Test-Time Fast Adaptation for Dynamic Scene Deblurring via Meta-Auxiliary Learning

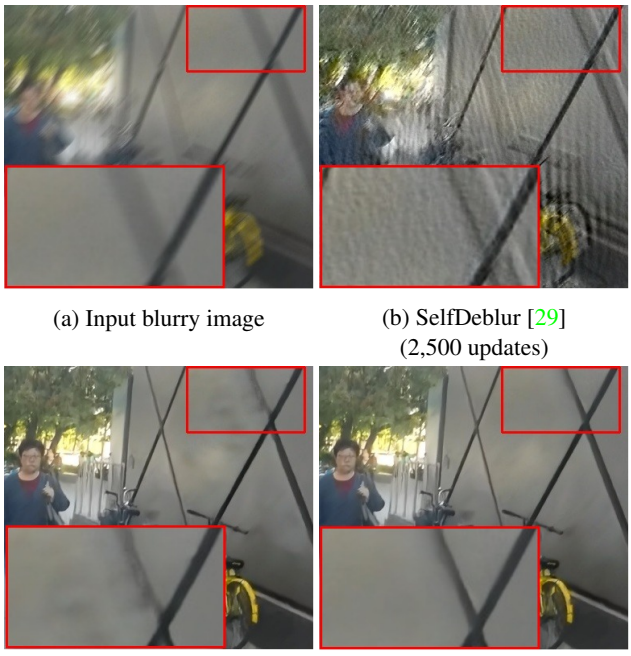
Zhixiang Chi[1], Yang Wang[1,2], Yuanhao Yu[1], Jin Tang[1]
[1]Noah's Ark Lab, Huawei Technologies [2]University of Manitoba, Canada
{zhixiang.chi, yang.wang3, yuanhao.yu, tangjin}@huawei.com

Abstract

In this paper, we tackle the problem of dynamic scene deblurring. Most existing deep end-to-end learning approaches adopt the same generic model for all unseen test images. These solutions are sub-optimal, as they fail to utilize the internal information within a specific image. On the other hand, a self-supervised approach, SelfDeblur, enables internal training within a test image from scratch, but it does not fully take advantage of large external datasets. In this work, we propose a novel self-supervised meta-auxiliary learning to improve the performance of deblurring by integrating both external and internal learning. Concretely, we build a self-supervised auxiliary reconstruction task that shares a portion of the network with the primary deblurring task. The two tasks are jointly trained on an external dataset. Furthermore, we propose a meta-auxiliary training scheme to further optimize the pre-trained model as a base learner, which is applicable for fast adaptation at test time. During training, the performance of both tasks is coupled. Therefore, we are able to exploit the internal information at test time via the auxiliary task to enhance the performance of deblurring. Extensive experimental results across evaluation datasets demonstrate the effectiveness of test-time adaptation of the proposed method.

1. Introduction

Images taken in dynamic scenes are often degraded by objectionable blur caused by object motions and camera shake. Restoring latent clean images from such cases is challenging due to the spatially non-uniform property of the blur kernels. Despite its highly ill-posed characteristic, extensive research efforts have been devoted to remove the notorious blurry artifacts in the past decades [7, 6, 44, 46, 8, 9]. Recently, deep neural networks (DNNs) have been popular in this field. Various network structures have been proposed to achieve reliable quantitative results and generate visually pleasing clean images for dynamic scene deblurring [37, 1, 31, 24]. In particular, the end-to-end learning approaches have set the state-of-the-art by directly learning the statistical correlation between blurry and latent images on large-scale training datasets [40, 13, 27, 36, 47, 25, 23, 5, 2].



(a) Input blurry image (b) SelfDeblur [29] (2,500 updates)

(c) Meta-auxiliary learned (ours, no updates) (d) Meta-auxiliary learned (ours, *five* updates)

Figure 1: Sample deblurring results. Given a blurry input image (a), SelfDeblur [29] requires thousands of iterations to learn the internal information of the input image (b). Our approach uses meta-auxiliary learning to learn to adapt to the unique properties of the given image. Before adaption, the output has some artifacts due to the distribution shift between the training dataset and the test image (c). After five updates, our model quickly adapts to the internal information of the test image and removes the artifacts (d).

The main shortcoming for most of the existing DNN-

based methods is that the same set of trained weights are adopted for all unseen test images. However, the features learned from the external training data may not be optimal for the given test image [26]. The failure of exploiting the unique internal properties, such as depth variations and motion trajectories, leads to non-optimal solutions [29]. Therefore, the generalization highly depends on the distribution of training data and will likely deteriorate under distribution shift. To overcome this, SelfDeblur [29] explicitly captures the internal statistics of the given test image in a self-supervised manner. However, this method has some limitations. First, it assumes the blur is spatially uniform, which does not apply in dynamic scenes. Second, it fails to utilize broad external information.

