Automated Synthetic-to-Real Generalization

Wuyang Chen [1] **Zhiding Yu** [2] **Zhangyang Wang** [1] **Anima Anandkumar** [2 3]

Abstract

Models trained on synthetic images often face degraded generalization to real data. As a convention, these models are often initialized with ImageNet pretrained representation. Yet the role of ImageNet knowledge is seldom discussed despite common practices that leverage this knowledge to maintain the generalization ability. An example is the careful hand-tuning of early stopping and layer-wise learning rates, which is shown to improve synthetic-to-real generalization but is also laborious and heuristic. In this work, we explicitly encourage the synthetically trained model to maintain similar representations with the ImageNet pretrained model, and propose a *learning-to-optimize (L2O)* strategy to automate the selection of layer-wise learning rates. We demonstrate that the proposed framework can significantly improve the synthetic-to-real generalization performance without seeing and training on real data, while also benefiting downstream tasks such as domain adaptation. Code is available at: https://github.com/NVlabs/ASG.

1. Introduction

Training a deep convolutional neural network (DCNN) can require large amounts of labeled data in computer vision tasks such as segmentation (Ros et al., 2016; Richter et al., 2016; 2017), depth/flow estimation (Dosovitskiy et al., 2015; Mayer et al., 2016; Gaidon et al., 2016), object detection (Johnson-Roberson et al., 2016), visual navigation (Savva et al., 2019), and grasping (Coumans & Bai, 2016). When there is label scarcity, a popular approach is to resort to training with synthetic images, where full supervision can be obtained at a low cost. This finds applications in label-scarce domains such as robotics and autonomous driving where simulation can play an important role.

[1]Texas A&M University [2]NVIDIA [3]California Institute of Tech. Correspondence to: Wuyang Chen <wuyang.chen@tamu.edu>, Zhiding Yu <zhidingy@nvidia.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

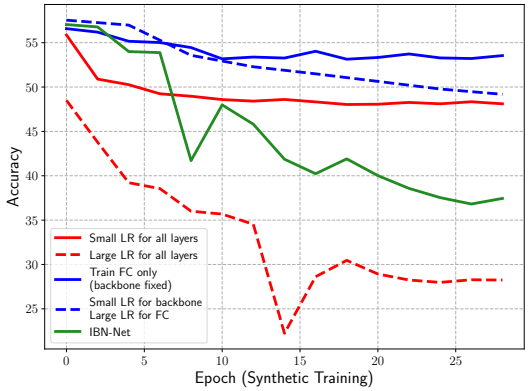

Figure 1. Both heuristic solutions (early stopping, small learning rates, etc.) and recent works (e.g. IBN-Net (Pan et al., 2018)) fall in poor generalization in synthetic-to-real transfer learning, which suffers from the huge appearance gap between the source and the target domain. Here, we studied different learning rates ("LR") or optimization strategies for the backbone and the last fully-connected classification layer ("FC"). All settings start with an ImageNet-pretrained backbone and a randomly initialized classification layer. Please see section 3.2 for experiment details.

However, there are many challenges to train with synthetic images. Models trained on synthetic images often face problems from degraded generalization on the real domain. Such a domain gap is usually caused by limitations on rendering quality, including unrealistic texture, appearance, illumination and scene layout, etc. As a result, networks are prone to overfitting to the synthetic domain with learned representations that differ from those obtained on real images. To this end, domain generalization methods (Li et al., 2017; Pan et al., 2018; Yue et al.) have been proposed to overcome the above domain gaps and improve model generalization on real target domains.

Synthetic-to-real transfer learning involves training a model only on synthetic images (source domain) without seeing any real ones, and targets on the generalization performance on unseen real images (target domain). Recent synthetic-to-real generalization algorithms often start with an ImageNet-pretrained model. To achieve the best generalization performance, it is a common practice to fine-tune the pretrained model on synthetic images for only a few epochs (i.e. **early-stopping**) with a **small learning rate**. Figure 1 illustrates the evaluation dynamics of several popular heuristic solutions on the VisDA-17 dataset (Peng et al., 2017). One could

clearly see the high performance in early epochs, and the improvements of fine-tuning with a small learning rate (or even a fixed backbone) over training with a large one (red dashed line). Similar behavior exists in recent works (e.g. IBN-Net (Pan et al., 2018)). This observation implies an important clue: all these heuristics try to retain the ImageNet domain knowledge during the synthetic-to-real transfer learning. It explains why the heuristic solutions in Figure 1 work: they allow the classifier to quickly adjust from ImageNet to the task defined by the synthetic images, while preventing the ImageNet-pretrained representations of natural images to be "washed out" due to catastrophic forgetting.

Unfortunately, existing solutions (e.g. IBN-Net) still face degraded generalization and are highly dependent on manual selections of training epochs and schedules (learning rates). Motivated by this open issue, we propose an Automated Synthetic-to-real Generalization (**ASG**) framework to improve synthetic-to-real transfer learning. This method is automated from two aspects: (1) It stably improves the generalization during transfer learning, avoiding the difficulty of choosing epochs to stop. (2) It automates the complicated tuning of layer-wise learning rates towards better generalization. The core of our work is the intuition that a good synthetically-trained model should share similar representations with ImageNet-models, and we leverage this intuition as a proxy guidance to search layer-wise training schedules through learning-to-optimize (L2O).

Summary of Contributions:

- We examine the behaviors of various training heuristics, in order to study the role of the ImageNet domain knowledge in synthetic-to-real generalization, which is not thoroughly discussed by the literature to the best of our knowledge.
- We provide a novel perspective to address synthetic-to-real generalization, by formulating it as a lifelong learning problem. We enforce the representation similarity between synthetically trained models and ImageNet-pretrained model, and treat their similarity as a proxy guidance of generalization performance. An overall design is illustrated in Figure 2.
- We demonstrate that proxy guidance not only dramatically improves the generalization performance, but can also be easily integrated by existing transfer learning frameworks as a simple drop-in module, without requiring any additional training beyond synthetic images. Experiments also prove the cross-task generalizability of our proxy guidance, which magnifies the strength of synthetic-to-real transfer learning.
- We design a reinforcement learning based learning-to-optimize (RL-L2O) approach to make the synthetic-to-real generalization practically more convenient, by automating the complicated heuristic designs with layer-wise learning rates. We demonstrate that our RL-L2O method out-performs hand-crafted decisions and learns explainable learning rate strategy.

