Conditional Hyper-Network for Blind Super-Resolution with Multiple Degradations

Guanghao Yin*, Wei Wang*, Zehuan Yuan, Wei Ji, Dongdong Yu, Shouqian Sun, Tat-Seng Chua, Changhu Wang

Abstract—Although the single-image super-resolution (SISR) methods have achieved great success on the single degradation, they still suffer performance drop with multiple degrading effects in real scenarios. Recently, some blind and non-blind models for multiple degradations have been explored. However, these methods usually degrade significantly for distribution shifts between the training and test data. Towards this end, we propose a novel conditional hyper-network framework for super-resolution with multiple degradations (named CMDSR), which helps the SR framework learn how to adapt to changes in the degradation distribution of input. We extract degradation prior at the task-level with the proposed ConditionNet, which will be used to adapt the parameters of the basic SR network (BaseNet). Specifically, the ConditionNet of our framework first learns the degradation prior from a support set, which is composed of a series of degraded image patches from the same task. Then the adaptive BaseNet rapidly shifts its parameters according to the conditional features. Moreover, in order to better extract degradation prior, we propose a task contrastive loss to shorten the inner-task distance and enlarge the cross-task distance between task-level features. Without predefining degradation maps, our blind framework can conduct one single parameter update to yield considerable improvement in SR results. Extensive experiments demonstrate the effectiveness of CMDSR over various blind, and even several non-blind methods. The flexible BaseNet structure also reveals that CMDSR can be a general framework for a large series of SISR models. Our code is available at https://github.com/guanghaoyin/CMDSR.

Index Terms—Blind Super-resolution, Hyper-network, Meta-learning, Multi-degradation Shift.

I. INTRODUCTION

Single image super-resolution (SISR) has posed a long-standing challenge in low-level vision with numerous important applications. It is an ill-posed problem that aims to restore a High-Resolution (HR) image by adding the missing high-frequency information from a Low-Resolution (LR) image. Since the pioneering method by SRCNN [1], deep learning approaches [2]–[8] have exhibited impressive performance. However, most existing methods focus on a fixed degradation, *i.e*, bicubic down-sampling or single Gaussian blurring. Such fixed settings really limit their generalization ability. In addition to down-sampling, unknown blurring and noise may also be introduced during the acquisition of LR
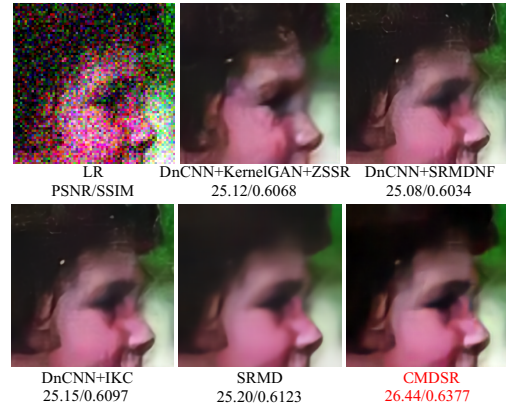
* Equal contribution.

Guanghao Yin, Shouqian Sun are with the Key Laboratory of Design Intelligence and Digital Creativity of Zhejiang Province, Zhejiang University, Hangzhou, China. E-mail: {ygh_zju, ssq}@zju.edu.cn.

Wei Ji, Tat-seng Chua are with School of Computing, National University of Singapore, Singapore. E-mail: {jiwei,dcscts}@nus.edu.sg.

Wei Wang, Zehuan Yuan, Dongdong Yu, Changhu Wang are with the Bytedance AI Lab, Beijing, China. E-mail: {wangwei.frank, yuanzehuan, yudongdong, wangchanghu}@bytedance.com.

Fig. 1. SR results (×4) with anisotropic Gaussian blur and AWGN (*Severe* degradation). Without anisotropic blur kernel for training, our CMDSR outperforms all the blind cascaded schemes, even better than non-blind method, SRMD [12], which was trained with anisotropic Gaussian blur.

images. When the data distributions at test time mismatch the training distributions (referred to as distribution shift [9], [10]), these learning-based models will suffer severe performance drop [11].

In recent years, several non-blind and blind super-resolution methods for multiple degradations have been proposed. The non-blind methods [12]–[15] usually take the ground truth (GT) degradation maps as an additional input to establish the LR-HR mapping. Although the non-blind models have achieved satisfactory performance with the guidance of predefined information, the problem with unknown realistic degradation largely limits their usage in real-world applications. Besides, the blind methods [16]–[19] only consider the blur and down-sampling in the degradation mode. Then, the cascaded schemes with blind denoising, blur estimation and SR methods are organized to restore the multi-degraded LR image [12], [20], [21]. However, each stage has a negative impact on each other, (*i.e*, the denoiser will make the LR image more blurred and lead to kernel mismatch, increasing the difficulty of deblurring, as seen in Fig. 4). Recently, there are some new attempts for blind SR. Several CycleGAN [22] based methods [21], [23]–[25] learn from unpaired LR-HR images, but they are more difficult to train. ZSSR [17] explores the zero-shot solution for the first time, where the CNN learns the mapping from the LR image and its downscaled versions (self-supervision). But it requires thousands of self-training iterations for each LR image. Recently, two optimization based meta-learning strategies, MZSR [26] and MLSR [27], have been proposed to accelerate the self-training steps from 1000
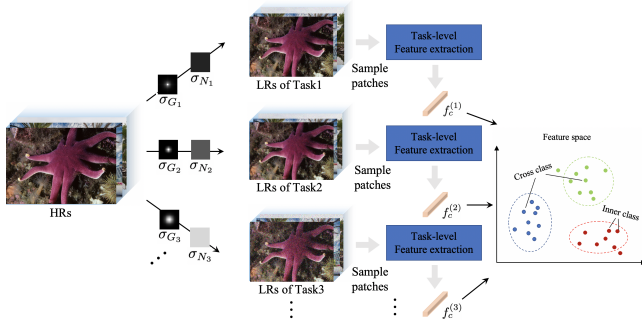
Fig. 2. Conditional feature extraction at task-level. There is a prior knowledge that degraded LR images from the same task have common degradation patterns (*i.e*, blur kernel width σ_G, noise level σ_N), which are different between different tasks. Therefore, we attempt to learn degradation prior at task-level and use the extracted conditional features to adapt the SR model to a new task.

to 10. But they have limitations when dealing with large scale factor because the self-downsampled image cannot provide enough information. Moreover, all those blind and non-blind methods cannot handle the degradation shift problem. When degradation distribution shifts between the training and test data, those SOTAs usually degrade significantly.

To address the problems above, our work tries to explore the following two challenges at the same time: (1) Can we propose a blind framework to effectively handle multiple degradations, especially when the accurate degradation estimation is very difficult? (2) Is it possible to overcome the distribution shift with an adaptive model, which can learn how to adapt its parameters to the unknown degraded LR images?

In this paper, we propose a conditional hyper-network for blind SR with multiple degradations (CMDSR) to largely overcome the aforementioned two challenges. For the first challenge, there is a prior knowledge which inspires us to handle it at task-level. As shown in Fig. 2, the LR images with different degradations obey different distributions. Although the accurate estimation of degradation is hard, images from the same task may contain similar implicit features to describe their common degradation patterns. Therefore, we group these LR images into different tasks and extract the degradation priors at task-level to describe the degradation patterns. Throughout this article, we define a group of LR patches with the same degradation pattern as a **support set**. For the second challenge, we use the distribution information extracted from the support set to make SR network adaptively adjust its parameters according to the distribution changes, such that our framework can handle distribution shifts.

Specifically, our CMDSR consists of two parts: the BaseNet and ConditionNet. As shown in Fig. 3, the shallow ConditionNet learns the feature representations of different tasks. Then, BaseNet multiplies its convolution weights with modulated conditional features in channel-wise. Finally, the adapted BaseNet restores the LR image. Inspired by recent contrastive learning [28], [29], we propose a task contrastive loss to decrease the distances of the conditional features from the same task and increase the distances of the conditional features from different tasks. Algorithm 1 presents the training stage,

where BaseNet and ConditionNet are alternately optimized with different steps and loss functions. Algorithm 2 presents the test stage, where the extracted degradation prior from ConditionNet adapts BaseNet to handle distribution shift.

