalization, with training and test data drawn from the same distribution, the literature has revealed a relationship between the flatness of the loss landscape and the generalization ability of deep learning models (Keskar et al., 2017; Dziugaite & Roy, 2017; Garipov et al., 2018; Izmailov et al., 2018; Jiang et al., 2020; Foret et al., 2021; Zhang et al., 2023). This relationship has then been leveraged by many recent works, demonstrating that a flatter minimum also improves Out-of-Distribution (OOD) performance (Cha et al., 2021; Ramé et al., 2023; Arpit et al., 2022). At the heart of all these DG methods lies the idea of weight averaging (Izmailov et al., 2018), which involves averaging weights from several trained models or at various stages of the training process.

In this work, we demonstrate that flatter minima in the loss landscape can be effectively achieved through weight quantization using Quantization-aware Training (QAT), making it an effective approach for DG. By restricting the possible weight values to a lower bit precision, quantization imposes constraints on the weight space, introducing quantization noise into the network parameters. This noise, as discussed

[1]CVLab, EPFL, Switzerland. [2]Swiss Data Science Center, Switzerland.. Correspondence to: Mathieu Salzmann <mathieu.salzmann@epfl.ch>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

in prior works (An, 1996; Murray & Edwards, 1992; Goodfellow et al., 2016; Hochreiter & Schmidhuber, 1994), acts as a form of regularization that naturally encourages the optimization process to converge toward flatter minima. Furthermore, our results show that models trained with quantization not only generalize better across domains but also reduce overfitting to source domains. To the best of our knowledge, this is the first work to explicitly explore the intersection of quantization and domain generalization. Through both analytical reasoning and empirical validation, we provide strong evidence that QAT promotes flatter minima, leading to enhanced generalization performance on unseen domains.

The benefit of having fast and light-weight quantized models then further allow us to even make an ensemble of them, termed *Ensemble of Quantization* (EoQ). EoQ achieves superior performance while maintaining the computational efficiency of a single full-precision model. This stands in contrast to ensemble-based methods like (Ramé et al., 2023; Arpit et al., 2022), which require storing and running multiple full-precision models. With our approach, quantization not only improves generalization but also reduces the model's memory footprint and computational cost at inference. As shown in Figure 1, EoQ yields a model with a memory footprint similar to the state-of-the-art single-model DG approaches and much smaller than other ensemble-based methods, yet outperforms all its competitors in terms of accuracy.

Our contributions can be summarized as follows:

- We are the first to demonstrate that quantization-aware training, traditionally used for model compression, can serve as an implicit regularizer, with quantization noise enhancing domain generalization.
- We demonstrate that QAT promotes flatter minima in the loss landscape and provide an analytical perspective behind this effect. Additionally, we show that QAT stabilizes model behavior on OOD data during training.
- In contrast to traditional DG methods that often increase model size or computational cost, QT-DoG not only improves generalization but also significantly reduces the model size, enabling efficient deployment in real-world applications. EoQ, for instance, requires nearly 6 times less memory than Arpit et al. (2022) and 12 times less training compute compared to Ramé et al. (2023), which trains 60 models for diverse averaging.

2. Related Work

2.1. Domain Generalization

Numerous multi-source domain generalization (DG) methods have been proposed in the past. In this section, we review some of the recent approaches, categorizing them into different groups based on their methodologies.

2.1.1. Domain Alignment

The methods in this category focus on reducing the differences among the source domains and learn domain-invariant features (Arjovsky et al., 2019; Krueger et al., 2021; Rame et al., 2022a; Sun et al., 2016; Sagawa et al., 2020; Ganin et al., 2016; Li et al., 2023; Cheng et al., 2024). The core idea is that, if the learnt features are invariant across the different source domains, they will also be robust to the unseen target domain. For matching feature distributions across source domains, DANN (Ganin et al., 2016) uses an adversarial loss while CORAL (Sun & Saenko, 2016) and DICA (Muandet et al., 2013) seek to align latent statistics of different domains. Unfortunately, most of these methods fail to generalize well and were shown not to outperform ERM on various benchmarks (Gulrajani & Lopez-Paz, 2021; Ye et al., 2022; Koh et al., 2021).

2.1.2. Regularization

In the literature, various ways of regularizing models (implicit and explicit) have also been proposed to achieve better generalization. For example, invariant risk minimization (Arjovsky et al., 2019) relies on a regularization technique such that the learned classifier is optimal even under a distribution shift. Moreover, (Huang et al., 2020) tries to suppress the dominant features learned from the source domain and pushes the network to use other features correlating with the labels. Furthermore, (Krueger et al., 2021) proposes risk extrapolation that uses regularization to minimize the variance between domain-wise losses, considering that it is representative of the variance including the target domain.

2.1.3. Vision Transformers

Recent studies have increasingly utilized vision transformers for domain generalization (Sultana et al., 2022). Some approaches enhance vision transformers by integrating knowledge distillation (Hinton et al., 2015) and leveraging text modality from CLIP (Radford et al., 2021) to learn more domain-invariant features (Moayeri et al., 2023; Chen et al., 2024; Huang et al., 2023; Liu et al., 2024; Yu et al., 2024; Shu et al., 2023; Addepalli et al., 2024).

2.1.4. Ensembling

Ensembling of deep networks (Lakshminarayanan et al., 2017; Hansen & Salamon, 1990; Krogh & Vedelsby, 1995) is a foundational strategy and has consistently proven to be robust in the past. Many works have been proposed to train multiple diverse models and combine them to obtain better in-domain accuracy and robustness to domain shifts (Arpit et al., 2022; Thopalli et al., 2021; Mesbah et al., 2022; Li et al., 2022; Lee et al., 2022; Pagliardini et al., 2023). However, ensembles require multiple models to be stored

and a separate forward pass for each model, which increases the computational cost and memory footprint, especially if the models are large.

2.1.5. Weight Averaging

Combining or averaging weights from different training stages or models has emerged as a robust approach to improve OOD generalization (Wortsman et al., 2022b; Matena & Raffel, 2022; Wortsman et al., 2022a; Gupta et al., 2020; Choshen et al., 2022; Wortsman et al., 2021; Maddox et al., 2019; Benton et al., 2021; Cha et al., 2021; Jain et al., 2023; Ramé et al., 2023). Techniques like SWAD (Cha et al., 2021) leverage weight averaging to identify flat minima, reducing overfitting and enhancing generalization under distribution shifts. Similarly, DiWA (Rame et al., 2022b) combines weights from independently trained models to improve robustness through increased diversity.

Arpit et al. (2022) integrates ensembling with weight averaging, yielding superior performance compared to either method alone, albeit with significant memory and computational costs. To address these challenges, we demonstrate that quantization can improve generalization while reducing resource demands.

Although flatter minima are not universally indicative of better domain generalization (Andriushchenko et al., 2023), they remain a valuable tool for improving robustness in many scenarios. Moreover, recent findings (Mueller et al., 2023) highlight that selective application of SAM (Foret et al., 2021), such as restricting it to normalization layers, can further refine its effectiveness. The consistent empirical success of SAM underscores its reliability as a method for enhancing domain generalization, despite the nuanced relationship between flatness and performance across different settings.

2.2. Model Quantization

Model quantization is used in deep learning to reduce the memory footprint and computational requirements of deep network. In a conventional neural network, the model parameters and activations are usually stored as high-precision floating-point numbers, typically 32-bit or 64-bit. The process of model quantization entails transforming these parameters into lower bit-width representations, such as 8-bit integers or binary values. Existing techniques fall into two main categories. **Post-Training Quantization** (PTQ) quantizes a pre-trained network using a small calibration dataset and is thus relatively simple to implement (Nagel et al., 2020; Li et al., 2021; Frantar et al., 2022; Zhao et al., 2019; Cai et al., 2020; Nagel et al., 2019; Shao et al., 2024; Lin et al., 2024; Chee et al., 2023; Li* et al., 2025; Ramachandran et al., 2024; Shang et al., 2024; Zhang & Shrivastava, 2025). **Quantization-Aware Training** (QAT) retrains the network during the quantization process and thus better preserves the model's full-precision accuracy. Yang et al. (2023); Esser et al. (2020); Zhou et al. (2017); Bhalgat et al. (2020); Yamamoto (2021); Yao et al. (2020); Shin et al. (2023). In the next section, we provide some background on quantization and on the method we will use in our approach. Our goal in this work is not to introduce a new quantization strategy but rather to demonstrate the impact of quantization on generalization.

3. Domain Generalization by Quantization

We build our method on the simple ERM approach to showcase the effects of quantization on the training process and on the generalization to unseen data from a different domain. Despite the simplicity of this approach, we will show in Section 4.1 that it yields a significant accuracy boost on the test data from the unseen target domain. Furthermore, it stabilizes the behavior of the model on OOD data during training, making it similar to that on the in-domain data. In the remainder of this section, we focus on providing some insights on how quantization enhances DG.

3.1. Quantization

Let w be a single model weight to be quantized, s the quantizer step size, and Q_N and Q_P the number of negative and positive quantization levels, respectively. We define the quantization process that computes $\bar{w}$, a quantized and integer scaled representation of the weights, as

$$\bar{w} = \lfloor clip(w/s, -Q_N, Q_P) \rceil, \tag{1}$$

where the function $clip(k, r_1, r_2)$ is defined as

$$clip(k, r_1, r_2) = \begin{cases} \lfloor k \rceil & \text{if } r_1 < k < r_2 \\ r_1 & \text{if } k \leq r_1 \\ r_2 & \text{if } k \geq r_2 \end{cases} \tag{2}$$

Here, $\lfloor k \rceil$ represents rounding k to nearest integer. If we quantize a weight to b bits, for unsigned data $Q_N = 0$ and $Q_P = 2^b - 1$, and for signed data $Q_N = 2^{b-1}$ and $Q_P = 2^{b-1} - 1$.

Note that the quantization process described in Eq. 1 yields a scaled value. A quantized representation of the data at the same scale as w can then be obtained as

$$w_q = \bar{w} \times s. \tag{3}$$

This transformation results in a discretized weight space that inherently introduces noise. We demonstrate generalization ability of QT-DoG with different quantization methods in section 4.1.3.

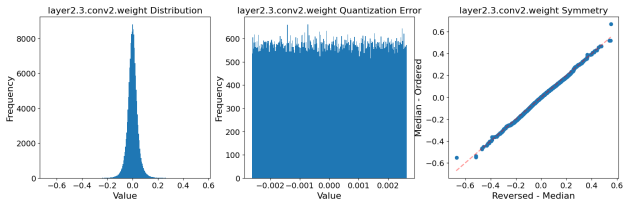

Figure 2. **Distribution of Quantization Noise and Weights**. We plot the weights(left), quantization noise(middle) and symmetry(right) of random layer in ResNet-50 model. We found KL-divergence to be **0.0009** between quantization noise and uniform distribution with same minimum and maximum value.

3.2. Quantization as uniform noise

Quantization can be modeled as additive noise under certain assumptions (Gray & Neuhoff, 1998). The noise or error introduced by the quantizer is bounded within $\left[-\frac{s}{2}, \frac{s}{2}\right]$ where s represents the quantization step size, as described in the previous section. When s is small relative to the dynamic range of the weight distribution, the quantization noise can be well approximated by a uniform distribution (Boncelet, 2009).

This observation is empirically validated in Figure 2, where we analyze the weight distribution of a randomly selected layer and apply 7-bit quantization. We then compute the Kullback-Leibler divergence (Kullback & Leibler, 1951) between the resulting quantization noise and a uniform distribution with the same minimum and maximum values, finding it to be very low. We extend this analysis in Appendix J across multiple layers, we observe that neural network weights generally tend to exhibit smooth and symmetric distributions, reinforcing the conclusion that quantization noise/error generally follows a uniform distribution. Prior works (Zhang et al., 2024; Murray & Edwards, 1992) show that adding noise to network weights can improve generalization, and similarly, quantization-aware training with structured uniform noise also enhances generalization.

