KNOWLEDGE DISTILLATION WITH MULTI-GRANULARITY MIXTURE OF PRIORS FOR IMAGE SUPER-RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation (KD) is a promising yet challenging model compression technique that transfers rich learning representations from a well-performing but cumbersome teacher model to a compact student model. Previous methods for image super-resolution (SR) mostly are tailored to the specific teacher-student architectures. And the potential for improvement is limited, which hinders their wide applications. This work presents a novel KD framework for SR models, the multi-granularity mixture of prior knowledge distillation (MiPKD), that is universally applicable to a wide array of architectures at feature and block levels. The teacher's knowledge is effectively integrated with the student's feature via the Feature Prior Mixer, and the reconstructed feature propagates dynamically in the training phase with the Block Prior Mixer. Extensive experiments demonstrate the effectiveness of the proposed MiPKD method.

1 INTRODUCTION

Super-resolution (SR) is a fundamental yet challenging task in the field of computer vision (CV), restoring high-resolution (HR) images from their low-resolution (LR) counterparts (Dong et al., 2015; Liang et al., 2021; Chen et al., 2021). In the past decade, the convolutional neural network (CNN) (Dong et al., 2014; Kim et al., 2016; Lim et al., 2017) and the Transformer (Chen et al., 2021; Liang et al., 2021; Wang et al., 2022c; Zamir et al., 2022) have demonstrated exceptional success for SR. However, it is impractical to directly deploy these models on resource-limited devices due to their heavy computation overload (Zhang et al., 2021b). Consequently, there has been a growing interest in model compression for SR models to facilitate their real-world applications.

Knowledge distillation, emerging as an effective model compression method, can significantly reduce computation overload, facilitating the student by transferring dark knowledge from the well-performed but cumbersome teacher model to the compact student model (Zhang et al., 2021a; Luo et al., 2021; Hui et al., 2019; Lee et al., 2020). Compared with other model compression techniques, such as quantization (Li et al., 2020; Hong et al., 2022; Ma et al., 2019), pruning (Wang et al., 2021a;b), compact block design (Ahn et al., 2018; Song et al., 2021; Nie et al., 2021; Wang et al., 2022a), and neural architecture search (NAS) (Zoph & Le, 2016; Wan et al., 2020; Ren et al., 2021), KD is a widely recognized method that can be combined with these techniques to further improve the compactness of the student model. KD for SR has also attracted wide attention recently and has gained remarkable progress (Li et al., 2020; Lee et al., 2020; Zhang et al., 2021a; He et al., 2020; Wang et al., 2021b). These methods can be roughly classified into the response-based KD and the feature-based KD, the former uses the output of the teacher model to supervise the student model, while the latter aligns the hidden representations between the teacher model and the student model (Gou et al., 2021; Wang et al., 2021b; He et al., 2020).

Although previous KD methods show promising results in SR, several issues hinder their wide applications. First, existing KD techniques for SR are tailored to specific teacher-student architectures. They support network depth (Figure 1(a)) or network width (Figure 1(b)) compression (He et al., 2020), and deteriorate the student dramatically when they are adopted into another setting. For instance, FAKD (He et al., 2020) boosts the student model in depth compression but deteriorates the student when applied to a width compression circumstance. CSD (Wang et al., 2021b) improves the

