Neural Maximum a Posteriori Estimation on Unpaired Data for Motion Deblurring

Youjian Zhang , Chaoyue Wang , and Dacheng Tao , Fellow, IEEE

Abstract—Real-world dynamic scene deblurring has long been a challenging task since paired blurry-sharp training data is unavailable. Conventional Maximum A Posteriori estimation and deep learning-based deblurring methods are restricted by handcrafted priors and synthetic blurry-sharp training pairs respectively, thereby failing to generalize to real dynamic blurriness. To this end, we propose a Neural Maximum A Posteriori (NeurMAP) estimation framework for training neural networks to recover blind motion information and sharp content from unpaired data. The proposed NeurMAP consists of a motion estimation network and a deblurring network which are trained jointly to model the (re)blurring process (i.e. likelihood function). Meanwhile, the motion estimation network is trained to explore the motion information in images by applying implicit dynamic motion prior, and in return enforces the deblurring network training (i.e. providing sharp image prior). The proposed NeurMAP is an orthogonal approach to existing deblurring neural networks, and is the first framework that enables training image deblurring networks on unpaired datasets. Experiments demonstrate our superiority on both quantitative metrics and visual quality over State-of-the-art methods.

Index Terms—Image deblurring, maximum a posteriori, unpaired data.

I. INTRODUCTION

DYNAMIC scene deblurring, which aims to restore the sharp content from a real-world blurry image, is a classic low-level vision task and would benefit serials of perceiving tasks, e.g. image classification and object detection. However, due to the blind and non-uniform nature of dynamic blur, it has long been a challenging research topic.

Conventional deblurring methods usually treat deblurring as a maximum a posteriori (MAP) estimation problem [1, 2, 3, 4, 5, 6]. Given a blurry image, the estimated blur kernels and the deblurred result are optimized alternately to approach the blurring process (Fig. 1(a)). To reduce the intrinsic ill-posedness, different image priors and blur kernel priors [6, 7, 8, 9, 10] are proposed. However, experiments show that the complex blurring process of real dynamic scenes cannot be accurately modeled by handcrafted priors. Moreover, performing MAP optimization on each blurry image is time-consuming and computationally inefficient, which further impedes these methods being widely used in real scenarios.

Capturing real dynamic scenes with a high-frame-rate video camera, pseudo blurry images can be synthesized by averaging multiple adjacent frames [11]. Meanwhile, the middle frame of the sequence is regraded as the sharp ground truth. As shown in Fig. 1(b), with paired training data, deep neural networks [11, 12, 13, 14, 15, 16, 17] are trained to directly restore sharp content from blurry inputs. Taking advantages of the powerful learning capability, deep learning-based methods can achieve impressive dynamic scene deblurring results on test set and largely reduce the inference time. However, since domain gaps exist [18], these well-trained networks may easily overfit synthetic blurry data and generalize poorly to unseen blurriness, especially real-world dynamic blurriness.1 As shown in Fig. 2, several most representative deep deblurring methods fail to recognize and handle real-world blurry images.

In this paper, we aim to explore useful information contained in unpaired blurry and sharp images, then utilize the learned knowledge to improve deblurring performance. Inspired by conventional MAP estimation, a Neural Maximum A Posteriori (NeurMAP) estimation framework is proposed to explore the accurate deblurring posterior \( p(S, K|B) \) from unpaired data, where \( B, S, K \) represent the blurry image, the sharp image and blur kernels. As shown in Fig. 1(c), we employ a deblurring network and a motion estimation network to recover the sharp content and blur kernels from blurry images. First, the deblurred result and the estimated blur kernels can be used to simulate the (re)blurring process, i.e. modeling the likelihood probability \( p(B|S, K) \). Meanwhile, the motion estimation network is trained to predict dynamic motion information (i.e. pixel-wise blur kernels) of an image. For example, the motion map estimated from a sharp image is penalized to be a zero field, and we enforce the motion estimation network to continually explore unremoved blur/motions contained in deblurred images. In contrast, the deblurring network is trained to minimize the remaining blurriness explored by the motion estimation network. We show such an adversarial training process actually enforces the sharp image prior \( p(S) \) and kernel prior \( p(K) \) during training. These training objectives work

---

1Different from synthetic training pairs, current filming equipment cannot acquire paired real blurry images and sharp ground truth.
Fig. 1. The schematic pipelines of (a) conventional MAP optimization; (b) previous deep deblurring methods; (c) the proposed NeurMAP framework. Unlike (a), neural networks presented in (b) and (c) are trained using a dataset.

Fig. 2. Comparisons on real-world image deblurring. The test image is sampled from both RealBlur dataset [19] and our collected Unpaired Real-world Blur (URB) dataset. Existing deep learning-based methods can only be trained on synthetic blurry-sharp pairs, thus failing to remove unseen real-world blur patterns. Our proposed NeurMAP successfully explored deblurring posterior probability from unpaired data and largely improved the deblurring results.

together to maximize the posteriori probability over unpaired training data, i.e. \( p(S, K | B) \propto p(B | K, S) p(S) p(K) \).

There are also some works that propose a neural network to construct some parts of the MAP estimation framework, however, none of them have achieved a full MAP estimation with both blur kernel \((K)\) and sharp image \((S)\) unknown, i.e. estimating the \(K\) and \(S\) at the same time. For example, [20], [21] both utilize a pretrained optical flow network to pre-calculate the motion between consecutive frames as an approximation of the blur kernel. In other words, \(p(K)\) is known and fixed during the inference process. Also, since these methods need to approximate the blur kernel from neighboring frames, they cannot be adopted to image deblurring. In the defocus deblurring/super-resolution work [22], [23], the reblurring loss takes a down-sampled sharp image as the input, where the sharp/high-resolution images \(S\) are known and fixed.

Overall, we make the following contributions:

- A semi-supervised NeurMAP framework is proposed for training the deblurring model using unpaired data. Although synthetic blurry-sharp image pairs are provided for training stabilization, to our best knowledge, our method is the first framework that enables training a MAP estimation on unpaired data.

- We build a direct connection between MAP estimation and training deblurring neural networks. Different from previous handcrafted priors and synthetic training pairs, the proposed NeurMAP explores deblurring posterior from unpaired data, which overcomes the overfitting problem on synthetic data.

- Comprehensive experiments show that the proposed NeurMAP succeeds in modeling unseen blurring processes (both kernel-synthesized data and real-world data), and achieved significant improvement on real-world dynamic scene deblurring. In addition, even with only unpaired data,
our NeurMAP is capable of recovering dynamic motions from a blurry image.

- We discussed the effects of different generative adversarial networks (GANs) that are employed in deblurring models, which further explain principles of adversarial training in the proposed NeurMAP framework.

II. BACKGROUND

A. Conventional MAP-Based Deblurring Methods

Conventional MAP usually performs an optimization process on both sharp content and blur kernel iteratively. Since the optimization process is relatively fixed, the conventional MAP estimation methods focused on designing different priors to get a better deblurring performance. For example, Gaussian scale mixture priors \(1, l_1/l_2\)-norms [24], \(l_0\)-norms [8], total variation priors [9] and dark channel priors [6] have been proposed during the past decade. Also, some specific camera motion constraints were introduced as regularization terms to shrink the potential solution space of blur kernels, such as ego motion [25], 3D camera rotation [7], [26], and camera forward [27]. Besides handcrafted priors, recently, [10], [28], [29], [30] proposed to utilizing a neural network to explore kernel/image priors for MAP optimization, yet they still require an alternative optimization process. Also, none of them can explicitly represent the spatial-variant blur kernel in dynamic scenes.