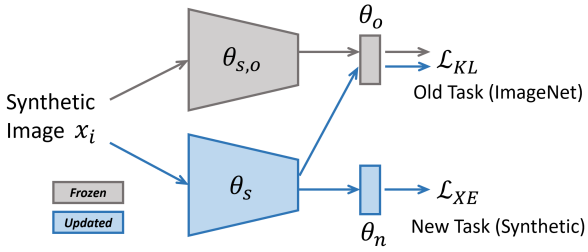

Figure 2. We formulate the synthetic-to-real transfer learning as a lifelong learning problem: training on synthetic images (new task) while still memorizing ImageNet classification (old task), acting as our proxy guidance during the transfer learning.

2. Automated Syn-to-Real Generalization

In our work, we propose an automated framework to address the synthetic-to-real transfer learning, dubbed Automated Synthetic-to-real Generalization (ASG). We assume an ImageNet-pretrained model as our starting point. Our target is to maximize the performance of the model on a target domain which consists of unseen real images, by utilizing only synthetic images from the source domain.

2.1. Syn-to-Real Generalization with Proxy Guidance

The accessibility to model pretrained on ImageNet (Deng et al., 2009) implicitly provides the domain knowledge of real images. As we are transferring a model trained on synthetic data to unseen real images, retaining the ImageNet domain knowledge is potentially beneficial to the generalization. Motivated by this, we force the model to memorize how to capture the representation learned from ImageNet while training on synthetic images, to maintain both the domain knowledge on real images and task-specific information provided by the synthetic data.

We start with an ImageNet-pretrained model $\mathcal{M}$, and formulate our transfer learning as a life-long learning problem: training on synthetic images as the new task while still memorizing the old ImageNet classification task. While updating the model $\mathcal{M}$ with synthetic images, we also keep a copy of the original ImageNet-pretrained model $\mathcal{M}_o$ which is frozen during the training. In addition to the cross-entropy loss $\mathcal{L}_{\text{XE}}$ calculated on the synthetic dataset, we also forward the synthetic images through $\mathcal{M}_o$ and minimize the KL divergence $\mathcal{L}_{\text{KL}}$ between the output of $\mathcal{M}_o$ and $\mathcal{M}$. Formally, we leverage the minimization of $\mathcal{L}_{\text{KL}}$ as a proxy guidance during our transfer learning process:

$$\theta_s^*, \theta_n^* \leftarrow \arg\min_{\theta_s, \theta_n}(\mathcal{L}) \tag{1}$$

$$\mathcal{L} = \mathcal{L}_{\text{XE}} + \lambda \mathcal{L}_{\text{KL}} \tag{2}$$

$$\mathcal{L}_{\text{XE}} = -\frac{1}{N_B}\sum_{i=1}^{N_B} \mathbf{y}_i \log(\mathcal{M}(\mathbf{x}_i, \theta_s, \theta_n)) \tag{3}$$

$$\mathcal{L}_{\text{KL}} = -\frac{1}{N_B}\sum_{i=1}^{N_B} \mathcal{M}_o(\mathbf{x}_i, \theta_{s,o}, \theta_o) \log(\mathcal{M}(\mathbf{x}_i, \theta_s, \theta_o)) \tag{4}$$

Here, λ is a balancing factor that controls how much ImageNet domain knowledge the model should retain. θ_n denotes the parameters for the synthetic-to-real transfer learning $\mathcal{L}_{\text{XE}}$ (i.e. the classifier layers for the new task), θ_o denotes the parameters for ImageNet classifier which will output the predicted probabilities on the ImageNet domain. θ_s denotes the parameters for the feature extractor (a.k.a. backbone) updated for the new tasks, and $\theta_{s,o}$ denotes the parameters for the feature extractor which is frozen with ImageNet-pretrained weights. θ_s and $\theta_{s,o}$ share the same structure. N_B is the current batch size, $\mathbf{x}_i$, and $\mathbf{y}_i$ are sample and ground truth from the new task in the current batch. This synthetic-to-real transfer learning with proxy guidance is illustrated in Figure 2. The new task and the old ImageNet task are jointly optimized during the training.

Cross-task proxy guidance: It is important to note that, the new task is not necessarily limited to be also for the image classification purpose. For some models in semantic segmentation (e.g. ResNet based FCN (Long et al., 2015a)), a pixel-wise $\mathcal{L}_{\text{XE}}$ provides a much denser supervision than the image-wise $\mathcal{L}_{\text{KL}}$ in Eq. 4. To spatially balance $\mathcal{L}_{\text{XE}}$ and $\mathcal{L}_{\text{KL}}$, we also make $\mathcal{L}_{\text{KL}}$ denser by applying it on cropped feature map patches:

$$\mathcal{L}_{\text{KL}}^{\text{dense}} = -\frac{1}{N_B}\frac{1}{N}\sum_{i}^{N_B}\sum_{j}^{N} \mathcal{M}_o(\mathbf{x}_{i,j}, \theta_{s,o}, \theta_o) \log(\mathcal{M}(\mathbf{x}_{i,j}, \theta_s, \theta_o)). \tag{5}$$

Here, $\mathbf{x}_{i,j}$ $(j = 1, \cdots, N)$ are cropped patches from $\mathbf{x}_i$. Later in section 3.4 we will demonstrate that this formulation also works well for cross-task training.

2.2. Automate LR Selection via Learning-to-Optimize

As observed in Figure 1, different convolution blocks contribute differently to the generalizability. This leads to a question: *does different layers in a deep network require different training strategy towards optimal synthetic-to-real generalization performance during the transfer learning?*

To avoid manually tuning the hyperparameters, we propose a reinforcement learning based learning-to-optimize (RL-L2O) framework to automatically adjust the learning rates for layers. In the RL-L2O framework, we aim to learn a parameterized policy π to dynamically control the learning rates given the training statistics of our model $\mathcal{M}$ during transfer learning.

Generally, the goal of the reinforcement learning algorithm is to learn a policy π^* that maximizes the total expected reward r over time. More precisely,

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_T}\left[\sum_{t=0}^{T} r_t\right] \tag{6}$$

where the expectation is taken over the sequence of states (or observations) and actions. In short, an action $\boldsymbol{a}_t$ produced by π will update the learning rates for $\mathcal{M}$ in the RL-L2O framework. A state $\boldsymbol{s}_t$ contains optimization related statistics of the model $\mathcal{M}$ during the transfer learning, and the reward r_t measures how well the optimization performs.

