Back to the Source: Diffusion-Driven Adaptation to Test-Time Corruption

Jin Gao*[1], Jialing Zhang*[1], Xihui Liu[3], Trevor Darrell[4], Evan Shelhamer†[5], Dequan Wang†[1,2]
[1]Shanghai Jiao Tong University [2]Shanghai Artificial Intelligence Laboratory
[3]The University of Hong Kong [4]University of California, Berkeley [5]DeepMind

Abstract

Test-time adaptation harnesses test inputs to improve the accuracy of a model trained on source data when tested on shifted target data. Most methods update the source model *by (re-)training on each target domain. While re-training can help, it is sensitive to the amount and order of the data and the hyperparameters for optimization. We update the target* data *instead, and project all test inputs toward the source domain with a generative diffusion model. Our diffusion-driven adaptation (DDA) method shares its models for classification and generation across all domains, training both on source then freezing them for all targets, to avoid expensive domain-wise re-training. We augment diffusion with image guidance and classifier self-ensembling to automatically decide how much to adapt. Input adaptation by DDA is more robust than model adaptation across a variety of corruptions, models, and data regimes on the ImageNet-C benchmark. With its input-wise updates, DDA succeeds where model adaptation degrades on too little data (small batches), on dependent data (correlated orders), or on mixed data (multiple corruptions).*

1. Introduction

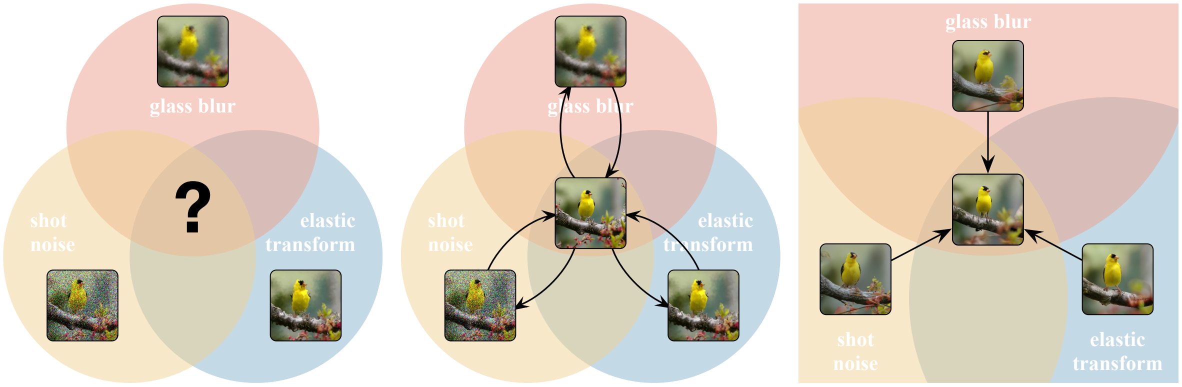
Deep networks achieve state-of-the-art performance for visual recognition [3,8,25,26], but can still falter when there is a *shift* between the source data and the target data for testing [38]. Shift can result from corruption [10, 27]; adversarial attack [7]; or natural shifts between simulation and reality, different locations and times, and other such differences [17, 36]. To cope with shift, adaptation and robustness techniques update predictions to improve accuracy on target data. In this work, we consider two fundamental axes of adaptation: what to adapt—the model or the input—and how much to adapt—using the update or not. We propose a test-time input adaptation method driven by a generative diffusion model to counter shifts due to image corruptions.

The dominant paradigm for adaptation is to train the model by joint optimization over the source and target [6,13, 44, 53, 54]. However, train-time adaptation faces a crucial issue: not knowing how the data may differ during testing. While train-time updates can cope with known shifts, what if new and different shifts should arise during deployment? In this case, test-time updates are needed to adapt the model (1) without the source data and (2) without halting inference. Source-free adaptation [15,19,20,23,51,55] satisfies (1) by re-training the model on new targets without access to the source. Test-time adaptation [46,51,56,58] satisfies (1) and (2) by iteratively updating the model during inference. Although updating the model can improve robustness, these updates have their own cost and risk. Model updates may be too computationally costly, which prevents scaling to many targets (as each needs its own model), and they may be sensitive to different amounts or orders of target data, which may result in noisy updates that do not help or even hinder robustness. In summary, most methods update the source *model*, but this does not improve all deployments.

We propose to update the target *data* instead. Our diffusion-driven adaptation method, DDA, learns a diffusion model on the source data during training, then projects inputs from all targets back to the source during testing. Figure 1 shows how just one source diffusion model enables adaptation on multiple targets. DDA trains a diffusion model to replace the source data, for source-free adaptation, and adapts target inputs while making predictions, for test-time adaptation. Figure 2 shows how DDA adapts the input then applies the source classifier without model updates.

Our experiments compare and contrast input and model updates on robustness to corruptions. For input updates, we evaluate and ablate our DDA and compare it to DiffPure [30], the state-of-the-art in diffusion for adversarial defense. For model updates, we evaluate entropy minimization methods (Tent [56] and MEMO [58]), the state-of-the-art for online and episodic test-time updates, and BUFR [5], the state-of-the-art for source-free offline updates. DDA achieves higher robustness than DiffPure and MEMO across ImageNet-C and helps where Tent degrades

* indicates equal contribution, † indicates corresponding author.