Overall, MAP estimation build a theoretical foundation for image deblurring tasks, and achieved reasonable deblurring performance. However, most methods still rely on predefined priors or strong assumptions on the latent image or the causes of the motion blur. Thus, these methods usually fail in handling more complicated spatial-variant blur in real scenarios [11]. Moreover, existing MAP-based methods have to perform the optimization on each blurry image, which could be time-consuming and inefficient.

B. Learning-Based Deblurring Methods

More recently, benefited from the release of synthetic blurry/sharp datasets [11], [31], end-to-end deblurring networks [11], [12], [32], [33], [34], [35], [36] are trained directly to restore the latent sharp content from a blurry input. Among these methods, [11], [12], [13] adopted the multi-scale restoration strategy, they explored the different parameter sharing schemes and skip-connections. Moreover, a multi-input multi-output U-Net (MIMO-U-Net) [37] is proposed to compress the multi-scale strategy in a single U-Net structure. Despite multi-scale, multi-temporal [17]/multi-stage [35] strategy has also been studied to boost the deblurring performance. [38] attempted to learn a variant RNN to model spatial-variant blurs. [15], [16], [39], [40] also designed a spatial-variant convolutional module for the non-uniform feature of blurry region. With the transformer-based methods starting to show its power in high-level vision tasks, several transformer-based methods [41], [42], [43] have also been proposed to explore the benefit of non-local feature aggregate in image restoration tasks. In addition to the methods which aim to find a strong and effective network architecture, [32], [44], [45] combined deblurring with generative adversarial networks (GANs). For example, Kupyn et al. proposed DeblurGANs [32], [44] that valid the effectiveness of different backbones and GANs.

However, all these learning-based methods rely heavily on the paired training dataset, and usually fail to recognize unseen blur patterns such as the real-world blur. There are methods [18], [19] that want to fix the generalization problem by obtaining more realistic pseudo training pairs. For example, Zhang et al. [18] proposed a learning-to-Blur GAN (BGAN) for generating blurry images with more realistic blur. Rim et al. [19] collected an unaligned real-world dataset in low-light environment with a well-designed dual-camera system. However, the acquisition process of these training pairs is complicated and time-consuming. Compared with [18], [19], the proposed NeurMAP can be directly trained on easily acquired unpaired data, and demonstrated better performance on unseen blur.

Another line of works adopt test-time adaptation [46]/meta-learning-based methods [47], [48] for a better generalization performance on unseen data distributions. They mainly focus on performing a domain shift for a single test image (test-time adaptation [46]) or a small support set (meta learning [47], [48]), while our method aims to fit a larger data domain concerning the unpaired dataset. However, current test-time adaptation/meta learning methods cannot be directly adopted in training unpaired dataset. Specifically, [46] conducts the test-time adaption as a refinement for the deblurred results; [47] needs to introduce ground truth during the meta-auxiliary training process; [48] requires a video set as input for synthesizing pseudo blurry/sharp image pairs. Nevertheless, they may serve as orthogonal approaches to help further improve the performance of our proposed NeurMAP.

III. METHOD

A. Preliminaries: Image Blurring & Deblurring

Mathematically, an image blurring process can be formulated as [1]

\[
B = K * S + \text{noise},
\]

where \(B\) and \(S\) represent blurry and sharp images, \(K\) is the blur kernel that records relative motions during the exposure period. As an inverse process, image deblurring aims to recover sharp content \(S\) given a blurry image \(B\).

**Maximum a posteriori probability (MAP) deblurring:** In Bayesian statistics [49], the deblurring task can be treated as a MAP estimation problem [1]. Given the blurry image \(B\), MAP estimation aims to retrieve both the latent sharp image \(S\) and blur kernels \(K\) to maximize the posterior distribution

\[
p(K,S|B) \propto p(B|K,S)p(S)p(K),
\]

where likelihood \(p(B|K,S)\) measures the (re)blurring fidelity; \(p(S)\) and \(p(K)\) define the priors of sharp images and blur kernels, which is designed to reduce the ill-posedness and enforce the smoothness of deblurred images and blur kernels. As shown in Fig. 1(a), in conventional MAP-based deblurring methods, given a blurry image \(B\), the optimization process is performed.
to iteratively optimize its sharp content \( S \) and blur kernels \( K \). Specifically, the objective of optimizing \( K \) is to maximize \( p(B|K, S)p(K) \), while the objective of optimizing \( S \) is to maximize \( p(B|K, S)p(S) \). The whole optimization process needs only a blurry image, yet not its corresponding sharp ground truth.

Deep neural networks trained using paired data: With the paired blurry/sharp dataset, many deep neural networks are trained directly to maximize the probability \( p(S|B) \) over a dataset, as shown in Fig. 1(b). Specifically, the deblurred results \( \hat{S} \) is enforced to be identical to ground truth sharp images \( \{S_i\} \).

The well-trained neural network largely improved the deblurring efficiency, yet usually fails to generalize to unseen blur patterns. Apparently, this pipeline cannot handle unpaired data.

Training neural networks to maximize posteriori probability (NeurMAP): To tackle the problem of the absence of sharp ground truth, our proposed NeurMAP aims to simultaneously obtain the sharp image \( S \) and blur kernels \( K \) in an end-to-end manner. Inspired by conventional MAP, the posteriori possibility is achieved by several training objectives. Differently, NeurMAP substitutes all the calculation and estimation with differentiable neural networks (Fig. 1(c)). Also, we design multiple loss functions as learnable image priors/kernel priors, instead of pre-defined priors in conventional methods. Besides paired data, the proposed NeurMAP is capable of learning deblurring posterior from unpaired data. More details are illustrated in the following Sections III-B and III-C.

B. The Proposed NeurMAP Framework

Overall, our framework consists three trainable neural networks: a deblurring network \( D \), a motion estimation network \( M \) and an natural image discriminator \( N \); and a non-trainable module: the reblurring module \( R \). In this section, we first introduce the functionality of these neural networks (module), our MAP-inspired training scheme is illustrated in Section III-C.

Deblurring network \( D \): Similar to existing deep deblurring methods, the proposed deblurring network \( D \) takes blurry images \( B \) as inputs, and is trained to output the deblurred result \( \hat{S} = D(B) \). Our proposed NeurMAP is an orthogonal framework to existing deep neural networks. Therefore, the network \( D \) can adopt different architectures. In this paper, we validate the proposed NeurMAP on two representative deblurring backbone networks (Section IV).

Motion estimation network \( M \): Given an input image such as \( B, \hat{S} \) or \( S \), the motion estimation network \( M \) is trained to recover the motion information contained in the image, i.e. how the image is blurred. Since the dynamic blur is usually spatial-variant, \( M \) predicts a motion map with the same resolution as its input. Similar to [50], the blur kernel under the linear assumption can be described as a 2-dimensional motion vector. In our NeurMAP, the motion map can be written as \( M(p) = (u_p, v_p) \), where \( u_p \) and \( v_p \) represent the horizontal and vertical components of the motion at the coordinate \( p \). Through training \( M \), we aim to get an accurate motion representation from the blurry image. There are two benefits of learning an accurate motion representation in our method: i) such motion representation can be applied as a strong image prior for sharp image (sharp image tends to have small blur kernel) to enforce the deblurring process; ii) the relative difference between a blurry image \( B \) and its deblurred result \( \hat{S} \) can be utilized to reblur \( \hat{S} \), i.e. \( M_{rel} \) in Fig. 3, thereby optimizing the fidelity term \( p(B|K, S) \).