Design of Optimization Coordinates: One challenge in applying reinforcement learning in our setting is that we want to be able to control the training schedules of a deep network of up to a hundred layers (ResNet-101), each of them requiring an action from our policy. As layers may have strong correlations during the optimization (Ghiasi et al., 2018), the policy may fall into sub-optimal solutions in this large scale action space. To avoid this difficulty and simplify our policy training, we leverage the underlying structures in current deep networks. Specifically, layers in $\mathcal{M}$ with similar input resolution will be grouped into a block, named as an *optimization coordinate*. Taking the ResNet family as an example, we group layers into a new coordinate whenever the feature map resolution is reduced. This grouping strategy keeps the action space of the policy small, and speeds-up the L2O training.

Design of Action Space: Intuitively, our policy could directly output learning rate for each coordinate. However, the model $\mathcal{M}$ could be very sensitive to the learning rate (as observed in Figure 1), and the learning rate usually resides in a small value range (e.g. $10^{-4} \sim 10^{-3}$). Directly predicting the value of the learning rate could be very unstable. Instead, we propose a learning rate scaling factor as the action. We first provide the policy a base learning rate η_{base}. In the following steps, π outputs discrete coordinate-wise learning rate scale factors as its actions $\boldsymbol{a}_t = [a_{1,t}, ..., a_{C,t}]$ where C is the number of optimization coordinates in $\mathcal{M}$. We formulate $\boldsymbol{a}_t$ as categorical actions, where each learning rate scale factor $a_{c,t} \in [0, 0.1, 0.2, ..., 0.9, 1]$. The learning rate for each coordinate is set to be $\eta_{c,t} = a_{c,t} \cdot \eta_{\text{base}}$, and we leverage the gradients and momentums calculated by stochastic gradient descent (SGD) (Rumelhart et al., 1986) to update the parameters in $\mathcal{M}$.

Design of Observation Space and Reward: At each step, the state (observation) $\boldsymbol{s}_t$ for π includes: current $\mathcal{L}_{\text{XE},t}$ (Eq. 3) and $\mathcal{L}_{\text{KL},t}$ (Eq. 4, Eq. 5), the training progress of $\mathcal{M}$ (i.e. $\frac{t}{T}$, where T equals to total training steps (i.e., "total epochs"×"iterations per epoch")), the mean and standard

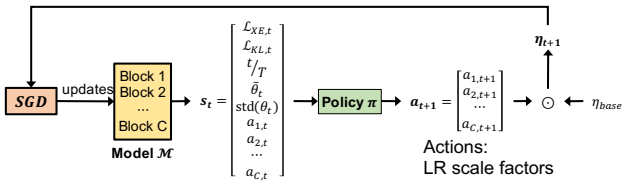

Figure 3. Workflow of the proposed L2O framework. $\boldsymbol{a}_t = [a_{1,t}, ..., a_{C,t}]^T$ is the learning rate scale factor for the coordinates, and η indicates the learning rate. $\odot$ is dot product.

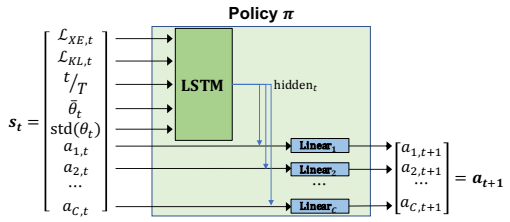

Figure 4. Architecture of the policy network.

deviation of the weights of the classifier ($\bar{\theta}_{n,t}$ and $\mathrm{std}(\theta_{n,t})$), and finally the scale factors from the last step $\boldsymbol{a}_{t-1}$. The policy learning is guided by reward $r_t = \mathcal{L}_{t-1} - \mathcal{L}_t$.

Policy Training: We update our LSTM policy π via the REINFORCE algorithm (Williams, 1992) to minimize:

$$\mathcal{L}_\pi = -\frac{1}{U}\sum_{t\in \mathrm{U}} r_t \cdot \log(p_\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)), \tag{7}$$

where U is the unroll length for LSTM. Algorithm 1 illustrate the procedure of our RL-L2O framework.

Once we obtained the learned policy π, we then freeze and apply it to the synthetic-to-real transfer learning of $\mathcal{M}$ together with SGD, as illustrated in Figure 3.

Algorithm 1: RL-L2O: policy (π) learning to control group-wise learning rates.

Input: base learning rate η_{base}, parameters $\theta_{n,0}, \theta_{s,0}$, hidden state $\boldsymbol{h}_0 = \mathbf{0}$, policy π, unroll length U, total training steps T.
Calculate $\mathcal{L}_0, \mathcal{L}_{\mathrm{XE},0}, \mathcal{L}_{\mathrm{KL},0}$ for $\theta_{n,0}, \theta_{s,0}$
Initialize storage
for $t = 0, \ldots, T-1$ **do**
  $\mathrm{prob}(\boldsymbol{a}_{t+1}), \boldsymbol{a}_{t+1}, h_{t+1} =$ $\pi(\mathcal{L}_{\mathrm{XE},t}, \mathcal{L}_{\mathrm{KL},t}, \frac{t}{T}, \bar{\theta}_{n,t}, \mathrm{std}(\theta_{n,t}), \boldsymbol{a}_t, h_t)$
  $(\theta_{n,t+1}, \theta_{s,t+1}) =$ $\mathrm{SGD}(\nabla\mathcal{L}_t, \boldsymbol{a_{t+1}}, \eta_{\mathrm{base}}, \theta_{n,t}, \theta_{s,t})$
  Calculate $\mathcal{L}_{t+1}, \mathcal{L}_{\mathrm{XE},t+1}, \mathcal{L}_{\mathrm{KL},t+1}$ for $\theta_{n,t+1}, \theta_{s,t+1}$
  $r_{t+1} = \mathcal{L}_t - \mathcal{L}_{t+1}$
  storage.append($\mathrm{prob}(\boldsymbol{a}_{t+1}), r_{t+1}$)
  if $(t+1)\%U == 0$ **then**
    $\pi = \mathrm{REINFORCE}(\pi, \mathrm{storage})$
    Initialize storage
return *final learned policy* π.

3. Experiments

3.1. Datasets

VisDA-17 (Peng et al., 2017) We perform ablation study on the VisDA-17 image classification benchmark. The VisDA-17 dataset provides three subsets (domains), each with the same 12 object categories. Among them, the training set (source domain) is collected from synthetic renderings of 3D models under different angles and lighting conditions, whereas the validation set (target domain) contains real images cropped from the Microsoft COCO dataset (Lin et al., 2014).