3.3. Quantization Leads to Flat Minima

In the literature (Rame et al., 2022b; Arpit et al., 2022; Krueger et al., 2021; Cha et al., 2021; Rame et al., 2022b; Foret et al., 2021), it has been established that a model's generalization ability can be increased by finding a flatter minimum during training. This is the principle we exploit in our work, but from the perspective of quantization, and provide an analytical view into how it contributes to achieving flatter minima. In practice, ERM can have several solutions with similar training loss values but different generalization ability. Even when the training and test data are drawn from the same distribution, the standard optimizers, such as SGD and Adam (Kingma & Ba, 2015), often lead to sub-optimal generalization by finding sharp and narrow minima (Keskar et al., 2017; Dziugaite & Roy, 2017; Garipov et al., 2018; Izmailov et al., 2018; Jiang et al., 2020; Foret et al., 2021). This has been shown to be prevented by introducing noise in the model weights during training (An, 1996; Murray & Edwards, 1992; Goodfellow et al., 2016; Hochreiter & Schmidhuber, 1994). Here, we argue that quantization inherently induces such noise and thus helps to find flatter minima. Here, we argue that quantization inherently induces noise, which aids in finding flatter minima. To support this, we use a second-order Taylor series expansion to analyze how quantization-induced perturbations affect the curvature of the loss landscape, leading to flatter minima.

Let $\hat{y}_i = f(\boldsymbol{x}, \boldsymbol{w})$ represent the predicted output of the network f, which is parameterized by the weights $\boldsymbol{w}$. A quantized network can then be represented as

$$f(\boldsymbol{x}, \boldsymbol{w}_q) = f(\boldsymbol{x}, \boldsymbol{w} + \Delta) = \hat{y}_i{}^q,$$

where $\boldsymbol{w}_q$ denotes the quantized weights and $\hat{y}_i{}^q$ the corresponding prediction. The quantized weights can thus be thought of as introducing perturbations (Δ) to the full-precision weights, akin to noise affecting the weights.

Such noise induced by the weight quantization can also be seen as a form of regularization, akin to more traditional methods. For small perturbations, (An, 1996; Murray & Edwards, 1992; Goodfellow et al., 2016) show that this type of regularization encourages the parameters to navigate towards regions of the parameter space where small perturbations of the weights have minimal impact on the output, i.e., flatter minima.

When noise is introduced via quantization, second-order Taylor series approximation of the loss function for the perturbed weights $\boldsymbol{w} + \Delta$ can be expressed as

$$\mathcal{L}(\boldsymbol{w} + \Delta) \approx \mathcal{L}(\boldsymbol{w}) + \nabla\mathcal{L}(\boldsymbol{w})^\top \Delta + \frac{1}{2}\Delta^\top \mathcal{H} \Delta, \quad (4)$$

where $\mathcal{L}(\boldsymbol{w})$ is the loss at the original weights $\boldsymbol{w}$, $\nabla L(\boldsymbol{w})$ is the gradient of the loss at $\boldsymbol{w}$, and $\mathcal{H} = \nabla^2 L(\boldsymbol{w})$ is the Hessian matrix, which contains second-order partial derivatives of the loss function with respect to the weights, representing the curvature of the loss surface.

Eq. 4 shows how the quantization noise Δ interacts with the curvature $\mathcal{H}$ of the loss function. In regions with large curvature (sharp minima), the Hessian $\mathcal{H}$ has large eigenvalues, and even small perturbations Δ result in large increases in the loss (Dinh et al., 2017). In contrast, in flat regions (small eigenvalues of $\mathcal{H}$), the loss remains nearly unchanged for small perturbations. Quantization noise acts as an implicit regularizer by introducing perturbations Δ that disrupt the model's weight updates. In sharper minima, where the Hessian $\mathcal{H}$ eigenvalues are large, small noise significantly increases the loss, causing the model to "escape" these regions and search for flatter, more stable minima. In flatter regions, where the Hessian $\mathcal{H}$ eigenvalues are small, the noise has less impact, helping the model settle into these

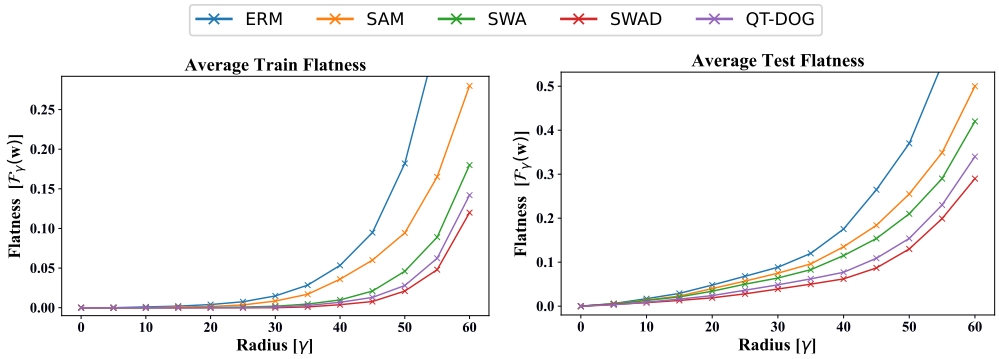

Figure 3. **Local Flatness Comparison:** We plot the average training (left) and testing (right) local flatness $\mathcal{F}\gamma(\boldsymbol{w})$ (Eq. 5) for ERM (Gulrajani & Lopez-Paz, 2021), SAM (Foret et al., 2021), SWA (Izmailov et al., 2018) and SWAD (Cha et al., 2021) by varying the radius γ on different domains of PACS. We evaluate the training flatness $\mathcal{F}\gamma^S(\boldsymbol{w})$ on the seen domains (left) and the test flatness $\mathcal{F}_\gamma^T(\boldsymbol{w})$ on the unseen domains (right).

regions with lower loss. This encourages convergence to solutions that are less sensitive to small changes in the input or model parameters, which is beneficial for out-of-distribution (OOD) generalization.

In the case of quantization-aware training, the induced noise Δ is influenced by the quantization bin width or the quantizer step size s, and thus ranges between $-\frac{s}{2}$ and $+\frac{s}{2}$. This s is directly dependent on the quantization levels or the bit-width chosen for weight quantization. As the number of bits per weight decreases, the amount of induced noise increases. Hence, the impact of the additional noise can be weighed by choosing an optimal bit-width. As will be shown in Section 4, certain bit-widths thus yield better and flatter minima that enhance generalization. However, if we induce too much noise(very low bit-precision), it introduces over-regularization. This excessive noise can overly restrict the search space, preventing the model from reaching a good solution. Instead, the optimization process may focus on minimizing the loss in a way that avoids sharp regions, but sacrifices the ability to find true minimum of the loss function. This is also evident in Table 10 in the appendix.

Moreover, Rissanen (1978); Hochreiter & Schmidhuber (1997) show that a flatter minimum corresponds to a low complexity network and requires fewer bits of information per weight. More importantly, Hochreiter & Schmidhuber (1997) demonstrates the importance of the bit-precision of the network weights and adds a regularization term in the loss function that seeks to lower the weight bit-precision to lead to flatter minima. Here, by using quantization, we are explicitly reducing bit-precision of the network weights, thus achieving the same goal.

3.4. Empirical Analysis of Quantization-aware Training and Flatness

In this section, we demonstrate that a flatter minimum is reached when incorporating quantization in the ERM process. Similar to (Dinh et al., 2017; Cha et al., 2021), we interpret flat minima as "a large connected region in weight space where the error remains approximately constant," as defined by (Hochreiter & Schmidhuber, 1997). Our loss flatness analysis shows that QT-DoG can find a flatter minimum in comparison to not only ERM but also SAM (Foret et al., 2021) and SWA (Izmailov et al., 2018).

Following the approach in Cha et al. (2021), we quantify local flatness $\mathcal{F}_\gamma(\boldsymbol{w})$ by measuring the expected change in loss values between a model with parameters $\boldsymbol{w}$ and a perturbed model with parameters $|\boldsymbol{w}'| = |\boldsymbol{w}| + \gamma$, where $\boldsymbol{w}'$ lies on a sphere of radius γ centered at $\boldsymbol{w}$.. This is expressed as

$$\mathcal{F}_\gamma(\boldsymbol{w}) = \mathbb{E}_{\|\boldsymbol{w}'\|}[\mathcal{E}(\boldsymbol{w}') - \mathcal{E}(\boldsymbol{w})], \tag{5}$$

where $\mathcal{E}(\boldsymbol{w})$ denotes the accumulated loss over the samples of potentially multiple domains.

For our analysis, we will evaluate flatness in both the source domains and the target domain, and thus $\mathcal{E}(\boldsymbol{w})$ is evaluated using either source samples or target ones accordingly.

As in Cha et al. (2021), we approximate $\mathcal{F}_\gamma(\boldsymbol{w})$ by Monte-Carlo sampling with 100 samples. In Figure 3, we compare the $\mathcal{F}_\gamma(\boldsymbol{w})$ of QT-DoG to that of ERM (Gulrajani & Lopez-Paz, 2021), SAM (Foret et al., 2021), SWA (Izmailov et al., 2018) and SWAD (Cha et al., 2021) for different radii γ. QT-DoG not only finds a flatter minimum than ERM, SAM and SWA but also yields a comparable flatness to SWAD's despite being 75% smaller in model size.

3.5. Stable Training Process

Here, we demonstrate the robustness of out-of-domain performance to model selection using the in-domain validation set. Specifically, we seek to show that accuracy on the in-domain validation data is a good measure to pick the best model for out-of-domain distribution. Therefore, we assume that during training, the model selection criterion based on

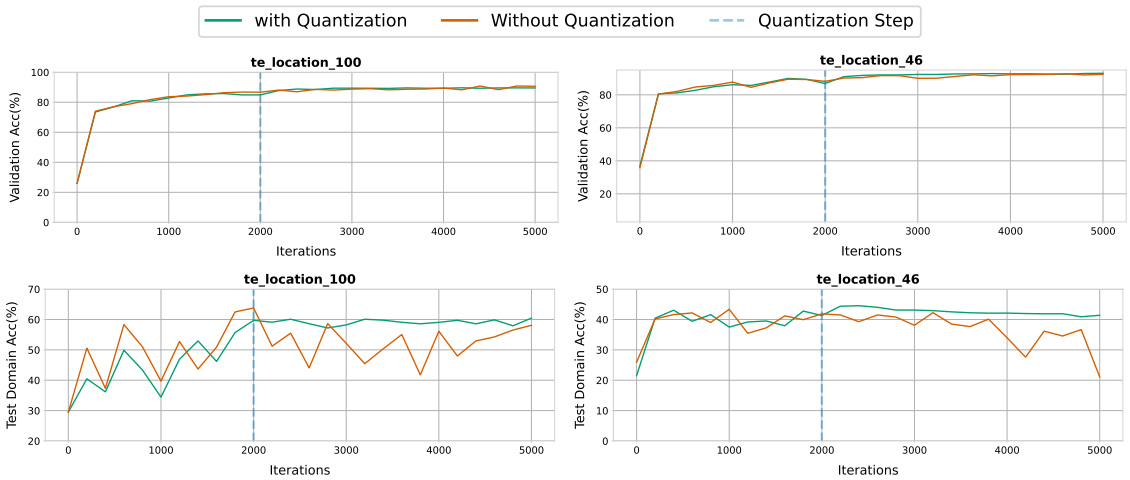

Figure 4. **Model Quantization improves out-of-domain performance as well as training *stability***. The plots were computed using the TerraInc dataset with domain L100 (left) and L46 (right) as test domain, and the other domains as training/validation data. The top two plots illustrate in-domain validation accuracy, while the bottom two represent out-of-domain test accuracy. The network used for these plots was a ResNet-50. For our quantized models, shown in blue in each plot, we quantized the model after 2000 steps. Note that the model accuracy is not only better with quantization but also much more stable for out of distribution data after the quantization step.

this validation data can select the best model for the OOD data even if the model starts to overfit. In other words, it is expected that the out-of-domain evaluation at each point of the training phase should improve or rather stay stable if the model is close to overfitting to the in-domain data. For these experiments, we use the TerraIncognita dataset (Beery et al., 2018) and consider the same number of iterations as for the DomainBed protocol (Gulrajani & Lopez-Paz, 2021).