Our proposed approach combines the ideas from two different machine learning paradigms, namely meta-learning (also known as learning to learn) and auxiliary-learning. Meta-learning enables fast adaptation at test time via a few training examples [14, 42, 39, 28, 21, 3, 33, 20]. In particular, model agnostic meta-learning (MAML) [3] has been successfully adopted to other image restoration problems, such as super-resolution (SR) [34, 26]. The methods in [34, 26] first conduct large-scale training on external SR datasets. Then the meta-learning scheme further optimizes the pre-trained model so that it can quickly adapt to unseen images via internal learning. The key point is that the supervision at test time can be simulated by further downscaling the low-resolution images. Thus, internal learning can be achieved to explore the specific patch-recurrence property. However, for deblurring problem, such setting is impractical as patch-recurrence diminishes across scales [19]. One could re-blur the blurry image and treat the original blurry image as the $clean$ counterpart. However, the re-blurring process requires accurate blur kernel estimation, which is challenging. Moreover, it may break the mapping between blurry image and latent clean image. So the application of MAML to the deblurring problem is not as straightforward.

Another practicable approach is to introduce auxiliary-learning by defining an auxiliary task alongside the primary deblurring task [16, 41, 50, 10]. These two tasks can share some parameters. The auxiliary task is often designed in a self-supervised way so that the auxiliary loss can be used to update the model weights at test time. Ideally, the updated shared weights can also improve the performance of the primary task [38]. However, we empirically observe that naively updating the model via the auxiliary task on the pre-trained model can lead to *catastrophic forgetting* [18] where the performance of the primary deblurring task drops.

We propose to integrate meta-learning and auxiliary-learning to exploit their respective strength for the deblurring problem. Inspired by [17], we propose to use self-supervised image reconstruction as the auxiliary task. Its loss can be defined at test time as it does not require any manual labeling. During meta-training, we have access to a labeled dataset consisting of pairs of blurring images and their ground-truth clean counterparts. We consider each image pair as a "task" using the meta-learning terminology [3]. For each task, we update the model parameters using the auxiliary loss defined on the blurry image. The performance of the updated parameters is measured by the deblurring quality via the primary loss. The goal of meta-learning is to learn the model parameters so that the deblurring output using the updated parameters better matches the ground-truth clean image. Note that for different input images, the corresponding updated parameters will be different. In other words, our model is adapted to each input image to better capture its internal information. See Fig. 1 for a qualitative example of our method compared with other alternatives.

The contributions of this paper are manifold. First, we propose to use self-reconstruction as an auxiliary task for the primary deblurring task. The jointly trained model already outperforms existing state-of-the-art. Second, we introduce novel meta-auxiliary learning to enable effective and fast test-time model adaptation. During testing, the model is updated using the self-supervised auxiliary task, which does not require extra labels. Third, our model is learned in a way that facilitates fast adaption with only a few gradient updates during testing. To the best of our knowledge, this is the first attempt to apply meta-auxiliary learning to low-level computer vision problems. Unlike SR [34, 26] using meta-learning, our approach does not require surrogate training pairs during testing. Although we focus on dynamic scene deblurring in this paper, our method can potentially be applied in other image restorations where surrogate training pairs cannot be obtained at test time.

2. Related Works

2.1. Deep learning-based dynamic scene deblurring

In recent years, DNNs have been widely employed for image deblurring. Early works substituted some modules in the conventional optimization-based framework with DNNs [31, 37, 5, 1]. Chakrabarti [1] applied the DNNs to predict the complex Fourier coefficients of the blur kernel. Sun *et al.* [37] explicitly estimated the blur kernel at patch level. Gong *et al.* [5] utilized DNNs to estimate the motion flow from blurry images. The clean images were obtained via non-blind deconvolution.

Nah *et al.* [23] adopted a kernel free method to generate a large-scale dynamic scene deblurring dataset by averaging the consecutive frames in high-speed videos. Furthermore, they proposed a multi-scale architecture to progressively restore the latent sharp image. Since then, various networks were proposed under end-to-end manner and set the state-of-the-art. That includes: deep hierarchical multi-patch network [48], selective sharing scheme [4], incremental tempo-

ral training [25], efficient pixel adaptive and feature attentive design [36]. However, those methods are sub-optimal since the same generic model is applied to every test image and fail to explore the specific internal information. Ren *et al.* [29] developed an unconstrained neural optimization solution to naturally explore the internal information of each input image. However, it did not take advantage of large-scale external datasets which contain rich blurry information. On the other hand, it is inefficient at inference as thousands of iterations are required for each image.

2.2. Auxiliary and meta learning

Auxiliary-learning aims to improve the generalization of the primary task [41, 17]. Lu *et al.* [17] utilized image reconstruction as the auxiliary task to provide the semantic cues for the depth completion. Sun *et al.* [38] proposed a test-time training scheme, where the model is updated by a self-supervised auxiliary task before making a decision.

[34, 26] adopted the meta-learning scheme from MAML [3] for super-resolution. It allows the pre-trained model to be optimized in a way such that it can quickly adapt to any test image. Due to exploiting both external and internal information, superior results are achieved. Our work is also related to [15], where meta-auxiliary learning framework (MAXL) is proposed for image classification, aiming to automatically discover optimal auxiliary labels to improve the primary task. The proposed method also differs from MAXL in another two aspects: our auxiliary task is self-supervised reconstruction, and our main goal is to activate test-time adaptation.

3. Proposed Method

Given a blurry image taken in the dynamic scene as I^b, our goal is to restore its clean counterpart I^c. Most existing approaches directly learn a mapping function (e.g. a DNN) $f : I^b \rightarrow I^c$ from a training dataset consisting of N examples $\{I^b_n, I^c_n\}_{n=1}^N$ where I^b_n is the n-th blurry image in the dataset and I^c_n is the corresponding latent clean image.