GTA5 (Richter et al., 2016) is a vehicle-egocentric image dataset collected in a computer game with pixel-wise semantic labels. It contains 24,966 images with a resolution of 1052×1914. There are 19 classes that are compatible with the Cityscapes dataset.

Cityscapes (Cordts et al., 2016) contains urban street images taken on a vehicle from some European cities. There are 5,000 images with pixel-wise annotations. The images have a resolution of 1024×2048 and are labeled into 19 semantic categories.

3.2. Implementation

Image classification: For VisDA-17, we choose ResNet-101 (He et al., 2016) as the backbone, and one fully-connected layer as the classifier. Backbone is pre-trained on ImageNet (Deng et al., 2009), and then fine-tuned on source domain, with learning rate = 1×10^{-4}, weight decay = 5×10^{-4}, momentum = 0.9, and batch size = 32. The model is trained for 30 epochs and λ for $\mathcal{L}_{\mathrm{KL}}$ is set as 0.1. In section 3.3, we will additionally study how to choose λ.

Semantic segmentation: We study both FCN with ResNet-50 and FCN with VGG-16 (Long et al., 2015a). Backbones are pre-trained on ImageNet. Our learning rate is 1×10^{-3}, weight decay is 5×10^{-4}, momentum is 0.9, and batch size is six. We crop the images into patches of 512×512 and train the model with multi-scale augmentation ($0.75 \sim 1.25$) and horizontal flipping. The model is trained for 50 epochs, and λ for $\mathcal{L}_{\mathrm{KL}}$ is set as 75. Note that λ in segmentation is considerably larger since $\mathcal{L}_{\mathrm{XE}}$ is a pixel-wise dense loss.

RL-L2O policy: We set the learning rate for policy training as 0.5. The size of the hidden state vector $\boldsymbol{h}$ is set to 20, and the unroll length $U = 5$. We train π for 50 epochs. For the ResNet family, we follow the convention (He et al., 2016) to group the layers into $C = 7$ coordinates: $\mathrm{conv1}, \mathrm{bn1}, \mathrm{conv2}, \mathrm{conv3}, \mathrm{conv4}, \mathrm{conv5}$,

and the classifier. For VGG-16 (Long et al., 2015a), we also group the layers into $C = 7$ coordinates: conv1, conv2, conv3, conv4, conv5, conv6&7, and the remaining projection_upsampling layers.

Proxy guidance: For all backbones we studied (ResNet-50, ResNet-101, and VGG-16), we forward the feature maps extracted by group conv5 into the ImageNet classifier (parameterized by θ_o) to calculate $\mathcal{L}_{\text{KL}}$.

3.3. ASG for Image Classification

We first perform the ablation studies on the VisDA-17 image classification task[1].

Generalization with Proxy Guidance. To evaluate the effect of our proxy guidance, we apply our $\mathcal{L}_{\text{KL}}$ loss on different learning rate settings we studied in Figure 1. As demonstrated in Figure 5, once we force the model to memorize the ImageNet domain knowledge, we achieve stably increasing and eventually better generalization performance for each setting we explored in Figure 1. The relative ranking still holds among the different learning rate settings, while the degraded generalizability is addressed. Early stopping is no longer needed, as models enjoy improved generalization given sufficient training epochs. This ablation study validates the contribution of retaining the ImageNet domain knowledge during the synthetic-to-real transfer learning. It is also worth noting that our proxy guidance can be also applied to different networks (e.g. the IBN-Net (Pan et al., 2018), green line in Figure 5), which demonstrate the easy integration of our approach as a simple drop-in module with existing synthetic-to-real generalization works, without requiring any additional training beyond synthetic images.

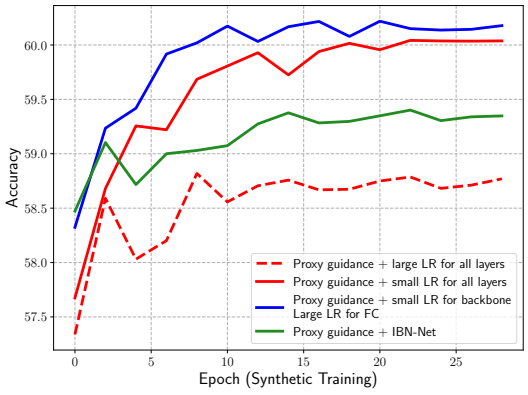

Figure 5. The degraded generalization during the synthetic-to-real transfer learning (studied in Figure 1) can be solved by forcing the model to retain the ImageNet domain knowledge via our proxy guidance[2] Task: ResNet-101 VisDA-17 Classification. $\lambda = 0.1$.

[1]There is no previous synthetic-to-real transfer work on VisDA-17 classification task, only domain adaptation works.

Moreover, a vital conclusion from Figure 5 is that, only reporting the (final) performance as a number is far from sufficient for analyzing and comparing synthetic-to-real transfer learning methods. Instead, the curve of the target performance during training can better demonstrate how well a model's generalizability is. Meanwhile, a stably increasing training curve implies that, the model is both better leveraging synthetic images and retaining ImageNet domain knowledge, instead of overfitting on synthetic appearance and leaving the domain gap an open issue.

How to choose λ: We also study the effect of different strengths of the proxy guidance loss $\mathcal{L}_{\text{KL}}$ by adjusting λ in Equation 2 for a ResNet-101 model trained with a small learning rate for the backbone and a large one for the classification layer (blue line in Figure 5). In Table 1, we adjust λ in a wide range from 0.01 to 1. While we obtain the best generalization accuracy with $\lambda = 0.1$, we can see that our proxy guidance is very robust to different strength of $\mathcal{L}_{\text{KL}}$. Therefore, choosing λ is much easier than tuning hyperparameters in heuristic solutions like epochs.

Table 1. Ablation of λ for the proxy guidance loss $\mathcal{L}_{\text{KL}}$. Model: ResNet-101. Task: VisDA-17 Classification.