Non-trainable reblurring module \( R \) : Differentiable (re)blurring is studied in serveral works [18], [20], [21], [51], [52]. Here, we follow the blur creation module proposed in [52], since it is computationally efficient and easy to implement. Specifically, the relative motion vector \( M(p) \) is discretized to \( N \) steps and then utilized for warping the deblurred/sharp image. Taking the reblurring process as an example

\[
\hat{B} = R(\hat{S}, M_{rel}) = \frac{1}{N} \sum_{n=0}^{N-1} \mathcal{W}(\hat{S}, \frac{n}{N-1}(M_{rel} - \frac{M_{rel}}{2}))
\]

(3)

\( \mathcal{W}(\cdot, \cdot) \) is a warping function, which can help us obtain \( N \) discretized frames during the blurring process. Given a relatively large \( N \), the average of these discretized frames approaches the real blurring process, i.e. \( \hat{B} = K * S \).

Compared to trainable reblurring model BGAN\_G proposed in [18], our fixed reblurring module, in conjunction with the pixel-wise motion map, is designed to generate spatially-varying motion-blurred images, where the spatial variability of the motion map results in corresponding variations in the degree of blurriness. The 4-dimensional motion vector used in BGAN\_G is not suitable to meet the requirement for spatial non-uniformity. Second, in the optimization for MAP estimation \( p(K, S|B) \propto p(B|K, S)p(K)p(S) \), we already includes two trainable modules for estimating \( p(K) \) and \( p(S) \). Making the reblurring module \( p(B|K, S) \) also trainable would significantly increase the difficulties in training the entire framework. We have replaced the reblurring module using the trainable network, yet the training failed to converge.

Natural image discriminator \( N \): Considering the ill-posedness of the restored sharp image and blur kernel, the sharp image prior imposed by motion estimation network is not enough to regularize the restored image to be visually realistic. In fact, our experiments show this training using unpaired data may easily lead to unnatural artifacts. So we employ an natural image discriminator to regularize the deblurred results so that they are in the natural image domain. Training the natural image discriminator is equivalent to learn and optimize an image prior term. Similar to DeblurGAN (v2) [32], [44], the proposed \( N \) adopts the PatchGAN structure [53] and aims to enforce deblurred images to have the same prior distribution with real-world images. Differently, the natural image discriminator only distinguishes the natural images (both blurry and sharp images) from generated ones (deblurred results are regraded as fake images).

C. NeurMAP Training Scheme

In this section, we introduce (i) the losses for training aforementioned networks, and (ii) how the training enforces maximizing a posteriori probability. Similar to conventional MAP-based deblurring methods, the proposed training scheme of NeurMAP...
aims to maximize \( p(K, S|B) \) (2). Differently, as shown in Fig. 3, our NeurMAP is trained on the whole training dataset (including both paired and unpaired data).

**Data term.** \( p(B|S, K) \): Giving the estimated blur kernels and deblurred results, the data term models the deblurring process, i.e. maximizes likelihood probability \( p(B|S, K) \). In the proposed NeurMAP, we calculate the reconstruction losses between a reblurred image \( \hat{B} \) with its blurry input \( B \)

\[
\mathcal{L}_{\text{reblur}} = \| B - \hat{B} \|^2 = \| B - \mathcal{R}(\hat{S}, M_{\text{rel}}) \|^2, \tag{4}
\]

where \( M_{\text{rel}} \) denotes the relative motion map. Since the reblurring module \( \mathcal{R} \) is fully-differentiable, the data term \( \mathcal{L}_{\text{reblur}} \) is used to optimize both deblurring network \( \mathcal{D} \) and motion estimation network \( \mathcal{M} \), which is essential for maintaining content consistency.

**Kernel prior term.** \( p(K) \): Since we do not have the ground truth for neither deblurred outputs nor estimated motion maps, the model only trained with the data term will soon converge to a trivial solution, e.g. the deblurred \( \hat{S} \) is identical with \( B \) and the estimated motion is always zero. Enforcing kernel priors contributes to getting rid of such cases. Specifically, we concluded following implicit priors: (i) The motions \( M_B \) estimated from a blurry image should be relatively large. We enforce \( M_B(p) \to (\alpha, \alpha) \), where the \( \alpha \) serves as an empirical upper bound for filtering outlier estimates of blurry motions. Based on our empirical evaluations \([1, 50, 54]\), for typical image sizes, it is unlikely for the blurry motions to exceed 40. (ii) The motion estimated from a sharp image should be a zero field, i.e. we penalize the motion vector of each pixel estimated from a sharp image to be as close to zero as possible (\( M_B(p) \to (0, 0) \)); (iii) During training, outputs of the deblurred network usually contains unremoved blurriness, and we hope the network \( \mathcal{M} \) can explore those unremoved blurriness. It is reasonable to assume the unremoved motion is scaling down from the motion of blurry input, i.e. \( M_S(p) \to M_B(p) \). Overall, these objectives can be written as

\[
\mathcal{L}_{\text{kernel}} = \mathcal{L}_{M_B} + \mathcal{L}_{M_S} + \mathcal{L}_{M_{\hat{S}}} = \min_M ||\mathcal{M}(B) - (\alpha, \alpha)|| + ||\mathcal{M}(S) - (0, 0)|| + ||\mathcal{M}(\mathcal{D}^*(B)) - \text{detach}(\mathcal{M}(B))||. \tag{5}
\]

Here, \( \mathcal{D}^* \) means the module \( \mathcal{D} \) is fixed during back-propagation. The intention of these losses is to push the motion of the blurry image and the sharp image in two opposite directions. Meanwhile, to estimate varying degrees of blurriness in the images, we incorporate a reblurring loss in the training process. This loss helps to guide the estimated motions towards more reasonable values. Thus, both \( \alpha \) and the reblurring loss work together to determine the accurate extent of motion in the blurry images.

In addition, since the blur kernels of dynamic blur are usually smooth along with the space, we apply the total variation (TV) regularization to encourage spatial smoothness of the estimated relative motion map

\[
\mathcal{L}_{\text{tv}}(M_{\text{rel}}) = \frac{1}{(w-1)h} \sum_{i=0}^{w-1} |M_{\text{rel}}(i, j) - M_{\text{rel}}(i + 1, j)| + \frac{1}{w(h-1)} \sum_{j=0}^{h-1} |M_{\text{rel}}(i, j) - M_{\text{rel}}(i, j + 1)|, \tag{6}
\]

where \((i, j)\) denotes the pixel location in a map with resolution of \( w \times h \). \( M \) represents different motion maps estimated from images.

**Image prior term.** \( p(\hat{S}) \): Since image deblurring is highly ill-posed, the regularization/priors on \( \hat{S} \) are often the key to the success of deblurring. In our NeurMAP, two image priors, i.e. sharp image prior and natural image prior, are learned and enforced by the motion estimation network and natural image discriminator, respectively.

Fig. 3. Our proposed framework consists of 3 trainable modules (\( \mathcal{D}, \mathcal{M}, \mathcal{N} \)) and one non-trainable reblurring module (\( \mathcal{R} \)). Specifically, \( \mathcal{D} \) is trained with reblurring term and image prior term to generate a deblurred image. Meanwhile, \( \mathcal{M} \) is trained to estimate the accurate motion maps of the images, and \( \mathcal{N} \) is trained to distinguish whether a image is visually natural. In this figure, networks represented using dash line is fixed during training, networks with solid line is trained using corresponding losses, \( \mathcal{L}(\cdot) \).
Recall that the motion estimation network $M$ is trained to estimate an accurate motion map from the input image. When training the deblurring network $D$, we penalize the motion map estimated from the deblurred results to be as sharp as possible (sharp image prior), i.e. $M_S(p) \to E[M_S]$ 

$$
\mathcal{L}_D^D = \min_D ||M^*(D(B)) - E[M^*(S)]||.
$$

(7)

We use the expectation $E[M^*(S)]$ since we do not have the sharp ground truth of the input blurry image, the expectation value is regarded as the target motion value of all sharp images. Note that (7) works together with (5) which training $M$ to explore unremoved blur as much as possible, and these two losses form an adversarial training scheme.