In this section, we present our proposed approach that has the following main novelties. First, the self-supervised auxiliary task, particularly, image reconstruction, will be introduced in addition to the primary deblurring task. The auxiliary task can be trained together with the primary task and acts as a regularization. The two tasks share most of the model parameters. Second, for a given test image, we can update the model parameters specifically to adapt to this test image based on the auxiliary branch since the auxiliary task is self-supervised and its loss can be readily computed for a test image. We call this *test-time adaptation* [38]. However, we have found that a naive test-time adaption does not perform well. In this paper, we propose a meta-auxiliary learning scheme so that the model parameters are learned in a way that facilitates effective test-time adaptation.

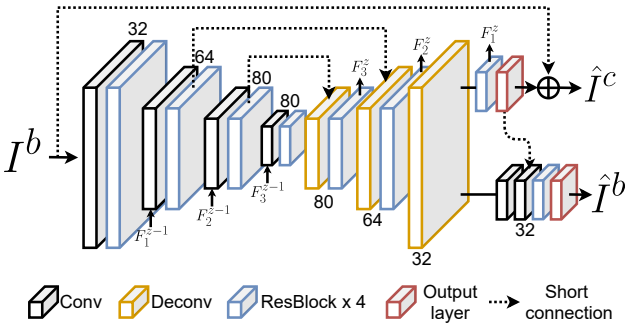


Figure 2: Illustration of the proposed architecture. Given an input blurry image I^b, the primary branch tries to produce the clean image $\hat{I}^c$, while the auxiliary branch aims to produce a reconstructed $\hat{I}^b$ of the input blurry image. These two branches share most of the model parameters.

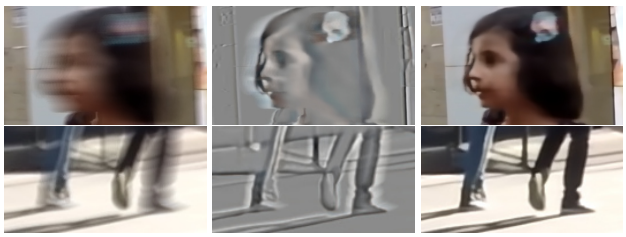
3.1. Model architecture

In the following, we introduce our two-branch architecture. It consists of a primary branch for the deblurring task and an auxiliary branch for the self-reconstruction task.

Primary deblurring network: This network takes a blurry image I^b as its input and produces a clean image counterpart $\hat{I}^c$. This network is based on the multi-scale structure [23]. For each scale, we adopt the similar U-Net architecture [30, 40, 25], which consists of conv, deconv layers and ResBlocks [23] as shown in Fig. 2. Inspired by [25], we enable feature recurrence among different scales. For each scale z, the feature maps $\{F_1^z, F_2^z, F_3^z\}$ after the ResBlocks in the decoder are passed to the encoder of the finer scale. As pointed by the arrows in Fig. 2, the recurrent features are concatenated with the features after the conv layers at corresponding places. As the blurriness diminishes at coarser scales, it is easier to optimize [23]. Therefore, the recurrent features from the decoder at coarse levels are helpful for deblurring at finer levels.

Self-supervised auxiliary network: A properly chosen auxiliary task can complement the primary task in a way such that the extra features learned provide broader interpretation of input data [15]. In our case, specific blurry characteristics are supposed to be captured by the auxiliary task to support the primary deblurring task. In addition, the auxiliary task should be self-supervised so that it can be used for test-time model adaptation.

Our choice of auxiliary task comes from the observation of residual learning. Residual learning has been widely used for effective deblurring [48, 4, 25, 27]. In residual learning, the output of the model is a residual image that aims to remove the low-frequency blur and produce high-frequency components to compensate the blurry areas, as shown in Fig. 3. Therefore, the extracted intermediate fea-



(a) Blurry image (b) Learned residual (c) Restored image

Figure 3: Examples of learned residual images. The learned residual images contain blurry information that is needed for both deblurring and reconstruction tasks.

tures contain blurry information that is also needed for reconstructing the blurry image. Hence, we propose to use the self-supervised reconstruction as the auxiliary task, where it reconstructs the input blurry image as $\hat{I}^b$. We empirically show in the experiment section that adding this self-reconstruction task during the training phase behaves as a regularizer to complement the deblurring task. On the other hand, reconstructing the blurry image at test time allows the model to learn specific blurry information, such as location of motion blur and motion trajectory. More importantly, the loss function of this auxiliary task only requires the blurry image itself, so it can be used for test-time adaptation.

As shown in Fig. 2, we utilize residual learning for the deblur branch and let the reconstruction branch to learn the RGB blurry image. The primary and auxiliary tasks share most of the parameters except the last few layers. The residual image generated at the primary branch is passed to the auxiliary branch to complement the self-reconstruction task. More importantly, the primary branch can also get updated during the test-time adaptation stage based on specific test images. This is different from the architecture in [38] where the primary branch is always freezed.

We then use the multi-scale L1 loss for both tasks as:

$$\mathcal{L}_{Pri} = \sum_{z=1}^{3} \left\| I_z^c - \hat{I}_z^c \right\|_1, \mathcal{L}_{Aux} = \sum_{z=1}^{3} \left\| I_z^b - \hat{I}_z^b \right\|_1. \quad (1)$$

where z denotes the corresponding scale level. The above losses are normalized by the image dimensions.

