Leveraging Proxy of Training Data for Test-Time Adaptation

Juwon Kang [1] **Nayeong Kim** [1] **Donghyeon Kwon** [1] **Jungseul Ok** [1 2] **Suha Kwak** [1 2]

Abstract

We consider test-time adaptation (TTA), the task of adapting a trained model to an arbitrary test domain using unlabeled input data *on-the-fly* during testing. A common practice of TTA is to disregard data used in training due to large memory demand and privacy leakage. However, the training data are the only source of supervision. This motivates us to investigate a proper way of using them while minimizing the side effects. To this end, we propose two lightweight yet informative *proxies* of the training data and a TTA method fully exploiting them. One of the proxies is composed of a small number of images synthesized (hence, less privacy-sensitive) by data condensation which minimizes their domain-specificity to capture a general underlying structure over a wide spectrum of domains. Then, in TTA, they are translated into labeled test data by stylizing them to match styles of unlabeled test samples. This enables virtually supervised test-time training. The other proxy is inter-class relations of training data, which are transferred to target model during TTA. On four public benchmarks, our method outperforms the state-of-the-art ones at remarkably less computation and memory.

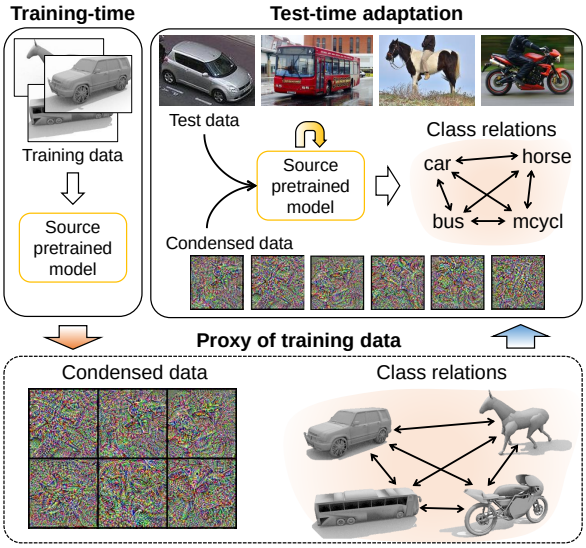


Figure 1. An overview of our test-time adaptation (TTA) framework. Before TTA, it (1) condenses training data into a tiny number of synthetic images and (2) extracts inter-class relations the training data exhibit. During TTA, the condensed data are used as labeled data for test-time training and the inter-class relations are used to regularize predictions of the model.

1. Introduction

Supervised learning with large-scale training data has driven remarkable improvement in a variety of machine learning tasks. In spite of its great success, however, it often suffers from limited generalization performance due to the distribution shift between training and test data (Long et al., 2015; Ganin & Lempitsky, 2015; Li et al., 2017; Dai & Van Gool, 2018; Hendrycks & Dietterich, 2019). In real-world deployment, such a distribution shift is inevitable since it is practically impossible to collect and annotate data for all possible environments in advance of training. A large body of research has alleviated the distribution shift problem via domain adaptation and generalization. Although they have provided impressive performance gain on realistic benchmarks, there is still a large gap between their problem settings and practical application scenarios: Domain adaptation relies on an impractical assumption that test domain data are available in training (Long et al., 2015; Sun & Saenko, 2016; Tzeng et al., 2017; Ganin & Lempitsky, 2015; Tsai et al., 2018; Chang et al., 2019), and domain generalization does not exploit test data at all although they are available in testing (Muandet et al., 2013; Li et al., 2017; Huang et al., 2020; Iwasawa & Matsuo, 2021; Choi et al., 2021; Shi et al., 2021; Kang et al., 2022).

To address these limitations of domain adaptation and generalization, test-time adaptation (TTA) has been proposed recently (Sun et al., 2020; Liu et al., 2021; Sarkar et al., 2022; Wang et al., 2021; Iwasawa & Matsuo, 2021; Wang

[1]Department of Computer Science and Engineering, POSTECH, Pohang, Korea [2]Graduate school of Artificial Intelligence, POSTECH, Pohang, Korea. Correspondence to: Suha Kwak <suha.kwak@postech.ac.kr>.

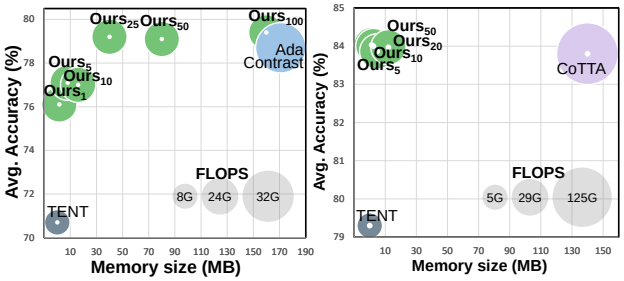
Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

et al., 2022b; Chen et al., 2022; Boudiaf et al., 2022; Shin et al., 2022). The goal of TTA is to adapt a trained model to the test domain using unlabeled input data *during testing*, and a convention in the literature is to disregard training data in that adaptation process. This practice makes sense since it is usually impractical to distribute a model along with training data whose scale is prohibitively vast in these days. Also, distributing raw training data may lead to privacy leakage issues when the data contain critical private information. Nevertheless, we believe the practice limits further improvement of TTA since it ignores the only source of supervision. The adaptation process using only unlabeled test data could be unreliable or even deteriorate performance due to the absence of accurate supervision incurring confirmation bias towards inaccurate predictions.

Motivated by this, we propose a new TTA method using *proxy* of training data without the side effects, *i.e.*, memory footprint and privacy leakage. Two types of the proxy in our method are illustrated in Figure 1. The first proxy is a tiny set of condensed training data for supervised test-time training. Before TTA, training data are condensed into a small number of synthetic images that are less domain-specific and loose private information as well as capturing as much information of the entire dataset as possible. During TTA, the condensed images are used as labeled (synthetic) test data for test-time training, which demands only small computation and memory while alleviating the privacy leakage issue thanks to the condensation. Furthermore, to close the domain gap between the condensed and test images, neural styles of the condensed images are replaced with those of test images in a feature space *on the fly*, which gives illusions of test images with ground-truth labels.

The second proxy is inter-class similarity relations extracted from training data in advance and transferred to the target model during TTA. Unlike individual features or classes, pairwise relations between classes are less sensitive to domain shifts and thus well-transferable (Park et al., 2019; Seo et al., 2021). Based on this observation, we propose a new variant of knowledge distillation, dubbed *class-relation knowledge distillation* (CRKD). CRKD forces inter-class similarities estimated in the test domain to approximate those of training data. Since the inter-class similarities are highly abstract, CRKD demands negligible memory and does not transfer private information of training data.

It is worthy to note that, thanks to the effective use of training data through the proxies, our work demands substantially less computation and memory than the state-of-the-art techniques using no training data (Wang et al., 2022b; Chen et al., 2022) as well as surpassing their performance as demosntrated in Figure 2. Regarding that models are often deployed in systems with limited computing power and memory for testing, such efficiency is of great importance



(a) VisDA-C (b) CIFAR10-C

Figure 2. Performance comparison in accuracy, computational cost, and memory requirement. The memory size is either the size of the momentum network or the amount of condensed data required by each method. The subscript of 'Ours' indicates the number of condensed images per class. Our method clearly outperformed the state of the art (AdaContrast and CoTTA) with 4.28 times and 116.7 times less memory footprint on VisDA-C and CIFAR10-C, respectively.

in practical TTA scenarios. Moreover, the efficiency of our method suggests that it successfully resolves the main objection to the use of training data for TTA (Wang et al., 2021). The major contribution of this paper is four-fold:

- We present the first attempt to utilize lightweight and informative proxies of training data for TTA without large memory demand or leaking privacy.
- We design a dataset condensation technique dedicated to TTA. Unlike conventional methods, it aims to generate synthetic images which loose privacy information of training data and whose styles are well replaced with those of test data on the fly.
- We propose CRKD, a knowledge distillation technique dedicated to TTA. CRKD extracts and transfers inter-class relations of training data, which are insensitive to domain shifts and do not hold private information.
- Our method outperformed every prior art on four TTA benchmarks. Moreover, it demands less computation and memory footprint than the state-of-the-art methods.

Limitation: First, it is not straightforward to apply our method to large-scale datasets due to the complexity of dataset condensation that quickly increases as the number of images and that of pretrained classes increas. We anticipate this issue will be resolved by advances in dataset condensation and development of deep learning hardwares. Second, ours cannot be applied when training data of the model in hand are latent. However, we believe this is an unusual case since most of publicly available models were trained on public datasets, and commercial providers have no reason not to distribute proxies of training data that are cheap and expected to prevent privacy leakage.