λ	0.01	0.05	0.1	0.5	1
Accuracy (%)	58.9	59.4	**60.1**	58.5	59.7

Automated Syn-to-Real Generalization. We next evaluate the performance of our RL-L2O framework. Specifically, we want to make sure the policy learned by our RL-L2O can perform better than both the random policy and the best hand-tuned learning rate policy we explored in Figure 5. A random policy means that the controller will always randomly pick an action as the learning rate scale factor. In all these three settings we start from the same base learning rate $\eta_{\text{base}} = 1 \times 10^{-4}$. Figure 6 demonstrates that, while the hand-tuned learning rate strategy is better than a random policy, our RL-L2O framework can even out-perform the human-designed one (blue line).

Additional Ablation Study on VisDA-17. We conduct additional ablation studies on VisDA-17 to further analyze the learning behaviors of ASG. Specifically, as both the proxy guidance and the RL-L2O frameworks are motivated to carefully preserve the ImageNet representations while targeting updates from the new tasks on synthetic data, it is interesting and important to connect the relation between the level of retained ImageNet knowledge and the synthetic-to-real generalization. In our experiment, we compute ImageNet validation accuracy as well as the generalization performance on Visda-17 target domain for the classification task.

[2]We could not utilize the proxy guidance when the backbone is fixed ("Train FC Only" blue dashed curve in Figure 1). The $\mathcal{L}_{\text{KL}}$ is always zero in this case as the group conv5 is not updated.

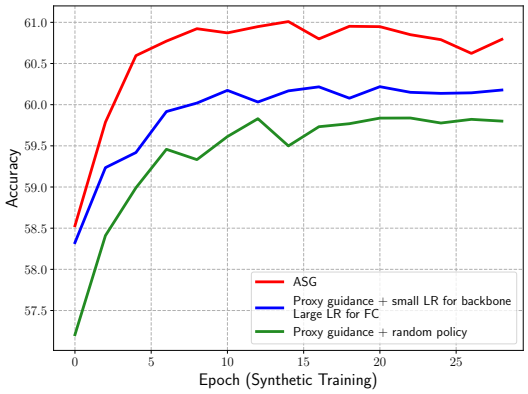

Figure 6. Our RL-L2O framework can out-perform both the random policy and a carefully hand-tuned learning rate strategy. All three settings include $\mathcal{L}_{\mathrm{KL}}$ with the same $\lambda = 0.1$ during training. Model: ResNet-101. Task: VisDA-17 Classification.

Table 2 demonstrates two conclusions: 1) Heuristic solutions that retain more ImageNet domain knowledge achieve higher synthetic-to-real generalization (#3 versus #1), i.e., using hand-crafted small learning rates to prevent the ImageNet-pretrained representations of natural images from being "washed out" due to catastrophic forgetting; 2) By leveraging Proxy Guidance, the generalization performance on VisDA-17 is dramatically improved, while the ImageNet accuracy is also maintained with almost no drop. It is interesting that Proxy Guidance leads to learned model parameters that achieve high accuracy simultaneously on both ImageNet and VisDA-17. In contrast, naively freezing the backbone and only fine-tuning the classifier layer ("Oracle" #5) results in inferior synthetic-to-real generalization despite high ImageNet performance.

Table 2. Our Proxy Guidance improves the synthetic-to-real generalization (Visda-17) by retaining the ImageNet domain knowledge. Learning rate (LR) settings were studied in Figure 1 and 5. FC: the last fully-connected classification layer. Top1 accuracies are in percentage (%). Model: ResNet-101.

#	Model	VisDA-17	ImageNet
1.	Large LR for all layers	28.2	0.8
2.	+ our Proxy Guidance	58.7 (+30.5)	76.2 (+75.4)
3.	Small LR for backbone and large LR for FC	49.3	33.1
4.	+ our Proxy Guidance	60.2 (+10.9)	76.5 (+43.4)
5.	Oracle on ImageNet[3]	53.3 (+4.0)	**77.4**
6.	ROAD (Chen et al., 2018)	57.1 (+7.8)	**77.4**
7.	Vanilla L2 distance	56.4 (+7.1)	49.1
8.	SI (Zenke et al., 2017)	57.6 (+8.3)	53.9
9.	ASG (ours)	**61.1**	76.7

[3]Oracle is obtained by freezing the ResNet-101 backbone while only training the last new fully-connected classification layer on the Visda-17 source domain (the FC layer for ImageNet remains unchanged). We use the PyTorch official model of ImageNet-pretrained ResNet-101.

In addition, we compare ASG with several other lifelong learning algorithms, including both feature-level ℓ_2 regularization (Chen et al., 2018) and weight-level importance-reweighted ℓ_2 constraints (Zenke et al., 2017). Row #5∼8 in Table 2 shows that although the three comparing methods indeed retain ImageNet domain knowledge while improving over the baseline (49.3%), they are not performing as well as the proxy guidance (60.2%) under the same LR policy.

3.4. ASG for Semantic Segmentation

We also conduct experiments to evaluate the generalization performance of ASG on semantic segmentation. In particular, we treat GTA5 as the synthetic source domain and train segmentation models on it. We then treat the Cityscapes validation/test sets as target domains where we directly evaluate the synthetically trained models.

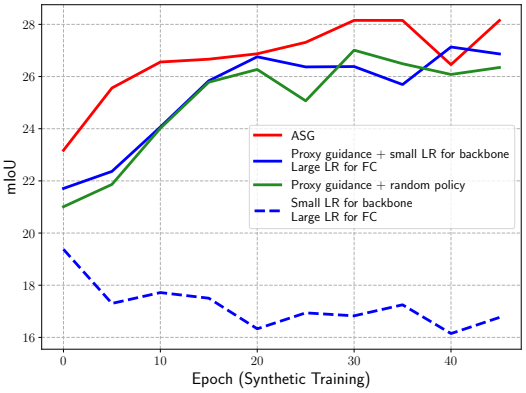

Figure 7. Dynamics of evaluation accuracy with training epochs. Models are trained on GTA5 and directly tested on the Cityscapes validation set. We use FCN-VGG16 as the backbone for segmentation models. In addition, $\mathcal{L}_{\mathrm{KL}}$ in all comparing methods share the same parameter $\lambda = 75$ during synthetic source training.

Figure 7 shows the dynamics of evaluation accuracy on the Cityscapes validation set. Again, ASG demonstrates significantly improved generalization performance on semantic segmentation over naive synthetic training. In addition, integrating proxy guidance with RL-L2O also consistently outperforms baselines where proxy guidance is integrated with other policy strategies. Note that in this case, both θ_o and $\mathcal{L}_{\mathrm{KL}}$ are oriented to the classification task, while θ_n and $\mathcal{L}_{\mathrm{XE}}$ designed for segmentation. This showcases the ability of ASG to generalize across different tasks.