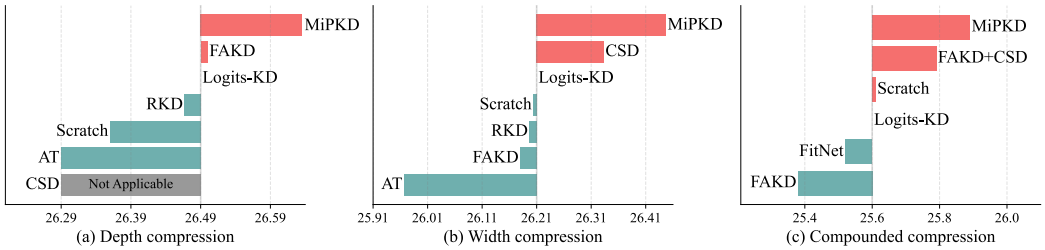


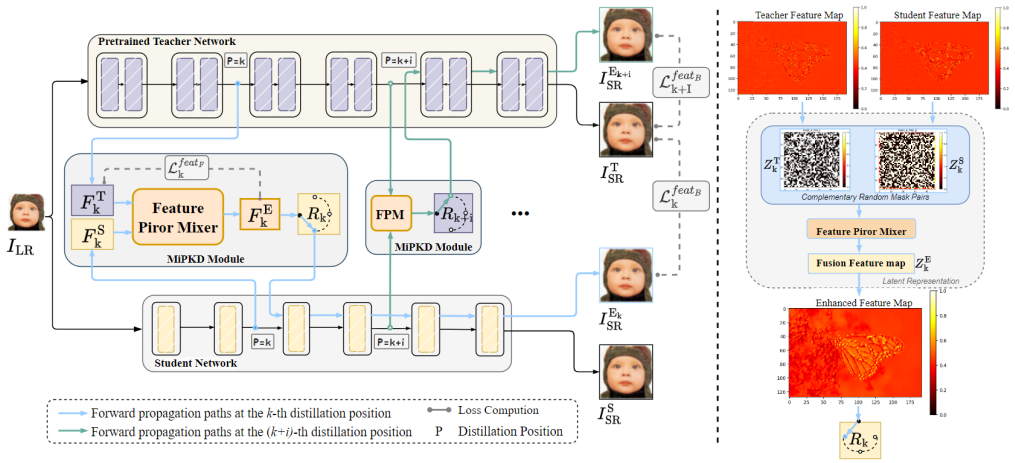
Figure 1: The PSNR of student models on Urban100 testset under different compression settings. In the depth compression (a), there are barely KD methods outperforming vanilla logits-KD. For width compression (b), CSD performs well but only satisfies this setting. For compounded compression, almost all KD underperforms training without KD.

student model significantly (Figure 1(b)) but is not compatible with depth compression in Figure 1 (a). It's necessary to propose a more flexible KD framework which is closer to real-world application. While few methods have discussed compounded compression on both depth and width dimensions, which is a much more general but challenging scenario. The existing KD methods for SR and those feature-based methods introduced from high-level CV, *e.g.* RKD (Park et al., 2019), AT (Zagoruyko & Komodakis, 2016), and FitNet (Romero et al., 2014) hardly benefit the student model. Figure 1 shows that the previous depth and channel distillation methods can just obtain a marginal performance gain or even deteriorate the student in most cases. To alleivate these issues, in this paper, we present a novel knowledge distillation framework for SR models, the multi-granularity mixture of prior knowledge distillation (MiPKD), that is universally applicable to a wide array of teacher-student architectures at feature and block levels. Specifically, the feature prior mixer dynamically combines priors from the teacher and student models' intermediate feature maps. Then its output enhanced feature map is supervised by the teacher model's feature map. The block prior mixer adopts a coarser-grained prior mixture at the network block level that dynamically and stochastically switches the normal forward propagation path to the teacher or the student. The output SR image of this ensembled sub-network is supervised by the teacher's output. In summary, the main contributions of this paper are as follows:

- We present MiPKD, a KD framework for efficient SR, transferring the teacher model's prior knowledge from both network width and depth levels. It's flexible and applicable to a wide array of teacher-student architectures.
- We propose the feature and block prior mixers to reduce the capacity disparity between teacher and student models for better alignment. The former combines the feature maps in a unified latent space, while the latter assembles dynamic combination of network blocks from teacher and student models.
- Extensive experiments on various benchmarks show that the proposed MiPKD framework significantly outperforms the previous arts.

2 Related Work

Deep SISR Models. Deep neural networks (DNNs) have made significant strides for image super-resolution (SR). Dong et al. (2014) introduced a convolutional neural network (CNN) architecture comprised of just three CNN layers for SR task. This was further advanced by (Kim et al., 2016) with the introduction of residual learning in the "Very Deep Super Resolution" (VDSR) model, which greatly expanded the architecture. Building on this foundation, Lim et al. (2017) proposed the "Enhanced Deep Super-Resolution Networks" (EDSR) model where the convolution layer is replaced with a simpler design. Zhang et al. (2018) then proposed the Residual Channel Attention Network (RCAN), which was much deeper than previous works. Recently, there has been a surge of interest in utilizing Transformers for image restoration tasks. Liang et al. (2021) introduced the Swin Transformer to low-level CV tasks, facilitating the deep feature extraction. Moreover, Zamir et al. (2022) proposed Restormer, which features a multi-scale hierarchical architecture that efficiently optimized the self-attention and MLP modules. Similarly, Wang et al. (2022c) introduced the Uformer model, which utilizes a LeWin Transformer block tailored for image restoration tasks. Despite the remarkable performance of both CNNs and Transformers in single image super-resolution (SISR),



(a) Overall framework of MiPKD (b) Feature Prior Mixer

Figure 2: Framework of the MiPKD method. MiPKD utilizes the multi-granularity prior mixture to constrain the KD process. At the k and $(k+i)$-th distillation position, the feature prior mixer dynamically combines priors from the teacher and student model, and the block prior mixer adopts a coarser-grained prior mixture at the network block level.

these approaches are often hindered by substantial memory requirements and computational overhead.