2. Related work

2.1. Test-time adaptation

Early approaches to TTA has been frequently addressed by self-supervised learning (SSL) on unlabeled test data. The method of Sun et al. (2020) relies on a proxy task of predicting the type of image rotation, but this unconstrained self-supervised adaptation could be easily overfitted to the auxiliary task. To resolve this issue, Liu et al. (2021) aligned test features by minimizing the distance between feature statistics of training and test domains, and Sarkar et al. (2022) proposed a consensus prediction strategy. All of these methods alter training process and/or model architecture for learning with the auxiliary SSL tasks.

On the other hand, Wang et al. (2021) proposed a fully test-time adaptation method with no such alteration by minimizing the prediction entropy for test data. Also, Iwasawa & Matsuo (2021) adapted a model to test samples in a domain generalization setting through a backpropagation-free adaptation method. They modify only the linear classifier and classify each sample based on its distance to the pseudo-prototypes. Also, Boudiaf et al. (2022) employed a parameter-free method for TTA by modifying output probabilities of the classifier. Meanwhile, recent studies on TTA for single instance (Khurana et al., 2021; Zou et al., 2022) have also been proposed, assuming a realistic scenario where on-demand inference is demanded. Other recent studies (Chen et al., 2022; Wang et al., 2022b) adopt momentum encoders for accurate pseudo labeling of test data, but the encoders require nontrivial memory overhead additionally.

Our method adapts any pretrained model to test data without such heavy auxiliary modules where all test data are streamed sequentially. More importantly, unlike all the previous work, it is designed to utilize training data in an efficient, effective, and privacy-preserving manner.

2.2. Neural style representation

Gatys et al. (2016) demonstrated that statistics of convolutional features capture the style of an image. Ulyanov et al. (2017) introduced instance normalization (IN) for image stylization, and thereafter IN has been used also for normalizing image styles (Nam & Kim, 2018; Dumoulin et al., 2017; Lee et al., 2022a). AdaIN (Huang & Belongie, 2017) transfers channel-wise mean and standard deviation of convolutional features for neural style transfer. Recent studies on domain generalization (Nam et al., 2019; Kim et al., 2021; Zhou et al., 2021; Kang et al., 2022) have used these style representation schemes under the assumption that a visual domain is characterized by styles of its images.

Our strategy for test-style injection to condensed training data is motivated by these methods. It is the first attempt to exploit test style representation for TTA and to investigate the impact of style injection on the use of condensed images.

2.3. Dataset condensation

Dataset condensation aims to synthesize a small set of informative data so that a network trained on them can achieve comparable performance to one trained with the original training dataset. Wang et al. (2018) introduced a meta-learning method considering condensed data as learnable parameters and optimize network parameters and the condensed data alternatively. Recently, gradient matching (Zhao et al., 2021; Cazenavette et al., 2022; Lee et al., 2022b) and distribution matching (Zhao & Bilen, 2021; Wang et al., 2022a) have been studied for the purpose.

We for the first time introduce data condensation for TTA to enable supervised test-time training with a small set of condensed training data, demanding only affordable computation and memory footprint.

2.4. Rehearsal-based learning

Catastrophic forgetting has been a chronic issue in continual learning, and regarding TTA as a combination of continual learning and unsupervised domain adaptation, it can deteriorate TTA performance too. One popular way of alleviating this issue is to replay memory of a few samples of the previous task (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Guo et al., 2020), which is called rehearsal-based learning. For example, GEM (Lopez-Paz & Ranzato, 2017) and A-GEM (Chaudhry et al., 2019) keep losses of examples stored in episodic memory low enough, and MEGA (Guo et al., 2020) generalizes the memory-based methods by novel loss-balancing updating rules.

Motivated by these methods, our work stores and replays condensed training data to prevent the model from being degraded by unlabeled test data in TTA.

2.5. Knowledge distillation

Knowledge distillation (KD) aims at transferring knowledge of a model (teacher) to another model (student), and has been employed for model compression and regularization (Ba & Caruana, 2014; Romero et al., 2014; Hinton et al., 2014; Zagoruyko & Komodakis, 2017; Yim et al., 2017; Polino et al., 2018; Crowley et al., 2018; Park et al., 2019; Nayak et al., 2019). By mimicking the teacher model, the student model can effectively learn semantic similarity between categories from the teacher.

Motivated by these methods, we propose to transfer interclass relations as knowledge of training data. Since interclass relations are highly abstract and less domain-sensitive, they are well transferred, demand only a tiny amount of memory, and do not raise privacy issues.

3. Method

Our method consists of two parts: (i) *dataset condensation* (DC) to synthesize a proxy of the training data in advance of TTA, and (ii) *test-time adaptation* (TTA) to adjust the target model with virtual supervision from converting the proxy into the test domain style. The synthesized proxy is lightweight as it consists of only a few synthetic data, and would raise fewer privacy concerns than raw training data. Especially, neural styles of test data are injected into the condensed ones on-the-fly in a feature level so that they behave like test domain data with ground truth labels. To this end, our condensation method aims to synthesize images that are appropriate for style injection as well as preserving as much information of the entire training dataset as possible. In addition, our method extracts the inter-class similarities learned from training data as the second proxy and forces such similarities estimated in the test domain to approximate those of training data. These two different proxies of training data demand only a small amount of additional memory and prevent privacy leakage.

The remainder of this section presents details of our dataset condensation method (Section 3.1) and our TTA framework using the proxies of training data (Section 3.2).

3.1. Style-normalized dataset condensation

Following recent DC approaches (Zhao & Bilen, 2021; Wang et al., 2022a), we synthesize a tiny set of condensed images by matching feature distributions of training and condensed data. In addition, our method has two distinctive features dedicated to TTA. First, since a pretrained network is employed and adapted during testing in TTA, we utilize a fixed pretrained network as the feature extractor for DC unlike conventional methods accompanying randomly initialized networks. Second, we aim to build condensed images preserving *style-normalized* contents of training data so that the condensed images become less domain-specific and better simulate test data when test styles are injected.

To be specific, the condensed images are optimized by minimizing the empirical estimate of the maximum mean discrepancy (Zhao & Bilen, 2021) between the style-normalized feature distribution of real training data and the feature distribution of condensed ones for each class; the discrepancy is measured on the embedding space of a pretrained model. Let $f := f_3 \circ f_2 \circ f_1$ be the pretrained network comprising three parts, lower part of encoder f_1, upper part of encoder f_2, and classifier f_3, as shown in Figure 3. We sample a batch of real training data B_c and that of condensed data $\bar{B}_c$ for every class c, and motivated by recent work on style transfer (Huang & Belongie, 2017; Gatys et al., 2016), apply instance normalization only for B_c to the output of f_1 to obtain style-normalized features of training data. Hence, our loss for DC is given by

$$\mathcal{L}_{\text{DC}} = \sum_{c=1}^{C} \left\| \frac{1}{|B_c|} \sum_{\mathbf{x} \in B_c} f_2 \circ \text{IN} \circ f_1(\mathbf{x}) - \frac{1}{|\bar{B}_c|} \sum_{\bar{\mathbf{x}} \in \bar{B}_c} f_2 \circ f_1(\bar{\mathbf{x}}) \right\|^2, \tag{1}$$

where C indicates the number of classes and IN stands for the instance normalization.

To be more specific, the instance normalization aims to remove domain characteristics of an image in a feature space so that the condensed data simulate a domain-invariant version of the training set. This approach is motivated by the fact that a visual domain has been characterized by styles of images in that domain and global statistics of low-level features have been widely used as style descriptors (Huang & Belongie, 2017; Zhou et al., 2021; Kang et al., 2022).

3.2. Test-time adaptation using proxies of training data

Our adaptation framework utilizes knowledge of training data in two different ways. First, the condensed data whose neural styles are replaced by those of test samples are used for supervised learning. Second, inter-class relations learned from training data are transferred to predictions for test data. An overview of our framework is illustrated in Figure 3, and its details are presented in the remainder of this section.

3.2.1. Supervised learning with condensed data

We propose to use a few condensed training data for facilitating TTA by preventing confirmation bias due to inaccurate supervision for unlabeled test data. A challenge lying in this direction is the distribution gap between training and test domains; using the condensed training data as-is for supervised learning may hinder the adaptation towards the test distribution due to the domain gap.