Since the shallow ConditionNet only uses the small size of support patches (*i.e*, 48×48), the time and computation cost of conditional feature extraction will be very little compared with BaseNet reconstruction. Without designing new complicated SR network, the proposed framework simply uses 10 res-blocks as BaseNet (called SRResNet-10) and achieves superior performance with blind methods. For complicated degradations, CMDSR even outperforms the non-blind models. It should be noted that our framework has no strict restrictions on the BaseNet structure. The ablation experiments in Table X demonstrates that CMDSR can be extended to other SISR models. To the best of our knowledge, the proposed CMDSR is the first hyper-network framework for blind SISR with multiple degradations.

In summary, our overall contribution is three-fold:

1) We present the first blind hyper-network framework to adaptively handle distribution shifts for the SISR task with multiple degradations in task-level.
2) We propose an unsupervised task contrastive loss to extract more discriminative task-level features.
3) Our proposed framework is hot-pluggable, efficient and flexible. We can replace the modules with stronger models for better performance. Hence, our framework can be applied as a general super-resolution framework.

II. RELATED WORK

Blind Single-Image Super-Resolution. Compared with typical SISR models [1], [2], [4], [5] which are tailored to specific single downsampler, blind SISR is a more challenging task, which assumes that the blur kernels are unavailable at test time. Previous methods usually combine the well-designed kernel-estimation and typical SISR methods. Michaeli *et al.* [16] mined the internal patch recurrence to estimate the blur kernel. Bell *et al.* [19] proposed KernelGAN to learn the blur kernel distribution. In order to relieve the mismatch between the estimated kernel and the real kernel, IKC [18] iteratively trained the estimation and correction networks. Although the accuracy of estimated kernel is largely improved, it remains very challenging for severe degradation. Inspired by zero-shot and self-supervised learning, ZSSR [17] efficiently exploited the internal recurrence of information inside an image. But this image-specific model requires self-training for each LR image, which is time-comsuming and cannot be applied to deep structure.

Multiple Degradations. Few attention has been paid to SISR with multiple degradations despite its important for real applications. Based on the perspective from maximum a posteriori (MAP) framework, existing non-blind methods (such as SRMD [12] and UDVD [15]) concatenate LR image, predefined blur kernel and noise maps as the input. Thus, SR result closely depends on both the LR image and degradation patterns. The blind schemes [12], [21] are usually based on the sequential combinations of denoising [30], blur estimation [19]
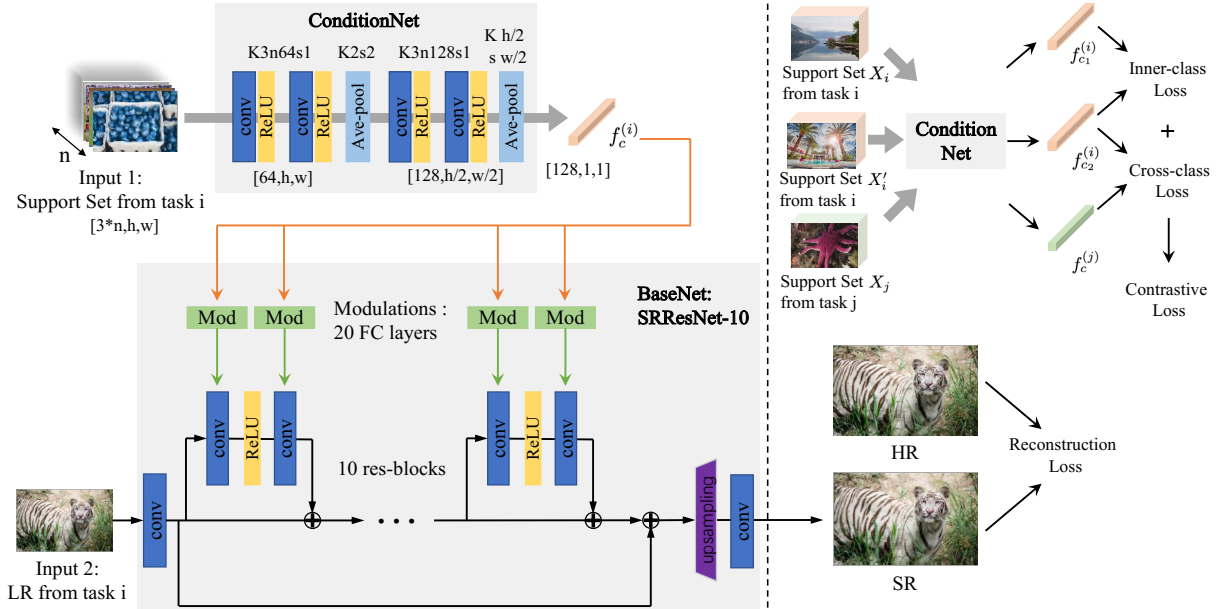
Fig. 3. Overall scheme of our proposed CMDSR. Left: Network structures of ConditionNet and BaseNet. ConditionNet extracts conditional feature $f_c^{(i)}$ from input support set. BaseNet adapts its parameters to task i according to the modulated features. Right: Loss functions optimization. BaseNet and ConditionNet are alternately trained by different loss configurations. It should be noted that although we simply use SRResNet-10 as BaseNet, CMDSR has no strict restrictions on the BaseNet structure and can be conveniently extended to other SR models without changing their structures.

and SR models [17]. Besides, CBSR [20] adapted a cascaded architecture, which can be jointly end-to-end learned from training data. All these methods will degrade for distribution shifts. Recently, the unpaired SISR methods [21], [23] conducted the domain transfer between the clean and real degraded domains. But it remains challenging to train a stable model for various shifts.

Meta-Learning. Meta-learning, commonly known as learning how to learn, refers to the process of improving a learning algorithm over multiple learning episodes. As pointed out by [31], [32], diverse meta-learning methods can be categorized into three groups: (1) Metric based methods [33]–[35] perform non-parametric learning in the metric space, which are largely restricted to the popular; (2) Optimization based methods [31], [36]–[39] use gradient descent to solve the optimization problem of meta-learner. A most famous example is MAML [37], which learns the transferable initial parameters, such that few gradient updates lead to performance improvement. And MTL [38] that leverages the advantages of transfer and meta learning. Recently, [39] proposes the Adaptive Risk Minimization to handle group distribution shift for image classification; (3) Network based methods [40]–[42] that use network to learn across tasks and rapidly updates its parameters to a new task.

There are few explorations of meta-learning for SISR. Recently, two gradient based meta-learning models have been proposed. They are the MZSR [26] and MLSR [27], which both employed the typical MAML framework [37] to accelerate the self-supervised training. Nevertheless, for large scale factors, the size of self-downsampled LRs on image becomes too small to provide enough information, which which limits their applications to real scenarios.

III. PROPOSED METHOD

A. CMDSR Setting

Our work focuses on blind SISR with multiple degradations, including blur, noise and down-sampling, which may simultaneously happen in a real-world case [11]. The degradation process is formulated as:

$$\mathbf{I}_{LR} = (\mathbf{I}_{HR} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}, \tag{1}$$

where $\mathbf{I}_{LR}$, $\mathbf{I}_{HR}$, $\mathbf{k}$, $\otimes$, $\downarrow_s$ and $\mathbf{n}$ respectively denote LR, HR image, blur kernel, convolution, decimation with scaling factor of s. and Gaussian noise. In this paper, we use the configuration in Eq. (1). to synthesize the LR images for training.

The key goal of our work is to develop a framework that can adapt and generalize in the face of degradation shift using only a small number of examples. To accomplish this, we need to find the representation that can describe the degradation prior of the LR image and guide the model to adapt to this degradation pattern. As explained in Section I, we observe that the LR images from the same task are degraded with the same pattern. This inspires us to deal with this problem in task-level, rather than in image-level. Therefore, we present a new approach to mine the implicit task-level semantics with different tasks. The extracted feature can be further used as a context prior to adapt the parameters of SR model.

In our framework, we provide two settings to access the training data: (1) The training data should be grouped into different tasks. We consider the multi-degradation distribution $p(\mathcal{T})$ over training tasks $\{\mathcal{T}_1, \mathcal{T}_2, ...\}$. For task $\mathcal{T}_i$, it consists of LR-HR pairs, where LR images $\{x_1^{(i)}, ..., x_m^{(i)}\}$ are synthesized from HR images $\{y_1^{(i)}, ..., y_m^{(i)}\}$ with ith degradation configuration. (2) ConditionNet extracts task-level feature from n

LR patches (named support set) belonging to the same task, $X_i = \{x_1^{(i)}, ..., x_n^{(i)}\}$ and BaseNet restores the single LR input $x_j^{(i)}$. With these settings, our framework can treat the training data at task-level.