Efficient SISR. To improve the model efficiency, there have been various approaches to make the SR model less redundant, such as neural architecture search (NAS) (Chu et al., 2021; Song et al., 2020), pruning (Wang et al., 2021a;b), low-bit quantization (Ma et al., 2019; Li et al., 2020; Hong et al., 2022), and compact net block design (Ahn et al., 2018; Song et al., 2021; Nie et al., 2021; Wang et al., 2022a;c; Zamir et al., 2022). The strength of NAS manifests in searching the optimal architecture but is time-consuming and computationally expensive due to the massive search space. Afterwards, compact SR model designs have attracted rising attention and achieved remarkable progress (Zhang et al., 2022; Hui et al., 2019; Ahn et al., 2018; Dong et al., 2016). ELAN, proposed by Zhang et al. (2022), incorporates the GMSA module that effectively exploits long-range image dependencies and achieves superior performance compared to transformer-based super-resolution models while being much less complex. Pruning (Wang et al., 2021a;b) and quantization (Ma et al., 2019; Li et al., 2020; Hong et al., 2022) are other two types of methods to remove model redundancy by sparsity and low-bit quantization mappings. Despite the considerable progress made by these lightweight networks, significant computational resources are still demanded.

Knowledge Distillation for SISR. Knowledge distillation is widely recognized as an effective neural network compression technique that is able to significantly reduce the computation overload and improve student's capability by transferring dark knowledge from the large teacher model to the lightweight student model (Gou et al., 2021; Yim et al., 2017; Hinton et al., 2015). Recently, several attempts have also been made for image super-resolution knowledge distillation. Lee et al. (2020) employ an trainable encoder-decoder network to perform information extraction, and use the statistics computed from the scale maps of the decoder to distill student models. He et al. (2020) proposed FAKD to distill the correlation information from the affinity matrix of feature maps. Wang et al. (2021b) proposed CSD that incorporates self-distillation and contrastive learning by introducing extra simply upsampled LR images as negative samples. However, none of the existing SRKD methods have discussed how to customize a proper teacher for a student with limited capacity or whether a stronger teacher consistently benefits the capacity-limited student. MTKDSR (Yao et al., 2022) employed two teacher models with different SR objectives (PSNR, perceptual) to guide the student model simultaneously. CrossKD (Fang et al., 2023) divides the teacher and student networks into two segments that are interchanged and connected to perform forward propagation. Furthermore, existing SRKD techniques for SR are tailored to specific teacher-student architectures, focusing on either network depth (Wang et al., 2021b) or channel compression (He et al., 2020), which is infeasible for practical compounded compression applications.

3 METHODOLOGY

3.1 PRELIMINARIES AND NOTATIONS

Given a low-resolution input image I_{LR}, the deep SR model $\mathcal{F}(\cdot)$ aims to reconstruct the high-resolution image $I_{SR} = \mathcal{F}(I_{LR}; \Theta)$ with fine details and consistent content with corresponding high-resolution image I_{HR}, where Θ denotes the model parameters. The logits-based KD method compels the student model $\mathcal{F}_S$ to produce the same prediction as the teacher model

$$\mathcal{L}_{logits} = \mathcal{D}_{logits}(I_{SR}^S,\ I_{SR}^T) \tag{1}$$

where $I_{SR}^S = \mathcal{F}_S(I_{LR}; \Theta^S)$ and $I_{SR}^T = \mathcal{F}_T(I_{LR}; \Theta^T)$ represent the output SR images of the student and teacher models, and $\mathcal{D}_{logits}$ is the loss function that measures the difference between two models' outputs, *e.g.* the L_1 or L_2 loss function. Similarly, the feature-based KD methods aim to mimic the rich implicit hidden representations between the teacher and the student, which also can be represented by the feature distillation loss

$$\mathcal{L}_{feat} = \mathcal{D}_{feat}(\mathcal{T}_s(\mathrm{F}_k^S),\ \mathcal{T}_t(\mathrm{F}_k^T)) \tag{2}$$

where F_k^S and F_k^T denote the feature maps of the student model and the teacher model at the k-th distillation position, respectively. $\mathcal{T}_t$ and $\mathcal{T}_s$ are the transformations applied on raw feature maps and $\mathcal{D}_{feat}$ is the loss function for feature distillation.