In order to effectively utilize the condensed data, we propose to inject neural styles of test data into them during TTA, which helps mitigate the distribution shift. Specifically, Huang & Belongie (2017) demonstrated that the style of an image can be represented by feature statistics of the image and be transferred to another image by replacing such statistics. Following previous work (Huang & Belongie, 2017; Zhou et al., 2021), we regard the channel-wise mean $\mu(\mathbf{Z}) \in \mathbb{R}^C$ and standard deviation $\sigma(\mathbf{Z}) \in \mathbb{R}^C$ of a low-layer feature map $\mathbf{Z} = f_1(\mathbf{x}) \in \mathbb{R}^{C \times H \times W}$ as the style of the data point $\mathbf{x}$, and extract them as follows:

$$\mu(\mathbf{Z}) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{Z}_{:,h,w}, \tag{2}$$

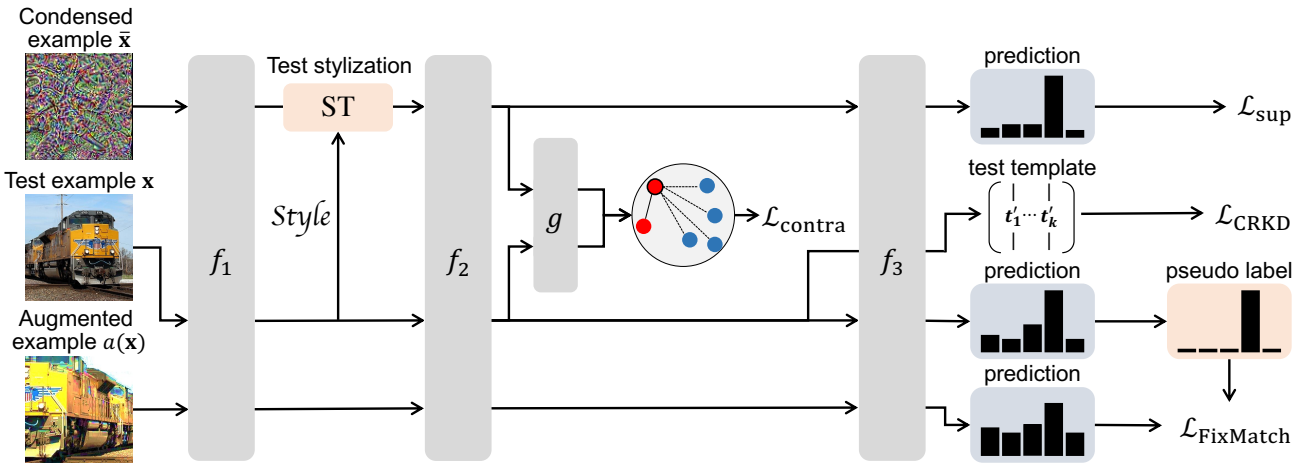


Figure 3. The overall architecture and training objectives of the proposed model. Condensed examples of training data are stylized by input test examples, and used for supervised test-time training with their labels ($\mathcal{L}_{\text{sup}}$). They are also used for contrastive learning so as to reduce the discrepancy between test-stylized condensed data and test data on a feature space ($\mathcal{L}_{\text{contra}}$). Meanwhile, inter-class relations of training data are used to regularize predictions for test examples through class-relation knowledge distillation so that inter-class relations of the predictions well approximate those of training data ($\mathcal{L}_{\text{CRKD}}$). Finally, we apply consistency regularization with augmented test examples to further boost performance by directly exploiting unlabeled test example for TTA ($\mathcal{L}_{\text{FixMatch}}$).

$$\sigma(\mathbf{Z}) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(\mathbf{Z}_{:,h,w} - \mu(\mathbf{Z}))^2}, \tag{3}$$

where H and W denote height and width of the feature map, respectively. After that, the neural styles of the condensed images are replaced with those of test images through test stylization function ST($\cdot$) inspired by AdaIN (Huang & Belongie, 2017). Given the intermediate feature map of a condensed image denoted by $\mathbf{Y} = f_1(\bar{\mathbf{x}})$, its test stylization is conducted by

$$\text{ST}(\mathbf{Y}) = \sigma(\mathbf{Z})\frac{\mathbf{Y} - \mu(\mathbf{Y})}{\sigma(\mathbf{Y})} + \mu(\mathbf{Z}), \tag{4}$$

where $\mathbf{Z}$ is a low-layer feature map of a randomly sampled test example.

Since it has been known that the content of an image is well preserved after such a stylization (Huang & Belongie, 2017; Zhou et al., 2021), the condensed data with styles of test data are supposed to still preserve their contents (*i.e.*, class labels). Thus, we use them for learning by minimizing the cross entropy loss:

$$\mathcal{L}_{\text{sup}} = \sum_i \text{CE}\Big(y_i, f_3 \circ f_2 \circ \text{ST} \circ f_1(\bar{\mathbf{x}}_i)\Big), \tag{5}$$

where $\bar{\mathbf{x}}_i$ is a condensed image and y_i is its class label.

In addition to the cross entropy loss, we also adopt contrastive learning to reduce the representation discrepancy between test-stylized condensed data and test data. To this end, we first get a pseudo label $\hat{y}_i = \arg\max f(\mathbf{x}_i)$ for a test sample $\mathbf{x}_i$. Then, the supervised contrastive loss (Khosla et al., 2020) is applied to each test feature $q_i = g \circ f_2 \circ f_1(\mathbf{x}_i)$ and stylized condensed data features $k_j = g \circ f_2 \circ \text{ST} \circ f_1(\bar{\mathbf{x}}_j)$, where g stands for a projection module, by discriminating positives and negatives determined by $\hat{y}_i$. Specifically, the loss is given by

$$\mathcal{L}_{\text{contra}} = -\sum_i \frac{1}{|K_i^+|}\sum_{k_l \in K_i^+} \log \frac{\exp(q_i^\top k_l/\tau_1)}{\sum_j \exp(q_i^\top k_j/\tau_1)}, \tag{6}$$

where K_i^+ is the set of positives for i-th test sample and τ_1 is a temperature parameter. This loss encourages aligning features of the same class closely regardless of data types.

3.2.2. Class-relation knowledge distillation

As another approach to leverage training data, we propose a knowledge distillation technique encouraging inter-class relations to be consistent between training and test domains. We believe that this approach is effective for TTA since, although low-level features or styles of an image such as color or texture vary substantially by domain shift, high-level semantics like inter-class relations usually remain consistent across different domains. Our method, called class-relation knowledge distillation (CRKD), transfers the mutual relations of classes learned in training domain to test domain.

When estimating inter-class relations of training data, we consider the k-th row of the weight matrix of the pretrained classifier as a template of class k. Let $\mathbf{t}_k$ denotes such a template for class k. Then the semantic affinity between classes k and k' is estimated as the similarity between their tem-

plates $\mathbf{t}_k$ and $\mathbf{t}_{k'}$. The inter-class relation is thus computed in a form of a similarity matrix as follows:

$$[\mathbf{M}]_{k,k'} = \frac{\mathbf{t}_k^\top \mathbf{t}_{k'}}{\|\mathbf{t}_k\|\|\mathbf{t}_{k'}\|}. \tag{7}$$

The k-th row vector of the matrix, $\mathbf{M}_{k,:}$, represents the similarities of class k with all the classes. Note that $\mathbf{M}$ is data-free knowledge derived from weight parameters of the pretrained model without either training data or additional inference, and thus is secure against privacy leakage.

We also establish class templates of test data to estimate inter-class relations in test domain. The test-domain template for class k, denoted by $\mathbf{t}'_k$, is estimated as a representative feature of test data classified into class k by the model f. To be specific, since the model learns with test data in an online manner, $\mathbf{t}'_k$ is updated progressively by exponential moving average (EMA) during testing as follows:

$$\mathbf{t'}_{\hat{y}_i} = \alpha \mathbf{t'}_{\hat{y}_i} + (1-\alpha)\mathbf{v}_i, \tag{8}$$

where α is the update ratio, $\mathbf{v}_i = f_2 \circ f_1(\mathbf{x}_i)$ is the feature of a test example $\mathbf{x}_i$, and $\hat{y}_i$ is the pseudo label of $\mathbf{x}_i$. By updating the templates through EMA, they change smoothly and reflect all the observed test data, which stabilizes supervisory signals given by CRKD. Then, we calculate the cosine similarity scores between input test example $\mathbf{x}_i$ and all the test-domain class templates in the form of a similarity score vector as follows:

$$[\mathbf{m}'_i]_k = \frac{\mathbf{t}_k'^\top \mathbf{v}_i}{\|\mathbf{t}'_k\|\|\mathbf{v}_i\|}. \tag{9}$$

Each element of $\mathbf{m}'_i$ represents the similarity between the test example $\mathbf{x}_i$ and a particular class. Our objective herein encourages the similarity scores $\mathbf{m}'_i$ to follow the inter-class similarities of training data through the KL divergence loss:

$$\mathcal{L}_{\text{CRKD}} = \sum_i \text{KL}\left(\text{softmax}\left(\frac{\mathbf{M}_{\hat{y}_i,:}}{\tau_2}\right), \text{softmax}\left(\frac{\mathbf{m}'_i}{\tau_2}\right)\right), \tag{10}$$

where τ_2 is a temperature parameter.