In Table 3, we compare our method with prior domain generalization methods for semantic segmentation. One can see that ASG achieves the best performance gain. Among the comparing methods, IBN-Net (Pan et al., 2018) im-

proves domain generalization by fine-tuning the mixed IN-BN residual building blocks, while (Yue et al.) transfers the styles from images in ImageNet to synthetic images. It is worth noting that (Yue et al.) requires ImageNet images during training and implicitly leverages ImageNet label information (i.e. "Auxiliary Domains") which brings potential advantages. In contrast, our method requires minimum extra information without using any additional images or labels, therefore can be conveniently applied to existing frameworks as a drop-in training strategy.

Table 3. Comparison to prior methods on domain generalization for semantic segmentation (GTA5→ Cityscapes).

Methods	Model	mIoU %	mIoU ↑ %
No Adapt IBN-Net (2018)	FCN-Res50	22.17 29.64	7.47
No Adapt Yue et al. ()	FCN-Res50	32.45 37.42	4.97
No Adapt Ours	FCN-Res50	23.29 31.89	**8.60**
No Adapt Yue et al. ()	FCN-VGG16	29.81 36.11	6.3
No Adapt Ours	FCN-VGG16	19.89 31.47	**11.58**

Policy Behaviors. Figure 8 shows clear and explainable behavior patterns of our policy for FCN-VGG16 on the segmentation task. In FCN-VGG16, groups $\mathrm{conv1}-5$ belong to the ImageNet-pretrained backbone, while $\mathrm{conv6\&7}$ and the remaining $\mathrm{projection\_upsampling}$ layers act as the classifier for the dense predictions. The feature map captured by $\mathrm{conv5}$ is forward into θ_o to calculate $\mathcal{L}_{\mathrm{KL}}$. As $\mathrm{conv5}$ is close to the calculation of $\mathcal{L}_{\mathrm{KL}}$, fixing $\mathrm{conv5}$ (i.e. selecting action = 0 which represents the learning rate scale factor = 0) can effectively minimize $\mathcal{L}_{\mathrm{KL}}$ and retain the ImageNet domain knowledge. As parameters from group $\mathrm{conv5}$ to $\mathrm{conv1}$ are gradually far from the $\mathcal{L}_{\mathrm{KL}}$ supervision, the corresponding selected actions also increase.

On the other hand, to perform dense prediction in semantic segmentation, the extracted feature maps are first forwarded to $\mathrm{conv6\&7}$ and then to $\mathrm{projection\_upsampling}$. In addition, similar trend holds for the classifier part: as $\mathrm{projection\_upsampling}$ is the closest group to $\mathcal{L}_{\mathrm{XE}}$, it is assigned with the highest scale factor for learning rate.

3.5. ASG for Unsupervised Domain Adaptation

The proposed ASG framework not only can improve the synthetic-to-real generalization performance, but also can considerably benefit downstream tasks such as unsupervised domain adaptation. Here we present synthetic-to-real domain adaptation results on VisDA-17 (Peng et al., 2017) in Table 4, where the model trained by ASG (which did not use any real target images during training) is leveraged as

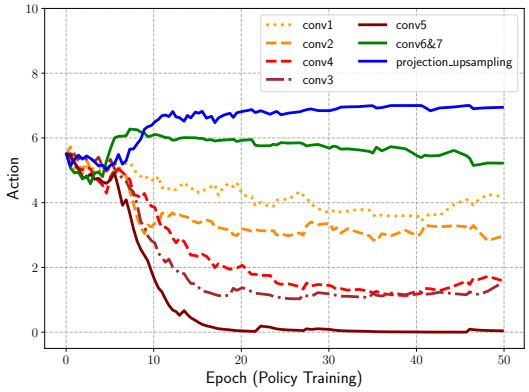

Figure 8. Action behavior of our RL-L2O framework during the policy training for $\mathcal{M}$ = FCN-VGG16 for the GTA5→Cityscapes segmentation transfer learning. Categorical actions are smoothed for better visualization purpose. Actions of $[0, 1, \cdots, 10]$ indicate learning rate scale factors $[0, 0.1, \cdots, 1.0]$.

the source model (i.e., starting point for the unsupervised domain adaptation training), and the CBST/CRST frameworks are adopted exactly following (Zou et al., 2018; 2019) for fair comparison purposes.

Starting from a much better initialization (our 61.1% compared with 51.6% in (Zou et al., 2019)), we significantly boost the adaptation performance over 6% compared with CBST/CRST, achieving 84.6% on Visda-17. It is important to emphasize that such improvement is obtained without any extra supervision and external knowledge. The only difference lies in smarter synthetic-to-real source training which ultimately leads to improved adaptation.

Table 4. Synthetic-to-real adaptation on Visda-17. We follow the same settings in (Zou et al., 2019) to set the weights as 0.1 and 0.25 for MRKLD and LRENT respectively, and report the averages and standard deviations (in brackets) of the evaluation results over five runs. Model: ResNet-101. "Tgt Img": whether the method leveraged target real images during training. Top-1 accuracies are in percentage (%).

Method	Tgt Img	Accuracy
Source (Saito et al., 2017)	✗	52.4
DANN (Ganin et al., 2016)	✓	57.4
MCD (Saito et al., 2018)	✓	71.9
ADR (Saito et al., 2017)	✓	74.8
SimNet-Res152 (Pinheiro, 2018)	✓	72.9
GTA-Res152 (2018)	✓	77.1
Source-Res101 (Zou et al., 2019)	✗	51.6
CBST (Zou et al., 2018)	✓	76.4 (0.9)
CRST (MRKLD) (2019)	✓	77.9 (0.5)
CRST (MRKLD + LRENT) (2019)	✓	78.1 (0.2)
Source-Res101 (ASG)	✗	61.1
ASG + CBST	✓	82.5 (0.7)
ASG + CRST (MRKLD)	✓	**84.6** (0.4)
ASG + CRST (MRKLD + LRENT)	✓	84.5 (0.4)