3.2 MIXTURE OF PRIOR KNOWLEDGE DISTILLATION

Inspired by MAE (He et al., 2022) that reconstructs the missing pixels from the masked input patches, we proposed the prior knowledge mixing mechanism for KD on SR tasks in both feature and block levels. The prior mixers are applied to the raw feature maps of the student and teacher models in order to encode them into a unified latent space, in which the models' prior knowledge is mixed. Subsequently, the mixed latent feature map is decoded to its original space, enabling the reconstruction of the enhanced feature map and the performance of distillation. While the purpose of the MAE is to reconstruct the masked pixels, the encoder-decoder in the feature prior mixer reconstructs the portion of the teacher model feature map that is replaced by the student's. This allows the student model's intermediate representations to have a similar distribution to the teacher model's. The block prior mixer modulates the network's capacity to process and represent information. This is achieved by means of a dynamic combination of blocks, whereby the resulting fusion information is transferred from the feature prior mixer to the enhanced network. The two granularity of prior mixtures follows the common idea of prior mixing and propagation, which effectively reduces the capacity disparity between the teacher and student.

Feature Prior Mixer. Figure 2 illustrates the hybrid prior knowledge framework at the feature level, including the encoder modules for the teacher and student feature maps and the decoder module for the fused latent representation. At the k-th feature distillation position, initially, the feature maps of both the student model $\mathbf{F}_k^S$ and teacher model $\mathbf{F}_k^T$ are fed into the respective encoder models to obtain the latent representations $\mathbf{Z}_k^S, \mathbf{Z}_k^T \in \mathbb{R}^{C\times H\times W}$ in a unified latent space, where C, H, W are the dimension of the feature maps. Subsequently, the encoded student and teacher feature maps are fused in accordance with a pair of randomly generated complementary masks. And the decoder reverts the fused feature map $\mathbf{Z}_k^S$ to the enhanced feature map representation $\mathbf{F}_k^E$ in the same space as raw feature maps as

$$F_k^E = \text{Decoder}(Z_k^E) = \text{Decoder}(\mathbf{Z}_k^S \odot (\mathbf{1} - \mathbf{I}^M) + \mathbf{Z}_k^T \odot (\mathbf{I}^M)), \tag{3}$$

where $\mathbf{I}^M \in \{0,1\}^{C\times H\times W}$ represents a random three-dimensional mask and $\odot$ denotes the element-wise product between matrices. The student's feature map is combined with the teacher's prior knowledge with the above mixing mechanism to reduce the discrepancy between them at the feature level. $\mathbf{F}_k^E$ is utilized as an input to the subsequent block level prior mixer module. The feature distillation loss $\mathcal{L}_k^{feat_F}$ of Feature Prior Mixer is computed between $\mathbf{F}_k^E$ and $\mathbf{F}_k^T$ as

$$\mathcal{L}_k^{feat_F} = \mathcal{D}_{feat}(\mathbf{F}_k^E,\ \mathbf{F}_k^T) \tag{4}$$

Additionally, in order to enhance the reconstruction capability of the decoder and ensure the stability of training, at the beginning of training, the auxiliary enhanced feature map $\mathbf{F}_k^{'E}$ is obtained by

directly passing the teacher's feature map to the teacher's encoder and decoder without applying the above masking and mixing strategy. The auxiliary "auto-encoder" loss $\mathcal{L}_k^{ae}$ is computed as

$$\mathcal{L}_k^{ae} = \mathcal{D}_{feat}(\mathbf{F}_k^{'E},\ \mathbf{F}_k^T). \tag{5}$$

It requires the encoder and decoder to serve as an auto-encoder structure, ensuring the decoded enhanced feature map is comparable with $\mathbf{F_k^T}$. The enhancement of the decoder contributes to the overall effectiveness of the feature prior mixer module.

Block Prior Mixer. Existing feature-based distillation methods on SR tasks mostly align the feature maps in the original representation space with Mean Absolute Error or Mean Square Error (MSE). The semantic information among the teacher and student networks are differently distributed (Liu et al., 2023). Solely aligning features at the present distillation node with the same magnitudes of distance can lead the student model to learn entirely different information. To tackle this issue, we propose to align the networks' ability of processing and representing information by assembling a dynamic combination of blocks and transmitting the fusion information from the Feature Prior Mixer to the enhanced network.