3.2.3. Total objective and details for training

In addition to the three loss functions in Eq. (5), Eq. (6), and Eq. (10), we further apply consistency regularization (Berthelot et al., 2020; Sohn et al., 2020) that has been known as an effective way of learning using unlabeled data. To this end, following FixMatch (Sohn et al., 2020), we perturb input test sample $\mathbf{x}_i$ and apply the standard cross entropy loss with its pseudo label $\hat{y}_i$ as follows:

$$\mathcal{L}_{\text{FixMatch}} = \sum_i \text{CE}(\hat{y}_i, f(a(\mathbf{x}_i))), \tag{11}$$

where $a(\cdot)$ is an augmentation operation. The overall objective of our framework is then given by

$$\mathcal{L} = \lambda \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{contra}} + \mathcal{L}_{\text{CRKD}} + \mathcal{L}_{\text{FixMatch}}, \tag{12}$$

where λ is a balancing parameter.

During TTA, we optimize batch normalization layers of the network as in previous work (Wang et al., 2021) and projection layers g for contrastive learning. Additionally, since feature vectors and templates are normalized when computing $\mathcal{L}_{\text{CRKD}}$, we also normalize the trained weights of the final fully-connected layer and its input feature vectors when computing $\mathcal{L}_{\text{sup}}$ of Eq. (5) for stable optimization; in other words, we adopt a cosine classifier for f_3.

4. Experiments

The proposed method is first evaluated on three benchmarks for common image corruptions (CIFAR10-C, CIFAR100-C, and TinyImageNet-C (Hendrycks & Dietterich, 2019)) and one benchmark for synthetic-to-real adaptation (VisDA-C (Peng et al., 2017)). Images of the corruption datasets are borrowed from the test sets of CIFAR10, CIFAR100, and TinyImageNet, and are intentionally degraded by 15 corruption types; each type has five severity levels with level 5 as the most severe condition. In all experiments on these datasets, our work is compared with previous methods in an online TTA task, in which a model is evaluated in an online manner. In other words, each test sample is processed only once, assuming streaming input data in testing. In addition, we demonstrate the effectiveness of our method for long-term adaptation in the online continual TTA setting (Wang et al., 2022b) on CIFAR10-C, where the target domain distribution continually changes over time.

4.1. Baselines

Common image corruptions. Our method is compared with four different baselines: 'Source', BN, PL, Tent (Wang et al., 2021), and CoTTA (Wang et al., 2022b). 'Source' stands for the model trained only on the source (training) domain and applied to the test domain as-is. BN means test-time normalization that uses the in-batch normalization statistics calculated from test data but freezes the other parameters during testing. PL indicates pseudo labeling, which optimizes the target model using test data with pseudo labels obtained by the source-pretrained model. Tent (Wang et al., 2021) modulates batch normalization layers by minimizing entropy on test data. Finally, CoTTA (Wang et al., 2022b) adopts a momentum encoder and diversely augments inputs to obtain accurate pseudo labels for them.

Synthetic-to-real. In addition to the TTA baselines, we further compare ours with three TTA methods, AdaContrast (Chen et al., 2022), ConjugatePL (Goyal et al., 2022)

and TTAC (Su et al., 2022), as well as two unsupervised domain adaptation (UDA) methods, CAN (Kang et al., 2019) and MCC (Jin et al., 2020), on VisDA-C. AdaContrast adopts self-supervised learning with a momentum encoder for accurate pseudo labeling of test data, and thus is more costly than ours in both space and time. TTAC utilizes the category-wise and global statistics from the training data as lightweight information to precisely align target domain features to source counterparts. The UDA methods are evaluated as they share a similar feature with ours, *i.e.*, using training data for test domain adaptation. However, UDA is substantially more favorable than online TTA since it gives full access to test data in training. Hence, the UDA methods are instead evaluated in a setting close to online TTA: They use as many training data as the condensed ones in our work, and are adapted to test domain only for a single epoch.[1]

4.2. Implementation details

We adopt ResNet (He et al., 2016) as the backbone network of our model, which performs normalization in batch normalization layers using test data (in-batch) statistics instead of updating batch normalization statistics as in previous work (Wang et al., 2021). Style normalization and injection are applied to the output of the second residual block of each network as previous studies (Zhou et al., 2021; Kang et al., 2022) have shown that the statistics of the block represent the style of images. For style-normalized dataset condensation, we set the number of synthetic images per class to 10 for CIFAR and TinyImageNet and to 50 for VisDA-C. For optimization in condensation, we initialize the synthetic images by random noise and optimize them with SGD. The resolution of a synthetic image is 32×32 for CIFAR, 64×64 for TinyImageNet, and 112×112 for VisDA-C, respectively. For optimization in test time, we adopt Adam (Kingma & Ba, 2015) for the common image corruption benchmark and SGD for VisDA-C. The balancing parameter λ is set to 0.1 for the corruption benchmark and 1.0 for the other. For temperature control, we set τ_1 to 0.1 and τ_2 to the inverse of the square root of the number of classes for each dataset. To apply $\mathcal{L}_{\text{FixMatch}}$, we use four augmented images for continual TTA and one augmented image for synthetic-to-real adaptation.

4.3. Robustness to common image corruption

Single target domain TTA. Table 1 compares our method with previous TTA baselines on three datasets for common image corruption. Following the convention of the previous online TTA setting (Wang et al., 2021), we report performance of our method during 1st epoch on the test set. Note that our results in the table were obtained without $\mathcal{L}_{\text{FixMatch}}$ for a fair comparison with baselines that do not utilize any augmentation. Our method consistently achieves the best average accuracy across all corruptions and severity levels on CIFAR10-C, CIFAR100-C, and TinyImageNet-C, regardless of the type of its backbone network; the records per corruption type are reported in the supplementary material. Moreover, considering the experimental setup in Table 1 where the adaptation process is conducted in a relatively short period of time (with just 10K test samples, compared to 150K in continual TTA and 55K in synthetic-to-real TTA), these results demonstrate the effectiveness of our method for fast adaptation.

[1]This setting is still more favorable since all test data are given at once unlike online TTA where test data are given sequentially.

Table 1. Results of TTA with ResNet backbones on CIFAR10-C, CIFAR100-C, and TinyImageNet-C, averaged across all 15 corruptions and 5 severity levels. We report average accuracy (%) and mark the best performance in **bold**. RN denotes ResNet. Note that $\mathcal{L}_{\text{FixMatch}}$ is not used in Ours for a fair comparison.

Method	CIFAR10-C		CIFAR100-C		TinyImageNet-C
	RN26	RN50	RN26	RN50	RN18
Source	71.97	81.75	40.42	53.89	30.93
BN	76.98	88.40	46.87	62.94	44.38
Tent (Wang et al., 2021)	80.95	89.39	52.43	66.03	45.95
Ours	**81.96**	**90.40**	**52.82**	**67.91**	**46.98**

Continual TTA. In Table 2, our method consistently improves performance in various corruption types and achieves the best average accuracy. Compared with CoTTA that refines pseudo labels by forwarding augmented inputs using a momentum network, our method effectively improves performance by consulting condensed training data without such additional network. This result suggests that ours not only adapts quickly to test domains, but also effectively prevents the model degeneration problem caused by confirmation bias with the aid of condensed data.

4.4. Adaptation from synthetic to real

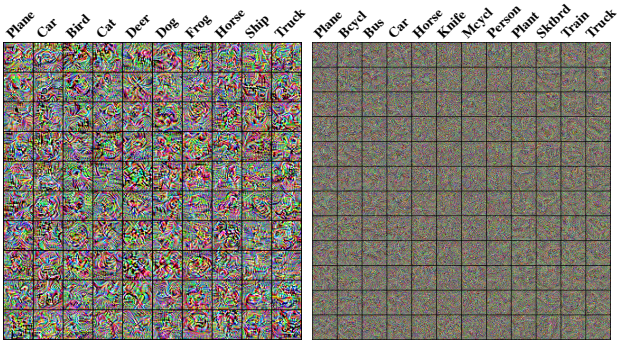
Table 3 presents quantitative results of our method, previous TTA methods, and UDA methods on VisDA-C with ResNet50 and ResNet101. Our method outperforms the all other online TTA methods in average accuracy. Specifically, our method using ResNet50 surpasses TTAC, which captures the feature distribution of training data precisely through running mean and covariance per cluster. While both TTAC and our method share the same purpose of capturing the distribution of training data, TTAC calculates feature statistics manually, whereas our method optimizes the condensed data in a fully data-driven manner. The performance improvement over TTAC demonstrates the effectiveness of our method. Meanwhile, our method outperforms AdaContrast, the previous state-of-the-art method, when using ResNet101. This indicates that our method effectively prevents confirmation bias by using condensed data with no heavy auxiliary module like momentum encoder. Addition-

Table 2. Classification accuracy (%) on CIFAR10-to-CIFAR10-C online continual TTA task. Results are evaluated on WideResNet-28 with the highest corruption severity level. We mark the best and second-best performance in **bold** and underline, respectively. * denotes the requirement on additional domain information of input data for resetting model.