(a) Setting: Multi-Target Adaptation (b) Cycle-Consistent Paired Translation (c) DDA (ours): Many-to-One Diffusion

Figure 1. **One diffusion model can adapt inputs from new and multiple targets during testing**. Our adaptation method, DDA, projects inputs from all target domains to the source domain by a generative diffusion model. Having trained on the source data alone, our source diffusion model for generation and source classification model for recognition do not need any updating, and therefore scale to multiple target domains without potentially expensive and sensitive re-training optimization.

due to limited, ordered, or mixed data. DDA is model-agnostic, by adapting the input, and improves across standard (ResNet-50) and state-of-the-art convolutional (ConvNeXt [26]) and attentional (Swin Transformer [25]) architectures without re-tuning.

Our contributions:

- We propose DDA as the first diffusion-based method for test-time adaptation to corruption and include a novel self-ensembling scheme to choose how much to adapt.
- We identify and empirically confirm weak points for online model updates—small batches, ordered data, and mixed targets—and highlight how input updates address these natural but currently challenging regimes.
- We experiment on the ImageNet-C benchmark to show that DDA improves over existing test-time adaptation methods across corruptions, models, and data regimes.

2. Related Work

Model Adaptation updates the source model on target data to improve accuracy. We focus on source-free adaptation—not needing the source while adapting—and on test-time adaptation—making predictions while adapting—because DDA is a source-free and test-time method.

Source-free adaptation [19, 20, 23, 51] makes it possible to respect practical deployment constraints on computation, bandwidth, and privacy. Nevertheless, most methods involve a certain amount of complexity and computation by altering training [4, 19, 20, 23, 51] and interrupt testing by re-training their model(s) offline on each target [5, 19, 20, 23, 33]. DDA is source-free, as it replaces the source data with source diffusion modeling. However, it differs by updating the data rather than the model(s). Furthermore, it does not alter the training of the classifier, as the diffusion model is trained on its own. By keeping its models fixed, DDA handles multiple targets without halting testing for model re-training, as source-free model adaptation does.

Test-time adaptation [31, 32, 46, 51, 52, 56, 59, 61] simultaneously updates and predicts. Such test-time model updates can be sensitive to their optimization hyperparameters along with the size, order, and diversity of the test data. On the contrary, DDA updates the data, which makes it independent across inputs, and thereby invariant to batches, orders, or mixtures of the test data. DDA can even adapt to a single test input, without augmentation, unlike test-time model adaptation.

Input Adaptation translates data between source and target. DDA adapts the input from target to source by test-time diffusion. Prior methods adapt during testing, but differ in their purpose and technique, or adapt during training, but cannot handle new target domains during testing.

During testing, translation from source to target enables the use of a source-only model. DiffPure [30] is the closest method to DDA because it applies diffusion to defense against the adversarial shift. However, DiffPure and DDA differ in their settings of adversarial and natural shift respectively, and as a result differ in their techniques. DDA differs in its conditioning of the diffusion updates and its self-ensembling of predictions before and after adaptation.

During training, translation from source to target provides additional data or auxiliary losses. Train-time translation includes style transfer [21, 37, 39, 57], conditional image synthesis [13, 14, 16, 34, 35, 40, 62], or adversarial generation [42] for robustness to shift. CyCADA [13] adapts by translating between source and target via generation with CycleGAN [62]. While CyCADA and DDA are generative, CyCADA needs paired source and target data for training, and cannot adapt to multiple targets during testing. DDA only trains one model on source to adapt to multiple targets.

Diffusion Modeling Diffusion [29, 41, 41, 47–50] is a strong, recent approach to generative modeling that samples by iteratively refining the input. In essence, diffusion learns to "reverse" noise to generate an image by gradient updates w.r.t. the input. The type of noise matters, and standard diffusion relies on Gaussian noise. In this work, we investigate how a strong diffusion model can project corrupted target data toward the source data distribution, even on corruptions that are highly non-Gaussian. We apply the denoising diffusion probabilistic model (DDPM) [11] in this new role of diffusion-driven adaptation. Guided diffusion models improve generation by optimization conditioned on class labels [2, 12], text [24, 28], and images [1], but test-time adaptation denies the data needed for their use as-is. DDA improves on the straightforward application of diffusion to achieve higher robustness to corruption during testing.

3. Diffusion-Driven Adaptation to Corruption

We propose diffusion-driven adaptation (DDA) to adopt a diffusion model to counter shifts due to input corruption. During training, we train a generation model (the diffusion model) with the source data, and train a recognition model (the classifier) with the source data and its labels. During inference, taking an example from the target domain as input, the diffusion model projects it back to the source domain, and then the classifier makes a prediction on the projected image. Figure 2 illustrates the projection and prediction steps of DDA inference.

Our DDA approach does not need any target data during training, and is able to accept arbitrary unknown target inputs during testing. Notably, this enables inference on a single image from the target domain. In contrast, previous model adaptation approaches, such as Tent [56] and BUFR [5], degrade on too little data (small batches), on dependent data (non-random order), or on mixed data (multiple corruptions). See Sec. 4.3 for our examination of these data regimes. In this way, DDA addresses practical deployments that are not already handled by model adaptation.