As can be seen in Figure 4, vanilla ERM (without quantization) quickly overfits to the in-domain validation/training dataset. That is, the OOD performance is highly unstable during the whole training process. By contrast, our quantized model is much more stable. Specifically, we quantize our model at 2000 steps, and it can be seen that the model performance on out-of-domain distribution is also unstable before that. Once the model weights are quantized, we see a regularization effect and the performance becomes much more stable on the OOD data. We provide training plots encompassing different domains as target settings for the sake of completeness. This inclusion serves to illustrate that quantization genuinely enhances stability in the training process. On the left, "te_location_100" is considered as target domain while "te_location_46" is used as the target domain for the plot on the right. These experiments evidence that model selection based on the in-domain validation set is much more reliable when introducing quantization into training.

3.6. Ensembles of Quantization

For our ensemble creation, we train multiple models independently from initialization, using random seeds to ensure diversity and, incorporate quantization into the training process to obtain smaller quantized models. We refer to this as the Ensemble of Quantization (EoQ). As Breiman (1996), we use the bagging method to combine the multiple predictions. Therefore, the class predicted by EoQ for an input $\mathbf{x}$ is given by

$$\hat{y} = \arg\max_k \ \text{Softmax}\left(\frac{1}{E}\sum_{i=1}^{E} f(\mathbf{x}; \boldsymbol{w}_q^i)\right)_k \tag{6}$$

where E is the total number of models in the ensemble, $\boldsymbol{w}_q^i$ denotes the parameters of the ith quantized model, and the subscript k denotes the kth element of the vector argument. Finally, we use the in-domain validation set performance to pick the best model state (weights) $\boldsymbol{w}_q^i$ of the ith quantized model used in the ensemble.

4. Experiments

We evaluate our approach on diverse datasets from Domainbed and WILDS (Koh et al., 2021) Benchmark. All implementation, datasets, metric details and various ablation studies are provided in the Appendix.

4.1. Results

In this section, we demonstrate the superior performance of our proposed approach by comparing it to recent state-of-the-art DG methods. We also present some visual evidence for the better performance of our quantization approach. Furthermore, we show how quantization not only enhances model generalization but also yields better performance on in-domain data.

4.1.1. Comparison with DG methods

Table 1 reports out-of-domain performances on five DG benchmarks and compares our proposed approaches to prior

Table 1. **Comparison with domain generalization methods.** Performance benchmarking on 5 datasets of the DomainBed benchmark. Highest accuracy is shown in bold, while second best is underlined. † do not report confidence interval and ensembles do not have confidence interval because an ensemble uses all the models to make a prediction. Our proposed method is colored in Gray. Average accuracies and standard errors are reported from three trials. For all the reported results, we use the same training-domain validation protocol as (Gulrajani & Lopez-Paz, 2021). $Models$ indicate the number of models trained, and $Size$ represents the relative network size.

Algorithm	Models	Size	PACS	VLCS	Office	TerraInc	DomainNet	Avg.
ResNet-50 (25M Parameters, Pre-trained on ImageNet)								
ERM	1	1x	84.7 ± 0.5	77.4 ± 0.3	67.5 ± 0.5	46.2 ± 0.4	41.2 ± 0.2	63.8
IRM	1	1x	84.4 ± 1.1	78.1 ± 0.0	66.6 ± 1.0	47.9 ± 0.7	35.7 ± 1.9	62.5
Group DRO	1	1x	84.1 ± 0.4	77.2 ± 0.6	66.9 ± 0.3	47.0 ± 0.3	33.7 ± 0.2	61.8
Mixup	1	1x	84.3 ± 0.5	77.7 ± 0.4	69.0 ± 0.1	48.9 ± 0.8	39.6 ± 0.1	63.9
MLDG	1	1x	84.8 ± 0.6	77.1 ± 0.4	68.2 ± 0.1	46.1 ± 0.8	41.8 ± 0.4	63.6
CORAL	1	1x	86.0 ± 0.2	77.7 ± 0.5	68.6 ± 0.4	46.4 ± 0.8	41.8 ± 0.2	64.1
MMD	1	1x	85.0 ± 0.2	76.7 ± 0.9	67.7 ± 0.1	49.3 ± 1.4	39.4 ± 0.8	63.6
Fish	1	1x	85.5 ± 0.3	77.8 ± 0.3	68.6 ± 0.4	45.1 ± 1.3	42.7 ± 0.2	63.9
Fishr	1	1x	85.5 ± 0.4	77.8 ± 0.1	67.8 ± 0.1	47.4 ± 1.6	41.7 ± 0.0	65.7
SWAD	1	1x	88.1 ± 0.4	<u>79.1 ± 0.4</u>	70.6 ± 0.3	50.0 ± 0.4	46.5 ± 0.2	66.9
MIRO	1	1x	85.4 ± 0.4	79.0 ± 0.0	70.5 ± 0.4	50.4 ± 1.1	44.3 ± 0.2	65.9
CCFP	1	1x	86.6 ±0.2	78.9 ±0.3	68.9 ±0.1	48.6 ±0.4	41.2 ± 0.0	64.8
ARM†	1	1x	85.1	77.6	64.8	45.5	35.5	61.7
VREx†	1	1x	84.9	78.3	66.4	46.4	33.6	61.9
RSC†	1	1x	85.2	77.1	65.5	46.6	38.9	62.7
Mixstyle†	1	1x	85.2	77.9	60.4	44.0	34.0	60.3
SagNet†	1	1x	86.3	77.8	68.1	48.6	40.3	64.2
QT-DoG (ours)	1	**0.22x**	87.8± 0.3	78.4± 0.4	68.9± 0.6	50.8± 0.2	45.1±0.9	66.2
ERM Ens. †	6	6x	87.6	78.5	70.8	49.2	<u>47.7</u>	66.8
DiWA†	60	1x	<u>89.0</u>	78.6	72.8	51.9	<u>47.7</u>	<u>68.0</u>
EoA†	6	6x	88.6	<u>79.1</u>	**72.5**	<u>52.3</u>	47.4	<u>68.0</u>
DART	4-6	4x-6x	78.5 ± 0.7	87.3 ± 0.5	70.1 ± 0.2	48.7 ± 0.8	45.8 ± 0.0	66.1
EoQ (ours)†	5	1.1x	**89.3**	**79.5**	<u>72.3</u>	**53.2**	**47.9**	**68.4**

works. These results demonstrate the superiority of EoQ across five DomainBed datasets, with an average improvement of 0.4% over the state-of-the-art EoA while reducing the memory footprint by approximately 75%. Compared to DiWA, we significantly reduce the computational burden and memory requirements for training, achieving a 12-fold reduction, as DiWA requires training 60 models for diverse averaging. EoQ achieves the most significant gain (7% improvement) on TerraIncognita (Beery et al., 2018), with nonetheless substantial gains of 3-5% w.r.t. ERM on PACS (Li et al., 2017) and DomainNet (Peng et al., 2019).

The results also demonstrate that simply introducing quantization into the ERM-based approach (Gulrajani & Lopez-Paz, 2021) surpasses or yields comparable accuracy to many existing works, although the size and computational budget of our quantization-based approach is significantly lower than that of the other methods. For our results in Table 1 and Figure 1, we employed 7-bit quantization on the network. Therefore, as shown in Figure 1, the model size is drastically reduced, becoming more than 4 times smaller than the other methods. Being smaller in memory footprint, our quantization-based approach can utilize ensembling without increasing the memory storage and computational resources.

Table 2. **Combination with other methods.** Results of PACS and Terra Incognita datasets incorporating QT-DoG with CORAL and MixStyle. C represents the compression factor of the model.

Algorithm	**PACS**	**TerraInc**	**C**
CORAL	85.5 ± 0.6	47.1 ± 0.2	-
CORAL + QT-DoG	**86.9 ± 0.2**	**50.6 ± 0.3**	**4.6x**
MixStyle	85.2 ± 0.3	44.0 ± 0.4	-
MixStyle + QT-DoG	**86.8 ± 0.3**	**47.7 ± 0.2**	**4.6x**

Moreover, quantization not only reduces the memory footprint but also the latency of the model. For example, running a ResNet-50 model on an **AMD EPYC 7302** processor yields a latency of 34.28ms for full-precision and 21.02ms for our INT8 quantized model.

4.1.2. Combinations with Other Methods

Since QT-DoG requires no modifications to training procedures or model architectures, it is universally applicable and can seamlessly integrate with other DG methods. As shown in Table 2, we integrate QT-DoG with CORAL (Sun et al., 2016) and MixStyle (Zhou et al., 2021). Both CORAL and MixStyle demonstrate improved performance when com-

Table 3. **Comparison between ERM and QT-DoG on the Amazon and Camelyon datasets.** We report the in-domain and out-of-domain accuracy with respective metrics as shown. C represents the compression factor of the model.

Dataset	Method	In-dist	Out-dist	C	Metric
Amazon	ERM	71.9 ± 0.1	53.8 ± 0.8	-	10th percentile acc
Amazon	QT-DoG	**79.2 ± 0.5**	**55.9 ± 0.6**	**4.6x**	10th percentile acc
Camelyon	ERM	93.2 ± 5.2	70.3 ± 6.4	-	Average acc
Camelyon	QT-DoG	**96.4 ± 2.1**	**78.4 ± 2.2**	**4.6x**	Average acc

Table 4. **Model quantization with different quantization algorithms.** We report the average target domain accuracy and the average source domain accuracy across all domains in PACS.

Algorithm	Type	In-domain	Out-domain
No quant	-	96.6 ± 0.2	84.7 ± 0.5
OBC	PTQ	96.8 ± 0.2	83.7 ± 0.4
INQ	QAT	97.1 ± 0.2	87.4 ± 0.3
LSQ	QAT	**97.3 ± 0.2**	**87.8 ± 0.3**

bined with QT-DoG, reinforcing our findings that QAT aids in identifying flat minima, thereby enhancing DG.

4.1.3. Different Quantization Methods

In this section, we perform an ablation study by replacing LSQ (Esser et al., 2020) with other quantization algorithms. We use INQ (Zhou et al., 2017) as another quantization-aware training method but also perform quantization using OBC (Frantar et al., 2022), that uses a more popular post-training quantization (PTQ) approach to quantize a network. We perform this ablation study on the PACS dataset, and the results are shown in Table 4. All the experiments are performed with 7-bit quantization. We observe that, while the QAT approaches tend to enhance generalization, the PTQ approach fails to do so. This is due to the fact that there is no training involved after the quantization step in PTQ. That is, with PTQ, we do not train the network with quantization noise to find a flatter minimum.

4.1.4. Results on Wilds Dataset

We performed experiments with 7 bit quantization on two datasets from the WILDS benchmark (Koh et al., 2021). We utilized the same experimental settings as outlined in the WILDS benchmark repository and incorporated quantization into the training process. The results presented in Table 3 confirm our findings on Domainbed benchmark. We used the same BERT model (Sanh et al., 2019) as in WILDS (Koh et al., 2021). These results highlight that QT-DoG generalizes well across both architectural variations and input modalities, including text.

4.1.5. Generality with Vision Transformer

In Table 5, we present the results of quantizing a vision transformer (ERM-ViT, DeiT-small) (Sultana et al., 2022) for domain generalization. We compare the performance of the baseline ERM-ViT to its quantized counterpart on the PACS and Terra Incognita datasets, demonstrating QT-DoG's effectiveness across different architectures. The results clearly show that QT-DoG also improves the performance of vision transformers.

Table 5. **Quantization of a Vision Transformer** Comparison of performance on PACS and TerraInc datasets with and without QT-DoG quantization of ERM_ViT (Sultana et al., 2022) with DeiT-Small backbone.