	$t \longrightarrow$															
Method	*Gaussian*	*shot*	*impulse*	*defocus*	*glass*	*motion*	*zoom*	*snow*	*frost*	*fog*	*brightness*	*contrast*	*elastic_trans*	*pixelate*	*jpeg*	Mean
Source	27.7	34.3	27.1	53.1	45.7	65.2	58.0	74.9	58.7	74.0	90.7	53.3	73.4	41.5	69.7	56.5
BN Stats Adapt	71.9	73.9	63.7	87.2	64.7	85.8	87.9	82.7	82.6	84.7	91.6	87.4	76.2	80.3	72.7	79.6
Pseudo-label	73.3	77.9	68.0	86.2	67.8	84.7	87.3	82.7	82.7	83.5	89.9	86.6	77.6	81.1	74.1	80.2
TENT-online* (Wang et al., 2021)	75.2	76.5	67.0	88.0	68.2	85.3	89.2	84.1	83.8	86.3	92.1	87.9	78.0	82.7	75.8	81.4
TENT-continual (Wang et al., 2021)	75.2	79.4	71.4	85.6	68.9	83.5	85.9	80.9	81.4	81.4	87.8	79.7	74.3	79.2	75.1	79.3
CoTTA (Wang et al., 2022b)	75.7	78.7	73.4	**88.4**	72.4	**87.8**	89.7	85.2	85.9	87.6	**92.5**	89.4	**81.7**	**86.6**	**82.7**	83.8
Ours	**76.8**	**81.6**	**75.0**	86.9	**73.5**	86.1	89.4	**85.6**	**86.6**	**87.9**	91.7	**90.2**	80.6	86.1	82.3	**84.0**

Table 3. Classification accuracy (%) on VisDA-C train → val. All methods use ResNet-101 backbone. We mark the best and second-best performance in **bold** and underline, respectively. † denotes offline unsupervised domain adaptation methods where the number of source images is equal to the number of condensed images in our method.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
ResNet50													
Source only	43.5	11.3	37.2	**69.4**	18.1	1.0	82.5	4.2	48.3	22.1	**79.6**	0.5	34.8
Tent (Wang et al., 2021)	86.6	22.0	**79.8**	51.9	78.3	17.6	**87.6**	64.2	78.9	23.6	65.9	1.1	54.8
ConjugatePL (Goyal et al., 2022)	-	-	-	-	-	-	-	-	-	-	-	-	61.6
TTAC(N-O) (Su et al., 2022)	81.5	59.8	64.2	36.9	76.2	60.4	84.5	58.7	77.0	53.4	74.8	32.2	63.3
Ours	**92.7**	**82.5**	**79.8**	65.3	**92.7**	**70.3**	80.7	**82.6**	**89.9**	**62.2**	77.2	**37.6**	**76.1**
ResNet101													
Source only	57.2	11.1	42.4	66.9	55.0	4.4	81.1	27.3	57.9	29.4	86.7	5.8	43.8
CAN† (Kang et al., 2019)	**95.7**	**88.8**	6.9	68.6	**94.5**	**94.8**	79.2	70.3	88.7	**80.6**	83.2	**51.7**	75.2
MCC† (Jin et al., 2020)	93.9	78.4	70.4	**74.3**	92.5	84.2	84.5	58.2	86.6	36.0	86.1	20.6	72.2
Tent (Wang et al., 2021)	91.1	45.6	**86.4**	66.4	88.7	75.1	**90.3**	76.4	84.4	47.1	83.6	13.7	70.7
AdaContrast (Chen et al., 2022)	95.0	68.0	82.7	69.6	94.3	80.8	**90.3**	79.6	90.6	69.7	**87.6**	36.0	78.7
Ours	92.5	82.4	85.8	74.2	92.7	88.5	83.9	**85.8**	**92.8**	62.5	75.2	32.5	**79.1**



(a) CIFAR10 (b) VisDA-C

Figure 4. Visualization of condensed data where each column presents 10 images per class of CIFAR10 and VisDA-C.

ally, our method outperformed the UDA methods working in a more favorable setting, which demonstrates the efficacy of our work and suggests that (online) TTA demands a dedicated solution even if training data are accessible.

Table 4. Ablation study on combinations of the losses and test stylization (ST) in online continual TTA on CIFAR10-C and single target domain online TTA on VisDA-C, respectively.

$\mathcal{L}_{\text{CRKD}}$	$\mathcal{L}_{\text{sup}}$	ST	$\mathcal{L}_{\text{contra}}$	$\mathcal{L}_{\text{FixMatch}}$	CIFAR10-C	VisDA-C
✓					80.6	72.9
✓	✓				73.4	73.1
✓	✓	✓			81.8	76.0
✓	✓	✓	✓		82.1	76.7
✓	✓	✓	✓	✓	**84.0**	**79.1**

4.5. Ablation study

Contribution of each component. Table 4 demonstrates the contribution of each proposed module on CIFAR10-C and VisDA-C. First, using $\mathcal{L}_{\text{CRKD}}$ only, even with no condensed data, clearly surpasses Tent, suggesting the effectiveness of transferring inter-class similarity for TTA. We then apply the cross entropy loss for the condensed data without test stylization, which leads to significant performance drop in the continual TTA setting. In contrast, the cross entropy loss applied to test-stylized condensed data improves the

Table 5. Ablation study on the number of condensed images per each class and the advantage of condensed images over original source (training) images. We report classification average accuracy (%) on VisDA-C train → val with ResNet-101 backbone, and mark the superior performance in **bold**.

Method	The number of data per class			
	10	25	50	100
Ours with condensed data	76.91	**79.17**	**79.07**	**79.39**
Ours with source data	**77.90**	77.85	78.40	78.36

Table 6. Impact of the condensed data and that of raw training data selected through various sampling strategies upon the final performance of our method. We report classification average accuracy (%) on VisDA-C train → val with ResNet-101 backbone.

Random	Per-class prototype	K-means	Condensed data
78.4	77.3	78.4	**79.1**

performance substantially on both benchmarks. These results show that the test-stylized condensed data can help the model adapt to the test domain consistently. In addition, $\mathcal{L}_{\text{contra}}$ boosts the performance on both benchmarks. Lastly, adding $\mathcal{L}_{\text{FixMatch}}$ further improves the performance and the final model outperforms all existing TTA methods.

Impact of the number of condensed data. As shown in Table 5, our method achieved outstanding performance and was not sensitive to the the number of condensed data when the number is larger than or equal to 25. A sufficient quantity of condensed data can capture the distribution of the training data and allow the model to leverage the knowledge of the distribution during test-time adaptation effectively.

Advantage of condensed data over raw training data. Table 5 also demonstrates that condensed data enable to achieve better performance than randomly sampled raw training data given a moderately large number of the data. The result suggests that condensed data better provide knowledge of the entire training set as intended in the data condensation. Furthermore, we explored two additional sampling strategies as alternatives of the random sampling and evaluated their impact on performance in Table 6. The first is to select a few samples closest to the mean feature per class (per-class prototype), and the second is to sample those closest to K-means of the entire source data. Our method using condensed data also outperforms all the three variants, highlighting the effectiveness of the condensed data that allow a model to observe the entire source data indirectly.

4.6. Visualization

Figure 4 shows the examples of condensed images for CIFAR10 and VisDA-C. These images show a large discrepancy visually from the training data and do not look like real-looking images. Nevertheless, as demonstrated in our experimental results, they help improve TTA performance significantly. Moreover, it looses privacy information of original training data, which is another advantage that alleviates the privacy leakage issue.