Joint training: We decompose the parameters of the entire network as $\theta = \{\theta^S, \theta^{Pri}, \theta^{Aux}\}$, where θ^S denotes the shared weights, θ^{Pri} and θ^{Aux} are the task-specific weights for the primary deblurring branch and the auxiliary reconstruction branch, respectively. The predicted clean image $\hat{I}^c$ and the reconstructed input image $\hat{I}^b$ can be obtained by:

$$\hat{I}^c = f_{pri}(I^b; \theta^S, \theta^{Pri}), \hat{I}^b = f_{aux}(I^b; \theta^S, \theta^{Aux}, \theta^{Pri}), \quad (2)$$

where $f_{pri}(\cdot)$ and $f_{aux}(\cdot)$ are the primiary deblurring branch and the auxiliary reconstruction branch, respectively. Note that θ^{Pri} is also needed for the auxiliary task, since the auxiliary task uses the output from the primary task. This is crucial for the test-time adaptation (Sec 3.2).

A straightforward way to train the model is to jointly minimize the combination of primary and auxiliary losses:

$$\mathcal{L}_{Pri}(\hat{I}^c, I^c; \theta^S, \theta^{Pri}) + \mathcal{L}_{Aux}(\hat{I}^b, I^b; \theta^S, \theta^{Aux}, \theta^{Pri}) \quad (3)$$

In our experiments, we call the model learned from Eq. 3 the pre-trained model. We use the pre-trained model as the initialization for the meta-auxiliary learning in Sec. 3.2.

3.2. Meta-auxiliary learning

The model obtained from the joint training (Eq. 3) is sub-optimal since it only exploits the external data and does not take advantage of internal information from test images. We propose a meta-auxliary learning to learn model parameters to facilitate test-time adaptation. For a test image, our model is updated and adapted to this specific test image.

Our method is partially inspired by [38], where test-time training is leveraged via a self-supervised auxiliary loss. The goal is to explore the distribution of each test sample, such that the updated parameter is tailored to that distribution. However, we have found that naively applying test-time training as in [38] leads to *catastrophic forgetting* as shown in Fig. 7. As the performance of two tasks are not connected, the weights updated via the auxiliary loss is more biased to only improve the reconstruction quality, not the primary deblurring quality.

Meta-auxiliary training: To overcome the foregoing issues, we propose to integrate the auxiliary-learning and meta-learning. Concretely, at the meta-training phase, we enforce the constraint that the parameter update via the auxiliary loss should improve the primary deblurring task. Given a pair of training images (I_n^c, I_n^b) and the pre-trained model θ, we first perform adaptation on I_n^b via a small number of gradient updates based on only the auxiliary loss:

$$\tilde{\theta}_n \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{Aux}(\hat{I}_n^b, I_n^b; \theta), \quad (4)$$

where α is the adaptation learning rate. Here $\tilde{\theta}_n = \{\tilde{\theta}_n^S, \tilde{\theta}_n^{Pri}, \tilde{\theta}_n^{Aux}\}$ can be seen as the model parameters adapted to the input I_n^b via internal-learning. Note that the adaptation step in Eq. 4 involves all the parameters.

Ideally, we would like the updated $\{\tilde{\theta}_n^S, \tilde{\theta}_n^{Pri}\}$ to enhance the deblurring task and minimize the primary loss. Accordingly, the meta-objective is defined as:

$$\min_{\theta^S, \theta^{Pri}} \sum_{n=1}^{N} \mathcal{L}_{Pri}(\hat{I}_n^c, I_n^c; \tilde{\theta}_n^S, \tilde{\theta}_n^{Pri}), \quad (5)$$

Note that $\mathcal{L}_{Pri}(\cdot)$ in Eq. 5 is a function of $\tilde{\theta}_n$, but the optimization is over θ. The meta-objective in Eq. 5 can be minimized by performing gradient descent as:

$$\theta \leftarrow \theta - \beta \sum_{n=1}^{N} \nabla_\theta \mathcal{L}_{Pri}(\hat{I}_n^c, I_n^c; \tilde{\theta}_n^S, \tilde{\theta}_n^{Pri}), \quad (6)$$

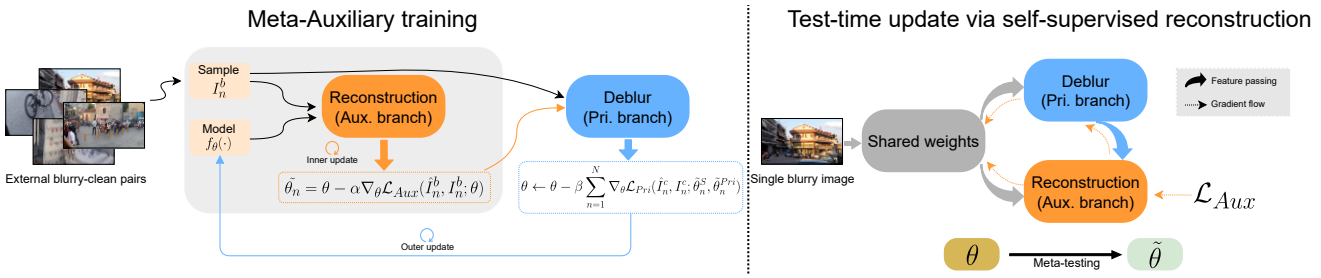


Figure 4: Illustration of the proposed meta-auxiliary learning. At the meta-auxiliary training phase, we first obtain the adapted parameters based on the auxiliary loss in the inner loop. Then, we evaluate the updated weights on the primary task. Finally, the model weights are updated at the outer loop based on the primary loss computed on adapted parameters. At meta-testing, we simply apply the adaptation step to update the model specifically to each test blurry image.

where β represents the meta-learning rate. In practice, we use a mini-batch in Eq. 6 instead of the entire training set. The entire training procedure is elaborated in Algorithm 1 and Fig. 4. Note that, since the primary branch involves θ^S and θ^{Pri}, only those two sets of parameters will be updated at the outer loop. θ^{Aux} will be updated at the inner step.