The sharp image prior focuses on minimizing point-wise dense motions of the deblurred images. However, it may be cheated by artifacts in texture-level. This is why we further introduce an natural image discriminator $N$, which aims to distinguish whether a deblurred result is a natural image. Fig. 4 gives a heuristic example. The utilization of only $\mathcal{L}_D^D$ as the sharp image prior (as shown in Fig. 4(a)) may result in deblurred images with low training losses but poor perceptual quality, such as the presence of artifacts. On the other hand, incorporating the natural image GAN loss (as depicted in Fig. 4(b)) serves to push generated images with artifacts towards higher energy, since the GAN training measures the discrepancy between the training and generated distributions. Specifically, we adopt patchGAN [53] as our backbone and LSGAN’s training losses [55]. We define natural images being with the numerical label ‘1’, fake images being with the numerical label ‘0’, then the adversarial losses is defined as

$$
\begin{align*}
\mathcal{L}_N^N &= \min_N \mathbb{E}_{I \in \{B\}} ||N(D^*(I)) - 0||^2 + \mathbb{E}_{I \in \{B,S\}} ||N(I) - 1||^2, \\
\mathcal{L}_D^D &= \min_D \mathbb{E}_{I \in \{B\}} ||N^*(D(I)) - 1||^2.
\end{align*}
$$

(8)

It is worth noting that the natural image manifold includes sharp images and blurry images, $\{B,S\}$, since we observe that the deblurring process outputs blurry images in the beginning. As shown in our experiments, only using sharp images may lead to unstable training.

Other than improving the naturalness, we apply a total variation (TV) loss to encourage the smoothness of the deblurred image. Since the sparse/uneven optimization on the motion map will cause the unsmoothness of deblurred image, we choose to smooth the motion map of the deblurred image $S$

$$
\begin{align*}
\mathcal{L}_{tv}(M_S) &= \frac{1}{(w-1)h} \sum_{i=0}^{w-1} |M_S(i,j) - M_S(i + 1,j)| \\
&+ \frac{1}{w(h-1)} \sum_{j=0}^{h-1} |M_S(i,j) - M_S(i,j + 1)|,
\end{align*}
$$

(9)

where $(i,j)$ denotes the pixel location in a map with resolution of $w \times h$. $M_S$ represents the motion map estimated from the deblurred image.

All these aforementioned image priors/regularizations is to encourage the deblurred results to be sharp and artifact-less.

NeurMAP training scheme: Till now, we introduced how to optimize each factor/term of the proposed NeurMAP framework. Before we conclude the overall training scheme, we hope to point that, in practical, our NeurMAP is trained in a semi-supervised manner. Similar to previous deep deblurring methods, the synthetic paired blurry/sharp images are employed to enforce the content loss, $\mathcal{L}_{content} = ||S_{pair} - D(B_{pair})||^2$, which enables our deblurring network $D$ to have a baseline deblurring capability (More discussion about the function of pair loss and paired data can be found in Section IV-E). Furthermore, we utilize the easily acquired unpaired data to train our NeurMAP framework, which can generalize deblurring to unseen blurring processes and achieve better deblurring results. Overall, in each training iteration, we apply content loss on paired data and aforementioned MAP losses on unpaired data. For the motion estimation network, its training loss is a weighted sum of data term and kernel prior term

$$
\mathcal{L}^M = \lambda \cdot \mathcal{L}_{reblur} + \lambda^M_{kernel} + \mathcal{L}_{tv}(M_{rel}).
$$

(10)

For the deblurring network $D$, we summarize data term and image prior term as the overall training loss

$$
\mathcal{L}^D = \lambda \cdot \mathcal{L}_{reblur} + \mathcal{L}^D_{sharp} + \beta \cdot \mathcal{L}_{natural} + \mathcal{L}_{tv}(M_S).
$$

(11)

Finally, the natural image discriminator $N$ is trained to provide natural image prior

$$
\mathcal{L}^N = \mathcal{L}_N^N.
$$

(12)

$\lambda$ and $\beta$ are only two hyperparameters that balance these training losses. The detailed analysis of setting hyperparameters is provided in implementation details (Section IV-A).

In contrast to the conventional MAP estimation that optimizes variables (i.e. $S,K$) directly and iteratively, we employ the principle of MAP estimation as the training objective of deep neural networks, thus we termed our method as NeurMAP. Different from conventional MAP estimation, the maximization optimization of NeurMAP occurs during the model training. Specifically, the deblurring network $D$ aims to predict the sharp...
content from a blurry input, corresponding to \( p(S|B) \); the motion estimation network \( \mathcal{M} \) predicts the blurry kernel (i.e., the reason, relative motions in this motion, that causing blur), corresponding to \( p(K|B) \); and the reblurring module reproduces the blurring process, corresponding to \( p(B|K,S) \).

When we estimate sharp image \( S \) with (11), the image prior loss \((p(S))\) and the reblurring loss manage to optimize the deblurring network, which can be derived from the following:

\[
\text{arg max}_S p(S|B,K^*) = \frac{p(B|S,K^*)p(S)p(K^*)}{p(K^*,B)} = \frac{p(B|S,K^*)p(S)p(K^*)}{p(K^*)p(B)} \propto p(S)p(B|S,K^*),
\]

(13)

where \( \mathcal{L}_{reblur} \) is proposed for maximizing \( p(B|S,K^*) \), and \( \mathcal{L}_{sharp}, \mathcal{L}_{natural}, \mathcal{L}_{tv}(M_S) \) is designed for maximizing \( p(S) \). Similarly, the kernel prior loss \((p(K))\) and the reblurring loss formulated in (10) aim to optimize the motion estimation network, i.e., \( \text{arg max}_K p(K|B,S^*) \propto p(K)p(B|S^*,K) \). Instead of iterative optimization, these two estimation parts together form the final objective, i.e. \( \text{max}_p p(K,S|B) \), in an end-to-end training. The overall process is shown in Algorithm 1. After all these networks trained under the MAP estimation, only the deblurring network is needed for inference, since it has been trained to fit the unpair image domain.

**IV. EXPERIMENTS**

**A. Implementation Details**

We provide layer-wise details of our motion offset estimation networks in Table I. Basically, the network is an encoder-decoder structure with skip-connections. The input image size is \( H \times W \times 3 \), and the size of output motion map is \( H \times W \times 2 \) for linear motion.

Compared to previous deep deblurring methods, our proposed NeurMAP consists of several loss terms. To ensure the robustness of the proposed framework, we set hyperparameters in a heuristic way. Specifically, we first keep the coefficients for \( \mathcal{L}_{kernel}, \mathcal{L}_{sharp} \) and \( \mathcal{L}_{tv} \) the same, since all of them penalize on estimated motion maps. Our preliminary experiments illustrated that their weights have little influence on deblurring performance. Then, the weight of reblurring loss \( \mathcal{L}_{reblur} \) and paired content loss \( \mathcal{L}_{pair} \) are set as \( \lambda \). We found that such reconstruction loss should be much larger than the other adversarial losses, where we set 100 times. Finally, the balance between the sharp image prior \( \mathcal{L}_{sharp} \) and the natural image prior \( \mathcal{L}_{natural} \) is controlled by \( \beta \). Based on the experimental comparisons presented in Section IV-D, \( \beta \) is set to 0.1 to have a best trade-off between deblurring effect and eliminating artifact.

In addition, we set \( 5 \times 10^{-5}, 10^{-4} \) and \( 10^{-4} \) as the initial learning rate for deblurring network, motion estimation network and natural image discriminator, respectively. All these learning rates linearly decayed to 0 at the end of the training. We use Adam [56] solver for optimization, with \( \beta_1 = 0.9, \beta_2 = 0.999 \) and \( \epsilon = 10^{-8} \). All discriminators are initialized using Xavier [57], and bias is initialized to 0.