Algorithm	PACS	TerraInc	Compression
ERM_ViT	84.3 ± 0.2	43.2 ± 0.2	-
ERM-SD_ViT	**86.3 ± 0.2**	44.3 ± 0.2	-
ERM_ViT + QT-DoG	86.2 ± 0.3	**45.6 ± 0.4**	**4.6x**

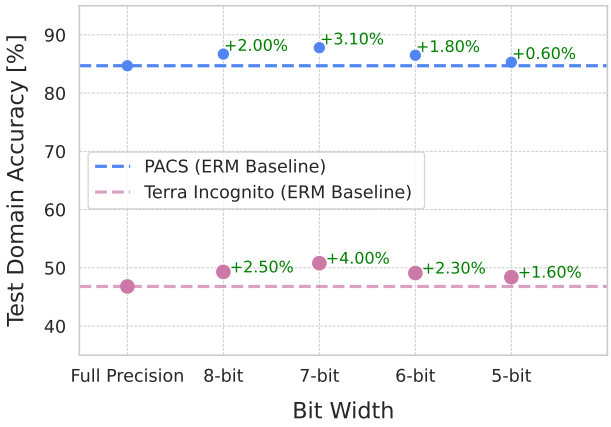

Figure 5. **Bit precision analysis for efficient quantization.** We show results on out-of-domain test accuracy with two different datasets, i.e., PACS and TerraIncognita. For each bit precision, we report the increase in the test domain accuracy averaged across all domains. The 7-bit quantized model exhibits the maximum increase for both datasets. We quantize the model at 2000 steps.

4.1.6. Bit Precision Analysis

Here, we empirically analyze the effect of different bit-precisions for quantization on the generalization of the model. We perform experiments with four different bit levels and present an analysis in Figure 5 on the PACS (Li et al., 2017) and TerraIncognita (Beery et al., 2018) datasets. We report the test domain accuracy averaged across all domains. For both datasets, 7-bit precision was found to be the

Table 6. **Performance comparison with CLIP-based methods.** We report accuracy on DomainNet, TerraIncognita, and Office datasets, as well as the average performance (AVG). QT-DoG achieves competitive accuracy while offering substantial compression.

Algorithm	Backbone	DomainNet	TerraInc	Office	Avg.	C
ERM	CLIP	59.9 ± 0.1	60.9 ± 0.2	83.0 ± 0.1	67.9	-
CLIPood	CLIP	63.5 ± 0.1	60.5 ± 0.4	87.0 ± 0.1	70.3	-
QT-DoG	CLIP	63.1 ± 0.2	61.9 ± 0.3	86.7 ± 0.2	**70.6**	**4.6x**

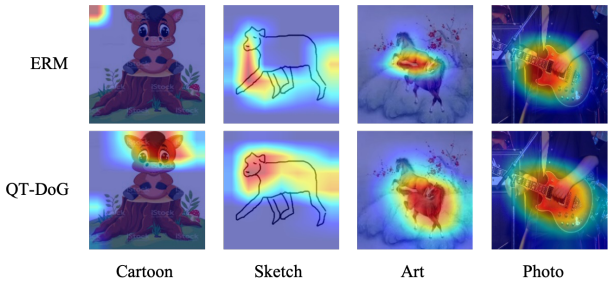

Figure 6. **GradCAM visualization for ERM (Gulrajani & Lopez-Paz, 2021) and QT-DoG.** We show results on the PACS dataset (Li et al., 2017) and consider a different domain as test domain in each run, indicated by the different columns.

optimal bit precision to have the best out-of-domain generalization while maintaining in-domain accuracy. Nonetheless, 8 bits and 6 bits also show improvements, albeit smaller than with 7-bit quantization. These results evidence that, even with a 6 times smaller model, quantization still yields better out-of-domain performance without sacrificing the in-domain accuracy.

4.1.7. Generality with CLIP-Based Methods

We compare QT-DoG with existing CLIP-based domain generalization methods using the ViT-B/16 backbone to maintain consistency in architecture. As shown in the table below, QT-DoG achieves higher average accuracy than both ERM (Gulrajani & Lopez-Paz, 2021) and CLIPood (Shu et al., 2023) across the DomainNet, TerraIncognita, and Office datasets. It provides a 0.3 percent improvement over the best baseline, CLIPood. Importantly, this performance gain comes with a 4.6 times reduction in model size, showing that QT-DoG improves both generalization and efficiency.

4.2. Visualizations

In Figure 6, we present some of the examples[1] from the PACS dataset and show GradCAM (Gildenblat & contributors, 2021) results in the target domain. We perform four different experiments by considering a different target domain for each run, while utilizing the other domains for training. We use the output from the last convolutional layer of the models with and without quantization. These visualizations evidence that quantization focuses on better regions than ERM, and with a much larger receptive field. In certain cases, ERM does not even focus on the correct image region. It is quite evident that quantization pushes the model to learn more generalized patterns, leading to a model that is less sensitive to the specific details of the training set.

[1] More examples are provided in the appendix.

5. Conclusion

We introduced QT-DoG, a novel generalization strategy based on neural network quantization. Our approach leverages the insight that QAT can find flatter minima in the loss landscape, serving as an effective regularization method to reduce overfitting and enhance the generalization capabilities. We empirically demonstrated, supported by analytical insights, that quantization not only enhances generalization but also helps stabilize the training process. Our extensive experiments across diverse datasets show that incorporating quantization with an optimal bit-width significantly enhances domain generalization, yielding performance comparable to existing methods while reducing the model size. Additionally, we proposed EoQ, a powerful ensembling strategy that addresses the challenges of memory footprint and computational load by creating ensembles of quantized models. EoQ outperforms state-of-the-art methods while being approximately four times smaller than its full-precision ensembling counterparts.

Acknowledgements

We thank Soumava Kumar Roy, Chen Zhao, Ahmad Jarrar Khan and Muhammad Zakwan for their help. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 945363. Moreover, this work was funded in part by the Swiss National Science Foundation and the Swiss Innovation Agency (Innosuisse) via the BRIDGE Discovery grant No. 194729.

Impact Statement

In this paper, we presented our research work aiming to highlight the relation of model quantization for the domain

generalization problem setup. In considering the societal impact of our approach, a few concerns emerge, such as performance degradation in extreme-level quantization. Although the proposed approach provides competitive performance compared to previous state-of-the-art models on domain generalization benchmarks, we note that performance slightly varies in extreme quantization cases. Our future work is aimed at a thorough analysis of this aspect in safety-critical applications, such as medical imaging, autonomous driving, or surveillance.

References

Addepalli, S., Asokan, A. R., Sharma, L., and Babu, R. V. Leveraging Vision-Language Models for Improving Domain Generalization in Image Classification. In *CVPR*, June 2024.

An, G. The effects of adding noise during backpropagation training on a generalization performance. *Neural Comput.*, 1996.

Andriushchenko, M., Croce, F., Müller, M., Hein, M., and Flammarion, N. A Modern Look at the Relationship between Sharpness and Generalization. *ICML*, 2023.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Arpit, D., Wang, H., Zhou, Y., and Xiong, C. Ensemble of Averages: Improving Model Selection and Boosting Performance in Domain Generalization. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *NeurIPS*, 2022.

Bahng, H., Chun, S., Yun, S., Choo, J., and Oh, S. J. Learning De-biased Representations with Biased Representations. In *ICML*, 2020.

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *ECCV*, 2018.

Benton, G., Maddox, W., Lotfi, S., and Wilson, A. G. G. Loss Surface Simplexes for Mode Connecting Volumes and Fast Ensembling. In *ICML*, 2021.

Bhalgat, Y., Lee, J., Nagel, M., Blankevoort, T., and Kwak, N. LSQ+: Improving low-bit quantization through learnable offsets and better initialization. In *CVPR*, 2020.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. *NeurIPS*, 2011.

Boncelet, C. Chapter 7 - Image Noise Models. In *The Essential Guide to Image Processing*. Academic Press, 2009.

Breiman, L. Bagging predictors. *Machine learning*, 24(2): 123–140, 1996.

Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. Zeroq: A novel zero shot quantization framework. In *CVPR*, 2020.

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., and Park, S. SWAD: Domain generalization by seeking flat minima. In *NeurIPS*, 2021.

Chee, J., Cai, Y., Kuleshov, V., and De Sa, C. QuIP: 2-Bit Quantization of Large Language Models With Guarantees. In *NeurIPS*, 2023.

Chen, Z., Wang, W., Zhao, Z., Su, F., Men, A., and Meng, H. PracticalDG: Perturbation Distillation on Vision-Language Models for Hybrid Domain Generalization. In *CVPR*, 2024.

Cheng, D., Xu, Z., Jiang, X., Wang, N., Li, D., and Gao, X. Disentangled Prompt Representation for Domain Generalization. In *CVPR*, June 2024.

Choshen, L., Venezian, E., Slonim, N., and Katz, Y. Fusing finetuned models for better pretraining. *arXiv preprint*, 2022.

Dai, D. and Van Gool, L. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *ITCS*, 2018.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *ICML*, 2017.

Dong, Z., Yao, Z., Gholami, A., Mahoney, M., and Keutzer, K. HAWQ: Hessian AWare Quantization of Neural Networks with Mixed-Precision. In *ICCV*, 2019.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *UAI*, 2017.

Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned Step Size Quantization. *ICLR*, 2020.

Fang, C., Xu, Y., and Rockmore, D. N. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *ICCV*, 2013.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-Aware Minimization for Efficiently Improving Generalization. In *ICLR*, 2021.

Frantar, E., Singh, S. P., and Alistarh, D. Optimal Brain Compression: A Framework for Accurate Post-Training Quantization and Pruning. In *NeurIPS*, 2022.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *JMLR*, 2016.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In *NeurIPS*, 2018.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

Gildenblat, J. and contributors. PyTorch library for CAM methods. https://github.com/jacobgil/pytorch-grad-cam, 2021.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Gray, R. and Neuhoff, D. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, 1998. doi: 10.1109/18.720541.

Gulrajani, I. and Lopez-Paz, D. In Search of Lost Domain Generalization. In *ICLR*, 2021.

Gupta, V., Serrano, S. A., and DeCoste, D. Stochastic Weight Averaging in Parallel: Large-Batch Training That Generalizes Well. In *ICLR*, 2020.

Hansen, L. K. and Salamon, P. Neural network ensembles. *TPAMI*, 1990.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

Hinton, G., Vinyals, O., and Dean, J. Distilling the Knowledge in a Neural Network. In *NeurIPS*, 2015.

Hochreiter, S. and Schmidhuber, J. Simplifying neural nets by discovering flat minima. *NeurIPS*, 1994.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 1997.

Huang, Z., Wang, H., Xing, E. P., and Huang, D. Self-challenging improves cross-domain generalization. In *ECCV*, 2020.

Huang, Z., Zhou, A., Ling, Z., Cai, M., Wang, H., and Lee, Y. J. A Sentence Speaks a Thousand Images: Domain Generalization through Distilling CLIP with Language Guidance. In *ICCV*, 2023.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018.

Jain, S., Addepalli, S., Sahu, P. K., Dey, P., and Babu, R. V. DART: Diversify-Aggregate-Repeat Training Improves Generalization of Neural Networks. In *CVPR*, 2023.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *ICLR*, 2020.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In *ICLR*, 2017.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B. A., Haque, I. S., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

Krogh, A. and Vedelsby, J. Neural network ensembles, cross validation, and active learning. In *NeurIPS*, 1995.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 2021.

Kullback, S. and Leibler, R. A. On information and sufficiency. *Annals of Mathematical Statistics*, 1951.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Lee, Y., Yao, H., and Finn, C. Diversify and Disambiguate: Learning From Underspecified Data. *ICML Workshop*, 2022.