4.7. Model complexity

Unlike CoTTA (Wang et al., 2022b) and AdaContrast (Chen et al., 2022), the state-of-the-art TTA methods using momentum encoders, our work does not rely on such auxiliary modules costly in computation and memory, but instead utilizes condensed data for facilitating TTA. The efficacy and small size of the condensed data allows ours to outperform the previous work not only in adaptation performance but also in space-time complexity substantially. This is demonstrated in Figure 2, where ours and the two previous methods are compared in terms of the adaptation performance, the additional memory usage of condensed data or momentum encoder, and the total FLOPs. The superiority of our method in the trade-off between adaptation performance and complexity proves that ours is more suitable for practical TTA scenarios, *e.g.*, on edge devices such as robots and surveillance cameras, where reducing memory usage and computation overhead is crucial.

5. Conclusion

We have proposed a new method that extracts and exploits lightweight proxies of training data for TTA. The two types of proxy require only a small amount of additional memory and alleviate privacy leakage. By utilizing a few condensed samples stylized by test data and transferring the inter-class relations across domains, our method outperformed existing TTA methods on four benchmarks. Especially the proposed method surpassed the state of the art with fewer operations and smaller memory demand. The superiority of the proposed method in space-time complexity as well as adaptation performance suggests that it is more suitable to practical TTA scenarios. We anticipate that our research will inspire future research harnessing the knowledge of training data effectively, instead of blindly rejecting the use of training data in TTA scenarios.

Acknowledgement. This work was supported by the the Institute of Information & communications Technology Planning & Evaluation grant funded by Ministry of Science and ICT, Korea (IITP-2020-0-00842, IITP-2021-0-00739), Samsung Electronics Co., Ltd (IO201210-07948-01), and Samsung Research Funding & Incubation Center of Samsung Electronics (SRFC-IT1801-05).

References

Ba, J. and Caruana, R. Do deep nets really need to be deep? In *NeurIPS*, volume 27, 2014.

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., and Raffel, C. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *ICLR*, 2020.

Boudiaf, M., Mueller, R., Ben Ayed, I., and Bertinetto, L. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8344–8353, 2022.

Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y. Dataset distillation by matching training trajectories. In *CVPR*, pp. 4750–4759, 2022.

Chang, W.-G., You, T., Seo, S., Kwak, S., and Han, B. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, 2019.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

Chen, D., Wang, D., Darrell, T., and Ebrahimi, S. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Choi, S., Jung, S., Yun, H., Kim, J. T., Kim, S., and Choo, J. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *CVPR*, pp. 11580–11590, 2021.

Crowley, E. J., Gray, G., and Storkey, A. J. Moonshine: Distilling with cheap convolutions. In *NeurIPS*, volume 31, 2018.

Dai, D. and Van Gool, L. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3819–3824, 2018.

Dumoulin, V., Shlens, J., and Kudlur, M. A learned representation for artistic style. In *ICLR*, 2017.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *ICML*, pp. 1180–1189. PMLR, 2015.

Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *CVPR*, pp. 2414–2423, 2016.

Goyal, S., Sun, M., Raghunathan, A., and Kolter, Z. Test-time adaptation via conjugate pseudo-labels. *arXiv preprint arXiv:2207.09640*, 2022.

Guo, Y., Liu, M., Yang, T., and Rosing, T. Improved schemes for episodic memory-based lifelong learning. In *NeurIPS*, volume 33, pp. 1023–1035, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, June 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. In *Deep Learning, NeurIPS workshop*, 2014.

Huang, X. and Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pp. 1501–1510, 2017.

Huang, Z., Wang, H., Xing, E. P., and Huang, D. Self-challenging improves cross-domain generalization. In *ECCV*, pp. 124–140. Springer, 2020.

Iwasawa, Y. and Matsuo, Y. Test-time classifier adjustment module for model-agnostic domain generalization. In *NeurIPS*, volume 34, 2021.

Jin, Y., Wang, X., Long, M., and Wang, J. Minimum class confusion for versatile domain adaptation. In *ECCV*, pp. 464–480. Springer, 2020.

Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, pp. 4893–4902, 2019.

Kang, J., Lee, S., Kim, N., and Kwak, S. Style neophile: Constantly seeking novel styles for domain generalization. In *CVPR*, pp. 7130–7140, June 2022.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. In *NeurIPS*, volume 33, pp. 18661–18673, 2020.

Khurana, A., Paul, S., Rai, P., Biswas, S., and Aggarwal, G. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*, 2021.

Kim, N., Son, T., Lan, C., Zeng, W., and Kwak, S. Wedge: Web-image assisted domain generalization for semantic segmentation. *arXiv preprint arXiv:2109.14196*, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Lee, K., Kim, S., and Kwak, S. Cross-domain ensemble distillation for domain generalization. In *ECCV*, pp. 1–20. Springer, 2022a.

Lee, S., Chun, S., Jung, S., Yun, S., and Yoon, S. Dataset condensation with contrastive signals. In *ICML*, 2022b.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *ICCV*, pp. 5542–5550, 2017.

Liu, Y., Kothari, P., van Delft, B., Bellot-Gurlet, B., Mordan, T., and Alahi, A. Ttt++: When does self-supervised test-time training fail or thrive? In *NeurIPS*, volume 34, 2021.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *ICML*, pp. 97–105. PMLR, 2015.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. In *NeurIPS*, volume 30, 2017.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *ICML*, pp. 10–18. PMLR, 2013.

Nam, H. and Kim, H.-E. Batch-instance normalization for adaptively style-invariant neural networks. In *NeurIPS*, volume 31, 2018.

Nam, H., Lee, H., Park, J., Yoon, W., and Yoo, D. Reducing domain gap via style-agnostic networks. *arXiv preprint arXiv:1910.11645*, 2(7):8, 2019.

Nayak, G. K., Mopuri, K. R., Shaj, V., Radhakrishnan, V. B., and Chakraborty, A. Zero-shot knowledge distillation in deep networks. In *ICML*, pp. 4743–4751. PMLR, 2019.

Park, W., Kim, D., Lu, Y., and Cho, M. Relational knowledge distillation. In *CVPR*, pp. 3967–3976, 2019.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. Visda: The visual domain adaptation challenge. In *arXiv preprint arXiv:1710.06924*, 2017.

Polino, A., Pascanu, R., and Alistarh, D. Model compression via distillation and quantization. In *ICLR*, 2018.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *CVPR*, pp. 2001–2010, 2017.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *ICLR*, 2014.

Sarkar, A., Sarkar, A., and Balasubramanian, V. N. Leveraging test-time consensus prediction for robustness against unseen noise. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1839–1848, 2022.

Seo, M., Lee, Y., and Kwak, S. On the distribution of penultimate activations of classification networks. In *Uncertainty in Artificial Intelligence*, 2021.

Shi, Y., Seely, J., Torr, P. H., Siddharth, N., Hannun, A., Usunier, N., and Synnaeve, G. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

Shin, I., Tsai, Y.-H., Zhuang, B., Schulter, S., Liu, B., Garg, S., Kweon, I. S., and Yoon, K.-J. Mm-tta: Multi-modal test-time adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16928–16937, 2022.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, volume 33, pp. 596–608, 2020.

Su, Y., Xu, X., and Jia, K. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. *arXiv preprint arXiv:2206.02721*, 2022.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, pp. 443–450. Springer, 2016.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, pp. 9229–9248, 2020.

Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., and Chandraker, M. Learning to adapt structured output space for semantic segmentation. In *CVPR*, pp. 7472–7481, 2018.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *CVPR*, pp. 7167–7176, 2017.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, pp. 6924–6932, 2017.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Wang, K., Zhao, B., Peng, X., Zhu, Z., Yang, S., Wang, S., Huang, G., Bilen, H., Wang, X., and You, Y. Cafe: Learning to condense dataset by aligning features. In *CVPR*, pp. 12196–12205, 2022a.

Wang, Q., Fink, O., Van Gool, L., and Dai, D. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022b.

Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, pp. 4133–4141, 2017.

Zagoruyko, S. and Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

Zhao, B. and Bilen, H. Dataset condensation with distribution matching. *arXiv preprint arXiv:2110.04181*, 2021.

Zhao, B., Mopuri, K. R., and Bilen, H. Dataset condensation with gradient matching. In *ICLR*, 2021.

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

Zou, Y., Zhang, Z., Li, C.-L., Zhang, H., Pfister, T., and Huang, J.-B. Learning instance-specific adaptation for cross-domain segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pp. 459–476. Springer, 2022.

Leveraging Proxy of Training Data for Test-Time Adaptation

Supplementary material

This supplementary material presents additional experimental results (Section 6), detailed results for single domain TTA (Section 7), high-resolution visualization of condensed data (Section 8), and implementation details (Section 9) omitted from the main paper due to space limit.