Meta-auxiliary testing: The meta-learned parameters θ have been learned specifically to facilitate test-time adaptation, so they are less prone to the problem of *catastrophic forgetting*. In the meta-testing phase, given a test blurry image, the adapted parameter $\tilde{\theta}$ is obtained by simply applying Eq. 4. Then $\tilde{\theta}$ is adopted to deblur the given image.

Algorithm 1: Meta-Auxiliary training

Input: α, β: learning rates
Input: (I^b, I^c) pairs
Output: θ: meta-auxiliary learned parameters
Initialize the model with pre-trained weights:
$\theta = \{\theta^S, \theta^{Pri}, \theta^{Aux}\}$;
while *not converged* **do**
  Sample a batch of training pairs $\{I_n^b, I_n^c\}_{n=1}^N$;
  for *each n* **do**
    Evaluate auxiliary reconstruction loss $\mathcal{L}_{Aux}$;
    Compute adapted parameters with gradient descent: $\tilde{\theta_n} = \theta - \alpha\nabla_\theta\mathcal{L}_{Aux}(\hat{I}_n^b, I_n^b; \theta)$;
    Update: $\theta^{Aux} \leftarrow \theta^{Aux} - \alpha\nabla_\theta\mathcal{L}_{Aux}(\hat{I}_n^b, I_n^b; \theta^{Aux})$;
  end
  Validate the primary task and update: $\theta \leftarrow \theta - \beta\sum_{n=1}^N \nabla_\theta\mathcal{L}_{Pri}(\hat{I}_n^c, I_n^c; \tilde{\theta}_n^S, \tilde{\theta}_n^{Pri})$;
end

4. Experiments

4.1. Implementation details

We follow [36, 48, 23] to train our network on the GoPro training dataset [23], which consists of 2103 training pairs. We first perform joint training by optimizing the loss in Eq. 3 with Adam optimizer [11]. The initial learning rate is set to 10^{-4}, and reduced by a factor of 2 when the loss reaches a plateau. The training converges after 4000 epochs with a batch size of 6. During meta-auxiliary training, we fix the learning rates α and β to be 2.5×10^{-5}. For the adaptation step, we perform 5 gradient updates. We randomly crop 256 $\times$ 256 patches for data argumentation, as well as random horizontal/vertical flipping. During meta-testing, the loss is computed based on the whole image. The pixel values of all images are scaled to [-1, 1] and the activation function is set to LeakyReLU [45] with a slope of 0.1. All the experiments are conducted on Nvidia V100 GPUs.

4.2. Evaluation datasets and metrics

We evaluate the performance of the proposed method on widely used dynamic scene deblurring datasets, including GoPro test set [23] (1103 images) and HIDE [32] test set (2025 images). To further demonstrate the test-time adaptation capability of the proposed method under distribution shift, we also conduct comprehensive ablation studies on two video deblurring datasets: Adobe240 [35] (6708 images) and REDS validation set [22] (3000 images). For all datasets, the image resolution is 720 $\times$ 1280. We adopt PSNR and SSIM [43] as the evaluation metrics.

4.3. Comparison with the state-of-the-arts

We compare the proposed method extensively with the state-of-the-art learning-based methods: HumanAware[32], MS-CNN [23], DeblurGAN [12], DeblurGANV2 [13], SRN [40], DMPHN [48], MT-CNN [25], RADN [27], Suin *et al.* [36]. All of the above methods, including ours, are trained on GoPro dataset [23]. Therefore, the comparison is fair and faithful across different datasets. For each of these methods, we either obtain its result directly reported in the original paper or use the released official model with default parameters.

(a) Blurry image (b) Blurry patch (c) DeblurGAN-V2 (d) SRN (e) DMPHN (f) Ours (g) GT

Figure 5: Qualitative comparison with state-of-the-art approaches. The first two rows are from the GoPro dataset [23] and last two rows are from the HIDE dataset [32]. Our method yields sharper results than the state-of-the-art approaches.

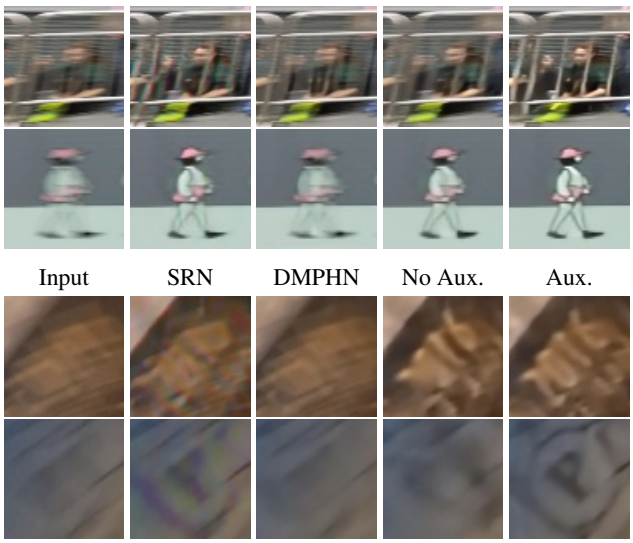


Figure 6: Qualitative examples to show the effectiveness of aux. task and test-time adaptation on real dataset[49].