Li, C., Zhang, D., Huang, W., and Zhang, J. Cross contrasting feature perturbation for domain generalization. In *ICCV*, 2023.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

Li*, M., Lin*, Y., Zhang*, Z., Cai, T., Li, X., Guo, J., Xie, E., Meng, C., Zhu, J.-Y., and Han, S. SVDQuant: Absorbing Outliers by Low-Rank Components for 4-Bit Diffusion Models. In *ICLR*, 2025.

Li, Y., Gong, R., Tan, Z., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., and Gu, S. BRECQ: Pushing the Limit of Post-Training Quantization by Block Reconstruction. *ICLR*, 2021.

Li, Z., Ren, K., Jiang, X., Li, B., Zhang, H., and Li, D. Domain Generalization using Pretrained Models without Fine-tuning. *arXiv preprint*, 2022.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. In *MLSys*, 2024.

Liu, F., Kim, M., Ren, Z., and Liu, X. Distilling CLIP with Dual Guidance for Learning Discriminative Human Body Shape Representation. In *CVPR*, June 2024.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. In *NeurIPS*, 2019.

Matena, M. and Raffel, C. Merging Models with Fisher-weighted averaging. In *NeurIPS*, 2022.

Mesbah, Y., Ibrahim, Y. Y., and Khan, A. M. Domain Generalization Using Ensemble Learning. In *ISWA*, 2022.

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., and Brendel, W. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *NeurIPS*, 2019.

Moayeri, M., Rezaei, K., Sanjabi, M., and Feizi, S. Text-To-Concept (and Back) via Cross-Model Alignment. In *ICML*, 2023.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *ICML*, 2013.

Mueller, M., Vlaar, T. J., Rolnick, D., and Hein, M. Normalization layers are all that sharpness-aware minimization needs. In *NeurIPS*, 2023.

Murray, A. and Edwards, P. Synaptic Weight Noise During MLP Learning Enhances Fault-Tolerance, Generalization and Learning Trajectory. In *NeurIPS*, 1992.

Nagel, M., van Baalen, M., Blankevoort, T., and Welling, M. Data-Free Quantization Through Weight Equalization and Bias Correction. *ICCV*, 2019.

Nagel, M., Amjad, R. A., van Baalen, M., Louizos, C., and Blanevoort, T. Up or Down? Adaptive Rounding for Post-Training Quantization. *ICML*, 2020.

Pagliardini, M., Jaggi, M., Fleuret, F., and Karimireddy, S. P. Agree to Disagree: Diversity through Disagreement for Better Transferability. In *ICLR* , 2023.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021.

Ramachandran, A., Kundu, S., and Krishna, T. Clamp-vit: Contrastive data-free learning for adaptive post-training quantization of vits. In Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., and Varol, G. (eds.), *ECCV*, 2024.

Rame, A., Dancette, C., and Cord, M. Fishr: Invariant Gradient Variances for Out-of-distribution Generalization. In *ICML*, 2022a.

Rame, A., Kirchmeyer, M., Rahier, T., Rakotomamonjy, A., Gallinari, P., and Cord, M. Diverse weight averaging for out-of-distribution generalization. In *NeurIPS*, 2022b.

Ramé, A., Ahuja, K., Zhang, J., Cord, M., Bottou, L., and Lopez-Paz, D. Model Ratatouille: Recycling Diverse Models for Out-of-Distribution Generalization. In *ICML*, 2023.

Rissanen, J. Modeling by shortest data description. *Automatica*, 1978.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally Robust Neural Networks. In *ICLR*, 2020.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *NeurIPS Workshop*, 2019.

Shang, Y., Liu, G., Kompella, R., and Yan, Y. Enhancing Post-Training Quantization Calibration Through Contrastive Learning. 2024.

Shao, W., Chen, M., Zhang, Z., Xu, P., Zhao, L., Li, Z., Zhang, K. Z., Gao, P., Qiao, Y., and Luo, P. OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models. *ICLR*, 2024.

Shin, J., So, J., Park, S., Kang, S., Yoo, S., and Park, E. NIPQ: Noise Proxy-Based Integrated Pseudo-Quantization. In *CVPR*, 2023.

Shu, Y., Guo, X., Wu, J., Wang, X., Wang, J., and Long, M. CLIPood: Generalizing CLIP to Out-of-Distributions. In *ICML*, 2023.

Sultana, M., Naseer, M., Khan, M. H., Khan, S., and Khan, F. S. Self-Distilled Vision Transformer for Domain Generalization. In *ACCV*, 2022.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

Sun, B., Feng, J., and Saenko, K. Return of Frustratingly Easy Domain Adaptation. In *AAAI*, 2016.

Thopalli, K., Katoch, S., Thiagarajan, J. J., Turaga, P. K., and Spanias, A. Multi-Domain Ensembles for Domain Generalization. In *NeurIPS Workshop*, 2021.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Wortsman, M., Horton, M., Guestrin, C., Farhadi, A., and Rastegari, M. Learning Neural Network Subspaces. *ICML*, 2021.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., and Schmidt, L. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022a.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. Robust fine-tuning of zero-shot models. In *CVPR*, 2022b.

Xiao, K. Y., Engstrom, L., Ilyas, A., and Madry, A. Noise or Signal: The Role of Image Backgrounds in Object Recognition. In *ICLR*, 2020.

Yalniz, I. Z., Jégou, H., Chen, K., Paluri, M., and Mahajan, D. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

Yamamoto, K. Learnable Companding Quantization for Accurate Low-bit Neural Networks. In *CVPR*, 2021.

Yang, J., Tang, C., Yu, C., and Lv, J. GWQ: Group-Wise Quantization Framework for Neural Networks. In *Asian Conference on Machine Learning*, 2023.

Yao, Z., Dong, Z., Zheng, Z., Gholami, A., Yu, J., Tan, E., Wang, L., Huang, Q., Wang, Y., Mahoney, M. W., and Keutzer, K. HAWQV3: Dyadic Neural Network Quantization. In *ICML*, 2020.

Ye, N., Li, K., Hong, L., Bai, H., Chen, Y., Zhou, F., and Li, Z. OoD-Bench: Benchmarking and Understanding Out-of-Distribution Generalization Datasets and Algorithms. In *CVPR*, 2022.

Yu, X., Yoo, S., and Lin, Y. CLIPCEIL: Domain Generalization through CLIP via Channel rEfinement and Image-text aLignment. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *NeurIPS*, 2024.

Zhang, H. R., Li, D., and Ju, H. Noise stability optimization for finding flat minima: A hessian-based regularization approach. *Transactions on Machine Learning Research*, 2024.

Zhang, T. and Shrivastava, A. LeanQuant: Accurate and Scalable Large Language Model Quantization with Loss-error-aware Grid. In *ICLR*, 2025.

Zhang, X., Xu, R., Yu, H., Dong, Y., Tian, P., and Cui, P. Flatness-Aware Minimization for Domain Generalization. In *ICCV*, October 2023.

Zhao, R., Hu, Y., Dotzel, J., De Sa, C., and Zhang, Z. Improving Neural Network Quantization without Retraining using Outlier Channel Splitting. *ICML*, June 2019.

Zhou, A., Yao, A., Guo, Y., Xu, L., and Chen, Y. Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights. In *ICLR*, 2017.

Zhou, K., Yang, Y., Qiao, Y., and Xiang, T. Domain Generalization with MixStyle. In *ICLR*, 2021.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. Domain Generalization: A Survey. *TPAMI*, 2022.

A. Datasets and Metrics

We demonstrate the effectiveness of our proposed method on diverse classification datasets used for evaluating multi-source Domain Generalization:

PACS (Li et al., 2017) is a 7 object classification challenge encompassing four domains, with a total of 9,991 samples. It serves to validate our method in smaller-scale settings. **VLCS** (Fang et al., 2013) poses a 5 object classification problem across four domains. With 10,729 samples, VLCS provides a good benchmark for close Out-of-Distribution (OOD), featuring subtle distribution shifts simulating real-life scenarios. **OfficeHome** (Venkateswara et al., 2017) comprises a total of 15,588 samples. It presents a 65-way classification challenge featuring everyday objects across four domains. **TerraIncognita** (Beery et al., 2018) addresses a 10 object classification challenge of animals captured in wildlife cameras, with four domains representing different locations. The dataset contains 24,788 samples, illustrating a realistic use-case where generalization is crucial. **DomainNet** (Peng et al., 2019) provides a 345 object classification problem spanning six domains. With 586,575 samples, it is one of the largest datasets. Furthermore, we present results on WILDS benchmark datasets to demonstrate the effectiveness of our approach in real-world applications.

We report out-of-domain accuracies for each domain and their average, i.e., a model is trained and validated on training domains and evaluated on the unseen target domain. Each out-of-domain performance is an average of three different runs with different train-validation splits for the quantized models. We then combine the predictions of the different quantized models for our EoQ results.

B. Implementation Details

We use the same training procedure as DomainBed (Gulrajani & Lopez-Paz, 2021), incorporating additional components from quantization. Specifically, we adopt the default hyperparameters from DomainBed (Gulrajani & Lopez-Paz, 2021), including a batch size of 32 (per-domain). We employ a ResNet-50 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015) as initial model and use a learning rate of 5e-5 along with the Adam optimizer, and no weight decay. Following SWAD(Cha et al., 2021), the models are trained for 15,000 steps on DomainNet and 5,000 steps on the other datasets. In the training process, we keep a specific domain as the target domain, while the remaining domains are utilized as source domains. During this training phase, 20% of the samples are used for validation and model selection. We validate the model every 300 steps using held-out data from the source domains, and assess the final performance on the excluded domain (target).

We use LSQ (Esser et al., 2020) and INQ (Zhou et al., 2017) for model quantization, with the same configuration as existing quantization methods (Esser et al., 2020; Bhalgat et al., 2020; Dong et al., 2019; Yao et al., 2020; Zhou et al., 2017), where all layers are quantized to lower bit precision except the last one. We quantize the models at 8,000 steps for DomainNet and 2,000 steps for the other datasets. Moreover, each channel in a layer has a different scaling factor s.

C. Discussion and Limitations

Despite showing success and surpassing the state-of-the-art methods in terms of performance, EoQ also has some limitations. First, it requires training multiple models like Rame et al. (2022b); Arpit et al. (2022), to create diversity and form an ensemble. This ensemble creation increases the training computational load. Nevertheless, our quantized ensembling models are much smaller in size.

Another limitation of this work is the challenge of determining the optimal bit precision for achieving the best performance in OOD generalization. In our experiments on the DomainBed benchmark, we identified 7 bits as the optimal precision. However, this may not hold true for other datasets. A potential future direction is to utilize a small number of target images to identify the optimal bit precision, which would significantly reduce the computational overhead associated with this process.

Lastly, given our utilization of a uniform quantization strategy, it would be interesting to investigate whether specific layers can be more effectively exploited than others through mixed-precision techniques to have even better domain generalization performance.

D. Per-Domain Performance Improvement

We also report per-domain performance improvement for PACS (Li et al., 2017) and Terra Incognito (Beery et al., 2018) dataset. We choose the best model based on the validation set and report the results in 7 and 8. The results with quantization correspond to 7 bit-precision and we perform quantization after 2000 steps. Table 7 and 8 show that EoQ is consistently better than the current state-of-the-art methods across domains for different datasets.

Algorithm	Art	Cartoon	Painting	Sketch	Avg.
ERM (our runs)	89.8	79.7	96.8	72.5	84.7
SWAD	89.3	83.4	97.3	82.5	88.1
EoA	90.5	83.4	98.0	82.5	88.6
DiWA	90.6	83.4	**98.2**	83.8	89.0
QT-DoG	89.1	82.4	96.9	82.3	87.8
EoQ	**90.7**	**83.7**	**98.2**	**84.8**	**89.3**

Table 7. **Per-Domain Accuracy Comparison for PACS.** We report the accuracy for each domain of the PACS dataset along with the average across all domains. Our proposed quantization is shaded in Gray.