B. Networks of CMDSR

Our framework consists of ConditionNet and BaseNet, which are shown in Fig. 3. It should be noted that our framework has no strict restrictions on the BaseNet structure. In this paper, the backbone of BaseNet is simply designed as SRResNet-10, which consists of 10 res-blocks.

First, ConditionNet, denoted as F_c, extracts the conditional feature $f_c^{(i)}$, which describes the degradation pattern i of the input support set $X^{(i)}$. It is formulated as:

$$f_c^{(i)} = F_c(X^{(i)}; \phi) = F_c(x_1^{(i)}, ..., x_n^{(i)}; \phi), \quad (2)$$

where ϕ is the parameters of ConditionNet and n denotes the size of support size at each step. In order to extract the task-level feature, we design a shallow ConditionNet with 2 average pooling layers and 4 convolution layers followed with ReLU and keep the input sample size unchanged during the training and test phrase. The internal channels of convolution layers are $c_{in} * n$, 64, 64, 128, 128.

Then, BaseNet, denoted as F_{sr}, adapts its original parameters θ to θ' with the conditional feature $f_c^{(i)}$. Since our model aims to be a general framework, we use depth-wise scaling to adapt convolutions without changing the CNN structures. Specifically, we adapt the parameters of 20 conv-layers of internal 10 res-blocks. We use 20 full-connected layers to generate adaptive coefficients with $f_c^{(i)}$ as input. The FC modulation layers change the number of $f_c^{(i)}$ channels to match convolution weights and also adjusts $f_c^{(i)}$ for each conv-layer. Then, the modulated features multiply with the weight of convolution in depth-wise manner:

$$w'_{pq} = w_{pq} * f'^{(i)}_{pq}, \quad (3)$$

where w_{pq} and w'_{pq} are the original and adapted weights, $f'^{(i)}_{pq}$ is the modulated variable corresponding to the qth channel of pth conv-layer. Finally, the adapted BaseNet restores the input LR image $x_j^{(i)}$ to the SR image $\hat{y}_j^{(i)}$. The whole process of BaseNet is formulated as:

$$\hat{y}_j^{(i)} = F_{sr}(x_j^{(i)}, f_c^{(i)}; \theta) = F_{sr}(x_j^{(i)}; \theta'). \quad (4)$$

C. Loss Functions

Owing to the fact that ConditionNet and BaseNet serve different purposes, they have different sensitivity to learning rate and loss functions. Hence, we optimize them alternately with different learning rates and optimization objectives. ConditionNet is trained after every t steps of BaseNet training. The details of loss functions are listed as follows.

Reconstruction Loss for BaseNet. Similar to most SISR models [2]–[5], we adopt a supervised reconstruction loss to calculate the L_1 distance between HR image and output SR image of BaseNet in pixel-wise,

$$L_{res} = \mathbb{E}_{(\mathbf{I}_{HR}, \mathbf{I}_{LR}) \sim p(\mathcal{T})} \parallel \mathbf{I}_{HR} - F_{sr}(\mathbf{I}_{LR}; \theta) \parallel_1 . \quad (5)$$

Combined Unsupervised Task Contrastive Loss and Supervised Reconstruction Loss for ConditionNet. As the prior knowledge explained before, our ConditionNet should output the conditional features, which are similar to those from the same degradation and dissimilar to others from different degradations. Instead of matching an input to a fixed target, recent works of contrastive learning [28], [29] measure the similarities of sample pairs in a representation space. Inspired by them, we propose a task contrastive loss, which shortens the inner-task distance and enlarges the cross-task distance between different conditional features.

For the inner-task loss, we sample two support sets from the same task, each containing n LR patches, represented as X_i, X'_i. And ConditionNet $F_c(\cdot; \theta)$ extracts features $f_{c_1}^{(i)}$, $f_{c_2}^{(i)}$ from X_i, X'_i. The inner-task loss is calculated as:

$$L_{inner} = \mathbb{E}_{X_i, X'_i \sim p_x(\mathcal{T}_i)} \parallel F_c(X_i; \theta) - F_c(X'_i; \theta) \parallel^2$$
$$= \mathbb{E}_{X_i, X'_i \sim p_x(\mathcal{T}_i)} \parallel f_{c_1}^{(i)} - f_{c_2}^{(i)} \parallel^2 . \quad (6)$$

For the cross-task loss, we resample n LR images from another task, denoted as support set X_j, which show different degradation distribution from X_i. Also, ConditionNet $F_c(\cdot; \theta)$ extracts conditional features $f_c^{(i)}$, $f_c^{(j)}$ from X_i, X_j. Then, the cross-task loss can be calculated as

$$L_{cross} = \mathbb{E}_{X_i \sim p_x(\mathcal{T}_i), X_j \sim p_x(\mathcal{T}_j), i \neq j} \parallel F_c(X_i; \theta) - F_c(X_j; \theta) \parallel^2$$
$$= \mathbb{E}_{X_i \sim p_x(\mathcal{T}_i), X_j \sim p_x(\mathcal{T}_j), i \neq j} \parallel f_c^{(i)} - f_c^{(j)} \parallel^2 . \quad (7)$$

Finally, we use the Logarithm and Exponential transformations to combine L_{inner} and L_{cross}. These transformations can smoothly optimize ConditionNet to shorten the inner-task distance and enlarge the cross-task distance. When L_{inner} is small and L_{cross} is large, the combined L_{con} will be close to 0. The task contrastive loss is formulated as:

$$L_{con} = \ln(1 + e^{-L_{cross}}) + \ln(1 + e^{L_{inner}}). \quad (8)$$

During experiments, we find that if we only train ConditionNet by the task contrastive loss in an unsupervised way, the output feature may not be entirely beneficial for the generalization of SISR (as shown in Table VI). To make a balance between task-level feature extraction and SR reconstruction, we combine the reconstruction loss L_{res} in Eq. (5) and task contrastive loss L_{con} in Eq. (8) with coefficient λ to constraint ConditionNet, which is formulated as:

$$L_{com} = L_{con} + \lambda * L_{res} \quad (9)$$

D. CMDSR Algorithm

Training Stage. Algorithm 1 shows the training procedure of CMDSR model. ConditionNet and BaseNet are alternately trained until they converge. In line 4, k tasks are randomly sampled from degradation distribution $p(\mathcal{T})$ for each step. In Line 3-9, BaseNet is adapted and supervised with HR-LR pairs. In Line 10-18, for every t_0 steps, ConditionNet is optimized with the combined unsupervised L_{con} of Line 16 and supervised L_{res} of line 7.

Testing Stage. Algorithm 2 shows the training procedure of CMDSR model. For test support set X, we can randomly sample patches from other LR images, which have the same

---

Algorithm 1: CMDSR Training

Data: Distribution over tasks: $p(\mathcal{T})$
Input: ConditionNet and BaseNet parameters: ϕ, θ
Input: Task size: k, support size: n, update step: t_0,
loss coefficient: λ, learning rates: α, β