To construct an enhanced network ($\mathcal{F}_E^{block}$) at the distillation position k, according to the Block Prior Mixing Option R_k randomly sampled from $\{0, 1\}$, the output of Feature Prior Mixer $\mathbf{F}_k^E$ is forwarding propagated to the student network ($\mathrm{R}_k = 1$) or teacher network ($\mathrm{R}_k = 0$), as the propagation path exemplified in Figure 2. The $\mathcal{B}_{S_{(k)}}$ and $\mathcal{B}_{T_{(k)}}$ represent the block from student and teacher models after the current position respectively. $\mathcal{B}_{O_{(k)}}$ represents the mixed block at the current position based on R_k, which can be computed as

$$\mathcal{B}_{O_{(k)}} = \mathrm{R}_k \mathcal{B}_{S_{(k)}} + (1 - \mathrm{R}_k)\mathcal{B}_{T_{(k)}}. \tag{6}$$

Based on this process, denote the output of such concatenated network as $I_{SR}^{E_k}$,

$$I_{SR}^{E_k} = \mathcal{F}_E^{block}(I_{LR}; \Theta^S) = \mathcal{B}_{O_{(k)}}(\mathbf{F}_k^E) \tag{7}$$

The feature knowledge distillation loss based on Block Prior Mixer is derived through the combined network's final output with the teacher model's output:

$$\mathcal{L}_k^{feat_B} = \mathcal{D}(I_{SR}^{E_k},\ I_{SR}^T) \tag{8}$$

In addition, $\mathcal{L}_k^{feat_B} = 0$ if the k-th feature distillation position is dropped out. It is anticipated that there will be an attainment of interchangeability between the corresponding teacher and student network blocks, allowing the student to inherit and replicate the capabilities of the teacher model.

The Whole Pipeline. Compared to conventional feature-based KD methods, MiPKD uses the enhanced feature maps and networks to impose more constraints on the student model. In general, for each feature distillation position k, based on the pair of $\mathbf{F}_k^T$ and $\mathbf{F}_k^S$ as the input of Feature Prior Mixer, the random masked feature maps are fused in a unified latent space. And the $\mathcal{L}_k^{feat_F}$ is computed to align the enhanced feature map $\mathbf{F}_k^E$ with the initial teacher feature map in the same representation space. Subsequently, the randomly sampled R_k determines the propagation option of $\mathbf{F}_k^E$, the networks' blocks are randomly exchanged and the knowledge is transmitted from the teacher to student model, as shown in Figure 2. Besides logits-KD loss $\mathcal{L}_{logits}$, reconstruction loss $\mathcal{L}_{rec}$, the feature losses in block and feature levels are accumulated:

$$\mathcal{L}_{total} = \lambda_{kd}\mathcal{L}_{logits} + \lambda_{rec}\mathcal{L}_{rec} + \sum_{k \le K}(\lambda_{feat}\mathcal{L}_k^{feat_F} + \lambda_{block}\mathcal{L}_k^{feat_B}). \tag{9}$$

where λ_{kd}, λ_{rec}, λ_{feat}, λ_{block} represent the weights for logits-kd loss, reconstruction loss, feature prior mixer and block prior mixer respectively. The teacher's prior knowledge is effectively transferred through this multi-level distillation process.

4 Experimental Results

4.1 Experiment Setups

Backbones and Evaluation. We use EDSR (Lim et al., 2017), RCAN (Zhang et al., 2018), and SwinIR (Liang et al., 2021) as backbone models to verify the effectiveness of MiPKD and compare it

Table 1: SR model specifications on $\times 4$ experimental settings. The #Params, FLOPs and FPS are calculated with a 256×256×3 input image and FPS is computed on a single NVIDIA V100 GPU.

Model	Role	Network			FLOPs (G)	#Params (M)	FPS
		Channel	Block	Group			
EDSR	Teacher	256	32	-	3293.35	43.09	3.2
	Student 1	64	32	-	207.28	2.70	33.958
	Student 2	64	16	-	129.97 (25.3×)	1.52 (28.3×)	53.3
RCAN	Teacher	64	20	10	1044.03	15.59	6.3
	Student	64	6	10	366.98	5.17	12.3
SwinIR	Teacher	180	6	-	861.27	11.90	0.459
	Student	60	4	-	121.48	1.24	0.874

with prior KD methods on $\times 2$, $\times 3$, and $\times 4$ super-resolving scales. The SR network specifications and some statistics are presented in Table 1, including the number of channels, residual blocks and residual groups (RCAN), number of parameters (#Params), FLOPs, and inference speed (frame per second, FPS).