Algorithm	L100	L38	L43	L46	Avg.
ERM (our runs)	58.2	38.3	57.1	35.1	47.2
SWAD	55.4	44.9	59.7	39.9	50.0
DiWA	57.2	50.1	60.3	39.8	51.9
EoA	57.8	46.5	**61.3**	43.5	52.3
QT-DoG	60.2	46.4	55.2	41.4	50.8
EoQ	**61.8**	**48.2**	59.2	**43.7**	**53.2**

Table 8. **Per-Domain Accuracy Comparison for Terra Incognito.** We report the accuracy for each domain of the Terra Incognito dataset along with the average across all domains. Our proposed quantization is shaded in Gray.

E. Bit Precision Analysis Extended

In contrast to main manuscript, Table 9 provides all the results in a tabular form. We show how quantization outperforms the vanilla ERM approach. This shows the superior performance of quantization over ERM despite being more than 6 times smaller in the case of 5 bit-precision.

Algorithm	**Compression**	**PACS**		**TerraInc**	
		In-domain	Out-domain	In-domain	Out-domain
ERM (our runs)	-	96.9 $\pm$ 0.1	84.7 $\pm$ 0.5	91.7 $\pm$ 0.2	47.2 $\pm$ 0.4
QT-DoG(8)	4x	97.0 $\pm$ 0.1	85.0 $\pm$ 0.1	90.9 $\pm$ 0.2	49.1 $\pm$ 0.1
QT-DoG(7)	4.6x	**97.3 $\pm$ 0.2**	**87.8 $\pm$ 0.3**	**92.3 $\pm$ 0.2**	**50.8$\pm$ 0.2**
QT-DoG(6)	5.3x	97.1 $\pm$ 0.1	86.5 $\pm$ 0.1	91.1 $\pm$ 0.0	49.0 $\pm$ 0.3
QT-DoG(5)	6.4x	97.0 $\pm$ 0.1	85.3 $\pm$ 0.4	91.0 $\pm$ 0.1	48.4 $\pm$ 0.2

Table 9. **Model quantization with different bit-precisions vs vanilla ERM.** We report the average target domain accuracy as well as the average source domain accuracy across all domains for the PACS (Li et al., 2017) and TerraIncognita (Beery et al., 2018) datasets. Quantization not only enhances the generalization ability but also retains the source domain performance. QT-DoG(x) indicates a model quantized with x bit-precision.

However, as shown in Table 10, decreasing bit-precision through quantization does not always improve performance above the baseline; after a point, there is a tradeoff between compression and generalization. Specifically, our experiments with 4-bit precision and lower did not yield satisfactory results. Finding the sweet spot for balancing speed and performance can

be an interesting research direction. Our results evidence that there exist configurations that can improve both speed and performance.

Algorithm	Bit-Precision	PACS
ERM	32	84.7 ± 0.5
QT-DoG	7	**87.8 ± 0.3**
	6	86.5 ± 0.1
	5	85.3 ± 0.4
	4	84.3 ± 0.3
	3	83.3 ± 0.4
	2	82.8 ± 0.2

Table 10. **Effect of aggressive quantization.** Performance comparison between ERM and QT-DoG with varying bit-precision on PACS.

F. Experiments with larger pre-training datasets

We also show experimental results with ResNeXt-50-32x4 in Table 11. Note that both ResNet-50 and ResNeXt-50-32x4d have 25M parameters. However, ResNeXt-50-32x4d is pre-trained on a larger dataset i.e Instagram 1B images(Yalniz et al., 2019). It is evident from Table 11 that incorporating quantization into training consistenlty improve accuracy even when a network is pre-trained on a larger dataset. Furthermore, EoQ again showed superior performance in comparison to other methods across five DomainBed datasets.

Algorithm	M	S	PACS	VLCS	Office	TerraInc	DomainNet	Avg.
ResNeXt-50 32x4d (25M Parameters, Pre-trained 1B Images)								
ERM	1	1x	88.7 ± 0.3	79.0 ± 0.1	70.9 ± 0.5	51.4 ± 1.2	48.1 ± 0.2	67.7
SMA	1	1x	92.7 ± 0.3	79.7 ± 0.3	78.6 ± 0.1	53.3 ± 0.1	53.5 ± 0.1	71.6
QT-DoG (ours)	1	1x	92.9 ± 0.3	79.2 ± 0.4	78.9 ± 0.3	54.1 ± 0.2	53.9 ± 0.2	71.8
ERM Ens.†	6	6x	91.2	80.3	77.8	53.5	52.8	71.1
EoA†	6	6x	<u>93.2</u>	**80.4**	<u>80.2</u>	<u>55.2</u>	<u>54.6</u>	<u>72.7</u>
EoQ† (ours)	**5**	**1.1x**	**93.5**	<u>80.3</u>	**80.3**	**55.6**	**54.8**	**72.9**

Table 11. **Comparison with other methods for ResNeXt-50.** Performance benchmarking on 5 datasets of the DomainBed benchmark. Highest accuracy is shown in bold, while second best is underlined. Ensembles† do not have confidence interval because an ensemble uses all the models to make a prediction. Our proposed method is colored in Gray. Average accuracies and standard errors are reported from three trials. For all the reported results, we use the same training-domain validation protocol as (Gulrajani & Lopez-Paz, 2021). M corresponds to the number of models trained during training and S corresponds to the relative network size.

G. In-domain Performance Improvement using Quantization

We further study the in-domain test accuracy of our quantization approach without ensembling on PACS and TerraIncognita datasets. As (Cha et al., 2021), we split the in-domain datasets into training (60%), validation (20%), and test (20%) sets. We choose the best model based on the validation set and report the results on the test set in Table 12. The results with quantization correspond to 7 bit-precision.

QT-DoG also enhances the in-domain performance. The regularization effect introduced by quantization prevents the model from overfitting to edge cases and pushes it to learn more meaningful and generalizable features, which we also demonstrate in Section 4.2. As the training data consists of various domains and the quantization limits the range of weight values, it discourages the model from becoming overly complex and overfitting to the noise in the training data. Therefore, the model is more robust to minor input fluctuations.

Method	PACS	TerraInc	Compression
ERM	96.6 ± 0.2	90.1 ± 0.2	-
SAM	97.3 ± 0.1	90.8 ± 0.1	-
SWA	97.1 ± 0.1	90.7 ± 0.1	-
SMA	96.8 ± 0.2	90.7 ± 0.4	-
SWAD	**97.7 ± 0.2**	90.8 ± 0.3	-
QT-DoG	97.3 ± 0.2	**91.1 ± 0.2**	**∼4.6x**

Table 12. **Comparison between generalization methods on PACS and TerraInc for IID settings.** We report the accuracy averaged across all domains. Our proposed approach is shaded in Gray. Highest accuracy is shown in bold, while second best is underlined.

H. Ablation on Layerwise and Channelwise scale

We conducted an ablation study where we set s at the layer level, rather than on a per-channel basis. We see that Channelwise s can lead to 1.5% accuracy as compared to layerwise s. The results of this experiment on the PACS dataset with 7 bit quantization are shown below:

Scale	OOD Accuracy
No quantization	84.7 ± 0.5
Layerwise	86.3 ± 0.4
Channelwise	87.8 ± 0.3

Table 13. OOD Accuracy with channelwise vs layerwise Scaling factor for quantization.

I. Ablation on Quantization Steps

We conducted an ablation study on the PACS dataset to identify the optimal number of steps after which quantization should be applied. We perform 7-bit quantization and the results are summarized below:

Quantization Step	OOD Accuracy
No quantization	84.7 ± 0.5
1000	86.2 ± 0.4
2000	87.8 ± 0.3
3000	86.9 ± 0.4
4000	85.1 ± 0.3

Table 14. OOD Accuracy across different quantization steps.

J. Adding Uniform Noise to weights

Quantization noise and uniform weight noise share similarities in that both introduce perturbations to the model's parameters. However, quantization noise specifically arises from the discretization of the weights, which can lead to a more structured form of regularization due to the rounding or truncation during the quantization process. In contrast, uniform weight noise typically adds random perturbations with a uniform distribution, which may not exhibit the same structured regularization properties.

In Table 15, we provide the results of our ablation study on the PACS dataset with uniform noise with different minimum and maximum value:

Noise	OOD Accuracy
No noise	84.7 ± 0.5
Uniform(-0.0001, 0.0001)	82.9 ± 0.6
Uniform(-0.00005, 0.00005)	83.8 ± 0.5
Uniform(-0.00001, 0.00001)	85.1 ± 0.4
Uniform(-0.000005, 0.000005)	85.6 ± 0.3

Table 15. OOD Accuracy under different noise levels

K. Visualization

More GradCAM Results

In Figure 9, 10, 11, 12, we present some of the examples from the Terra dataset and show GradCAM (Gildenblat & contributors, 2021) results on the target domain. We use the output from the last convolutional layer of the models with and without quantization for GradCAM. Similar to our experiments on PACS dataset, we perform four different experiments by considering a different target domain for each run, while utilizing the other domains for training. Both models are trained with the similar settings as (Gulrajani & Lopez-Paz, 2021). For quantization method, we quantized the model after 2000 iteration and employ 7 bit-precision as it provides the best out-of-domain performance. Moreover, we present some more examples for PACS dataset in Figure 7.

These visualizations further proves that quantization pushes the model to be less sensitive to the specific details of the training set.

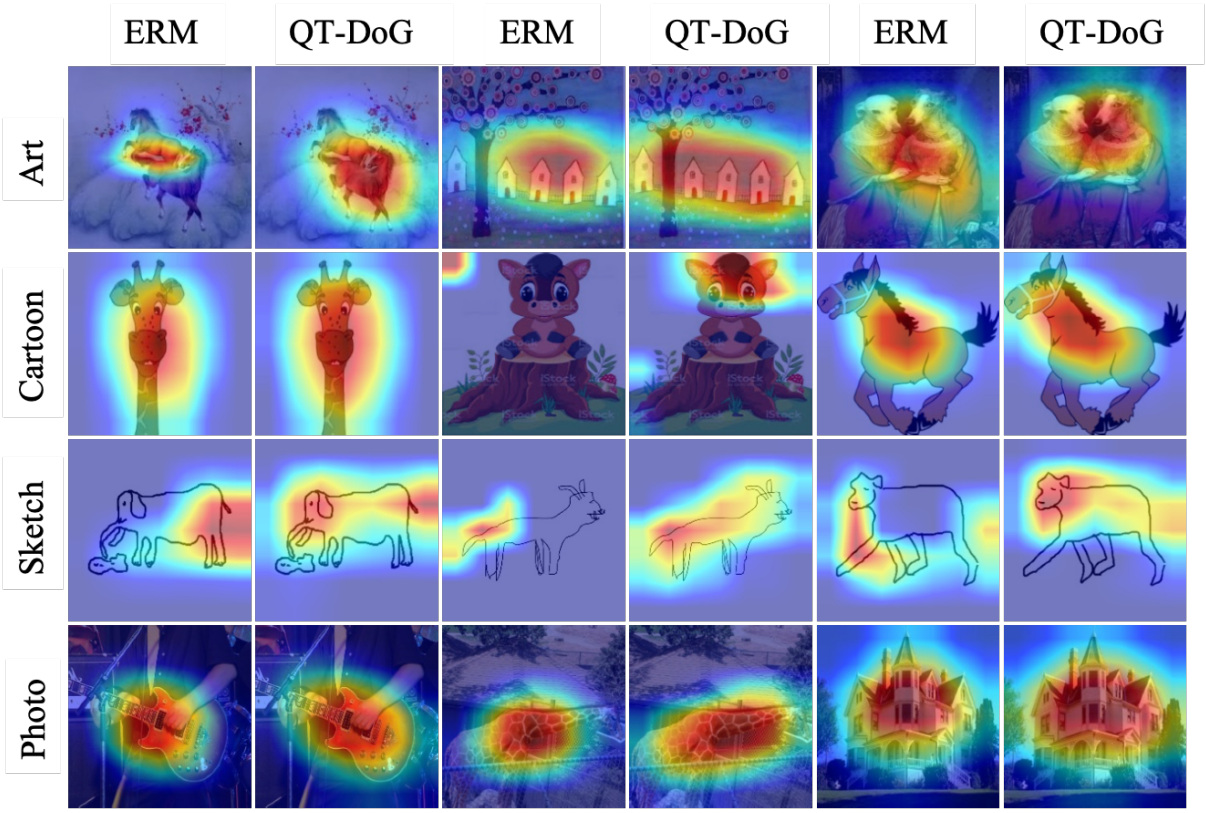

Figure 7. **GradCAM visualization for ERM (Gulrajani & Lopez-Paz, 2021) and QT-DoG.** We show results on the PACS dataset (Li et al., 2017) and consider a different domain as test domain in each run, indicated by the different rows in the figure.