3.1. Background: Diffusion for Image Generation

Diffusion models have recently achieved state-of-the-art image generation by iteratively refining noise into samples from the data distribution. Given an image sampled from the real data distribution $x_0 \sim q(x_0)$, the forward diffusion process defines a fixed Markov chain, to gradually add Gaussian noise to the image x_0 over T timesteps, producing a sequence of noised images $x_1, x_2, \cdots, x_T$. Mathematically, the forward process is defined as

$$\begin{aligned} q(x_{1:T}|x_0) &:= \prod_{t=1}^{T} q(x_t|x_{t-1}), \\ q\left(x_t \mid x_{t-1}\right) &:= N\left(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}\right), \end{aligned} \tag{1}$$

where the sequence, $\beta_1, ..., \beta_T$, is a fixed variance schedule to control the step sizes of the noise.

We can further sample x_t from x_0 in a closed form,

$$q(x_t|x_0) := \sqrt{\overline{\alpha}_t} x_0 + \epsilon\sqrt{1-\overline{\alpha}_t}, \epsilon \sim \mathcal{N}(0,1), \tag{2}$$

where $\alpha_t := 1 - \beta_t$ and $\overline{\alpha_t} := \prod_{s=1}^{t} \alpha_s$.

On the other hand, given the Gaussian noise sampled from the distribution $X_T \sim \mathcal{N}(0, \mathbf{I})$, the reverse diffusion process iteratively removes the noise to generate an image in T timesteps. The reverse process is formulated as a Markov chain with Gaussian transitions:

$$\begin{aligned} &p(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p(x_{t-1}|x_t), \\ &p_\theta\left(x_{t-1} \mid x_t\right) := N\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \sigma_t^2\left(x_t, t\right) \mathbf{I}\right). \end{aligned} \tag{3}$$

Denoising diffusion probabilistic models (DDPM) [11] set $\sigma_t\left(x_t, t\right) = \sigma_t \mathbf{I}$ to time-dependent constants. μ_θ is parameterized by a linear combination of x_t and $\epsilon_\theta(x_t, t)$, where $\epsilon_\theta(x_t, t)$ is a function that predicts the noise. The parameters of $\mu_\theta\left(x_t, t\right)$ are optimized by the variational bound on the negative log-likelihood $\mathbb{E}[-\log p_\theta(x_0)]$. With this parameterization and following DDPM [11], the training loss $\mathcal{L}_{\text{simple}}$ simplifies to the mean-squared error between the actual noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ in x_t and the predicted noise

$$\mathcal{L}_{\text{simple}} := ||\epsilon_\theta(x_t, t) - \epsilon||^2. \tag{4}$$

Since their loss derives from a bound on the negative log-likelihood $\mathbb{E}[-\log p_\theta(x_0)]$, diffusion models are optimized to learn a generative prior of the training data.

3.2. Diffusion for Input Adaptation

We now detail our diffusion-driven adaptation method. A diffusion model is trained on the source domain to learn a generative prior of the input distribution for a source classifier. Once trained, it can be applied to project single/multi-target domain data to the source domain, by running the forward process followed by the reverse process.

Given an input image x_0 from the target domain and an unconditional diffusion model trained on the source domain, we first run the forward process (Eqn. 2, the green arrow in Fig. 2) of the diffusion model, *i.e.*, perturb the image with Gaussian noise. We denote the image sequence derived by N forward steps as $x_0, x_1, \cdots, x_N$, where N is a hyper-parameter ("diffusion range") controlling the amount

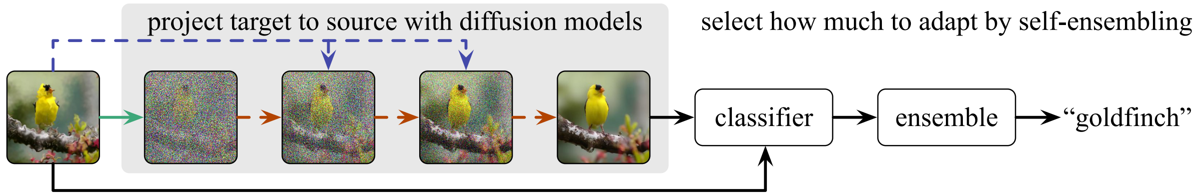

project target to source with diffusion models
select how much to adapt by self-ensembling
classifier
ensemble
“goldfinch”

Figure 2. **DDA projects target inputs back to the source domain.** Adapting the input during testing enables direct use of the source classifier without model adaptation. The projection adds noise (forward diffusion, green arrow) then iteratively updates the input (reverse diffusion, red arrow) with conditioning on the original input (guidance, purple arrow). For reliability, we ensemble predictions with and without adaptation depending on their confidence.

Algorithm 1 Diffusion-Driven Adaptation

```
Input: Reference image $x_0$
Output: Generated image $x_0^g$
$N$: diffusion range, $\phi_D(\cdot)$ : low-pass filter of scale D
Sample $x_N \sim q\left(x_N \mid x_0\right)$                ▷ perturb input
$x_N^g \leftarrow x_N$
for $t \leftarrow N \ldots 1$ do
    $\hat{x}_{t-1}^g \sim p_\theta\left(x_{t-1}^g \mid x_t^g\right)$        ▷ unconditional proposal
    $\hat{x}_0^g \leftarrow \sqrt{\frac{1}{\bar{\alpha}_t}} x_t^g - \sqrt{\frac{1}{\bar{\alpha}_t} - 1}\boldsymbol{\epsilon}_\theta(x_t^g, t)$
    $x_{t-1}^g \leftarrow \hat{x}_{t-1}^g - \boldsymbol{w}\nabla_{x_t}\left\|\phi_D\left(x_0\right) - \phi_D\left(\hat{x}_0^g\right)\right\|_2$
end for
return $x_0^g$
```

of noise added to the input image. Then the reverse process (Eqn. 3, the red dotted arrow in Fig. 2) starts with the noised image x_N, then removes noise for N steps to generate the denoised image sequence $x_{N-1}^g, x_{N-2}^g, \cdots, x_0^g$. Since the diffusion model has learned a generative prior of the source domain, the generated image x_0^g should be more likely under the distribution of the source data.