6. Additional experiments

6.1. Sensitivity analysis

We examined the sensitivity of our method to hyper-parameters λ, τ_1, and α on VisDA-C. Table 7 shows that our method was insensitive to λ, the balancing weight for the supervised learning loss, and τ_1, the temperature term in the contrastive learning loss. For the update ratio α in Eq. 8, the best performance was observed at 0.9995, indicating the effectiveness of smoothly updating the test template.

Table 7. Sensitivity analysis for hyper-parameters, the balancing loss weight λ, temperature scale τ_1, and momentum update ratio α, on VisDA-C with ResNet101.

λ	0.1	0.25	0.5	1.0
Accuracy(%)	78.5	79.0	**79.2**	79.1
τ_1	0.1	0.25	0.5	1.0
Accuracy(%)	79.1	**79.2**	79.1	79.0
α	0.5	0.9	0.9995	1.0
Accuracy(%)	75.1	78.3	**79.1**	76.2

6.2. Impact of contrastive learning with additional memory

To explore the advantages of utilizing a large batch size in contrastive learning, we introduce additional feature memory to increase the effective batch size for the contrastive loss. Table 8 shows the performance of the variants of our method with varying sizes of additional memory on VisDA-C using ResNet101 backbone. The use of additional memory enhances the performance of our method, with improvements of up to 0.5%p when its size is 1024.

Table 8. Ablation study on the size of additional memory for contrastive learning on VisDA-C with ResNet101.

Size of memory	0 (default)	128	256	512	1024	2048
Accuracy(%)	79.1	79.2	79.2	79.4	79.6	79.5

6.3. Impact of the number of condensed data

We provide an ablation study to investigate the effect of the number of condensed images per each class, denoted as IPC, on VisDA-C (Peng et al., 2017). The ablation study is conducted with $\{1, 5, 10, 25, 50, 100\}$ IPC, respectively. As shown in Table 9, the proposed method improves performance with an increase in the number of condensed data. Also, ours with 25 IPC instead of 50 (used in the main paper) still achieves the state-of-the-art performance, which indicates that ours can surpass the other methods with less overhead.

6.4. Effect of test stylization on condensed data

To empirically investigate the effect of test stylization for reducing the domain gap between condensed data and test data, we present an quantitative analysis with the Hausdorff distance between condensed data ($\mathbf{CD}$), test-stylized condensed data ($\mathbf{SD}$), and test data ($\mathbf{TD}$). Since test stylization is applied to the output of the second residual block of each network, the distance is measured on that features between different sets of data using ResNet50 on CIFAR10-C. As shown in Table 10, the distance between ($\mathbf{SD}$,$\mathbf{TD}$) is smaller than the distance between ($\mathbf{CD}$,$\mathbf{TD}$) for each test domain, which shows that the test stylization can reduce the domain gap.

7. Detailed results for single domain TTA

Due to the lack of space, we only reported the results averaged over all corruption types for single domain TTA on image corruption benchmarks in the main paper. Table 11 shows the detailed results for each corruption type on CIFAR10C and CIFAR100C with ResNet26 and ResNet50. Regardless of network type and corruption type, the proposed method consistently improves the adaptation performance except for a few cases (*e.g.*brightness).

8. Visualization of condensed data

Figure 5 and 6 show the high-resolution examples of condensed data on VisDA-C. They look like noise images that do not contain semantic information and the style of original images, which indicates that they are less domain-specific and loose private information.

9. Implementation details

For dataset condensation, we used the SGD optimizer with the learning rate 1.0 and momentum 0.5. During TTA, the batch size of condensed data is set to the same as that of test data for each dataset if the number of condensed data is sufficient. Otherwise, it consists of all condensed data. For the cosine classifier for f_3, we additionally apply temperature scaling for similarity logits with the value 0.07 for CIFAR10 and 0.05 for the other dataset.

Table 9. Classification accuracy (%) on VisDA-C train → val with ResNet-101 backbone. We mark the best performance in **bold**. IPC denotes the number of condensed images per each class. Gray row indicates the results reported in the main paper. The results of ours using raw source (training) images instead of condensed images are also reported to demonstrate the advantage of condensed images over raw source images.

Method	IPC	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
Ours	1	91.2	81.6	82.8	73.0	91.4	**91.3**	79.5	81.6	90.9	53.6	74.1	21.5	76.1
	5	92.3	81.7	84.9	74.2	91.4	89.3	82.3	84.6	92.8	55.2	72.4	24.7	77.1
	10	92.9	79.4	86.1	**76.0**	91.3	88.4	81.4	86.9	**93.1**	48.1	74.0	26.5	77.0
	25	92.9	81.6	85.7	74.4	91.8	90.1	83.3	86.7	92.6	65.5	74.2	31.2	79.2
	50	92.5	**82.4**	85.8	74.2	**92.7**	88.5	83.9	85.8	92.8	62.5	75.2	32.5	79.1
	100	**93.2**	81.3	**87.5**	73.7	92.6	88.2	**84.7**	**87.4**	92.0	65.7	74.8	31.6	**79.4**
Ours	10	92.9	80.8	82.9	68.0	91.0	90.3	82.4	81.6	90.5	59.9	**78.2**	**36.2**	77.9
with	25	92.6	80.1	83.4	68.6	91.5	90.4	83.3	82.0	90.6	61.1	76.1	34.5	77.9
source	50	92.6	79.4	83.8	68.2	91.8	89.6	83.8	82.2	91.0	**66.8**	77.7	33.9	78.4
images	100	93.1	79.7	82.7	67.7	91.6	90.1	84.4	82.1	90.9	64.8	78.7	34.6	78.4

Table 10. The Hausdorff distance between condensed data (CD), stylized condensed data (SD) and test data (TD) to empirically measure the domain gap between them.

Distance	gauss	shot	impulse	defocus	glass	motion	zoom	snow	frost	fog	bright	contrast	elastic	pixelate	jpeg	AVG
(CD,TD)	0.1535	0.1610	0.1520	0.1652	0.1583	0.1731	0.1626	0.1688	0.1641	0.1675	0.1768	0.1804	0.1604	0.1593	0.1557	0.1639
(SD,TD)	0.1475	0.1530	0.1449	0.1564	0.1478	0.1592	0.1527	0.1548	0.1573	0.1616	0.1629	0.1588	0.1491	0.1505	0.1477	0.1536

Table 11. Test accuracy (%) on CIFAR10C and CIFAR100C for each corruption type averaged over 5 severity levels.

	gauss	shot	impulse	defocus	glass	motion	zoom	snow	frost	fog	bright	contrast	elastic	pixelate	jpeg
							ResNet26								
CIFAR-10															
Source-only	54.13	62.20	49.61	80.08	56.53	76.00	72.57	78.88	73.47	84.62	**90.89**	75.52	81.37	68.54	75.21
BN Adapt	65.73	68.66	66.90	86.46	63.27	82.59	84.11	77.01	76.79	84.00	87.19	84.20	79.21	80.02	68.50
TENT	75.20	78.32	72.70	87.17	68.07	84.16	85.77	79.98	81.27	**87.37**	87.53	84.00	81.17	83.23	78.30
Ours	**76.78**	**79.09**	**74.01**	**87.38**	**69.41**	**84.53**	**86.15**	**80.25**	**81.62**	87.27	87.44	**85.90**	**81.66**	**83.43**	**79.29**
CIFAR-100															
Source-only	15.48	22.12	20.42	52.22	17.18	45.31	45.00	46.53	39.71	54.44	**65.30**	46.72	50.32	43.08	42.53
BN Adapt	31.02	34.27	35.96	58.47	34.34	53.55	55.55	46.69	44.79	53.69	59.99	55.24	50.50	52.10	36.92
TENT	41.00	44.52	42.74	61.65	40.02	56.10	59.77	51.57	51.10	59.96	62.15	58.82	53.88	57.13	46.00
Ours	**41.40**	**44.54**	**43.02**	**62.39**	**40.27**	**57.04**	**60.01**	**51.72**	**51.77**	**60.06**	62.67	**59.13**	**54.06**	**57.52**	**46.68**
							ResNet50								
CIFAR-10															
Source-only	66.51	73.73	68.37	92.71	57.10	84.78	91.87	82.81	82.74	87.22	**94.14**	92.28	88.22	79.11	84.59
BN Adapt	86.74	88.26	82.37	92.05	79.90	88.50	92.62	85.98	89.20	88.08	92.57	92.44	87.66	90.99	88.69
TENT	87.72	89.26	84.30	92.60	81.45	89.51	92.93	87.37	89.97	89.63	92.97	93.02	88.38	91.52	**90.17**
Ours	**88.67**	**90.25**	**86.14**	**93.07**	**83.56**	**90.89**	**93.45**	**89.08**	**90.85**	**91.30**	93.59	**93.60**	**89.56**	**92.03**	90.01
CIFAR-100															
Source-only	33.70	41.94	34.26	70.42	22.94	56.86	67.87	52.31	54.06	59.48	**73.89**	67.16	61.85	54.89	56.79
BN Adapt	59.02	62.32	53.98	69.44	50.36	63.53	69.46	58.64	63.42	60.83	69.90	69.47	61.87	67.35	64.49
TENT	62.47	65.31	58.37	71.57	54.88	66.74	71.63	62.15	66.11	65.84	72.31	71.37	64.93	69.80	66.94
Ours	**64.25**	**67.03**	**60.95**	**73.19**	**57.15**	**68.99**	**73.03**	**64.75**	**67.76**	**68.63**	73.75	**72.98**	**66.46**	**71.46**	**68.32**