1 **for** $t = 1, 2, ...$ **do**
2   Randomly sample k tasks $\mathcal{T}_i \sim p(\mathcal{T})$;
3   **foreach** $\mathcal{T}_i$ **do**
4   Random crop n LR-HR patches from n LR-HR images of $\mathcal{T}_i$ as $(X_i, Y_i) \sim p(\mathcal{T}_i)$;
5   Use all n LR patches of X_i as the support set to extract conditional feature $f_c^{(i)}$:
       $f_c^{(i)} = F_c(X_i; \phi) = F_c(x_{i_1}, ..., x_{i_n}; \phi)$;
6   Compute adapted parameters θ_i' of BaseNet:
       $F_{sr}(\cdot\ ; \theta_i') = F_{sr}(f_c^{(i)}; \theta)$;
7   Evaluate the reconstruction loss in Eq. (5):
       $L_{res}^{(i)} = \sum_{j=1}^{n} L_1(F_{sr}(x_{i_j}; \theta_i'), y_{i_j})$;
8   **end**
9   Update BaseNet with reconstruction loss:
     $\theta \leftarrow \theta - \alpha \nabla_\theta \sum_{i=1}^{k} L_{res}^{(i)}$;
10   **if** $(t\ Mod\ t_0) = 0$ **then**
11   Resample n LR patches for k support sets from k tasks $\mathcal{T}_i$ of line 4: $X_i' \sim p_x(\mathcal{T}_i)$;
12   Resample n LR patches for k support sets from another k tasks $\mathcal{T}_j$: $X_j \sim p_x(\mathcal{T}_j), j \neq i$;
13   **foreach** $\mathcal{T}_i, \mathcal{T}_j, j \neq i$ **do**
14   Evaluate inner-task loss as Eq. (6):
         $L_{inner}^{(i)} = L_2(F_c(X_i; \theta), F_c(X_i'; \theta))$;
15   Evaluate cross-task loss as Eq. (7):
         $L_{cross}^{(i)} = L_2(F_c(X_i; \theta), F_c(X_j; \theta))$;
16   Evaluate task contrastive loss as Eq. (8):
         $L_{con}^{(i)} = L_{con}(L_{inner}^{(i)}, L_{cross}^{(i)})$;
17   **end**
18   Update ConditionNet with combined loss:
       $\phi \leftarrow \phi - \beta \nabla_\phi \sum_{i=1}^{k}(L_{con}^{(i)} + \lambda L_{res}^{(i)})$;
19   **end**
20 **end**

Algorithm 2: CMDSR Testing

Data: LR test image: $\mathbf{I}_{LR}$
Input: Trained parameters of ConditionNet F_c and BaseNet F_{sr}: ϕ, θ
Output: Restored SR image: $\mathbf{I}_{SR}$

1 Randomly crop n patches of I_{LR} as the support set X;
2 Extract conditional feature $f_c = F_c(X; \phi)$;
3 Compute adapted parameters θ' of BaseNet:
   $F_{sr}(\cdot\ ; \theta') = F_{sr}(f_c; \theta)$;
4 **return** $\mathbf{I}_{SR} = F_{sr}(\mathbf{I}_{LR}; \theta')$

---

degradation pattern with $\mathbf{I}_{LR}$, or from $\mathbf{I}_{LR}$ itself. For unknown LR in real scenario, it's hard to acquire other images which have the same degradation pattern. Therefore, we randomly crop patches from LR itself (self-patches) to get the support set at test time in all our experiments. With the conditional feature extracted from the support set, BaseNet performs fast adaptation to the test distribution in one step and produces the restored SR image.

IV. EXPERIMENTS

A. Experimental Setting

As introduced before, the input of CMDSR consists of two parts: the support set for the ConditionNet and the LR image for the BaseNet. When training, n random patches are cropped from n randomly sampled LRs of the same task and this process is conducted for k training tasks. Hence, during training, the sizes of support sets and LR images are set as $[k, n*3, h, w]$ and $[k, 3, h, w]$ separately, where size 3 indicates RGB channels. When testing, the full LR image $[1, 3, H, W]$ is fed into BaseNet and n random patches are cropped from LR itself to compose the self-support set $[1, n*3, h, w]$ for testing. All uniform random croppings allow for overlapping. In training configurations, we set the task size as $k = 8$ and the size of support set n to 20. The patch size $h \times w$ is 48×48. The update step t_0 is 10, which means that the ConditionNet is joined for training after the BaseNet has been trained for 9 steps. The loss coefficient λ of Eq. (9) is 0.1. The initial learning rates α, β of the BaseNet and ConditionNet are set to 10^{-3} and 10^{-4}. The ADAM optimizer [43] is applied.

We use the LR-HR pairs of DIV2K [46] for training. Following previous works [12], [15], the degraded LR images of different tasks are synthesized based on Eq. (1). As shown in Eq. 1, the multi-degradation applies a sequence of degrading effects on HR images, following the order of blurring, bicubic downsampling and then adding noise. To ensure consistency with previous works [12], [15], we use 15×15 isotropic Gaussian blur kernels and the Additive White Gaussian Noise (AWGN). The blur kernel widths σ_G are discrete in range of $[0.2, s]$ for scale factor s with a stride of 0.1. For noise, we set the AWGN with the continuous noise levels σ_N in range [0, 75]. For each iteration, the discrete kernel width and continuous noise level are randomly sampled. The total task number of our training data is infinite. We set the scale factor $s = 4$ and all the experiments were conducted on NVIDIA Tesla-V100 GPUs.

It should be noted that previous methods [12], [18] choose the isotropic and anisotropic Gaussian blur kernels for generating blur degradation. Since we only use isotropic Gaussian blur kernels, our training degradation patterns (σ_G, σ_N) are the subspace of $(\sigma_G, \sigma_G^{ani}, \sigma_N)$ in SOTAs. Therefore, the training sets of SOTAs can cover all degradation information of ours. Although this is "unfair" for our model, we want to highlight the ability of our model for degradation shift in this "unfair" setting and verify the effectiveness of our work.

B. Experiments on Synthetic Multi-degradated Images

To demonstrate the effectiveness and generalization of our framework for multi-degradations, we evaluate the proposed CMDSR from the perspectives of matched degradation and shift degradation. First, we use the *Simple* and *Middle* testing sets, which are in range of training distribution. Since our

TABLE I
AVERAGE PSNR VALUES WITH SCALE FACTOR ×4 ON *Simple*, *Middle* AND *Severe* DEGRADATIONS. WE USE THE PROVIDED OPENSOURCE CODES OF SOTA MODELS TO COMPUTE THEIR RESULTS, EXCEPT RESULTS OF IRCNN [44] AND UDVD [15], WHICH ARE DIRECTLY EXTRACTED FROM THEIR PUBLICATIONS. BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN RED AND BLUE.

Degradation	Kernel	Noise	Model Type	Models	Datasets		
					Set5	Set14	BSD100
(A) *Simple* Degradation In-Distribution	$\sigma_G^{0.2}$	σ_N^{15}	BI-structured SR model	RCAN+ [5]	24.90	23.87	23.42
			Blind multi-degraded SR model	ZSSR [17]	25.40	24.30	24.05
				IRCNN [44]	28.35	-	-
			Blind denoising/deblurring + Blind SR model	DnCNN [30]+KernelGAN [19]+ZSSR [17]	27.02	25.46	25.34
				DnCNN [30]+MZSR [26]	26.57	25.10	24.72
				DnCNN [30]+IDN [45]+MLSR [27]	26.32	25.14	24.63
				DnCNN [30] + IKC [18]	28.16	26.11	25.68
			Blind denoising+ Gt blur kernel maps+ Non-blind SR model	DnCNN [30] + SRMDNF [12]	28.31	26.19	25.79
			Gt Degradation maps+ Non-blind SR model	SRMD [12]	28.79	26.48	25.95
				UDVD [15]	29.04	26.82	26.08
			Conditional hyper-network	Our CMDSR	28.35	26.23	25.83
(B) *Middle* Degradation In-Distribution	$\sigma_G^{2.6}$	σ_N^{15}	BI-structured SR model	RCAN+ [5]	23.32	22.54	22.61
			Blind multi-degraded SR model	ZSSR [17]	24.91	23.74	23.57
				IRCNN [44]	24.36	-	-
			Blind denoising/deblurring + Blind SR model	DnCNN [30]+MZSR [26]	25.86	24.31	24.40
				DnCNN [30]+IDN [45]+MLSR [27]	25.89	24.44	24.32
				DnCNN [30]+KernelGAN [19]+ZSSR [17]	26.08	24.66	24.65
				DnCNN [30] + IKC [18]	26.84	25.09	25.02
			Blind denoising+ Gt blur kernel maps+ Non-blind SR model	DnCNN [30] + SRMDNF [12]	23.85	21.04	21.79
			Gt Degradation maps+ Non-blind SR model	SRMD [12]	26.82	25.12	24.86
				UDVD [15]	26.98	25.33	24.96
			Conditional hyper-network	Our CMDSR	27.10	25.39	25.12
(C) *Severe* Degradation Out-of-Distribution	σ_G^{ani}	σ_N^{50}	BI-structured SR model	RCAN+ [5]	16.32	15.83	15.81
			Blind multi-degraded SR model	ZSSR [17]	17.89	17.46	17.79
			Blind denoising/deblurring + Blind SR model	DnCNN [30]+MZSR [26]	21.73	20.80	21.49
				DnCNN [30]+IDN [45]+MLSR [27]	21.31	20.46	21.32
				DnCNN [30]+KernelGAN [19]+ZSSR [17]	22.32	21.69	22.34
				DnCNN [30] + IKC [18]	22.18	21.63	22.23
			Blind denoising+ Gt blur kernel maps+ Non-blind SR model	DnCNN [30] + SRMDNF [12]	21.63	21.18	21.99
			Gt degradation maps+ Non-blind SR model	SRMD [12]	22.43	21.83	22.43
			Conditional hyper-network	Our CMDSR	23.07	22.14	23.03

framework is trained with isotropic Gaussian blur kernel, we also add the *Severe* testing set with anisotropic Gaussian blur kernel to validate whether CMDSR can handle degradation shift. More precisely, three testing sets are synthesized as:

(A) *Simple*: 7×7 isotropic Gaussian blur kernel with kernel width $\lambda = 0.2$ followed by BI downsampler ($\sigma_G^{0.2}$) and AWGN with noise level 15 (σ_N^{15}).