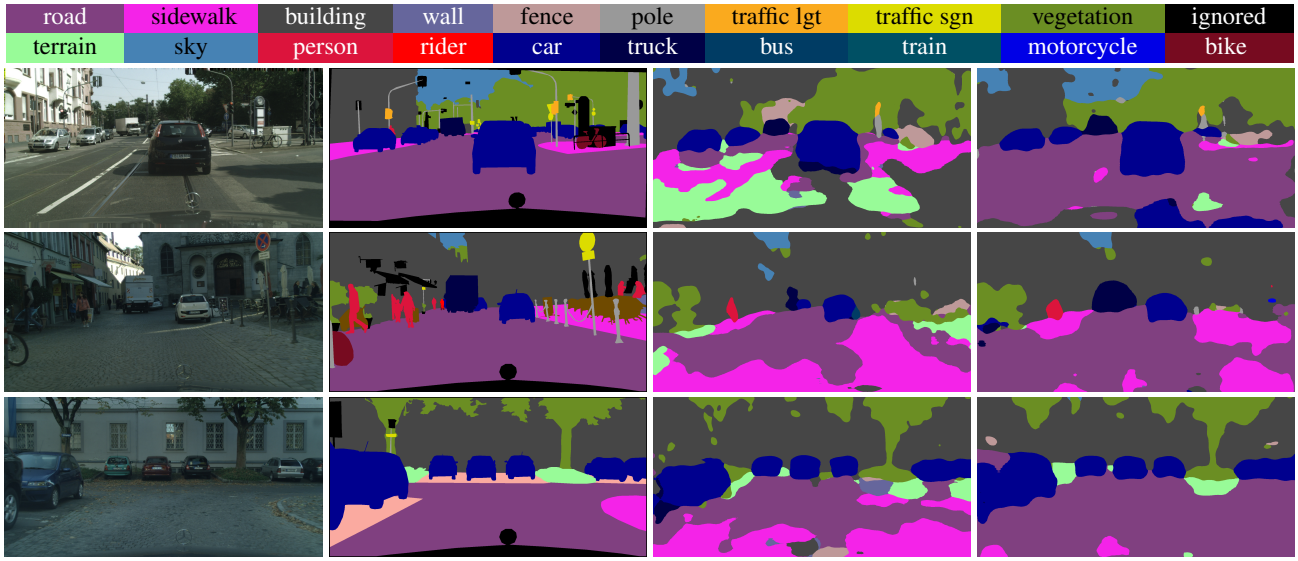

Figure 9. Generalization results on GTA5 → Cityscapes. Rows correspond to sample images in Cityscapes. From left to right, columns correspond to original images, ground truth, predication results of baseline (FCN-VGG16 (Long et al., 2015a)), and prediction by model trained with our ASG framework.

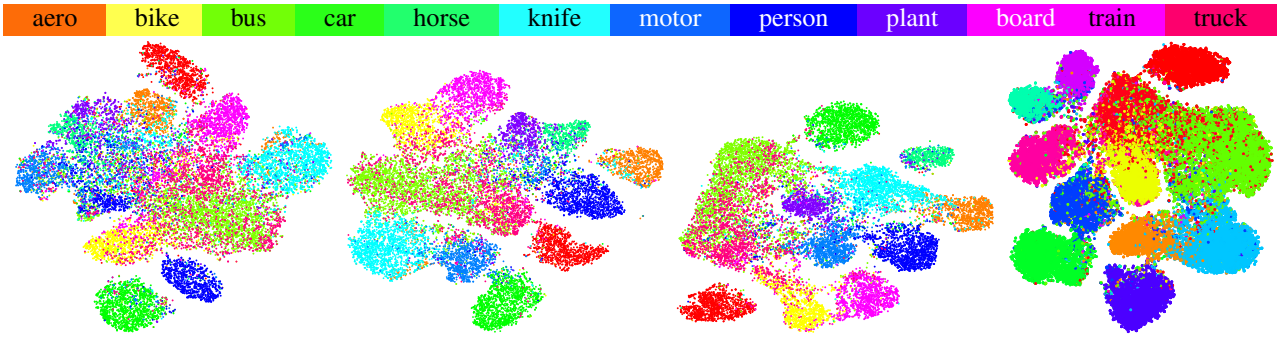

Figure 10. t-SNE visualization of feature embeddings of different models on the target domain of VisDA-17. From left to right: source model (Zou et al., 2019), CBST (Zou et al., 2018), CRST (MRKLD+LRENT) (Zou et al., 2019), and ASG + CRST (MRKLD+LRENT).

Feature visualization. We show the t-SNE visualization of the feature embeddings extracted by the backbone (ResNet-101) of different models in Fig. 10. Compared with Both CBST (Zou et al., 2018) and CRST (MRKLD+LRENT) (Zou et al., 2019), feature embeddings obtained by ASG + CRST form purer clusters in terms of semantic labels.

4. Related Work

4.1. Domain Generalization and Adaptation

Domain generalization considers the problem of generalizing a model on the unseen target domain without leveraging any target domain images (Gan et al., 2016; Muandet et al., 2013; Yuan et al., 2020). Muandet et al. (2013) proposed to use the MMD (Maximum Mean Discrepancy) to align the distributions from different domains and train the network with adversarial learning. Li et al. (2017) built separate networks for each source domain and used the shared parameters for the test. Li et al. (2018) improved the generalization performance by using a meta-learning approach on the split training sets. Pan et al. (2018) boosted a CNNs generalization by carefully integrating the Instance Normalization and Batch Normalization as building blocks.

Unsupervised domain adaptation (UDA) trains a model towards a specific target domain, where the (unlabeled) images from the target domain are available for training. One major idea is to learn domain invariant embeddings by minimizing the distribution divergence between the source and target domain (Long et al., 2015b; Sun & Saenko, 2016; Tzeng et al., 2014). Hoffman et al. (2017) reduced domain gap by first translating the source images into target style with a cycle consistency loss, and then aligning the feature maps of the

network across different domains through the adversarial training. Other works that leverage image level translation to bridge the domain gap include domain stylization (**?**) and DLOW (**?**). Besides image-level translation, a number of works also perform adversarial learning at feature or output level for improved domain adaptation performance.

More recently, Zou et al. (2018) proposed an Expectation-maximization like UDA framework based on an iterative self-training process, where the loss of the latent variable is minimized. This is achieved by alternatively generating pseudo labels on target data and re-training the model with the mixed source and pseudo target labels.

In contrast to existing domain generalization and adaptation methods, we resort to leveraging the ImageNet-pretrained model as a proxy guidance during the synthetic-to-real transfer learning, without any extra adversarial training or modification to model architecture.