Methods	GoPro [23]		HIDE [32]	
	PSNR	SSIM	PSNR	SSIM
MS-CNN [23]	29.08	0.914	26.81	0.890
DeblurGan [12]	28.70	0.858	24.51	0.871
DeblurGanV2 [13]	29.55	0.934	26.61	0.875
SRN [40]	30.26	0.934	28.36	0.915
HumanAware[32]	30.26	0.940	28.89	0.930
DMPHN [48]	31.20	0.940	29.09	0.924
MT-CNN [25]	31.15	0.945	-	-
RADN [27]	31.76	0.953	-	-
Suin *et al.* [36]	32.02	0.953	29.98	0.930
Pre-trained	32.30	0.955	30.35	0.932
Ours	**32.50**	**0.958**	**30.55**	**0.935**

Table 1: Comparison of our model with existing state-of-the-art. Our pre-trained model obtained by optimizing Eq. 3 already outperforms existing state-of-the-art. This demonstrates the effectiveness of using image construction as an auxiliary task during training. With meta-auxiliary learning, our approach further improves the results.

Quantitative comparison: Table 1 reports PSNR and SSIM measures on GoPro [23] and HIDE [32]. We also report the results from the pre-trained model optimized by Eq. 3. As shown in Table 1, our pre-trained model, which only employs auxiliary-learning, consistently outperforms the existing approaches. With meta-auxiliary learning scheme, our method enables test-time adaptation to utilize the internal information for every test image. Notably, with test-time adaptation, we observe 0.2dB improvement over the pre-trained model on both datasets. More importantly, it outperforms the best existing approach [36] by 0.48dB and 0.57dB on GoPro and HIDE, respectively.

Qualitative comparison: Fig. 5 shows the visual comparisons on challenging examples. Our method yields cleaner results, especially near the edge boundaries. For the areas that suffer from both camera shake and motion blur, the performance of the existing methods usually degrades. As shown in the 1^{st} and 4^{th} row of Fig. 5, artifacts are visibly observed from the results of existing approaches. Similar performance degradation is also observed on real blurry images as shown in Fig. 6. Due to the distribution shift,

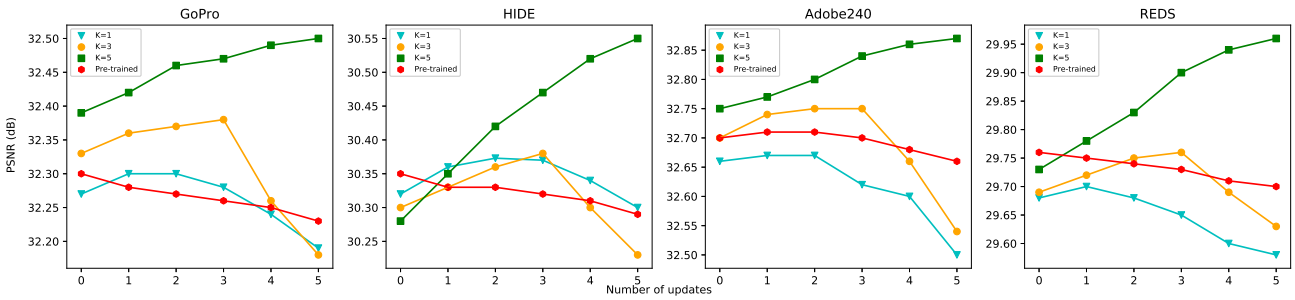


Figure 7: Illustration of PSNR after each gradient update for models with K=$\{1, 3, 5\}$, and pre-trained model. Without meta-auxiliary learning, the pre-trained model consistently performs worse. In contrast, the meta-auxiliary learned models are able to gain performance via test-time adaptation. And peak PSNR is achieved when K is matched during training and testing.

Methods	GoPro [23]		HIDE [32]	
	PSNR	SSIM	PSNR	SSIM
Single-scale	31.04	0.937	28.79	0.904
+ Aux. (share half encoder)	31.07	0.938	28.86	0.906
+ Aux. (share encoder)	31.07	0.938	28.87	0.908
+ Aux. (share until 2^{nd} deconv)	31.10	0.939	28.88	0.909
+ Aux. (share until last deconv)	31.14	0.940	28.92	0.910
Multi-scale	31.79	0.949	29.89	0.925
+ feature recurrence	32.03	0.953	30.22	0.930
+ feature recurrence + Aux.	32.30	0.955	30.35	0.932

Table 2: Ablation studies on network structures. Incorporating the auxiliary-learning via self-reconstruction improves the primary deblurring task. More improvement from the auxiliary task is observed when multi-scale is employed.