(B) *Middle*: 7×7 isotropic Gaussian blur kernel with kernel width $\lambda = 2.6$ followed by BI downsampler ($\sigma_G^{2.6}$) and AWGN with noise level 15 (σ_N^{15}).

(C) *Severe*: 7×7 anisotropic Gaussian blur kernel with kernel width $\lambda_1 = 4$, $\lambda_2 = 1$, angle $\Theta = -0.5$ followed by BI downsampler (σ_G^{ani}) and AWGN with noise level 50 (σ_N^{50}).

The *Simple* and *Middle* are in distribution of training data and the *Severe* is out of distribution. We follow [12], [15] to set the degradation parameters of three testing sets for fair comparison with other SOTA methods for consistent comparison.

To be specific, we compare the proposed framework with non-blind and blind methods. For non-blind methods, we make comparison with two latest models, SRMD [12] and UDVD [15], which utilize the accurate blur kernel and noise maps as the additional inputs. For blind methods, the SOAT BI structured SR model, RCAN [5] is first compared. Since most blind SR methods for multiple degradations have not been studied sufficiently, except ZSSR [17] (5000 steps) and IR-CNN [44], we follow [12], [19], [21] to add cascaded schemes, which combine SR models with blind denoising and deblurring methods: DnCNN [30] + KernelGAN [19] + ZSSR [17] (5000 steps), DnCNN [30] + IKC [18]. Although two previous meta SR methods [26], [27] have limitations with large scale factor (×4), we still add them as DnCNN [30] + MZSR [26] (10 steps), DnCNN [30] + IDN [45] + MLSR [27] (10 steps). To evaluate the mutual negative influence between cascaded stages, we also add a baseline by combining blind denoiser and non-blind SR model, DnCNN [30] + SRMDNF [12].

1) Degradation-in-Distribution: The Section A and B of Table I show the PSNR values on *Simple* and *Middle* degradations, where degradation patterns match the range of training data. Due to the unawareness of multiple degradations, the
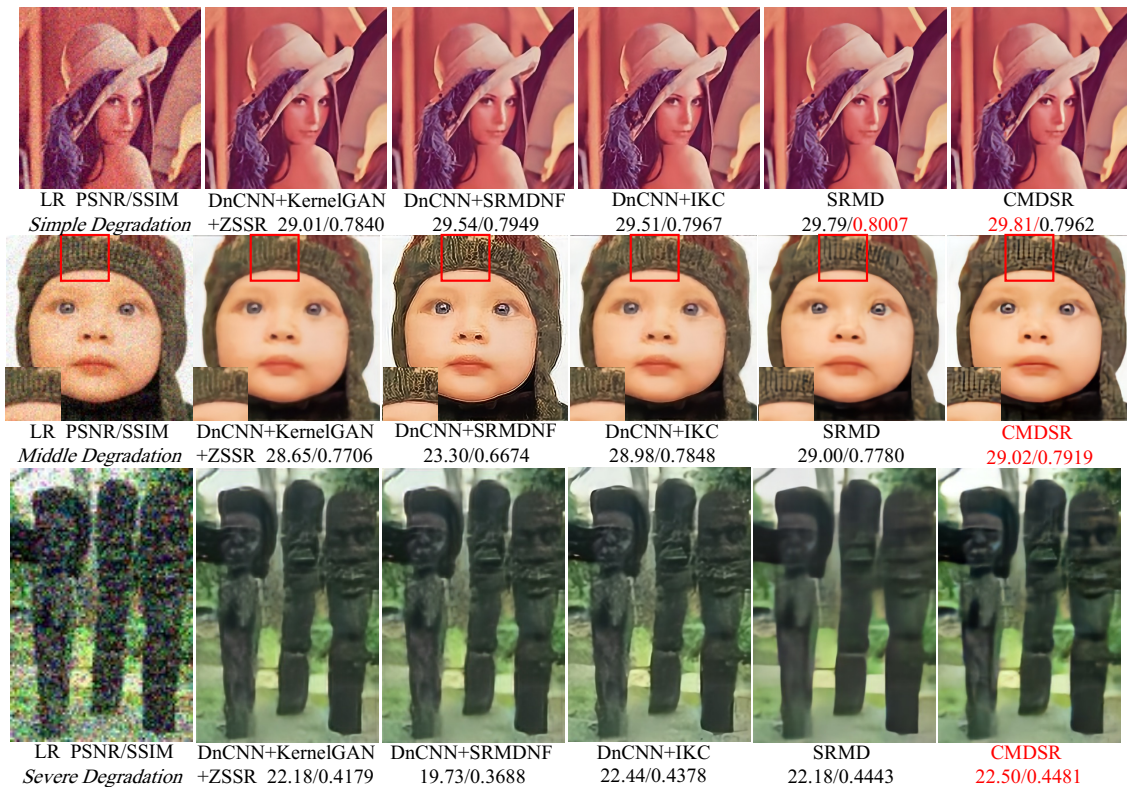
Fig. 4. SR perceptual results (×4) of images on *Simple*, *Middle* and *Severe* degradations. Best results are highlighted in red. For *Simple* degradation, CMDSR outperforms all the blind methods and is competitive with non-blind method, the SRMD. For *Middle* and *Severe* degradations, our CMDSR even outperforms the non-blind SRMD. Better to enlarge the images for more visual details.

BI structured model RCAN [5] produces the worse PSNR. When kernel width and noise level are increasing, the cascaded blind methods suffer the mutual negative influence between different stages. This is because the denoiser will make the LR image more blurred and lead to kernel mismatch, which increases the difficulty of deblurring. The severe performance drops of PSNR as shown in Table I and the over-sharp results of DnCNN [30] + SRMDNF [12] in Fig. 4 clearly illustrates this phenomenon. Our CMDSR achieves better PSNR than all blind schemes for *Simple* degradation, but a little lower than the non-blind methods, SRMD [12] and UDVD [15], because they take the accurate blur kernel and noise maps as the additional inputs. However, it is noted that when degradation is more complicated, the generalization of our adaptive framework becomes prominent. Our CMDSR achieves the best performance for *Middle* degradation, even better than the non-blind methods. As shown in Fig. 4, CMDSR produces sharper and clearer SR results. These results demonstrate that CMDSR is an effective blind framework for multiple degradations.

2) Degradation-Out-of-Distribution: The results in Section C of Table I show PSNR values on *Severe* degradation, where the degradation levels are higher and the blur kernel is out of the distribution of training data. The qualitative comparisons are presented in Fig. 1 and Fig. 4. Our CMDSR significantly outperforms all the blind and the non-blind methods, because the parameters of BaseNet are not fixed but adaptive for a new degradation when testing. It should be emphasized that non-blind SRMD [12] is trained with both isotropic and

TABLE II
AVERAGE PSNR VALUES WITH SCALE FACTOR ×4 ON NOISE-FREE DEGRADATIONS. BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN RED AND BLUE.

Models	Kernel	Datasets		
		Set5	Set14	BSD100
ZSSR		28.87	27.15	26.68
SRMDNF		31.96	28.35	27.49
IKC	$\sigma_G^{0;2}$	32.39	28.77	27.58
UDVD		32.31	28.78	27.70
CMDSR		31.82	28.53	27.60
ZSSR		27.69	26.06	25.92
SRMDNF		31.77	28.26	27.43
IKC	$\sigma_G^{2;6}$	32.05	28.55	27.47
UDVD		31.99	28.55	27.55
CMDSR		31.79	28.36	27.50
ZSSR		26.24	25.33	25.02
SRMDNF		30.48	27.23	26.85
IKC	σ_G^{ani}	30.51	27.31	26.70
CMDSR		29.94	27.29	26.95

anisotropic Gaussian blur kernel, while our blind CMDSR trained with isotropic blur kernels still achieves better qualitative and quantitative results. These results further demonstrate the generalization of our CMDSR to handle distribution shifts.