Table 2: Quantitative comparison of distilling EDSR (Lim et al., 2017) on the benchmark datasets. In these experiments, the EDSR student model of c64b32 is distilled by the teacher model of c256b32.

Scale	Method	Set5	Set14	BSD100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
x2	Teacher	38.20/0.9606	34.02/0.9204	32.37/0.9018	33.10/0.9363
	Scratch	38.00/0.9605	33.57/0.9171	32.17/0.8996	31.96/0.9268
	KD	38.04/0.9606	33.58/0.9172	32.19/0.8998	31.98/0.9269
	RKD	38.03/0.9606	33.57/0.9173	32.18/0.8998	31.96/0.9270
	AT	37.96/0.9603	33.48/0.9167	32.12/0.8990	31.71/0.9241
	FitNet	37.59/0.9589	33.09/0.9136	31.79/0.8953	30.46/0.9111
	FAKD	37.99/0.9606	33.60/0.9173	32.19/0.8998	32.04/0.9275
	CSD	38.06/0.9607	33.65/0.9179	32.22/0.9004	32.26/0.9300
	MipKD	**38.18/0.9611**	**33.82/0.9197**	**32.30/0.9011**	**32.56/0.9323**
x3	Teacher	34.76/0.929	30.66/0.8481	29.32/0.8104	29.02/0.8685
	Scratch	34.39/0.927	30.32/0.8417	29.08/0.8046	27.99/0.8489
	KD	34.43/0.9273	30.34/0.8422	29.10/0.8050	28.00/0.8491
	RKD	34.43/0.9274	30.33/0.8423	29.09/0.8051	27.96/0.8493
	AT	34.29/0.9262	30.26/0.8406	29.03/0.8035	27.76/0.8443
	FitNet	33.35/0.9178	29.71/0.8323	28.62/0.7949	26.61/0.8167
	FAKD	34.39/0.9272	30.34/0.8426	29.10/0.8052	28.07/0.8511
	CSD	34.45/0.9275	30.32/0.8430	29.11/0.8061	28.21/0.8549
	MipKD	**34.60/0.9288**	**30.50/0.8454**	**29.21/0.8079**	**28.52/0.8592**
x4	Teacher	32.65/0.9005	28.95/0.7903	27.81/0.744	26.87/0.8086
	Scratch	32.29/0.8965	28.68/0.7840	27.64/0.7380	26.21/0.7893
	KD	32.30/0.8965	28.70/0.7842	27.64/0.7382	26.21/0.7897
	RKD	32.30/0.8965	28.69/0.7842	27.64/0.7383	26.20/0.7899
	AT	32.22/0.8952	28.63/0.7825	27.59/0.7365	25.97/0.7825
	FitNet	31.65/0.8873	28.33/0.7768	27.38/0.7309	25.40/0.7637
	FAKD	32.27/0.8960	28.65/0.7836	27.62/0.7379	26.18/0.7895
	CSD	32.34/0.8974	28.72/0.7856	27.68/0.7396	26.34/0.7948
	MipKD	**32.45/0.8980**	**28.79/0.7865**	**27.71/0.7400**	**26.46/0.7968**

We compare MiPKD with the baselines: train from scratch, Logits-KD (Hinton et al., 2015), RKD (Park et al., 2019), AT (Zagoruyko & Komodakis, 2016), FitNet (Romero et al., 2014), FAKD (He et al., 2020), CrossKD (Fang et al., 2023), and CSD (Wang et al., 2021b). Since the CSD is a self-distillation method in the channel-spliting manner, it's not applicable to the RCAN experiments of network depth distillation. The results for ×4 EDSR trained with CSD are obtained by testing the provided checkpoint, and the ×2 and ×3 ones are reproduced by us since the checkpoints are unavailable. To evaluate quality of SR model's output, we calculate the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) on the Y channel of the YCbCr color space. We use 800 images from DIV2K (Timofte et al., 2017) for training and evaluate SR models on four

Table 3: Quantitative comparison on RCAN (Zhang et al., 2018) architecture on the benchmark datasets. In these experiments, the RCAN student model of c64b6 is distilled by the teacher model of c64b20.