Methods	GoPro [23]		HIDE [32]		Adobe240 [35]		REDS [22]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Pre-trained	32.30	0.955	30.35	0.932	32.70	0.944	20.76	0.87
K=1, N=5	32.30	0.957	30.37	0.932	32.67	0.943	29.70	0.85
K=3, N=5	32.38	0.957	30.38	0.934	32.75	0.945	29.76	0.87
K=5, N=1	32.37	0.957	30.41	0.934	32.65	0.944	29.83	0.88
K=5, N=3	32.43	**0.958**	30.50	**0.935**	32.71	0.945	29.92	0.88
K=5, N=5	**32.50**	**0.958**	**30.55**	**0.935**	**32.87**	**0.946**	**29.96**	**0.89**

Table 3: Evaluation of number of updates K and batch size N on different datasets. Larger K and N yield better deblurring results. Larger K allows the model to better exploit and adapt the internal information. While larger N prevents from overfitting to any specific training example.

the compared methods suffer from poor generalization with incomplete blurry removal and artifacts. In contrast, our method produces visually pleasing results. We believe this is because our model has learned to effectively adapt to the internal information of each test image.

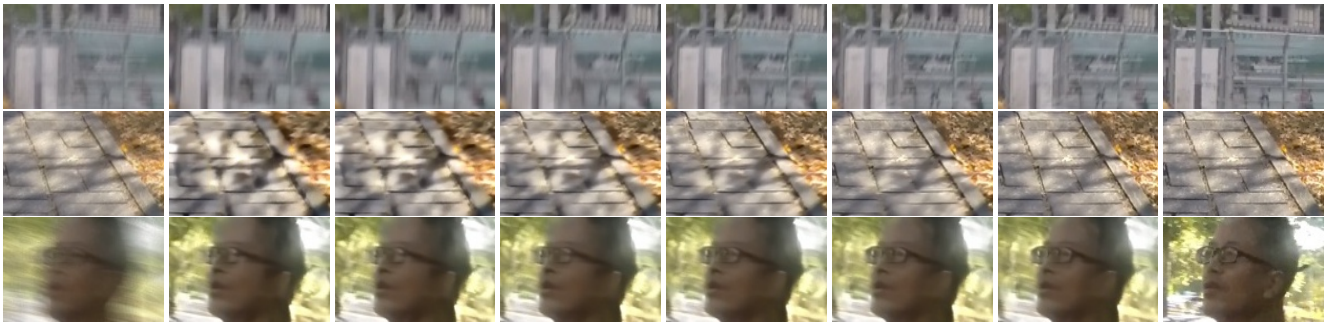
4.4. Ablation studies

We perform additional ablation experiments to further study various aspects of the proposed approach.

Network structures: In Table 2, we evaluate the impact of each component of the proposed network structure. Since the auxiliary task aims to complement the primary task, we first investigate the deblurring performance by changing the number of shared layers. We train 4 models where both tasks share the weights of 1) half of the encoder; 2) the entire encoder; 3) until the second or 4) until last deconv layer. As for the unshared parts, both tasks have the same structure, except the very last few layers, as shown in Fig. 2. Table 2 (top 5 rows) reveals that the auxiliary reconstruction better improves deblurring when more weights are shared between two tasks. The qualitative evaluation in Fig. 6 (first two rows) also shows the effectiveness of auxiliary task.

We also investigate the effect of using multi-scale. As reported in Table 2 (bottom 3 rows), the multi-scale structure boosts the deblurring performance by 0.75dB and 1.1dB on two datasets, respectively. As feature recurrence allows the features that may contain blur patterns to be shared among scales, it further improves the performance. Finally, adding the auxiliary task gives additional performance boost. More importantly, the deblurring task takes more advantage when it is integrated with the auxiliary task at multi-scale. The improvement made by auxiliary-learning is observed to be 0.27dB for multi-scale, but only 0.1dB for single-scale.

Number of gradient updates and batch size: In this study, we show the impacts of two factors in Algorithm 1: number of gradient updates in the inner loop and batch size. Table 3 shows the results of models that are trained with various number of gradient update $K = \{1, 3, 5\}$ and batch size $N = \{1, 3, 5\}$. Note the number of gradient updates is consistent during training and testing. Overall, we observe that larger K and N tend to produce better results. Our hypothesis is that larger K allows the model to better adapt to the internal structure of a test image, while larger N prevents



(a) Input (b) No update (c) 1 update (d) 2 updates (e) 3 updates (f) 4 updates (g) 5 updates (h) GT

Figure 8: Visual illustration of the unfolded adaptation process for model with K=5 on the GoPro dataset [23] (row 1-2) and the HIDE dataset [32] (row 3-5). With the test-time adaptation, the artifacts and incomplete blur removal suffered from distribution shift are resolved. Cleaner and sharper images are generated and are visually closer to the ground truth images.

the model from overfitting to any specific training examples. The results reported in Table 3 across 4 datasets are consistent with our assumption. However, we found that, further increasing K or N does not bring additional benefits.

4.5. Unfolding the adaptation process

To gain further insights of our model, we unfold the adaptation process for models with $K = \{1, 3, 5\}$ and the pre-trained model. The results after each gradient update on all 4 datasets are illustrated in Fig.7. We can make several interesting observations. First, naively adapting the pre-trained model degrades the performance instead of improving it. This is probably because the pre-trained model is not learned in a way that facilitates test-time adaptation. In contrast, the meta-auxiliary learned models are able to use test-time adaptation to improve the deblurring. Another interesting observation is that the number of gradient updates during test-time adaptation should match that during training. This is intuitively reasonable. If we use a particular K value (e.g. K=3) during training, the model has been trained to perform the most effective adaptation with 3 gradient updates. If we use a different K value (e.g. 1 or 5) at test time, the adaption process does not match what has been learned. So the performance might drop due to this distribution shift.