**B. Datasets**

Similar to most learning-based methods, we employ GoPro dataset [11] as our paired training data. In addition, we employ

---

**Algorithm 1:** Pseudo Code for Training NeurMAP (Unpair Data).

```plaintext
for number of training iterations do
    Sample minibatch of blurry images \( B \).
    Sample and maintain a pool of sharp images \( S \).
    Compute: \( \hat{S} = D(B) \)
    Fix parameters of deblurring network \( D \).
    Update the parameters of motion estimation network \( \theta_M \) with gradient descent:
    \( \nabla_{\theta_M} \mathcal{L}^M \)
    Update the parameters of natural image discriminator \( \theta_N \) with gradient descent:
    \( \nabla_{\theta_N} \mathcal{L}^N \).
    Fix \( M \) and \( N \), update the parameters of deblurring network \( \theta_D \) with gradient descent:
    \( \nabla_{\theta_D} \mathcal{L}^D \).
end for
```

---

**TABLE I**

| Stage   | Output Space to Depth | Layer Details         |
|---------|-----------------------|-----------------------|
| Conv1   | \( H/2 \times W/2 \) | \( 5 \times 5, 12, 16 \), stride 1 |
| ResBlock1 | \( H/2 \times W/2 \) | \( 5 \times 5, 16 \) |
| Conv2   | \( H/4 \times W/4 \) | \( 5 \times 5, 16, 32 \) |
| ResBlock2 | \( H/4 \times W/4 \) | \( 5 \times 5, 32 \) |
| Conv3   | \( H/8 \times W/8 \) | \( 5 \times 5, 32, 64 \) |
| ResBlock3 | \( H/8 \times W/8 \) | \( 5 \times 5, 64 \) |
| Bottleneck1 | \( H/8 \times W/8 \) | \( 1 \times 1, 64, 128 \) |
| Dconv1   | \( H/16 \times W/16 \) | \( 5 \times 5, 64, 32 \) |
| Bottleneck2 | \( H/16 \times W/16 \) | \( 1 \times 1, 32 + 32, 128 \) |
| Dconv2   | \( H/32 \times W/32 \) | \( 5 \times 5, 64, 16 \) |
| Bottleneck3 | \( H/32 \times W/32 \) | \( 1 \times 1, 16 + 16, 64 \) |
| Dconv3   | \( H/64 \times W/64 \) | \( 5 \times 5, 32, 32 \) |
| Conv4   | \( H/64 \times W/64 \) | \( 5 \times 5, 32, 4, 32 \) |

+ Denotes that a skip connection concatenates this layer with the corresponding layer in the encoder.
two kinds of unpaired data which have different types of blurring texture for testing the unpaired image deblurring ability.

**Kernel-synthesized blur dataset:** First, instead of the paired blur data created by averaging consecutive frames (GoPro dataset), our kernel-synthesized blur dataset is built by convolutional blur kernels. Specifically, we take the sharp images from dataset collected by [13], then convolving them with the blur kernels synthesized by the method proposed in Gong et al. [50]. These blur kernels are represented as a spatial-variant 2D vector map. Finally, we applied 6 different blur kernel maps to each of 300 sharp images, and obtained 1800 blurry images. We splitted them to 5:1 for training/test set. In training phase, the paired information between blurry and sharp images are ignored. This dataset is mainly used for our quantitative evaluation since it has ground truth sharp images and corresponding blur kernels.

**Real-world blur datasets:** Several real-world blur datasets are adopted to evaluate the effectiveness of unpair training on real-world blurry images. First, we have RealBlur dataset [19] which shot blurry image and its unaligned sharp image through a dual-camera system. However, images in RealBlur dataset are shot under the low-light environment and by a professional equipment. To obtain a dataset that is closer to everyday life, we collect a real-world blur dataset with a handheld smartphone, and call it Unpaired-Real-world Blur dataset (URB dataset). We film video clips of real dynamic scenes at 30 FPS, and extract every frame as blurry images. Meanwhile, we take multiple static photos for each scenes as sharp images. Unlike RealBlur dataset, all these blurry images taken in handheld smartphone have no groundtruth. Our data collection process is conducted in a light-sufficient environment, and we try to use a relatively moderate movement speed. Finally, we build a dataset of total 3211 blurry images filmed from 24 different scenes.

### C. Evaluation of Generalization Ability

To evaluate the generalization ability, we employed two popular deblurring backbone networks, i.e. DMPHN-(one stack) [14] and MPRNet [35], to test the proposed NeurMAP framework. Through comparing our models with several State-of-the-arts methods, we conduct both quantitative and qualitative comparisons on the GoPro dataset and the kernel-synthesized blur dataset since they provide ground truth sharp images. For Real-Blur datasets, we adopt geometric alignment [19] on deblurred images before conducting full-reference image quality assessment (PSNR, SSIM, LPIPS [58]). Due to the lack of ground truth in URB dataset, we can only provide quantitative metrics for no-reference image quality assessment (NIQE [59]) and qualitative comparisons including user studies to valid our performance on real-world test set.

Among all comparison methods, the method proposed by DM-PhN [14], MIMO-Unet [37] and MPRNet [35] are trained with only content loss, while DeblurGAN-V2 [44] and DBGAN [18] are trained with a joint loss of content, perceptual and adversarial loss. In addition, DMPHN, DeblurGAN-V2, MIMO-Unet and MPRNet are trained only using GoPro dataset. Besides GoPro dataset, DBGAN utilized their proposed Real-World Blurry Image (RWBI) dataset to synthesize pseudo training pairs. For a fair comparison, we use their official released well-trained models during inference.

### Comparisons on GoPro and kernel-synthesized blur datasets:

As shown in Table II, four metrics are utilized to conduct a quantitative comparison with existing methods for a comprehensive perspective. Among them, PSNR and SSIM mainly focus on the pixel values and structure similarity, while LPIPS [58] and NIQE [59] pay more attention to perceptual similarity and naturalness. As shown in Table II, we can see that both the DM-PhN [14] + NeurMAP and MPRNet [35] + NeurMAP outperform all the State-of-the-art methods on the kernel-synthesized blur dataset in four metrics. Although the backbone model DMPHN has a inferior performance on GoPro dataset, its performance can exceed MPRNet by training with our NeurMAP framework. Our method demonstrates a significant improvement of 0.93 dB in the kernel-synthesized blur test set (DMPHN versus DMPHN + NeurMAP) as shown in Table II. In contrast, if we compare MPRNet with DMPHN, we can observe a large improvement of 2.45 dB (32.66-30.21) on the GoPro dataset made by MPRNet. However, on the kernel-synthesized blur test set, MPRNet only get a 0.84 dB improvement (24.35-23.51). These results highlight the persistent issue caused by the domain gap between training and test images in the deblurring task. Our proposed NeurMAP approach addresses this challenge by utilizing unpaired data to significantly enhance the model’s generalization performance. The results demonstrate the efficacy of NeurMAP in reducing the domain gap and improving the overall accuracy of the deblurring models.

The comparison between the former pre-trained model and our model shows the generalization ability of a model well-trained on the paired dataset is inferior to the model trained with the unpaired dataset. Designing the training scheme that can utilize more general and easier acquired unpaired blurry/sharp data is our core motivation. To further validate the effectiveness of our method, we conducted additional comparisons as described in Table II (DMPHN + GAN and MPRNet + GAN). The results show that the combination of former methods with a GAN loss did not produce a promising deblurring result and in some cases even led to a performance drop. These results reinforce the superiority of our proposed training scheme. Also, a slight performance drop in PSNR can be observed when comparing the method MPRNet + NeurMAP and MPRNet + GAN with the original MPRNet in the GoPro dataset. Under the condition that keeping the model capacity constant while fitting an enlarged blurry image domain (i.e. GoPro dataset and unpaired dataset), there may be a trade-off between the model’s performance on the original GoPro dataset and the model’s generalization ability on the new unpaired dataset.