Scale	Method	Set5	Set14	BSD100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
x2	Teacher	38.27/0.9614	34.13/0.9216	32.41/0.9027	33.34/0.9384
	Scratch	38.13/0.9610	33.78/0.9194	32.26/0.9007	32.63/0.9327
	KD	38.17/0.9611	33.83/0.9197	32.29/0.9010	32.67/0.9329
	RKD	38.18/0.9612	33.78/0.9191	32.29/0.9011	32.70/0.9330
	AT	38.13/0.9610	33.70/0.9187	32.25/0.9005	32.48/0.9313
	FitNet	37.97/0.9602	33.57/0.9174	32.19/0.8999	32.06/0.9279
	FAKD	38.17/0.9612	33.83/0.9199	32.29/0.9011	32.65/0.9330
	CrossKD	38.18/0.9612	33.82/0.9195	32.29/0.9012	32.69/0.9331
	MiPKD	**38.26/0.9614**	**34.02/0.9210**	**32.35/0.9017**	**32.98/0.9357**
x3	Teacher	34.74/0.9299	30.65/0.8482	29.32/0.8111	29.09/0.8702
	Scratch	34.61/0.9288	30.45/0.8444	29.18/0.8074	28.59/0.8610
	KD	34.61/0.9291	30.47/0.8447	29.21/0.8080	28.62/0.8612
	RKD	34.67/0.9292	30.48/0.8451	29.21/0.8080	28.60/0.8610
	AT	34.55/0.9287	30.43/0.8438	29.17/0.8070	28.43/0.8577
	FitNet	34.21/0.9248	30.20/0.8399	29.05/0.8044	27.89/0.8472
	FAKD	34.63/0.9290	30.51/0.8453	29.21/0.8079	28.62/0.8612
	CrossKD	34.66/0.9291	30.50/0.8448	29.22/0.8082	28.64/0.8617
	MiPKD	**34.76/0.9299**	**30.61/0.8467**	**29.28/0.8090**	**28.89/0.8658**
x4	Teacher	32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/0.8087
	Scratch	32.38/0.8971	28.69/0.7842	27.63/0.7379	26.36/0.7947
	KD	32.45/0.8980	28.76/0.7860	27.67/0.7400	26.49/0.7982
	RKD	32.39/0.8974	28.74/0.7856	27.67/0.7399	26.47/0.7981
	AT	32.31/0.8967	28.69/0.7839	27.64/0.7385	26.29/0.7927
	FitNet	31.99/0.8899	28.50/0.7789	27.55/0.7353	25.90/0.7791
	FAKD	32.46/0.8980	28.77/0.7860	27.68/0.7400	26.50/0.7980
	CrossKD	32.45/0.8984	28.81/0.7866	27.69/0.7406	26.53/0.7992
	MiPKD	**32.58/0.8998**	**28.84/0.7875**	**27.75/0.7418**	**26.66/0.8029**

benchmark datasets: Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2012), BSD100 (Martin et al., 2001), and Urban100 (Huang et al., 2015).

Training Details. All models are trained using Adam (Kingma & Ba, 2014) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$, with a batch size of 16 and a total of 2.5×10^5 updates. The initial learning rate is set to 10^{-4} and is decayed by a factor of 10 at every 10^5 iteration. We set the loss weights λ_1 and λ_2 to 10 and 1, respectively. The proposed MiPKD is implemented by the BasicSR (Wang et al., 2022b) and PyTorch (Paszke et al., 2019) framework and train them using 4 NVIDIA V100 GPUs. The LR images used for training and evaluation were obtained by down-sampling the HR images with the bicubic degradation method. During training, the input images are randomly cropped into 48×48 patches and augmented with random horizontal/vertical flips and rotations.

4.2 Results and Comparison

Comparison with Baseline Methods. Quantitative results for training EDSR (Lim et al., 2017), RCAN (Zhang et al., 2018), and SwinIR (Liang et al., 2021) of three SR scales are presented in Table 2, Table 3 and Table 4, from which we can draw the following conclusions:

(1) Existing KD methods for SR have limited effects, some may even deteriorate the student model. The KD methods originally designed for high-level CV tasks (RKD, AT, FitNet), though applicable, hardly improve the SR models over training from scratch. For instance, AT and FitNet underperform the vanilla student models trained without KD among all settings.

(2) The presented MiPKD outperforms existing KD methods baselines for model compression. For example, MiPKD outperforms the vanilla student in the most challenging dataset Urban100 in EDSR$\times$2, $\times$3 and $\times$4 settings by **0.6 dB**, **0.53 dB**, **0.25 dB** in terms of PSNR, respectively as Table 2 shown. Compared with trainning from scratch, **0.35 dB**, **0.30 dB**, **0.30 dB** in terms of PSNR are improved, respectively, on Urban100 dataset in RCAN $\times$2, $\times$3, and $\times$4 settings as Table 3 shown.