However, we notice a trade-off between preserving classes while translating domains when choosing different diffusion ranges N. If N is too large and too much noise is added to the image, the diffusion model will not be able to preserve the class information in the input image. On the contrary, if N is too small and too little noise is added, there are not enough diffusion steps to project images from the target to the source. Our goal is to translate the domain from target to source, while preserving the class information as much as possible. Unfortunately, class and domain information are commonly entangled with each other, making it difficult to find a trade-off for sufficient domain translation and class preservation.

To address this trade-off, we provide structural guidance during the reverse process. We design an iterative latent refinement step (denoted by the purple dotted arrow in Fig. 2) conditioned on the input image in the reverse process, so that the image structure and class information can be preserved when translating images across domains.

Inspired by ILVR [1], we add a linear low-pass filter implemented by $\phi_D(\cdot)$, a sequence of downsampling and upsampling operations with a scale factor of D, to capture the image-level structure. We iteratively update the diffusion sample x_t^g to reduce the structural difference of generated sample as measured by D.

At each step of reverse process, we can obtain an estimate of x_0, $\hat{x}_0^g$, from the noisy image at the current step x_t^g.

$$\hat{x}_0^g = \sqrt{\frac{1}{\bar{\alpha}_t}} x_t^g - \sqrt{\frac{1}{\bar{\alpha}_t} - 1}\boldsymbol{\epsilon}_\theta(x_t^g, t). \tag{5}$$

Therefore, we can avoid conflicting with the diffusion update by using the direction of similarity between the reference image x_0 and $\hat{x}_0^g$, not the one between x_t and x_t^g. At each step t in the reverse process, we force x_t^g to move in the direction that decreases the distance between $\phi_D(x_0)$ and $\phi_D(\hat{x}_0^g)$:

$$x_{t-1}^g = \hat{x}_{t-1}^g - \boldsymbol{w}\nabla_{x_t}\left\|\phi_D\left(x_0\right) - \phi_D\left(\hat{x}_0^g\right)\right\|_2, \tag{6}$$

with a scaling hyperparameter $\boldsymbol{w}$ to control the step size of guidance. For simplicity, we neglect the difference between t and $t-1$ and update $\hat{x}_{t-1}^g$ based on x_t^g's gradient, and spare an extra reverse step.

In summary, we first perturb the input image from the target domain with noise in the forward process of the diffusion model, and then in the reverse process, we adapt the input with iterative guidance to generate an image that is more like source data without altering the class information too much. Algorithm 1 outlines the projection of target data back to the source by diffusion.

3.3. Self-Ensembling Before & After Adaptation

Adapting target inputs back to the source by diffusion helps our source-trained recognition model to make more reliable predictions. In most cases, diffusion generates an image that improves accuracy, because it has preserved the class information while projecting out the target shift (at least partially). However, the diffusion model is not perfect, and can sometimes generate an image that is less recognizable to the classifier than the original target input.

Motivated by this possibility, we propose a self-ensembling scheme to aggregate the prediction results from the original and adapted inputs. Since we have the test input x_0 and adapted input x_0^g from diffusion, we can run the classification model on both images. We make the final prediction based on the average confidence of both, *i.e.*, $\arg\max_c \frac{1}{2}\left(p_c + p_c^g\right)$, where $c \in \{1, \ldots, C\}$, and the confidence of the C categories is $p \in \mathbb{R}^C$ and $p^g \in \mathbb{R}^C$.

This self-ensembling scheme enables the automatic selection of how much to weigh the original and adapted inputs to further increase robustness.

4. Experiments

4.1. Setup

We summarize the data, settings, adaptation methods, and classification models studied in our experiments. Full implementation detail is provided by the code in the supplementary material and the documentation of hyperparameters in the Appendix.

Datasets ImageNet-C (IN-C) [10] and ImageNet-$\overline{\text{C}}$ (IN-$\overline{\text{C}}$) [27] are standard benchmarks for robust large-scale image classification. They consist of synthetic but natural corruptions (noise, blur, digital artifacts, and different weather conditions) applied to the ImageNet [43] validation set of 50,000 images. IN-C has 15 corruption types at 5 severity levels. IN-$\overline{\text{C}}$ has 10 more corruption types, selected for their dissimilarity to IN-C, at 5 severity levels. We measure robustness as the top-1 accuracy of predictions on the most severe corruptions (level 5) on IN-C and IN-$\overline{\text{C}}$. We evaluate DDA with the same hyperparameters across each dataset except as noted for ablation and analysis.

Adaptation Settings We consider two settings with more and less knowledge of the target domains. *Independent* adaptation is the standard setting for robustness experiments on ImageNet-C, where adaptation and evaluation are done independently for each corruption type. *Joint* adaptation is a more realistic and difficult setting, where adaptation and evaluation are done jointly over all corruptions by combining their data. Experimenting with both settings allows standardized comparison with existing work and exploration of adaptation without knowledge of target domain boundaries.