On the other hand, our method is shown to be very effective in improving LPIPS/NIQE metrics, as shown in kernel-synthesized blur dataset. The methods trained with NeurMAP can achieve much better score in LPIPS/NIQE compared to the method w/o NeurMAP, which suggests the results are visually clearer and closer to the sharp ground truth in terms of human visual perception. This inference can also be supported by the qualitative examples presented in the Fig. 5. Despite the fact that our method successfully recover the sharp content, we find that our results are significantly clearer on the edges, and the details of our results are visually more abundant compared to others. As discussed in [53], our experiments also suggest
that most existing deblurring models which trained with only the content loss would carry intrinsic blur brought by $L_1/L_2$ norm.

**Generalization ability on Real-world blur datasets:** We perform NeurMAP training on unpaired RealBlur and URB dataset, and evaluate the generalization ability on the test set. Quantitative comparisons can be found in Table III. For RealBlur dataset, PSNR, SSIM and LPIPS metrics are calculated for geometrically aligned deblurred images. Similar to the analysis carried out from kernel-synthesized blur dataset, the proposed NeurMAP performs better in LPIPS/NIQE metrics compared to State-of-the-art methods. It is noteworthy that our DMPHN + NeurMAP method achieves better results compared to MPRNet in terms of these two metrics, yet the former is more light-weight and use less inference time. We should point out that, since our method do not rely on any regression-based loss ($L_1/L_2$ losses) which is favored by pixel value metrics, i.e. PSNR/SSIM, it is very hard for our method to obtain a high PSNR/SSIM score. On the contrary, our method tends to synthesize misaligned high-frequency details which may be penalized by PSNR/SSIM [60], [61], [62]. Moreover, comparing the model trained with NeurMAP with model w/o NeurMAP, we can see the improvement on DMPHN is more obvious than

### Table II

**Quantitative Comparison on Both GoPro and Kernel-Synthesized Blur Datasets**

| Model                 | GoPro [11] | Kernel-synthesized blur | Size (MB) | Runtime (s) |
|-----------------------|------------|-------------------------|-----------|-------------|
|                       | PSNR↑ | SSIM↑ | LPIPS↓ | NIQE↓ | PSNR↑ | SSIM↑ | LPIPS↓ | NIQE↓ |            |
| DeblurGAN-V2 [44]     | 29.55 | 0.9340 | 0.2535 | 5.290 | 21.61 | 0.7422 | 0.3339 | 3.802 | 60.9 | 0.124 |
| Stack(4)-DMPHN [14]  | 31.20 | 0.9453 | 0.1162 | 5.418 | 23.90 | 0.8332 | 0.2214 | 3.879 | 86.8 | 0.424 |
| DBGAN [18]            | 31.10 | 0.9424 | 0.1197 | 5.198 | 23.59 | 0.8278 | 0.2387 | 3.425 | 46.5 | 0.379 |
| MIMO-Unet+ [37]       | 32.45 | 0.9572 | 0.0906 | 5.044 | 23.64 | 0.8484 | 0.2397 | 3.378 | 68.4 | 0.018 |
| DMPHN [14]            | 30.21 | 0.9345 | 0.1409 | 5.665 | 23.51 | 0.8187 | 0.2363 | 3.948 | 21.7 | 0.011 |
| DMPHN + GAN           | 29.99 | 0.9359 | 0.1030 | 5.103 | 21.49 | 0.735 | 0.3142 | 3.891 | 21.7 | 0.011 |
| DMPHN + NeurMAP       | 30.21 | 0.9389 | 0.1148 | 4.916 | 24.44 | 0.8500 | 0.1237 | 3.174 | 21.7 | 0.011 |
| MPRNet [35]           | 32.66 | 0.9591 | 0.0886 | 5.155 | 24.35 | 0.8499 | 0.2248 | 3.308 | 80.6 | 0.143 |
| MPRNet + GAN          | 32.10 | 0.9591 | 0.0625 | 4.333 | 21.15 | 0.7687 | 0.1733 | 3.221 | 80.6 | 0.143 |
| MPRNet + NeurMAP      | 32.30 | 0.9601 | 0.0539 | 4.110 | 24.81 | 0.8629 | 0.1192 | 3.052 | 80.6 | 0.143 |

The red and underlined values indicate the best and second best performance respectively.

### Table III

**Quantitative Comparison on Real-World Datasets**

| Model                 | RealBlur [19] | URB | PSNR↑ | SSIM↑ | LPIPS↓ | NIQE↓ |
|-----------------------|---------------|-----|-------|-------|--------|-------|
| DeblurGAN-V2 [44]     | 26.52 | 0.8052 | 0.1935 | 5.464 | 4.590 |
| Stack(4)-DMPHN [14]  | 27.78 | 0.8469 | 0.1920 | 5.728 | 5.354 |
| DBGAN [18]            | 24.93 | 0.7451 | 0.2945 | 5.323 | 5.050 |
| MIMO-Unet+ [37]       | 27.63 | 0.8386 | 0.1991 | 5.442 | 4.818 |
| DMPHN [14]            | 27.52 | 0.8369 | 0.1973 | 5.682 | 5.201 |
| DMPHN + NeurMAP       | 28.05 | 0.8444 | 0.1497 | 4.140 | 4.302 |
| MPRNet [35]           | 28.70 | 0.8729 | 0.1527 | 4.308 | 5.002 |
| MPRNet + NeurMAP      | 28.49 | 0.8603 | 0.1425 | 4.834 | 4.570 |

The red and underlined values indicate the best and second best performance respectively.
that on MPRNet, we infer the MPNet is harder to train for its heavy-weighted structure.

We can also observe a better visual quality results on real-world test data. As shown in Figs. 6 and 7, we provide examples from RealBlur and URB test set respectively. We can see nearly all the models trained on GoPro dataset fail to generalize to real-world blur. In fact, they tend to output an identical image when handling unseen bluriness. In addition, although DBGAN [18] utilize real-world blurry images in RWBI dataset as training data, it also fails to recover sharp content from real-world blurry images. Due to the lack of quantitative metric to evaluate the performance of our model on the Real-world data without ground truth. A user study is conducted to validate the visual quality of our well-trained DMPHN + NeurMAP model (Fig. 9). We selected 50 real-world blurry images, for each image, we asked 10 users on Amazon Mechanical Turk to compare the deblurring results of different methods. The result shows that about 70% to 80% of people think our results are visually better than the others.

Analysis of the estimated relative motion map: Our framework performs the deblurring and motion estimation in a collaborative manner. The more accurate motion information is estimated,
the better the deblurring performance will be achieved. Here, we validate that our well-trained motion estimation network can estimate accurate and dynamic motions/blur kernels in an explicit representation. First, we provide a quantitative evaluation on kernel-synthesized blur dataset since they have ground truth blur kernels. Specifically, we calculate the mean squared error (MSE) between estimated motion vectors and ground truth. Fig. 8(a) shows our estimated motion vectors are more accurate than a former supervised motion estimation method [50]. Note that we follow [50] to build the kernel-synthesized blur dataset and our NeurMAP did not employ any paired information or ground truth motions during the training. Also, we visualize the relative motion map estimated from both kernel-synthesized blur dataset and real-world dataset as shown in Fig. 8(b). Our estimated motion maps are highly consistent with the ground truth motion maps in both direction and magnitude. Although there is no ground truth for real-world test image, we observe the estimated motion maps mainly focus on the blurry edges, which partly explains how the motion estimation contributes to image deblurring.