L. Quantization Noise follows Uniform Distribution

It is a well established fact in the literature that the distribution of quantization error can be modeled as uniform distribution (Boncelet, 2009; Gray & Neuhoff, 1998). For uniform data distribution, it only needs to hold within bounds of quantization bins as in this case half of the samples will be floored while the other half will be ceiled. As the the distribution of data is uniform with a flat pdf, the error introduced by the rounding operation will also have the same distribution.

In many scenarios, however, the distribution of network weights is not uniform. Therefore, we extend this argument to more general data distributions. If the data follows a symmetric distribution with a reasonably smooth pdf, we can approximate the pdf of this distribution by a piecewise linear function, interpolating between the boundaries of the quantization bins. This approximation becomes more accurate as we increase the bit-width, which in turn increases the number of quantization bins. Although these linear pdfs within the quantization bins are not flat, the symmetry of the distribution ensures that there exists another bin with a similar but negatively sloped pdf. Thus, when considered as a whole, the quantization error becomes uniform, i.e $\epsilon \sim U\left[-\frac{s}{2}, \frac{s}{2}\right]$.

In order to model the quantization error as a uniform error, we need to have a symmetric weight distribution. Deep neural networks are overparameterized models that can have multiple optimal solutions. To guide the models toward specific distributions, weight priors are typically applied in the form of regularizers. These priors lead to trained weights that correspond to the specified distribution. L1 regularization encourages sparsity and promotes a Laplace distribution for the weights, which is symmetric and has sharper peaks around zero. On the other hand, L2 regularization promotes a Gaussian distribution, which is also symmetric but follows a smooth bell-shaped curve with heavier tails compared to the Laplace distribution.

These regularization techniques result in weight distributions that are symmetric and smooth, satisfying the necessary condition for quantization error to be modeled as a uniform distribution, where the quantization noise/error exhibits a very low KL divergence with respect to a uniform distribution.

We plot various layers' weight distributions along with the quantization noise in Figure 8 and demonstrate that weights generally exhibit a symmetric and smooth distribution. Additionally, the quantization noise/error shows a very low Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) with respect to a uniform distribution having the same minimum and maximum values. To assess symmetry, we employ reverse-order and forward-order medians, ensuring a balanced representation of the weight distribution. The reverse-order median captures the statistical properties of the lower half, while the forward-order median does the same for the upper half, providing a more robust symmetry evaluation.

M. Reproducibility

To guarantee reproducibility, we will provide the source code publicly along with the details of the environments and dependencies. We will also provide instructions to reproduce the main results of Table 1 in the main paper. Furthermore, we will also share instructions and code to plot the loss surfaces and GradCAM results.

Every experiment in our work was executed on a single NVIDIA A100, Python 3.8.16, PyTorch 1.10.0, Torchvision 0.11.0, and CUDA 12.1.

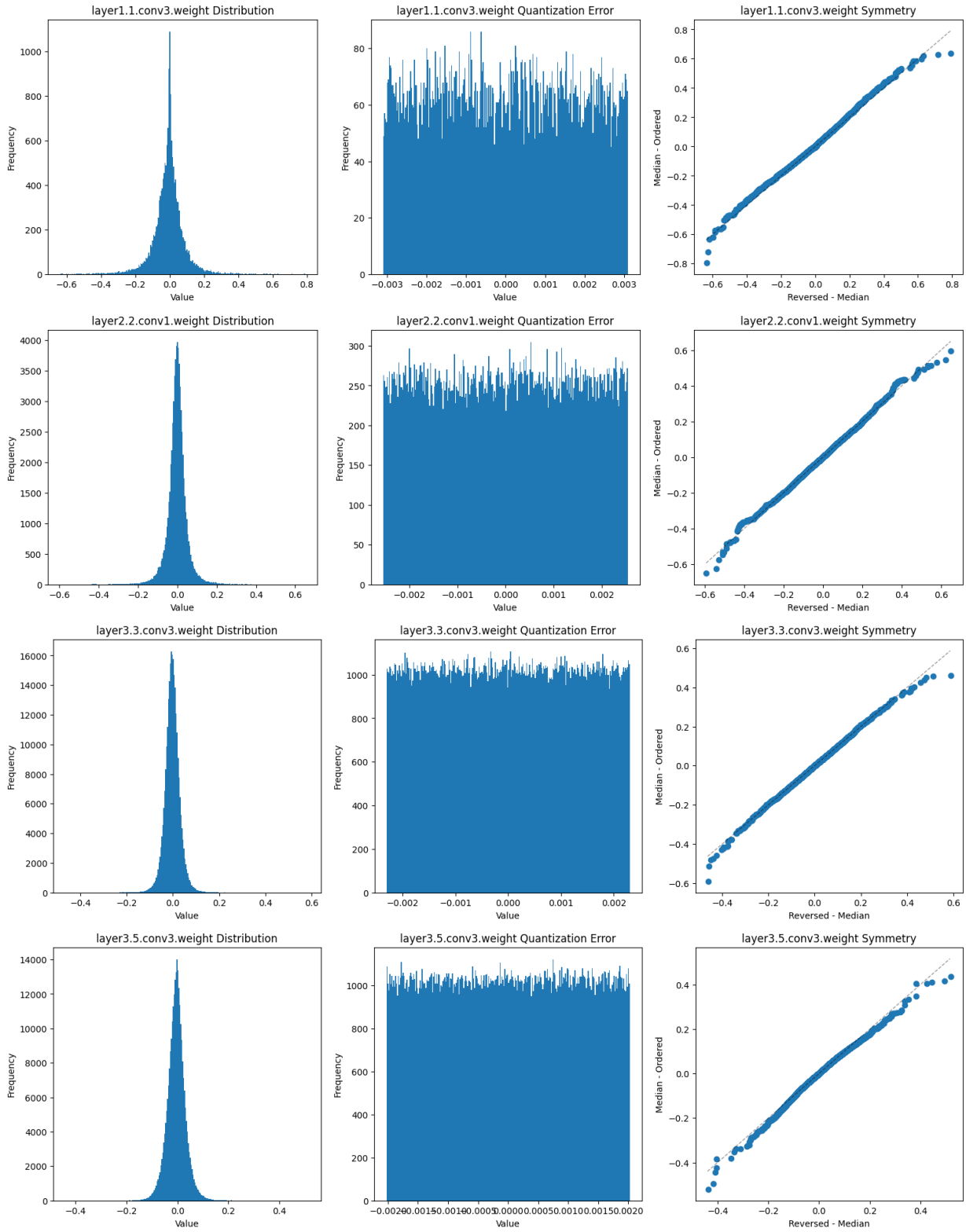

Figure 8. **Distribution of Quantization Noise and Weights**. We plot the weights(left), quantization noise(middle) and symmetry(right) of various layers in ResNet-50 model. We found KL-divergence to be [Top **{0.0061, 0.004, 00069, 0.00054}** Bottom] between quantization noise and uniform distribution with same minimum and maximum value.

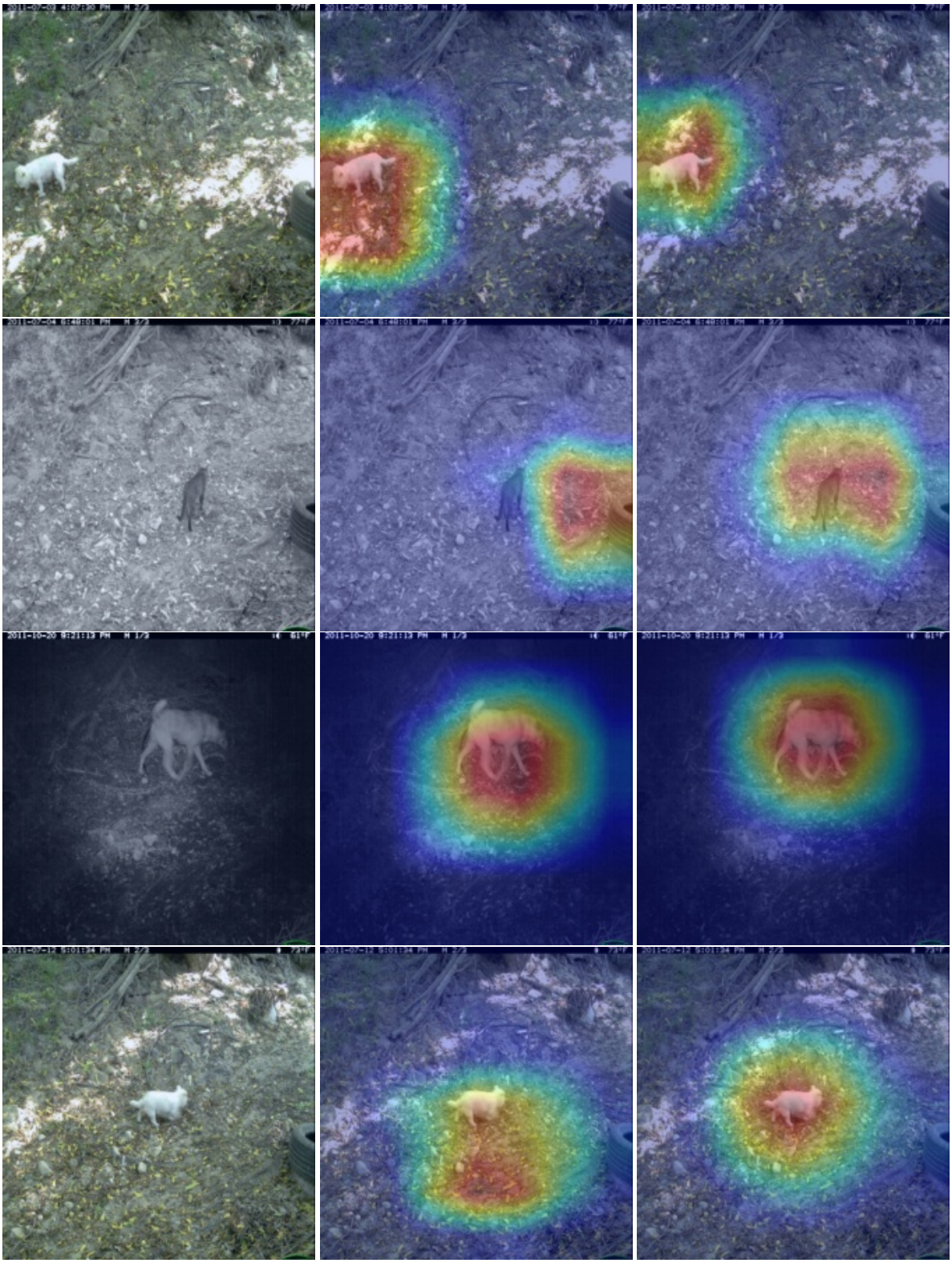

Figure 9. **Visualization of GradCAM results on the Terra Incognito dataset with L38 as test domain.** We show original image, GradCAM with ERM (Gulrajani & Lopez-Paz, 2021) and GradCAM with QT-DoG [Left to Right].