Methods We compare DDA to an ablation without self-ensembling, model adaptation by MEMO [58] and Tent [56], and input adaptation by the adversarial defense DiffPure [30]. MEMO adapts to each input by augmentation and entropy minimization: it minimizes the entropy of the predictions w.r.t. the model parameters over different augmentations of the input, then resets. By relying on data augmentation, MEMO avoids trivial solutions to optimizing so many parameters on a single input. Tent adapts on batches of inputs by updating a small number of statistics and parameters by entropy minimization, but unlike MEMO it does not reset, and its updates compound across batches. DiffPure and DDA rely on the same unconditional diffusion model [2] but differ in their reverse steps and guidance. DiffPure simply adds a given amount of noise ($t = 150$) and then reverses to $t = 0$.

Table 1. **DDA is more robust in the episodic setting.** Episodic inference is independent across inputs, and includes the source-only model without adaptation, model updates by MEMO, and input updates by DiffPure and DDA (ours). We evaluate accuracy on standard ImageNet and the corruptions of ImageNet-C.

	IN	ImageNet-C Accuracy			
Model	Acc.	Source-Only	MEMO	DiffPure	DDA
ResNet-50	76.6	18.7	24.7	16.8	**29.7**
Swin-T	81.2	33.1	29.5	24.8	**40.0**
ConvNeXt-T	82.1	39.3	37.8	28.8	**44.2**
Swin-B	83.4	40.5	37.0	28.9	**44.5**
ConvNeXt-B	83.9	45.6	45.8	32.7	**49.4**

Classifiers We experiment with multiple classifiers to assess general improvement. We select ResNet-50 [8] as a standard architecture, plus Swin [25] and ConvNeXt [26] to evaluate the state-of-the-art in attentional and convolutional architectures. Experimenting with Swin and ConvNeXt sharpens our evaluation of adaptation as these architectures already improve robustness.

4.2. Benchmark Evaluation: Independent Targets

Input updates are more robust than model updates with episodic adaptation. We begin by evaluating source-only inference (without adaptation), model adaptation with MEMO, and input adaptation with DiffPure or our DDA. Each method is "episodic", in making independent predictions for each input, for a fair comparison. Table 1 summarizes each source classifier and compares the robustness of each method. DDA achieves consistently higher robustness than MEMO and DiffPure. On the latest Swin-T and ConvNeXt-T models DDA still delivers a ∼5 point boost.

DDA consistently improves on IN-C corruption without catastrophic failure. Figure 3 analyzes robustness across each corruption type of IN-C. DDA is the most robust overall, although DDA without self-ensembling can improve over the source-only model on most high-frequency corruptions. As for low-frequency corruptions, our self-ensembling automatically selects how much to adapt, and

Table 2. **Diffusion vs. Other Corruptions.** We measure robustness to corruption on ImageNet-$\overline{\mathrm{C}}$, which is designed to differ from ImageNet-C, by accuracy at maximum severity (level 5).

Method	ResNet-50	Swin-T	ConvNeXt-T
Source-Only	25.8	44.2	47.2
DiffPure [30]	19.8	28.5	32.1
DDA (ours)	29.4	43.8	46.3

compensates for the current failures of diffusion to avoid drops on more global corruptions like fog and contrast.

Although DiffPure likewise adapts the input by diffusion, its specialization to adversarial attacks makes it unsuitable for input corruptions. Its average accuracy on IN-C is worse than the accuracy without adaptation. This drop underlines the need for the particular design choices of DDA that specialize it to natural shifts like corruptions, which are unlike the norm-bounded attacks DiffPure is designed for.

DDA is not sensitive to small batches or ordered data. The amount and order of the data for each corruption may vary in practical settings. For the amount, source-free methods use the entire test set at once, while test-time methods may choose different batch sizes. For the order of the target data, it is commonly shuffled (as done by Tent and other test-time methods). We evaluate at different batch sizes, with and without shuffling, to understand the effect of these data regimes. Figure. 4 plots sensitivity to these factors. DDA, MEMO, and DiffPure are totally unaffected, being episodic, but Tent is extremely sensitive. Controlling the amount and order of data during deployment may not always be possible, but Tent requires it to ensure improvement.

DDA maintains accuracy on the corruptions of IN-$\overline{\mathrm{C}}$. Table 2 compares input adaptation by DiffPure and DDA on IN-$\overline{\mathrm{C}}$. These corruption types are more difficult, as they are designed and selected to differ from natural images and the corruptions of IN-C. While DDA does not improve robustness in this case, it averts the large drops caused by DiffPure, which are even larger than its drops on IN-C.

4.3. Challenge Evaluation: Joint Targets

The joint adaptation setting combines the data for all corruption types to present a new challenge. The amount, order, and mixture of the data can be varied to complicate adaptation for methods that depend on the batching or ordering of domains. DDA and MEMO can both address small batches, ordered data, and mixed domains, because they are episodic methods, which adapt to each input independently. However, non-episodic methods like Tent have no such guarantee, because of its cumulative updates across inputs.

Table 3. **DDA is reliably more robust when the target data is limited, ordered, or mixed.** Deployment may supply target data in various ways. To explore these regimes, we vary batch size and whether or not the data is ordered by class or mixed across corruption types. We compare episodic adaptation by input updates with DDA (ours) and by model updates with MEMO against cumulative adaptation with Tent. DDA and MEMO are invariant to these differences in the data. However, Tent is highly sensitive to batch size and order, and fails in the more natural data regimes.