D. Evaluation of Training Strategies and Losses

To analyze the effectiveness of our proposed framework, we compare the deblurring networks trained using different strategies/losses. For fair comparison, all the comparison methods conduct a semi-supervised learning scheme.

a) Ablation studies on training strategies: We design several training strategies to validate the effectiveness of each proposed modules in the MAP framework. We conduct experiments on the three main parts of the proposed NeurMAP, i.e., motion estimation network \( \mathcal{M} \) for learning \( p(K) \), adversarial training on motion and image discriminators for learning sharp/natural image priors \( p(S) \) and reblurring module \( \mathcal{R} \) for calculating \( p(B|K,S) \). Quantitative results are evaluated with kernel-synthesized test data in DMPHN backbone (Table IV).

As shown in Table IV, Net1 employs a vanilla sharp image discriminator which distinguishes between sharp and blurry images and Net2 applies a natural image discriminator to encourage more natural image rather than generating artifact. The results show that model with only \( p(S) \) causes a significant drop in PSNR and SSIM scores since it cannot preserve the content. In Net3 and Net4, we add reblurring loss when training \( D \). By comparing Net1/Net2 with Net3/Net4, the results show that when pair loss is absent for training unpaired data, reblurring loss (for deblurring) is critical to maintaining content consistency. Based on Net3/Net4, we further add the motion estimation module for comparison, i.e. Net6 and NeurMAP. It can be observed that the experiments with kernel prior loss and reblurring loss (for motion estimation) can significantly improve the deblurring performance, since the accurately estimated motion provide a more accurate sharp image prior. However, since Net6 cannot handle artifacts well, we add the natural image discriminator to form the NeurMAP finally.

To summary, the experiments with only \( p(S) \) (Net1, Net2), with \( p(B|K,S) + p(S) \) (Net3, Net4), and with \( p(B|K,S) + p(K) \) (Net5) all fail to generate promising results. On the contrary, the experiments with all three parts (Net6, NeurMAP) can significantly improve the performance with respect to all the metrics, therefore, it is crucial that these three main parts work together to form the MAP framework.

We can also discuss the function of reblurring loss in our proposed framework here. Most existing works [20], [21] apply both the paired loss and the reblurring loss on the same paired training dataset. For the test data sampled from the same domain/distribution, reblurring loss acts as an auxiliary regularization. However, in our setting, paired loss and reblurring loss are applied on different training datasets. Reblurring loss in our method exists in both the training of the deblurring network training and the motion estimation network. When estimating sharp image, since \( p(K) \) is fixed, the reblurring loss serves

---

**Table IV**

| Method | \( \mathcal{M} \) | \( \mathcal{R}^{\mathcal{M}} \) | \( \mathcal{R}^{\mathcal{D}} \) | Image priors | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|---|---|---|
| Net1 | ✓ | | | Sharp | 18.99 | 0.7035 | 0.3676 |
| Net2 | | ✓ | ✓ | Natural | 20.82 | 0.7112 | 0.3132 |
| Net3 | ✓ | ✓ | | Sharp | 22.20 | 0.7787 | 0.1820 |
| Net4 | | ✓ | ✓ | Natural | 21.49 | 0.7330 | 0.3142 |
| Net5 | ✓ | ✓ | ✓ | (motion) | 21.94 | 0.7409 | 0.2274 |
| NeurMAP | ✓ | ✓ | ✓ | (motion) | 23.65 | 0.8280 | 0.1640 |

The bold face values indicate the best performance.

**Table V**

| Model (backbone – DMPHN) | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|
| \( \lambda_{\text{sharp}} : \lambda_{\text{natural}} = 1 : 0.5 \) | 22.16 | 0.7588 | 0.2205 |
| \( \lambda_{\text{sharp}} : \lambda_{\text{natural}} = 2 : 0.1 \) | 21.42 | 0.7312 | 0.2336 |
| \( \lambda_{\text{sharp}} : \lambda_{\text{natural}} = 1 : 0.1 \) | **24.44** | **0.8500** | **0.1237** |

The bold face values indicate the best performance.
as a part of $\max \ p(S|B, K^*) \propto p(S)p(B|S, K^*)$. Similarly, when estimating blur kernel ($p(S)$ is fixed), the reblurring loss serves as a part of $\max \ p(K|B, S^*) \propto p(K)p(B|S^*, K)$. The reblurring loss is an important component to constrain both optimizations.

b) Ablation studies on TV loss: In addition to validating the losses designed for each training module as a whole, we also provide visual comparisons of total variance loss. Fig. 11(b) shows that without TV regularization, the texture artifacts will arise. As shown in Fig. 12, we show the results come from different settings of hyperparameters of $\mathcal{L}_{\text{sharp}}^D$ and $\mathcal{L}_{\text{natural}}^D$.

We define the weight of $\mathcal{L}_{\text{sharp}}^D$ and $\mathcal{L}_{\text{natural}}^D$ in loss function (11) (main submission) as $\lambda_{\text{sharp}}$ and $\lambda_{\text{natural}}$, respectively. As we can see, $\lambda_{\text{sharp}} : \lambda_{\text{natural}} = 1 : 0.1$ generates reasonable deblurring results. When we raise $\lambda_{\text{natural}}$ to 0.5, there is little deblurring effect in the results. On the contrary, when we raise $\lambda_{\text{sharp}}$ to 2, the unsmooth artifact will arise in the deblurring results. The quantitative results on kernel-synthesized blur dataset (Table V) also show that all the other models have a poor performance on PSNR/SSIM, which means they fail in restoration fidelity. It is also interesting to observe that the kernel prior loss $\mathcal{L}_{\text{sharp}}^D$ is beneficial to improve the NIQE metric, which shows that $\mathcal{L}_{\text{sharp}}^D$ is effective in generating sharpened texture.

d) Comparison with former unsupervised methods: The proposed NeurMAP can achieve a better deblurring performance without paired data and pair loss, yet the performance is not stable (sometimes with artifact) and drops significantly compared to the model trained by only the paired dataset, especially in terms of PSNR/SSIM, so we are not satisfying about the results. Here, we further compare our NeurMAP with the former unsupervised methods CycleGAN [63] and UID-GAN [64]. Both CycleGAN [63] and UID-GAN [64] are cycleGAN-based methods that work toward learning unpaired image-to-image transfer between different domains. However, the boundary between blurry and sharp images is often vague (e.g. blurry to sharp is a continuous transformation, blurry/sharp textures are sometimes similar), and it is difficult for cycleGAN-based methods to distinguish whether an image is blurry or not. As a result, while [63], [64] can enable training on unpaired data, they cannot properly deblur the dynamic real-world datasets, e.g. real-world street-view like GoPro dataset.

UID-GAN [64] tries to improve upon CycleGAN [63] by disentangling the blur information, but the blur information is modeled as an implicit vector, which may not represent the complex and non-uniform blur in real-world scenes. Thus, UID-GAN only works on domain-specific tasks, i.e. face and text. In contrast, our method uses a pixel-wise map as the representation of blur information and a fixed and accurate reblurring module to complete the cyclic loss, which helps us to recover more dynamic and accurate blur information.

We conduct unsupervised and semi-supervised training for [63], [64] and compare the results with our method. Results in the Table VI shows that our method achieves a much better PSNR/SSIM in both fully unsupervised training scheme and semi-supervised scheme. We infer that the improvement comes from our accurate estimation of motion, and the sharp image prior imposed by the motion estimation. Visual comparison in Fig. 13 also shows the cycleGAN-based methods [63], [64]...
Fig. 12. Results of different hyperparameters of $\mathcal{L}^D_{\text{sharp}}$ and $\mathcal{L}^D_{\text{natural}}$. From left to right shows 3 different settings of hyperparameters. The first one generates reasonable results; the second one tends to generate blurry output; the third one will cause more artifacts.