4.2. Lifelong Learning

Lifelong learning (Thrun, 1998) focuses on flexibly appending new tasks to the model's training schedules, while maintaining the knowledge captured from previous old tasks. Li & Hoiem (2017) leverages only new task data to train the network while preserving the original capabilities by minimizing the outputs between the old network and the newly learned one. Lopez-Paz and Ranzato (2017) proposed a Gradient Episodic Memory (GEM) to alleviate the knowledge forgetting while transferring knowledge from previous tasks. Shin et al. (2017) developed a Deep Generative Replay framework, which is used to sample training data from previous tasks when training the new task. A number of other works on lifelong learning with related or similar applications include (Zenke et al., 2017; Kirkpatrick et al., 2017; Shafahi et al., 2019) where lifelong learning is shown to avoid catastrophic forgetting and benefit tasks such as incremental tasks learning, domain adaptation and adversarial defense. One work that is particularly related to our synthetic-to-real generalization theme is (Chen et al., 2018) where the authors propose a spatial aware adaptation scheme and also leverage a distillation loss to avoid overfitting to synthetic data. Our work differs from the above prior works by carefully looking into the important role played by layer-wise learning rate policies in synthetic-to-real transfer learning problems and accordingly propose a principled solution to automate the policy search.

4.3. Learning to Optimize

Andrychowicz et al. (2016) proposed the first learning-to-optimize framework, where both the optimizee's gradients and loss function values were formulated as the input features for a Recurrent neural network (RNN) optimizer. Their RNN optimizer adopted coordinate-wise weight sharing to alleviate the dimensionality challenge. Li and Malik (2016) used the gradient history and objective values as observations and step vectors as actions in their reinforcement learning framework. Chen et al. (2017) leveraged RNN to train a meta-optimizer to optimize black-box functions (e.g. Gaussian process bandits). Recently, Wichrowska et al. (2017) introduced an optimizer of multi-level hierarchical RNN architecture augmented with additional architectural features, in order to improve the generalizability of the optimization tasks. (Cao et al., 2019; You et al., 2020) further extended learned optimizers to handling Bayesian swarm optimization, and graph network training, respectively. In our work, we leverage the learning-to-optimize approach to control the layer-wise learning rates for the training of deep CNNs, where the deep CNN (i.e. optimizee) will be transferred from the synthetic source domain to the real target domain, extending the application range of the current learning-to-optimize methods.

5. Conclusion

In this paper, we present an Automated Synthetic Generalization (ASG) method for the synthetic-to-real transfer learning problem. We carefully analyzed the pitfall in existing generalization approaches where the ImageNet domain knowledge is catastrophically forgotten. By leveraging the minimization of predictions between ImageNet-pretrained model and the model for the new task as a proxy guidance, the generalization performance is dramatically improved during the whole training process. We further include a reinforcement learning based learning-to-optimize strategy to automate the layer-wise learning rates towards a better generalization performance. Our experiments demonstrate both the superior generalization performance and the automated learning schedules by our ASG framework.

6. Acknowledge

Work done during internship at NVIDIA. We appreciate the computing power supported by NVIDIA GPU infrastructure. We also thank for the discussion and suggestions from four anonymous reviewers and the help from Yang Zou for the domain adaptation experiments. The research of Z. Wang was partially supported by NSF Award RI-1755701.

References

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. In *NeurIPS*, 2016.

Cao, Y., Chen, T., Wang, Z., and Shen, Y. Learning to optimize in swarms. In *NeurIPS*, 2019.

Chen, Y., Hoffman, M. W., Colmenarejo, S. G., Denil, M., Lillicrap, T. P., Botvinick, M., and de Freitas, N. Learning to learn without gradient descent by gradient descent. In *ICML*, 2017.

Chen, Y., Li, W., and Van Gool, L. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *CVPR*, 2018.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Coumans, E. and Bai, Y. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

Gaidon, A., Wang, Q., Cabon, Y., and Vig, E. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

Gan, C., Yang, T., and Gong, B. Learning attributes equals multi-source domain generalization. In *CVPR*, 2016.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.

Ghiasi, G., Lin, T.-Y., and Le, Q. V. Dropblock: A regularization method for convolutional networks. In *NeurIPS*, 2018.

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

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv:1711.03213*, 2017.

Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S. N., Rosaen, K., and Vasudevan, R. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv:1610.01983*, 2016.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

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

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018.

Li, K. and Malik, J. Learning to optimize. *arXiv:1606.01885*, 2016.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE Trans. PAMI*, 40(12):2935–2947, 2017.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015a.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. *arXiv:1502.02791*, 2015b.

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

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

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

Pan, X., Luo, P., Shi, J., and Tang, X. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018.

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

Pinheiro, P. O. Unsupervised domain adaptation with similarity learning. In *CVPR*, 2018.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

Richter, S. R., Hayder, Z., and Koltun, V. Playing for benchmarks. In *ICCV*, 2017.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Saito, K., Ushiku, Y., Harada, T., and Saenko, K. Adversarial dropout regularization. *arXiv:1711.01575*, 2017.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. Habitat: A platform for embodied ai research. In *ICCV*, 2019.

Shafahi, A., Saadatpanah, P., Zhu, C., Ghiasi, A., Studer, C., Jacobs, D., and Goldstein, T. Adversarially robust transfer learning. *arXiv:1905.08232*, 2019.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. In *NeurIPS*, 2017.

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

Thrun, S. Lifelong learning algorithms. In *Learning to learn*, pp. 181–209. Springer, 1998.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv:1412.3474*, 2014.

Wichrowska, O., Maheswaranathan, N., Hoffman, M. W., Colmenarejo, S. G., Denil, M., de Freitas, N., and Sohl-Dickstein, J. Learned optimizers that scale and generalize. In *ICML*, 2017.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

You, Y., Chen, T., Wang, Z., and Shen, Y. L^2-gcn: Layer-wise and learned efficient training of graph convolutional networks. In *CVPR*, 2020.

Yuan, Y., Chen, W., Chen, T., Yang, Y., Ren, Z., Wang, Z., and Hua, G. Calibrated domain-invariant learning for highly generalizable large scale re-identification. In *WACV*, 2020.

Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., and Gong, B. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *ICCV*.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *ICML*, 2017.

Zou, Y., Yu, Z., Vijaya Kumar, B., and Wang, J. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.

Zou, Y., Yu, Z., Liu, X., Kumar, B., and Wang, J. Confidence regularized self-training. In *ICCV*, 2019.