Method	Mixed Classes	Mixed Types	Batch Size	ResNet-50	Swin-T	ConvNeXt-T
Source-Only	N/A		N/A	18.7	33.1	39.3
MEMO [58]				24.7	29.5	37.8
DiffPure [30]				16.8	24.8	28.8
DDA (ours)				**29.7**	**40.0**	**44.2**
Tent [56]	✗	✗	1 / 64	2.2 / 0.4	0.2 / 0.2	0.1 / 1.4
	✗	✓	1 / 64	1.6 / 0.5	0.2 / 0.5	0.3 / 1.6
	✓	✗	1 / 64	3.0 / **7.6**	0.1 / 43.3	0.2 / 48.8
	✓	✓	1 / 64	2.3 / 3.9	0.3 / **44.1**	0.3 / **51.9**

DDA is more robust on joint targets where Tent and other cumulative updates degrade. Table 3 compares episodic adaptation by DDA, MEMO, and DiffPure with cumulative adaptation by Tent in the joint setting. The reported results are an average under multiple experiments to avoid randomness though we find that there is almost no difference among different seeds.

While the episodic methods are all invariant to the joint setting, this is not the case for Tent. Tent can adapt the best when its assumptions of large enough batches and randomly ordered data are met, but it can otherwise harm robustness. In contrast, the accuracy of DDA is independent of batch size and data order, and helps robustness in each setting.

For further comparison to model updates in the joint setting, we evaluate batch normalization (BN) on the target data [46] and source-free adaptation of feature histograms by BUFR [5]. We evaluate with ResNet-50, as it is a standard architecture for IN-C, and the focus of [46]. Although BN is competitive in the independent setting, sharing the mean and variance across all corruptions in the joint setting cannot adapt well: it achieves worse than source model performance at 10.3% accuracy vs. the 29.7% accuracy of DDA. BUFR does not report results with ImageNet scale, nor with ResNet-50, and our tuning could not achieve better than source-only accuracy.

4.4. Analysis & Ablation of Diffusion Updates

Timing As diffusion models are computationally intense, we compare the time for model adaptation by MEMO and input adaptation by DiffPure and DDA. We measure the wall clock time for single input inference with ResNet-50 on the same hardware (GeForce RTX 2080 Ti) and average

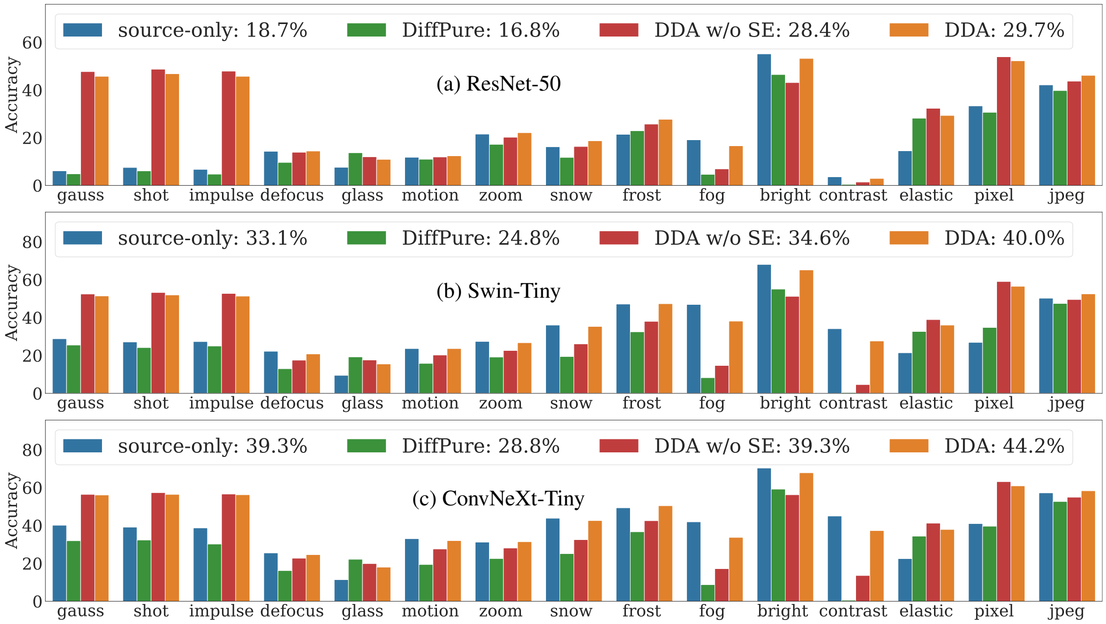


Figure 3. **DDA reliably improves robustness across corruption types.** We compare DDA with the source-only model, state-of-the-art diffusion for adversarial defense (DiffPure), and a simple ablation of DDA (DDA w/o Self-Ensembling (SE)). DDA is the best on average, strictly improves on DiffPure, and improves on simple diffusion in most cases. Our self-ensembling prevents catastrophic drops (on fog or contrast, for example).