Fig. 13. Visual comparison of unsupervised methods. Neither paired data nor pair loss is employed during training.

usually output different color or tune as the style they learned, since they cannot properly distinguish the expected style (blur/sharp).

E. Discussion and Limitation

Learning to deblur from unpaired data is a challenging task. Here, we attempt to further clarify the principle and limitations of our NeurMAP through three questions.

Q. Differences between GAN-based methods with NeurMAP?
A: GANs are widely used in image processing tasks to improve the realistic of generated images. However, former GAN-based methods [32], [44] need to apply GAN losses and pair loss on the same paired dataset. When the pair loss is absent for unpaired data, GANs can only penalize the similarity between two distributions (e.g. $p(S)$ and $p(\tilde{S})$), they cannot handle the unpaired training data without using other constraints. In the proposed NeurMAP, a reblurring term is presented to optimize the likelihood $p(B|K, S)$, which is critical to keep the content consistency and the physical meaning of kernels.

CycleGAN-based methods are also proposed to maintain the content consistency, which we have compared and analyzed in the Section IV.D.D.

Q. Can NeurMAP optimize a single image like traditional MAP optimization?
A: Our proposed NeurMAP cannot perform optimization on a single image, while we aim to learn the blur patterns in a dataset. Different from the conventional MAP which applies pre-defined
respectively. The blue curve shows an ideal and with absolute motion will with absolute $M^\hat{B}$ and synthesize blurry images with randomly generated $-M^\hat{Q}$. Is needed to define the natural image domain. When learning natural image priors, the dataset of sharp images should be learned from the unpaired sharp images. For example, our natural/sharp image priors for natural/sharp image, our natural/sharp image priors are comparable across these settings, with some slight drops settings as in Table II, yet the reblurring module is changed to synthesize quadratic motion blur. Correspondingly, the output vector of the motion estimation network changes from 2D to 4D, as a representation for quadratic motion [52]. The results in Table VII show that the improvement on PSNR/SSIM is trivial. Similar to the conclusions drawn from Table II, more pronounced improvement can be observed in terms of LPIPS and NIQE metrics. We can also observe a clearer edge recovered from our method in the visual comparison of Fig. 15.

**Q. Why the paired loss is needed in the NeurMAP training scheme? (limitation.)**

As we mentioned, the proposed NeurMAP can be trained only using unpaired data. Actually, we tried this fully unsupervised training scheme, and experiments (Fig. 13, Table VI) show that our model is still able to preserve the content consistency in color and most texture, and achieve better deblurring performance than former fully unsupervised methods [63, 64] in dynamic scene. However, the deblurring effect is not satisfactory and different artifacts will arise. These suggest that our NeurMAP is still troubled by ill-posedness under unsupervised training settings. Supervised training on paired data cannot improve deblurring performance on unpaired data, yet it should be a crucial term to insist the image prior $p(S)$. On the one hand, considering the domain gap between paired data and unpaired data, our experiments (Table VIII, Figs. 6, 7) demonstrate that the pair loss on paired data cannot contribute to learning $p(S|B_{un})$. We also decrease the ratio of paired data in one training batch from 50% (our default setting) to 25% and 12.5% respectively in Table IX. It can be observed that the results on unpaired dataset are comparable across these settings, with some slight drops when the ratio decrease to 12.5%. On the other hand, the pair loss can be regarded as enforcing the natural image prior for blurry images without sharp ground truth (i.e. unpaired data). It still plays an important role to enforce the deblurring network to output sharp natural images. We will conduct further research for unsupervised deblurring in the future.
TABLE VIII
QUANTITATIVE COMPARISON OF DIFFERENT COMBINATIONS OF TRAINING LOSS AND DATASET

| Training losses | Test on GoPro dataset | Test on K-syn dataset |
|-----------------|------------------------|-----------------------|
| Pair            | unpair                 | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| Exp1            | GoPro                  | 30.21 | 0.9345 | 0.1409 | 23.51 | 0.8187 | 0.2363 |
| Exp2            | GoPro GoPro            | 29.86 | 0.9307 | 0.0917 | 23.61 | 0.8225 | 0.2370 |
| NeurMAP         | GoPro K-syn            | 30.21 | 0.9389 | 0.1148 | 24.44 | 0.8500 | 0.1237 |

V. CONCLUSION
In this article, we propose a NeurMAP framework that consists of a deblurring network, a motion estimation network and an natural image discriminator. Based on the proposed NeurMAP estimation, all these networks are trained jointly to remove the blur patterns within unpaired data. Experiments show that our NeurMAP significantly improves the generalization ability to unseen blurliness. However, there still exist challenges, e.g. still requiring paired data during training. In future works, We hope to explore better image priors or introduce high-level semantic guidance to solve these problems.

ACKNOWLEDGMENTS
The authors would like to thank anonymous reviewers and the handling editor for their constructive comments.

REFERENCES
[1] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph,” ACM Trans. Graph., vol. 25, no. 3, pp. 787–794, 2006.
[2] S. Cho and S. Lee, “Fast motion deblurring,” ACM Trans. Graph., vol. 28, no. 5, 2009, Art. no. 145.
[3] J. Jia, “Single image motion deblurring using transparency,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2007, pp. 1–8.
[4] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Efficient marginal likelihood optimization in blind deconvolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2011, pp. 2657–2664.
[5] T. Hyun Kim, B. Ahn, and K. Mu Lee, “Dynamic scene deblurring,” in Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 3160–3167.
[6] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, “Blind image deblurring using dark channel prior,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1628–1636.
[7] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, “Non-uniform deblurring for shaken images,” in Proc. IEEE Int. Conf. Comput. Vis., 2012, pp. 168–186.
[8] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, “l_0-regularized intensity and gradient prior for deblurring text images and beyond,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 2, pp. 342–355, Feb. 2017.
[9] L. Xu and J. Jia, “Two-phase kernel estimation for robust motion deblurring,” in Proc. Eur. Conf. Comput. Vis., Springer, 2010, pp. 157–170.
[10] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, “Neural blind deconvolution using deep priors,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3341–3350.

[11] S. Nah, T. Hyun Kim, and K. Mu Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3883–3891.
[12] X. Zhang, Y. Gao, X. Jia, W. Zuo, and J. Jia, “Scale-recurrent network for deep image deblurring,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8174–8182.
[13] X. Gao, X. Tao, X. Shen, and J. Jia, “Dynamic scene deblurring with parameter selective sharing and nested skip connections,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3848–3856.
[14] H. Zhang, Y. Dai, H. Li, and P. Koniusz, “Deep stacked hierarchical multi-patch network for image deblurring,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5978–5986.
[15] M. Suin, K. Purohit, and A. Rajagopalan, “Spatially-attentive patch-hierarchical network for adaptive motion deblurring,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3606–3615.

TABLE IX
THE COMPARISON FOR DIFFERENT RATIO OF PAIRED DATA IN SEMI-SUPERVISED TRAINING

| Model            | GoPro dataset | K-syn dataset |
|------------------|---------------|---------------|
|                  | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| 12.5% pair data  | 29.89 | 0.9356 | 0.1131 | 24.36 | 0.8432 | 0.1388 |
| 25% pair data    | 29.96 | 0.9360 | 0.1057 | 24.34 | 0.8477 | 0.1336 |
| 50% pair data    | 30.21 | 0.9389 | 0.1148 | 24.44 | 0.8500 | 0.1237 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