C. Experiments on Synthetic Noise-free Images

As we have explained before, although existing methods can perform well on single degradation (*eg.* noise-free), they are difficult to deal with multiple degradations, and the multiple degradations are more complicated in real tasks. However, it
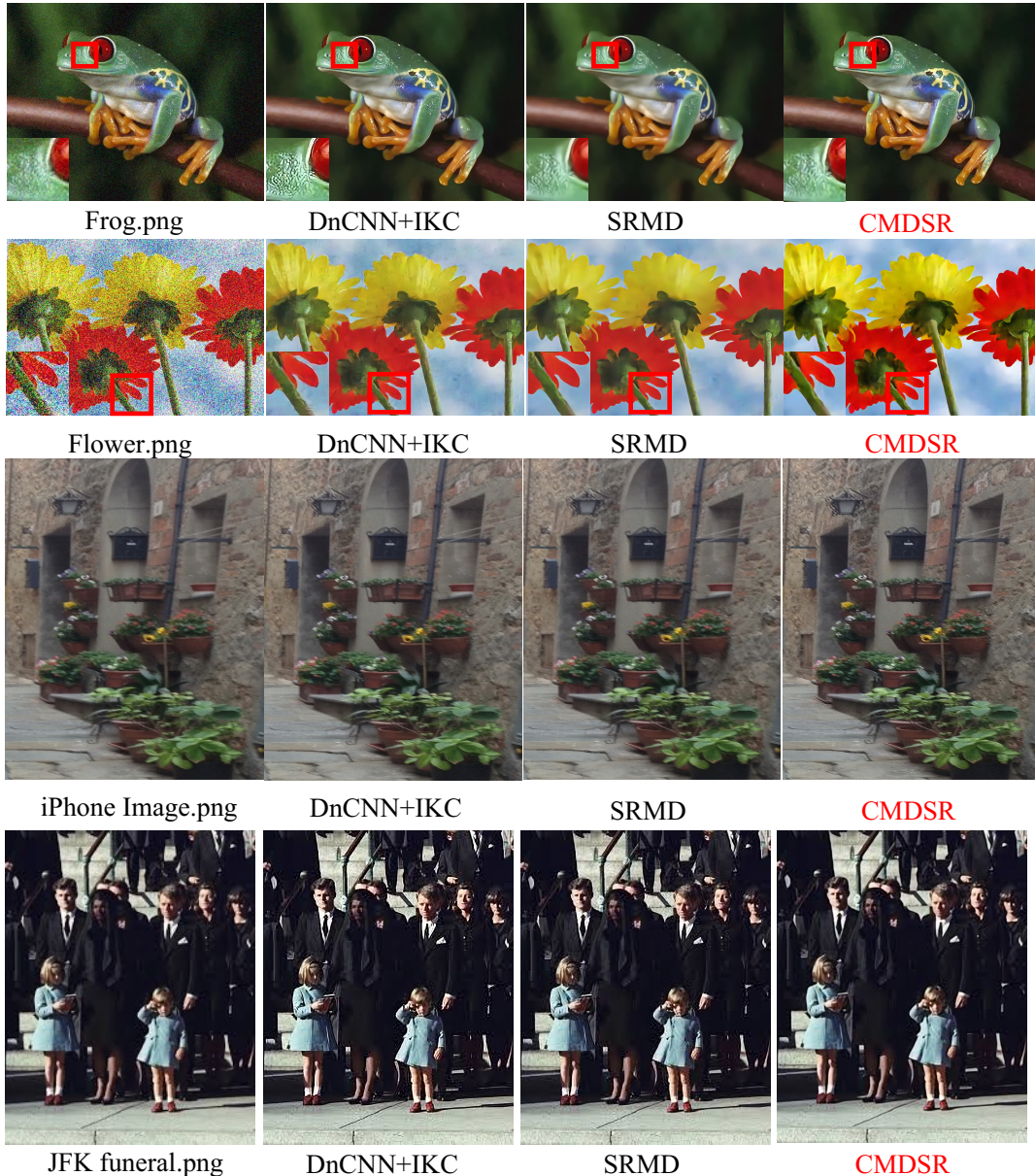
Fig. 5. SR perceptual results (×4) of real-captured images. Best results are highlighted in red. Our CMDSR produces best results with **less artifacts and brighter color**. Better to enlarge the images for more visual details.

is still a valid assessment on noise-free degradation. Here, we present the experimental results on noise-free degradation in Table II. Compare to well-designed SOTAs, such as blind IKC and non-blind UDVD, our CMDSR achieves little worse performance. When degradation is more complicated and out of distribution, the performance of our CMDSR can get improved, which also proves the adaptation of our model on noise-free degradation. It should be noted that since different degradation patterns are mixed together, it's a tough problem for existing blind SR models to conduct accurate degradation estimation on multi-degradations. Therefore, the superiority of adapted our CMDSR are more significant on multi-degraded images.

D. Perceptual Visualization on Real Images

We further extend the experiments to real images. The most representative blind and non-blind methods, DnCNN [30] + IKC [18] and SRMD [12] are compared with our framework. Since there are no GT degradation patterns for real images, SRMD [12] is searched by manual grid as in [12]. The qualitative results of real images [12], [47] are shown in Fig. 5. It is clear that the blind scheme produces over-sharp results and non-blind SRMD [12] fails to recover sharp edges. Overall, CMDSR produces the best results with less artifacts, sharper edges, and even brighter color.

E. Ablation Experiments

In this section, we present the systematic ablation studies to explain the important implementation details and validate the

effectiveness of our framework. For fair comparison, all the ablation studies were conducted with training data and settings as described in Section IV-A.

1) Computational and Parameter Comparison: We present the SOTA blind and non-blind SR models, RCAN [5], ZSSR [17], DnCNN [30] + KernelGAN [19] + ZSSR [17] (5000 steps), DnCNN [30] + IKC [18], SRMD [12], UDVD[1] [15], SRResNet-10 and CMDSR to conduct experiments. The model parameters, average inference time and PSNR results for $\times 4$ SISR of Set5 with *Middle* Degradation are listed in Table III. It is clear that our proposed CMDSR can achieve better results with fast speed and relatively less parameters.

2) BaseNet w/ and w/o ConditionNet: We first evaluate the performance of BaseNet with and without ConditionNet to show the importance of conditional feature. Although ConditionNet is not directly used for SISR, it involves more parameters. For a fair comparison, we add SRResNet-16, where the number of parameters nearly equals to the completed CMDSR. The SRResNet-16 is then trained with the same synthetic data as ours. As shown in Table IV, the PSNR result of CMDSR is much better, which demonstrates the significance of conditional hyper-network for unsupervised task-level feature extraction.
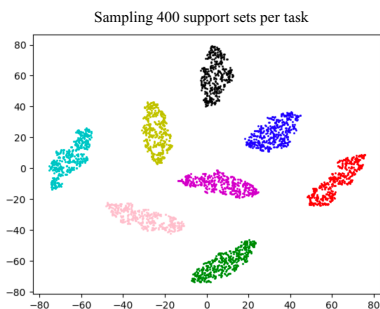

Fig. 6. The t-SNE visualization of modulated conditional features. Different colors represent different tasks.

3) Visualizations of Conditional Features: To show ConditionNet can efficiently extract task-level features, we compare the conditional features between inner-tasks and cross-tasks. Using DIV2K validation set, we randomly sample 8 different tasks and sample 400 support sets for each task. We choose the conditional features modulated by the first modulation layer and show the t-SNE [48] visualizations in Fig. 6. It is clear that the modulated features of inner-tasks are similar and those from cross-tasks are significantly different, which is consistent with the prior knowledge in Fig. 2.