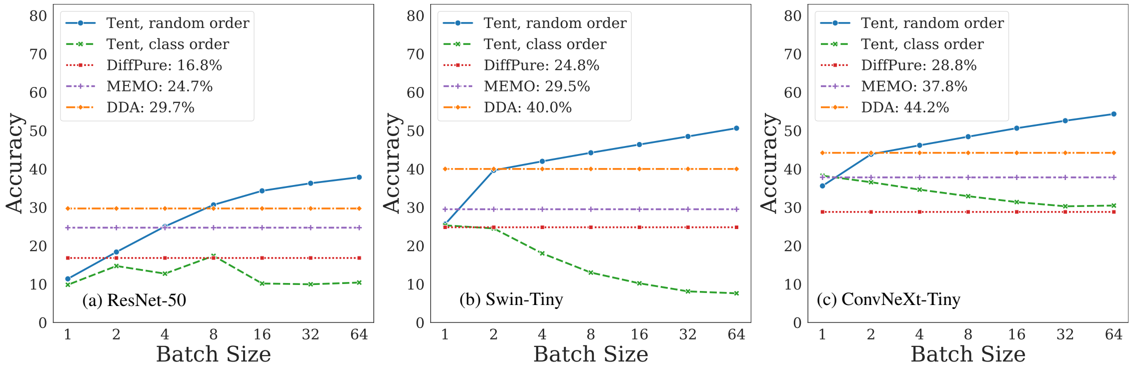


Figure 4. **DDA is invariant to batch size and data order while Tent is extremely sensitive.** To analyze sensivity to the amount and order of the data we measure the average robustness of independent adaptation across corruption types. DDA does not depend on these factors and consistently improves on MEMO. Tent fails on class-ordered data without shuffling and degrades at small batch sizes.

over the test set of $50,000$ inputs. Table 4 reports our profiling. While this experimentally verifies the current cost of diffusion modeling, it underlines the importance of design choices: DDA is more robust to corruption and faster than DiffPure. Furthermore, we confirm the potential for speed-up by applying accelerated sampling with DEIS [60].

Ablation We ablate the different diffusion steps that update the input. As described in Sec. 3.2, our diffusion-driven adaptation method is composed of a forward process, reverse process, and guidance. We experiment with three settings as follows: (1) We first run the forward process (*i.e.*, add Gaussian noise) on the input image and then run the reverse process of the diffusion model to denoise, without our iterative guidance. This setting is denoted as "*forward+reverse*". (2) We start from a random noise and run the reverse process of the diffusion model with iterative

Table 4. **DDA balances time and robustness.** Diffusion by DDA or DiffPure is slower than entropy minimization by MEMO, but DDA is the most robust and faster than DiffPure. Accelerating diffusion by DEIS can trade time and robustness for DDA.

	MEMO [58]	DiffPure [30]	DDA (ours)	DDA (+DEIS)
Runtime (s)	0.7	31.7	13.5	2.4
IN-C Acc. (%)	24.7	16.8	29.7	27.0

Table 5. **Ablation of diffusion updates justifies each step.** We ablate the forward, reverse, and refinement updates of DDA. We omit self-ensembling to focus on the input updates. Forward adds noise, reverse denoises by diffusion, and refinement guides the reverse updates. DDA is best with all steps, but forward and reverse or reverse and refinement help on their own.

		ResNet-50	Swin-T	ConvNeXt-T
(a) elastic	forward+reverse	24.5	24.9	25.8
	reverse+guidance	17.7	23.0	24.8
	DDA (ours)	**32.3**	**38.9**	**41.2**
(b) blur	forward+reverse	13.9	14.4	15.0
	reverse+guidance	7.6	11.5	13.4
	DDA (ours)	**12.0**	**17.6**	**19.9**
(c) noise	forward+reverse	19.5	20.2	21.0
	reverse+guidance	20.7	24.0	26.9
	DDA (ours)	**48.7**	**53.2**	**57.3**

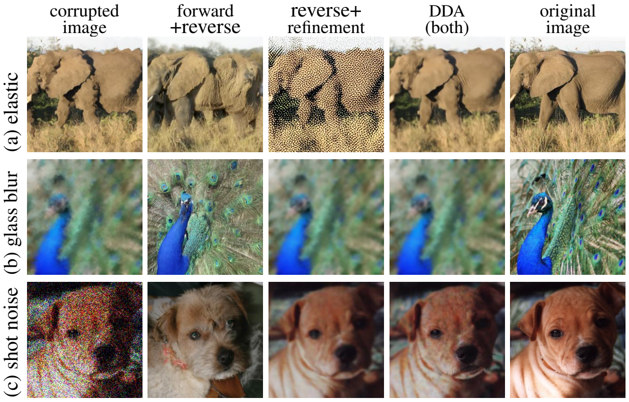


Figure 5. **Visualization of updates for ablations of diffusion.**

guidance, which we denote as "*reverse+guidance*". (3) Our DDA model combines both, *i.e.*, we run the forward process on the input image and then run the reverse process of the diffusion model with iterative guidance. Figure 5 shows the performance of "forward-reverse", "reverse-guidance", and our DDA approach which includes the forward process, reverse process, and iterative guidance. The results demonstrate that each step contributes to the robustness of adaptation.

5. Discussion

DDA mitigates shift by test-time input adaptation with diffusion modeling. Our experiments on ImageNet-C confirm that diffusing target data back to the source domain improves robustness. In contrast to test-time model adaptation, which can struggle with scarce, ordered, and mixed data, our method is able to reliably boost accuracy in these regimes. In contrast to source-free model adaptation, which can require re-training to each target, we are able to freely scale adaption to multiple targets by keeping our source models fixed. These practical differences are due to our conceptual shift from model adaptation to input adaptation and our adoption of diffusion modeling.