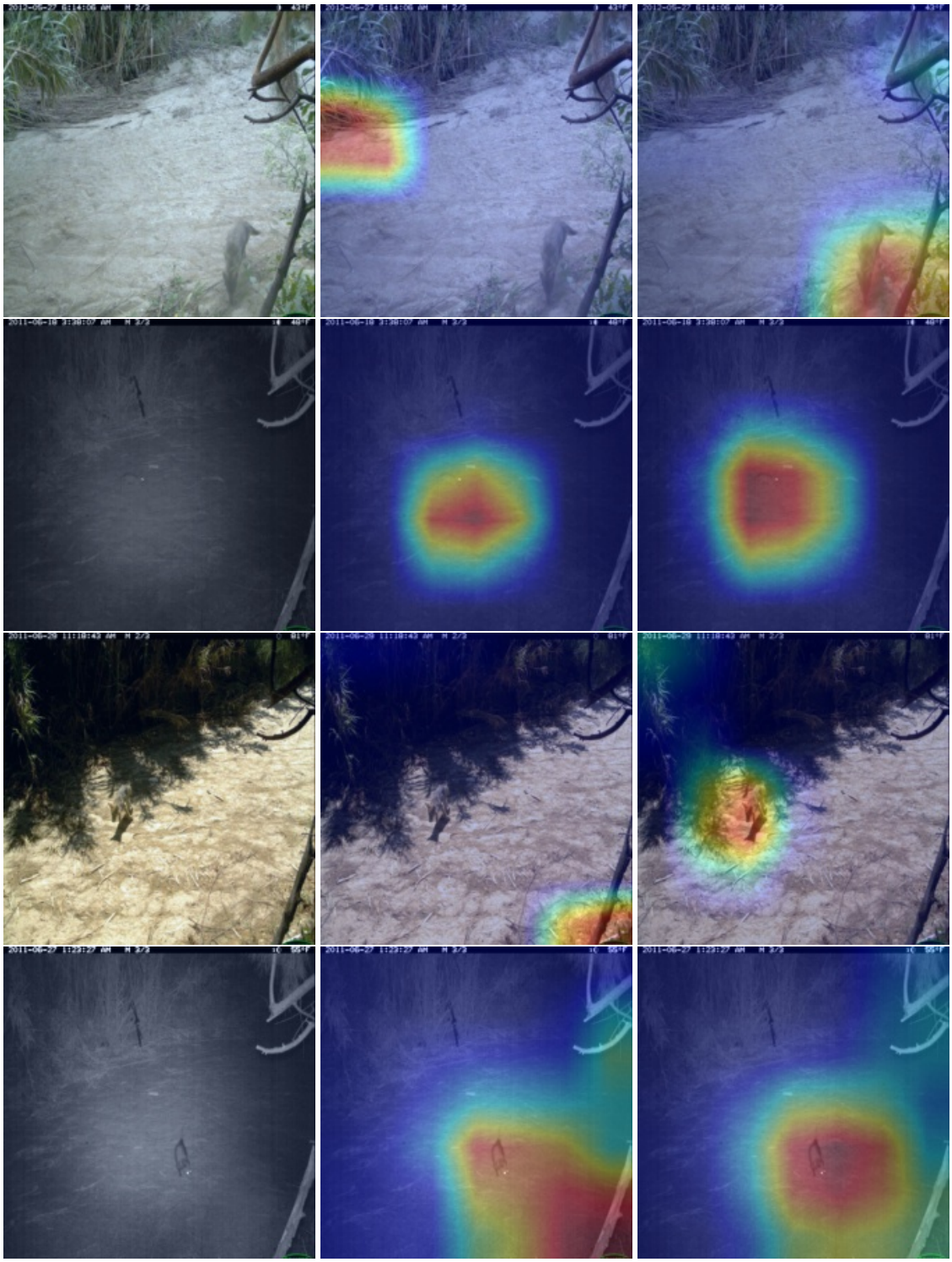

Figure 10. **Visualization of GradCAM results on the Terra Incognito dataset with L46 as test domain.** We show original image, GradCAM with ERM (Gulrajani & Lopez-Paz, 2021) and GradCAM with QT-DoG [Left to Right].

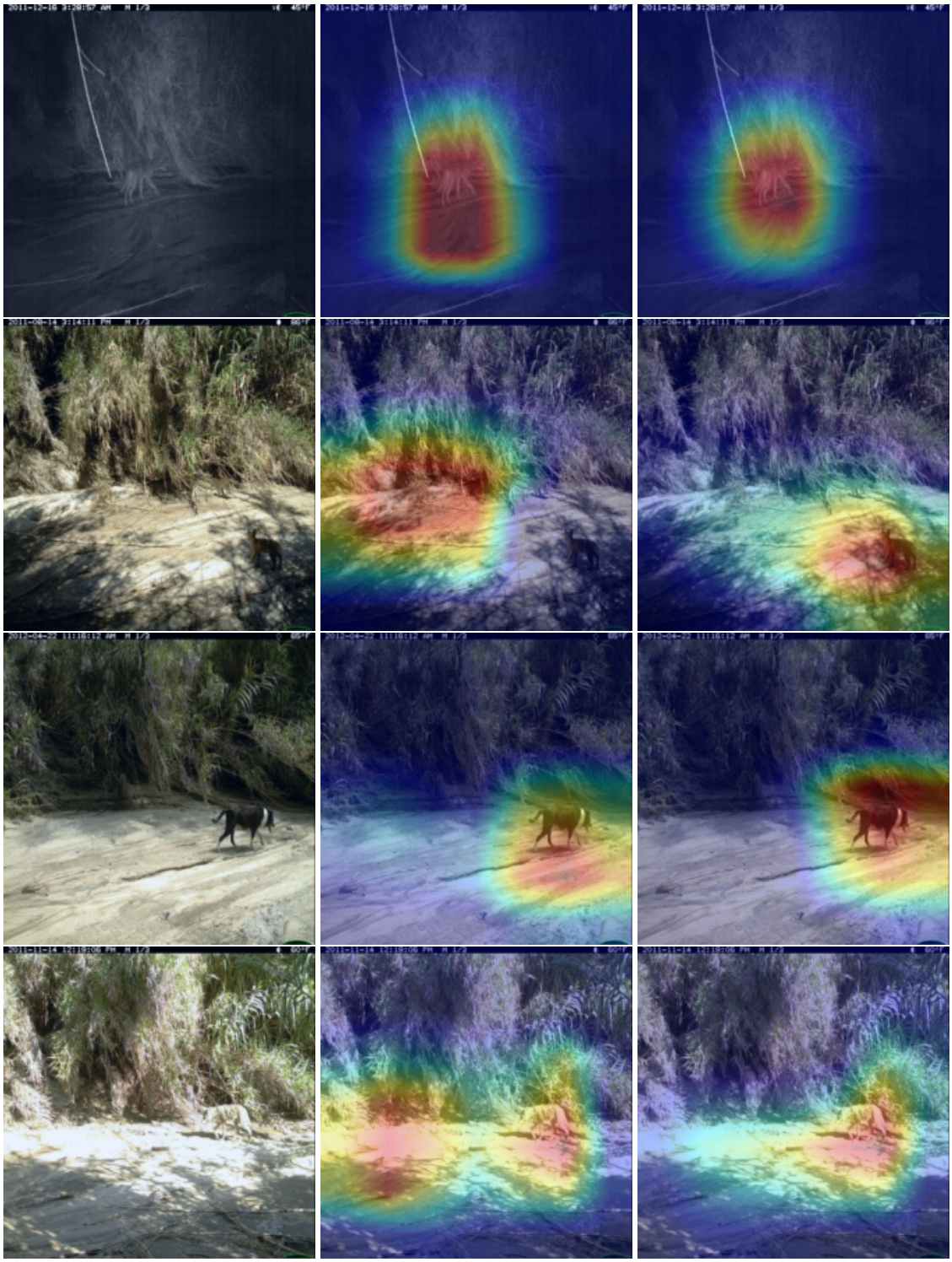

Figure 11. **Visualization of GradCAM results on the Terra Incognito dataset with L43 as test domain.** We show original image, GradCAM with ERM (Gulrajani & Lopez-Paz, 2021) and GradCAM with QT-DoG [Left to Right].

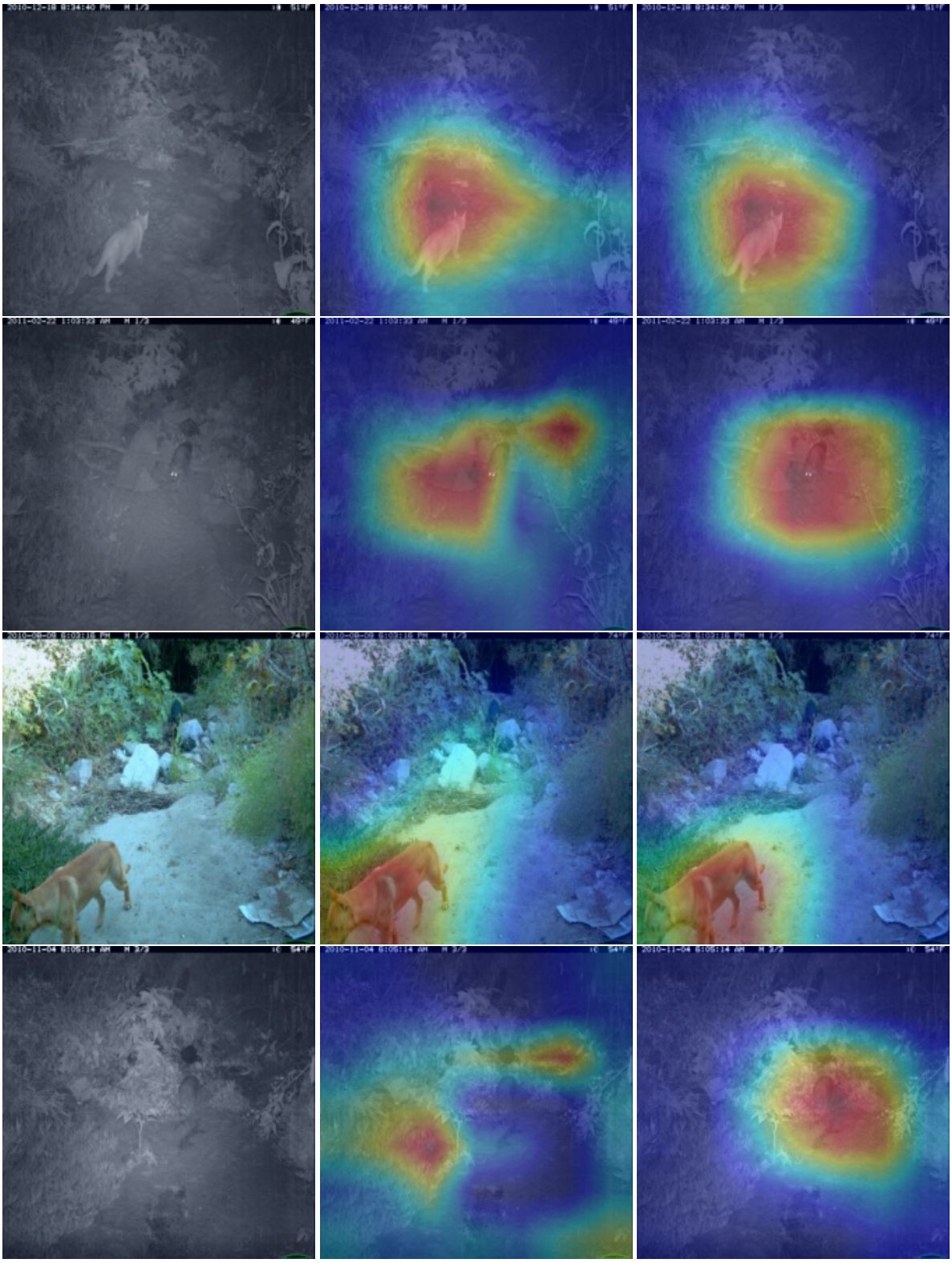

Figure 12. **Visualization of GradCAM results on the Terra Incognito dataset with L100 as test domain.** We show original image, GradCAM with ERM (Gulrajani & Lopez-Paz, 2021) and GradCAM with QT-DoG [Left to Right].