4) Stability with Different Support Sets: Our framework aims to extract the common degradation patterns from different LR images of the same task. Therefore, the conditional features should be content-invariant. To demonstrate this, we conduct experiments to evaluate the stability of SR results with different support sets. As shown in Fig. 7, we evaluated the PSNR values of *baby*.png using different support patches
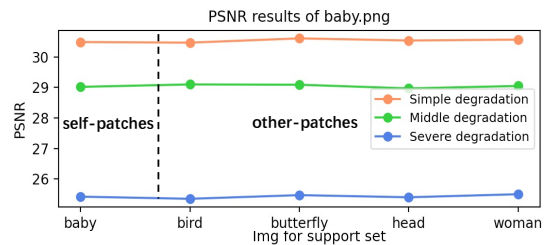

Fig. 7. PSNR results of *baby*.png using different support patches of Set5.

of Set5. We randomly cropped the self-support-patches from *baby*.png itself and other-support-patches from other LR images of Set5, separately. The self-support-patches and other-support-patches from the same task share the same degradation but have different contents. As shown in Fig. 7, it is clear that our framework achieves relatively stable performance. This further verifies that our model can extract degradation prior from different image contents at task-level.
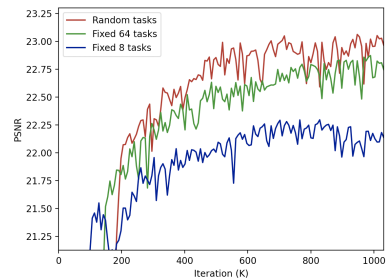

Fig. 8. PSNR curves comparison of *Severe* degradation with fixed 8 tasks, 64 tasks and random infinite tasks.

5) The Importance of Randomly Selected Tasks from A Infinite Degradation Pool : Why we use the random selected tasks from the infinite degradation pool for each iteration, but not fix them from a specific degradation pool at the beginning of training? We conduct experiments by using fixed 8 tasks, 64 tasks and random infinite tasks. The curves of PSNR results with *Severe* degradation are shown in Fig 8. As introduced in the experimental setting, the degradation distributions $p(\mathcal{T})$ of meta-training data is randomly sampled. For each training iteration with scale factor s, we sample k tasks, where the blur kernel widths σ_G are in the range of [0.2, s] and the noise levels σ_N are in the range of [0, 75]. Some previous meta-network frameworks [37], [39] fix the tasks at the beginning of training and feed all the tasks for one iteration. However, our framework cannot use the fix tasks because of the following two reasons: (1) we cannot define a certain number of tasks because the degradations of LR images are various and infinite; and (2) the fixed tasks will cause overfitting, which limits the generalization of CMDSR. As shown in Fig 8, we can observe that both models trained with fixed tasks get worse performance with distribution shifts. Moreover, when using 8 tasks, the curve of training loss even shows performance drop. The model trained with random tasks achieves the best performance at shifted degradation, which indicates that the setting of random tasks is significant for our framework.

TABLE III
COMPARISONS ON MODEL PARAMETERS, AVERAGE INFERENCE TIME AND PSNR RESULTS FOR ×4 SISR OF SET5 WITH *Middle* DEGRADATION. IT IS
CLEAR THAT OUR PROPOSED CMDSR CAN ACHIEVE BETTER RESULTS WITH FAST SPEED AND RELATIVELY LESS PARAMETERS.

Model	RCAN+ [5]	ZSSR [17] (5000 steps)	DnCNN [30] + KernelGAN [19] + ZSSR [17] (5000 steps)	DnCNN [30] + IKC [18] (10 iterations)
Params.	15.6M	0.23M	1.08M	2.16M
Sec.	507.32ms	183.85s	213.93s	780.40ms
PSNR	23.32	24.91	26.08	26.84

Model	SRMD [12]	UDVD [15]	SRResNet-10	CMDSR (SRResNet-10 + ConditionNet)
Params.	1.55M	4.83M	1.04M	1.48M
Sec.	6.28ms	42.72ms	29.84ms	39.19ms
PSNR	26.82	26.98	26.09	27.10

TABLE IV
AVERAGE ×4 PSNR OF SRRESNET-10, SRRESNET-16 AND
SRRESNET-10+CONDITIONNET ON SET5 WITH *Middle* DEGRADATION.

Model	Parameters	PSNR
SRResNet-10 w/o ConditionNet	1.04M	26.09
SRResNet-16 w/o ConditionNet	1.48M	26.62
SRResNet-10 w/ ConditionNet (Our CMDSR)	1.46M	27.10

6) The Number of Task Size, Support Size and Patch Size:
To evaluate the influence of task size k, support size n
and patch size $h \times w$, we conduct experiments by changing
the number of task size ($k = 2, 4, 8$), support size ($n = 1, 10, 15, 20, 25, 30$) and patch size ($h \times w = 8 \times 8, 32 \times 32, 48 \times 48$).

First, we fix the support size and increase the task size. The
number of task size defines the number of tasks which are
used for each training step. When we use a larger task size,
more tasks will be fed into CMDSR for each iteration. As
Table V shows, the CMDSR benefits from larger training task
size.

TABLE V
AVERAGE ×4 PSNR ON SET5 WITH *Middle* DEGRADATION, WHERE
CMDSR IS TRAINED WITH DIFFERENT TASK SIZES.

Model	Task Size	Support Size	PSNR
CMDSR	2	20	26.96
	4	20	27.03
	8	20	27.10

Then, we fix the task size and change the support size
and patch size. The number of support size defines the
number of LR images of the support set for each task. As
Fig. 9 shows, the CMDSR benefits from larger support size
and larger patch size. According to the network structure of
ConditionNet, the number of support size also defines the input
channel of ConditionNet. Larger support size and larger patch
size is beneficial for ConditionNet to extract the degradation
information more accurately from LR patch samples at task
level. And our settings (support set size=20, patch size=48) are
the most appropriate because further larger sizes bring only
marginal improvements.

7) Combination of Loss Functions: We compare the results
of CMDSR, where ConditionNet is separately trained with task
contrastive loss L_{con}, reconstruction loss L_{res} and combined
loss. The BaseNet is still optimized by the reconstruction
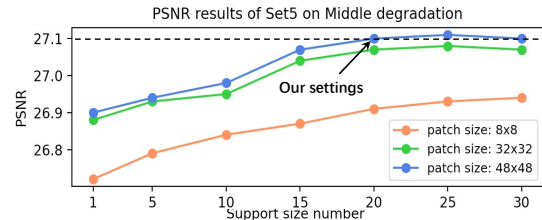loss L_{res} for SR. As shown in Table VI, if ConditionNet is

Fig. 9. PSNRs on *Middle* Set5 using different support sizes and patch sizes.

TABLE VI
AVERAGE ×4 PSNR ON SET5 WITH *Middle* DEGRADATION, WHERE
CONDITIONNET IS SEPARATELY TRAINED WITH THREE LOSSES.

Model	Loss of BaseNet	Loss of ConditionNet	PSNR
ConditionNet + BaseNet (CMDSR)	L_{res}	L_{res}	23.36
	L_{res}	L_{con}	26.84
	L_{res}	$L_{con} + \lambda * L_{res}$	27.10

trained with L_{res}, CMDSR gets collapsed to produce even
worse results than the single BaseNet. Only when using the
unsupervised L_{con}, the result is acceptable, but using the
combined loss achieves the best results; This is because it
makes a balance between task-level feature extraction and the
generalization of SISR.

*8) The Effect of Contrastive Loss and Reconstruction Loss
Ratio:* How to balance the effect of Contrastive Loss and
Reconstruction Loss to better optimize the ConditionNet?
We conduct experiments by changing the loss coefficient
($\lambda = 1, 0.1, 0.01$). The PSNR results with *Middle* degradation
are shown in Table VII. It is observed that the large ratio
of Reconstruction loss ($\lambda = 1$) is harmful for the task-level
feature extraction of ConditionNet. When we decrease the ratio
($\lambda = 0.01$), the performance of CMDSR is slightly decreased.
Therefore, we set the loss coefficient λ as 0.1.

TABLE VII
AVERAGE ×4 PSNR ON SET5 WITH *Middle* DEGRADATION, WHERE
CONDITIONNET IS SEPARATELY TRAINED WITH DIFFERENT LOSS
COEFFICIENTS.