Having examined whether to adapt by input updates or model updates, we expect that reconciling the two will deliver more robust generalization than either alone.

Limitations The strengths and weaknesses of input adaptation complement those of modal adaptation. Although our method can adapt to a single target input, it must adapt from scratch on each input, and so its computation cannot be amortized across the deployment. In contrast, model adaptation by TTT [51] or Tent [56] can update on each batch while cumulatively adapting the model more and more. Although diffusion can project many targets to the source data, and does so without expensive model re-training, it can fail on certain shifts. If these shifts arise gradually, then model adaptation could gradually update too [18], but our fixed diffusion model cannot.

We rely on diffusion, and so we are bound to the quality of generation by diffusion. Diffusion does have its failure modes, even though our positive results demonstrate its present use and future potential. In particular, diffusion models may not only translate domain attributes but other image content, given their large model capacity. Our use of image guidance helps avoid this, but at the cost of restraining adaptation on certain corruptions. New diffusion architectures or new guidance techniques specific to adaption could address these shortcomings.

At present, diffusion takes more computation time than classification, so ongoing work to accelerate diffusion is needed to reduce inference time [45]. Our design choices bring DDA to 19× the time as MEMO, while DiffPure takes $\sim 45\times$ the time, but both diffusion methods are still slower than model updates. Accelerating diffusion sampling by DEIS [60] reduces the time to $<4\times$ but sacrifices $\sim$3 points of robustness. Further speed-up may require more fundamental changes to diffusion sampling and training.

Societal Impact While our work seeks to mitigate dataset shift, we must nevertheless remain aware of dataset bias. Because our diffusion model is trained entirely on the source data, biases in the data may be reflected or amplified by the learned model. Having learned from biased data, the diffusion model is then liable to project target data to whatever biases are present, and may in the process lose important or sensitive attributes of the target data. While this is a serious concern, diffusion-driven adaptation at least allows for interpretation and monitoring of the translated images, since it adapts the input rather than the model. Even so, making good use of this capacity requires diligence and more research into automated analyses of generated images.

References

[1] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *ICCV*, 2021. 3, 4

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 3, 5

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 13

[4] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14340–14349, 2021. 2

[5] Cian Eastwood, Ian Mason, Chris Williams, and Bernhard Schölkopf. Source-free adaptation to measurement shift via bottom-up feature restoration. In *ICLR*, 2021. 1, 2, 3, 6, 14

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 1

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5, 13

[9] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021. 16

[10] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 1, 5, 15

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021. 3

[13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 1, 2

[14] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2

[15] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *NeurIPS*, 2021. 1

[16] Liming Jiang, Changxu Zhang, Mingyang Huang, Chunxiao Liu, Jianping Shi, and Chen Change Loy. Tsit: A simple and versatile framework for image-to-image translation. In *ECCV*, 2020. 2

[17] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021. 1

[18] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *ICML*, 2020. 8

[19] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *CVPR*, 2020. 1, 2

[20] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020. 1, 2

[21] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *ECCV*, 2018. 2

[22] Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. *arXiv preprint arXiv:2212.04825*, 2022. 15

[23] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020. 1, 2, 13

[24] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *WACV*, 2023. 3

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1, 2, 5, 13

[26] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 1, 2, 5, 13

[27] Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. In *NeurIPS*, 2021. 1, 5, 15

[28] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 3

[29] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 3

[30] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *ICML*, 2022. 1, 2, 5, 6, 7, 12, 14

[31] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022. 2

[32] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023. 2

[33] Prashant Pandey, Mrigank Raman, Sumanth Varambally, and Prathosh AP. Generalization on unseen domains via inference-time label-preserving target projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12924–12933, June 2021. 2

[34] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *ECCV*, 2020. 2

[35] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 2

[36] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 1

[37] François Pitié, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1-2):123–137, 2007. 2

[38] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. MIT Press, Cambridge, MA, USA, 2009. 1

[39] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 2

[40] Stephan R Richter, Hassan Abu Al Haija, and Vladlen Koltun. Enhancing photorealism enhancement. *TPAMI*, 2022. 2

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[42] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *ECCV*, 2020. 2

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 5

[44] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 1

[45] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2021. 8

[46] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020. 1, 2, 6

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[48] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 3

[49] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *NeurIPS*, 2020. 3

[50] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 3

[51] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. Test-time training for out-of-distribution generalization. In *ICML*, 2020. 1, 2, 8

[52] Yushun Tang, Ce Zhang, Heng Xu, Shuoshuo Chen, Jie Cheng, Luziwei Leng, Qinghai Guo, and Zhihai He. Neuro-modulated hebbian learning for fully test-time adaptation. *arXiv preprint arXiv:2303.00914*, 2023. 2

[53] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 1

[54] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1

[55] Thomas Varsavsky, Mauricio Orbes-Arteaga, Carole H Sudre, Mark S Graham, Parashkev Nachev, and M Jorge Cardoso. Test-time unsupervised domain adaptation. In *MICCAI*, 2020. 1

[56] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 1, 2, 3, 5, 6, 8, 12, 13, 14

[57] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *ICCV*, 2019. 2

[58] Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *NeurIPS*, 2021. 1, 5, 6, 7, 14

[59] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. In *NeurIPS*, 2021. 2

[60] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022. 7, 8

[61] Aurick Zhou and Sergey Levine. Bayesian adaptation for covariate shift. *Advances in Neural Information Processing Systems*, 34:914–927, 2021. 2

[62] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2