Fig. 8 visually shows the unfolded adaptation process. Initially, the network may suffer from artifacts or incomplete deblurring due to distribution shift. However, as more specific blur patterns are learnt during test-time adaptation, the model is more tailored to deblur every image. Therefore, the outputs are getting cleaner and sharper.

4.6. Dynamic adaptation for video deblurring

The results on different datasets reported in Table 3 and Fig. 7 consistently demonstrate the strong generalization of the proposed meta-auxiliary learning. However, for video deblurring, running the adaptation on every frame is tedious and inefficient. Thus, we propose a dynamic adaptation mechanism to improve efficiency. We apply the model adaptation on every j frames, and allow the next $j-1$ frames to use the same adapted weights. This is based on the observation that the consecutive video frames are highly correlated and are more likely drawn from the same distribution. They may also share similar low-level statistics and blur patterns. Reported in Table 4 are the results with various j values. As we can see, increasing j from 1 to 10 only drops 0.04dB for both datasets, but the inference time can be reduced dramatically. With $j = 10$, the model is still improved compared to the one without test-time adaptation.

Dataset	No update	$j = 1$	$j = 3$	$j = 5$	$j = 10$
Adobe240 [35]	32.75	32.87	32.84	32.84	32.83
REDS [22]	29.73	29.96	29.94	29.93	29.92

Table 4: Evaluation of dynamic adaptation for video deblurring with various j. Due to the high correlation between consecutive frames in videos, performing test-time adaption on every j frame can achieve better speed-quality tradeoff.

5. Conclusion

In this work, we have introduced novel meta-auxiliary learning for dynamic scene deblurring. We first built a self-reconstruction auxiliary task to share certain layers with the primary deblurring task. Integrated with meta-auxiliary learning, the model is constrained so that the update via the auxiliary task brings performance gain for the deblurring task. Thus, the model is endowed with the ability for fast adaptation. At test-time, adaptation is performed on test images to achieve superior deblurring. Extensive experiments show that the proposed method outperforms state-of-the-art methods by utilizing external and internal information.

References

[1] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European Confererence on Computer Vison*, 2016. 1, 2

[2] Zhixiang Chi, Xiao Shu, and Xiaolin Wu. Joint demosaicking and blind deblurring using deep convolutional neural network. In *IEEE International Conference on Image Processing*, 2019. 1

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. 2, 3

[4] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3

[5] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton Van Den Hengel, and Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[6] Ankit Gupta, Neel Joshi, C Lawrence Zitnick, Michael Cohen, and Brian Curless. Single image deblurring using motion density functions. In *European Conference on Computer Vision*, 2010. 1

[7] Stefan Harmeling, Hirsch Michael, and Bernhard Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *Advances in Neural Information Processing Systems*, 2010. 1

[8] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee. Dynamic scene deblurring. In *IEEE International Conference on Computer Vision*, 2013. 1

[9] Tae Hyun Kim and Kyoung Mu Lee. Segmentation-free dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1

[10] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2016. 2

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[12] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5, 6

[13] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *IEEE International Conference on Computer Vision*, 2019. 1, 5, 6

[14] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2

[15] Shikun Liu, Andrew Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. In *Advances in Neural Information Processing Systems*, 2019. 3

[16] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[17] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. From depth what can you see? depth completion via auxiliary image reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3

[18] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, 1989. 2

[19] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, 2014. 2

[20] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2017. 2

[21] Tsendsuren Munkhdalai and Hong Yu. Meta networks. *Proceedings of machine learning research*, 70:2554, 2017. 2

[22] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 5, 7, 8

[23] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 3, 5, 6, 7, 8

[24] Thekke Madam Nimisha, Akash Kumar Singh, and Ambasamudram N Rajagopalan. Blur-invariant deep learning for blind-deblurring. In *IEEE International Conference on Computer Vision*, 2017. 1

[25] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European Confererence on Computer Vison*, 2020. 1, 3, 5, 6

[26] Seobin Park, Jinsu Yoo, Donghyeon Cho, Jiwon Kim, and Tae Hyun Kim. Fast adaptation to super-resolution networks via meta-learning. In *European Confererence on Computer Vison*, 2020. 2, 3

[27] Kuldeep Purohit and AN Rajagopalan. Region-adaptive dense network for efficient motion deblurring. In *AAAI Conference on Artificial Intelligence*, 2020. 1, 3, 5, 6

[28] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. 2

[29] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015. 3

[31] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2015. 1, 2

[32] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *IEEE International Conference on Computer Vision*, 2019. 5, 6, 7, 8

[33] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017. 2

[34] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3

[35] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5, 7, 8

[36] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3615, 2020. 1, 3, 5, 6

[37] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2

[38] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, 2020. 2, 3, 4

[39] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[40] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 3, 5, 6

[41] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *2018 IEEE International Conference on Robotics and Automation*, 2018. 2, 3

[42] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016. 2

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[44] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision*, 98(2):168–186, 2012. 1

[45] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 5

[46] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1

[47] Yuan Yuan, Wei Su, and Dandan Ma. Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[48] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3, 5, 6

[49] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6

[50] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2