Model	Combined Loss of ConditionNet	coefficient	PSNR
CMDSR	$L_{con} + \lambda * L_{res}$	$\lambda = 1$	25.27
		$\lambda = 0.1$	27.10
		$\lambda = 0.01$	27.03

9) The Importance of Alternating Training Strategy: As we have explained before, the ConditionNet and BaseNet serve different purposes with different learning rates and loss functions. Here, we have conducted the experiment to show the importance of alternating training strategy with different steps t_0. Specifically, we alternately optimize the BaseNet and ConditionNet with different steps ($t_0 = 0, 1, 10, 20$). The results are presented in Table VIII. It can be seen that the alternative training strategy is quite important, since non-alternative training $t_0 = 0$ produces the worst performance. We speculate that this is caused by the inconsistent convergence speed of the ConditionNet and Basenet. To make a balance between the conditional feature extraction and SR restoration, we experimentally set the alternating training step $t_0 = 10$ to train ConditionNet and BaseNet alternatively.

TABLE VIII
AVERAGE ×4 PSNR ON SET5 WITH *Middle* DEGRADATION USING DIFFERENT ALTERNATING TRAINING STEP t_0.

Model	Alternating training step t_0	PSNR
CMDSR	0	26.86
	1	27.05
	10	27.10
	20	27.01

10) Can CMDSR Use Other Modulation?: Since our model aims to be a general framework, we use depth-wise scaling to adapt convolutions without changing the CNN structures or involving more network parameters. Hence, we applied the depth-wise scaling as our modulation. Here, we extend to use the novel Spatial Feature Transform (SFT) of IKC [18] as the modulation. As shown in Table IX, CMDSR can further benefit from SFT which concatenates the degradation and image features to apply an affine transformation in each middle layer. However, after using the SFT modulations, the parameters of CMDSR get greatly increased. With sufficient calculation resources, we believe more novel modulations can be applied to our framework in the future.

TABLE IX
AVERAGE ×4 PSNR ON SET5 WITH *Middle* DEGRADATION USING DIFFERENT MODULATIONS.

Model	Modulation	PSNR	Params.
CMDSR	depth-wise scaling	27.10	1.48M
	SFT [18]	27.53 ↑	5.92M

11) Can CMDSR Extend to Other SISR Structure?: As mentioned before, since our model uses depth-wise scaling to adapt convolutions without changing CNN structures, it has no strict restrictions on the BaseNet structure. Therefore, we attempt to extend our framework to other SISR structure. Specifically, we replace SRResNet-10 with other three SISR models, VDSR [2], EDSR [4] and IDN [45]. Moreover, we also use the same training data described in Section IV-A to train these models without ConditionNet. As shown in Table X, all joint models obtain significant improvement, and EDSR [5] achieves the best results with the largest number of parameters. We believe that our framework can be extended to more complicated structures in the future.

TABLE X
AVERAGE ×4 PSNR ON SET5 WITH *Middle* DEGRADATION USING OTHER STRUCTURES FOR BASENET.

Model	BaseNet Parameters	PSNR
VDSR [2] w/o ConditionNet	0.67M	26.49
VDSR [2] w/ ConditionNet		26.97↑
IDN [45] w/o ConditionNet	0.80M	26.53
IDN [45] w/ ConditionNet		27.03↑
EDSR [4] w/o ConditionNet	43M	26.81
EDSR [4] w/ ConditionNet		27.51↑

V. CONCLUSION

In this paper, we investigate the blind SISR problem with multiple degradations. Inspired by meta-learning, we design the first blind SR hyper-network that learns how to adapt to changes in input distribution. Specifically, we use a ConditionNet to extract the task-level features with batches of LR patches and use BaseNet to rapidly adapt its parameters according to the conditional features. Extensive experiments show that our framework can handle distribution shift by only performing one-step adaptation. For complicated and real scenes, our blind model even outperforms several non-blind models. Besides, our framework is flexible and efficient so that we can extend the structure of BaseNet to other SISR models. In future work, we will extend our general framework to more CNN models and more low-level vision tasks.

VI. ACKNOWLEDGEMENT

REFERENCES

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015. 1, 2

[2] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654. 1, 2, 4, 11

[3] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1637–1645. 1, 4

[4] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144. 1, 2, 4, 11

[5] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301. 1, 2, 4, 6, 7, 9, 10, 11

[6] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690. 1

[7] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0. 1

[8] C. Tian, Y. Xu, W. Zuo, B. Zhang, L. Fei, and C.-W. Lin, "Coarse-to-fine cnn for image super-resolution," *IEEE Transactions on Multimedia*, 2020. 1

[9] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. The MIT Press, 2009. 1

[10] D. Lazer, R. Kennedy, G. King, and A. Vespignani, "The parable of google flu: traps in big data analysis," *Science*, vol. 343, no. 6176, pp. 1203–1205, 2014. 1

[11] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," in *European conference on computer vision*. Springer, 2014, pp. 372–386. 1, 3

[12] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3262–3271. 1, 2, 5, 6, 7, 8, 9, 10

[13] G. Riegler, S. Schulter, M. Ruther, and H. Bischof, "Conditioned regression models for non-blind single image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 522–530. 1

[14] K. Zhang, W. Zuo, and L. Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1671–1681. 1

[15] Y.-S. Xu, S.-Y. R. Tseng, Y. Tseng, H.-K. Kuo, and Y.-M. Tsai, "Unified dynamic convolutional network for super-resolution with variational degradations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 496–12 505. 1, 2, 5, 6, 7, 9, 10

[16] T. Michaeli and M. Irani, "Nonparametric blind super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 945–952. 1, 2

[17] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3118–3126. 1, 2, 3, 6, 9, 10

[18] J. Gu, H. Lu, W. Zuo, and C. Dong, "Blind super-resolution with iterative kernel correction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 1604–1613. 1, 2, 5, 6, 8, 9, 10, 11

[19] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-gan," in *Advances in Neural Information Processing Systems*, 2019, pp. 284–293. 1, 2, 6, 9, 10

[20] P. Liu, H. Zhang, Y. Cao, S. Liu, D. Ren, and W. Zuo, "Learning cascaded convolutional networks for blind single image super-resolution," *Neurocomputing*, vol. 417, pp. 371–383, 2020. 1, 3

[21] S. Maeda, "Unpaired image super-resolution using pseudo-supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 291–300. 1, 2, 3, 6

[22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232. 1

[23] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a gan to learn how to do image degradation first," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 185–200. 1, 3

[24] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 701–710. 1

[25] Z.-S. Liu, W.-C. Siu, L.-W. Wang, C.-T. Li, and M.-P. Cani, "Unsupervised real image super-resolution via generative variational autoencoder," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 442–443. 1

[26] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3516–3525. 1, 3, 6

[27] S. Park, J. Yoo, D. Cho, J. Kim, and T. H. Kim, "Fast adaptation to super-resolution networks via meta-learning," in *16th European Conference on Computer Vision, ECCV 2020*, 2020, pp. 754–769. 1, 3, 6

[28] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742. 2, 4

[29] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738. 2, 4

[30] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. 2, 6, 7, 8, 9, 10

[31] Y. Lee and S. Choi, "Gradient-based meta-learning with learned layerwise metric and subspace," in *International Conference on Machine Learning*, 2018, pp. 2927–2936. 3

[32] H. Yao, X. Wu, Z. Tao, Y. Li, B. Ding, R. Li, and Z. Li, "Automated relational meta-learning," in *International Conference on Learning Representations*, 2019. 3

[33] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3630–3638, 2016. 3

[34] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4080–4090. 3

[35] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208. 3

[36] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," *arXiv preprint arXiv:1606.04474*, 2016. 3

[37] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017, pp. 1126–1135. 3, 9

[38] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412. 3

[39] M. Zhang, H. Marklund, A. Gupta, S. Levine, and C. Finn, "Adaptive risk minimization: A meta-learning approach for tackling group shift," *arXiv preprint arXiv:2007.02931*, 2020. 3, 9

[40] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, 2016, pp. 1842–1850. 3

[41] T. Munkhdalai and H. Yu, "Meta networks," in *International Conference on Machine Learning*, 2017, pp. 2554–2563. 3

[42] B. N. Oreshkin, P. Rodriguez, and A. Lacoste, "Tadam: task dependent adaptive metric for improved few-shot learning," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 719–729. 3

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015. 5

[44] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3929–3938. 6

[45] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 723–731. 6, 11

[46] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135. 5

[47] M. Lebrun, M. Colom, and J.-M. Morel, "The noise clinic: a blind image denoising algorithm," *Image Processing On Line*, vol. 5, pp. 1–54, 2015. 8

[48] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008. 9