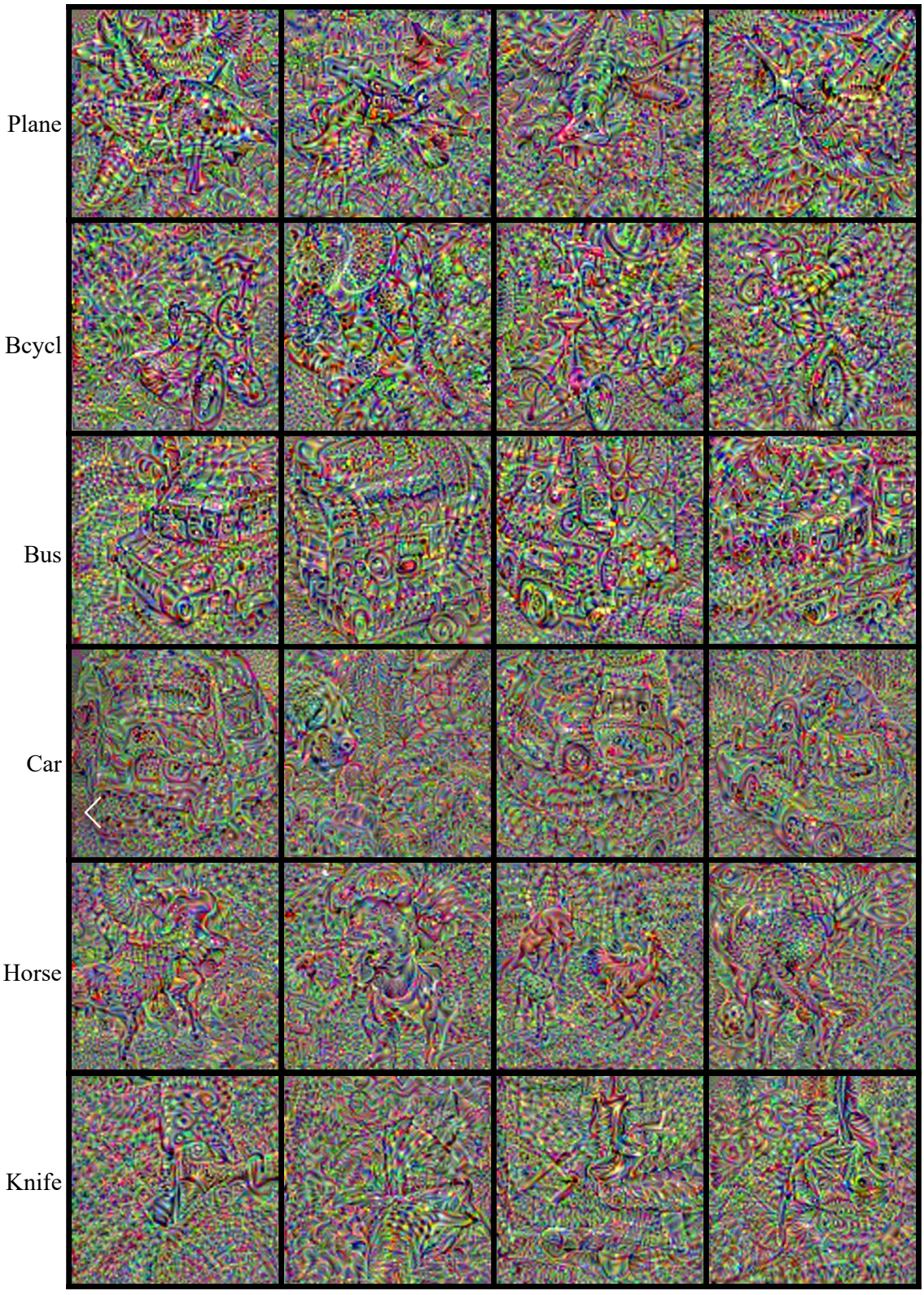


Figure 5. Visualization of condensed data where each row presents 4 images per class of VisDA-C (Part 1).

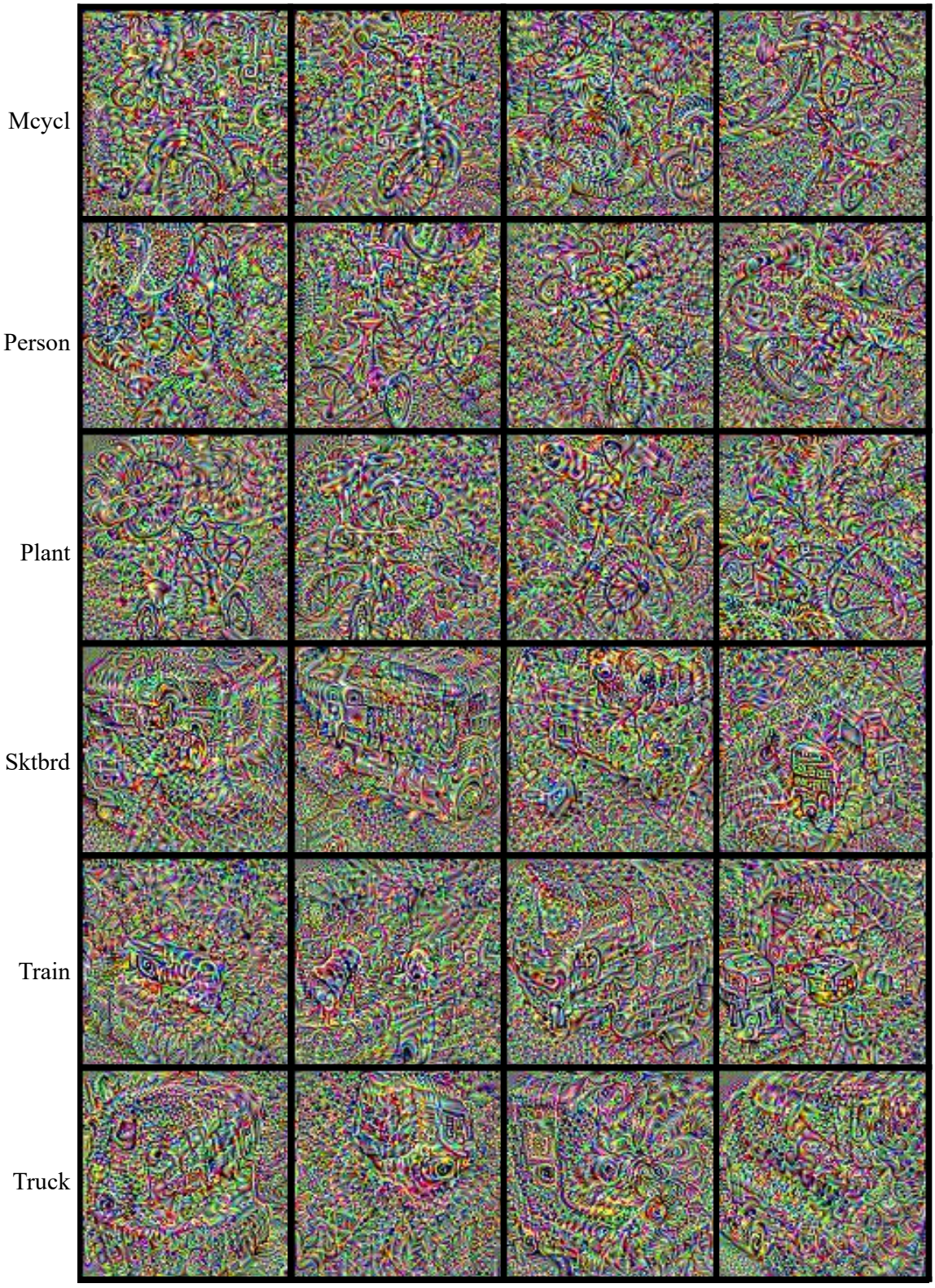


Figure 6. Visualization of condensed data where each row presents 4 images per class of VisDA-C (Part 2).