Table 4: Quantitative comparison of distilling SwinIR (Liang et al., 2021) on the benchmark datasets.

Scale	Method	Set5		Set14		BSD100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2	Teacher	38.36	0.9620	34.14	0.9227	32.45	0.9030	33.40	0.9394
	Scratch	38.00	0.9607	33.56	0.9178	32.19	0.9000	32.05	0.9279
	KD	38.04	0.9608	33.61	0.9184	32.22	0.9003	32.09	0.9282
	MipKD	**38.14**	**0.9611**	**33.76**	**0.9194**	**32.29**	**0.9011**	**32.46**	**0.9313**
3	Teacher	34.89	0.9312	30.77	0.8503	29.37	0.8124	29.29	0.8744
	Scratch	34.41	0.9273	30.43	0.8437	29.12	0.8062	28.20	0.8537
	KD	34.44	0.9275	30.45	0.8443	29.14	0.8066	28.23	0.8545
	MipKD	**34.53**	**0.9283**	**30.52**	**0.8456**	**29.19**	**0.8079**	**28.47**	**0.8591**
4	Teacher	32.72	0.9021	28.94	0.7914	27.83	0.7459	27.07	0.8164
	Scratch	32.31	0.8955	28.67	0.7833	27.61	0.7379	26.15	0.7884
	KD	32.27	0.8954	28.67	0.7833	27.62	0.7380	26.15	0.7887
	FitNet	32.08	0.8925	28.51	0.7800	27.53	0.7354	25.80	0.7779
	FAKD	32.06	0.8926	28.52	0.7800	27.53	0.7354	25.81	0.7780
	MipKD	**32.39**	**0.8971**	**28.76**	**0.7854**	**27.68**	**0.7403**	**26.37**	**0.7956**

(3) The MiPKD is applicable to the transformer network and able to boost the model's performance. Conventional feature-based KD methods are not directly applicable to the Transformer-type networks, so we compare MiPKD with training from scratch and the response-based KD (Hinton et al., 2015) in the experiments. The results in Table 4 indicate that the MiPKD could improve the transformer SR model by a large margin, further emphasizing its superior performance.

Table 5: Training expenses of KD methods for distilling ×2 EDSR model.

KD methods	Logits-KD	FitNet	FAKD	CSD	MiPKD
Time (s/step)	0.49	0.56	0.56	1.18	0.87
Urban100 PSNR	31.98	30.46	32.04	32.26	32.56

Visual Comparison. Figure 3 compares the output of ×4 EDSR models from the Urban100 dataset with various KD methods. For instance, for img_047, MiPKD can reconstruct much better fine details than all baseline works. FAKD are prone to artifacts in the left-bottom of the building and the vanilla student, Logits-KD, FAKD, and FitNet are over-blurred. In contrast, MiPKD alleviates the blurring artifacts and reconstructs much more structural details. Similar observations can be found in other cases, *e.g.* the characters and anisotropic textures in img_073. These visual comparisons are consistent with the quantitative results, demonstrating the superiority of MiPKD. More visual comparisons can be found in the supplementary materials.

Comparison of training costs: As shown in Table 5, MiPKD significantly outperforms Logits-KD by 0.12dB PNSR, while with an increase of only 0.38s training time per step. It indicates that our MiPKD achieves the best trade-off between performance and training time.

5 Ablation Study

To demonstrate the effectiveness of the proposed MiPKD scheme, we conduct detailed ablation studies on ×multiple scale RCAN and EDSR networks.

Ablation on the feature and block prior mixers for MipKD. There are two fine-grained prior mixer modules in MiPKD, namely, the feature and block prior mixers. Their individual effects are ablated in Table 6. The result shows that employing the feature prior mixers leads to significant performance improvement and the block prior mixer based on it could further boost the student model.

Ablation on the MiPKD feature prior mixer module. In the feature prior mixer module of MiPKD, the teacher and student models' feature maps are mapped to the latent space through corresponding encoders, then randomly mixed and stitched. We present an analysis on the encoders in Table 7, comparing MiPKD with 1) removing the encoders, aligning and utilizing the teacher's feature map directly and 2) sharing encoder between the teacher and student model. Removing the encoders would substantially deteriorate the student model's performance. Due to the different distribution of

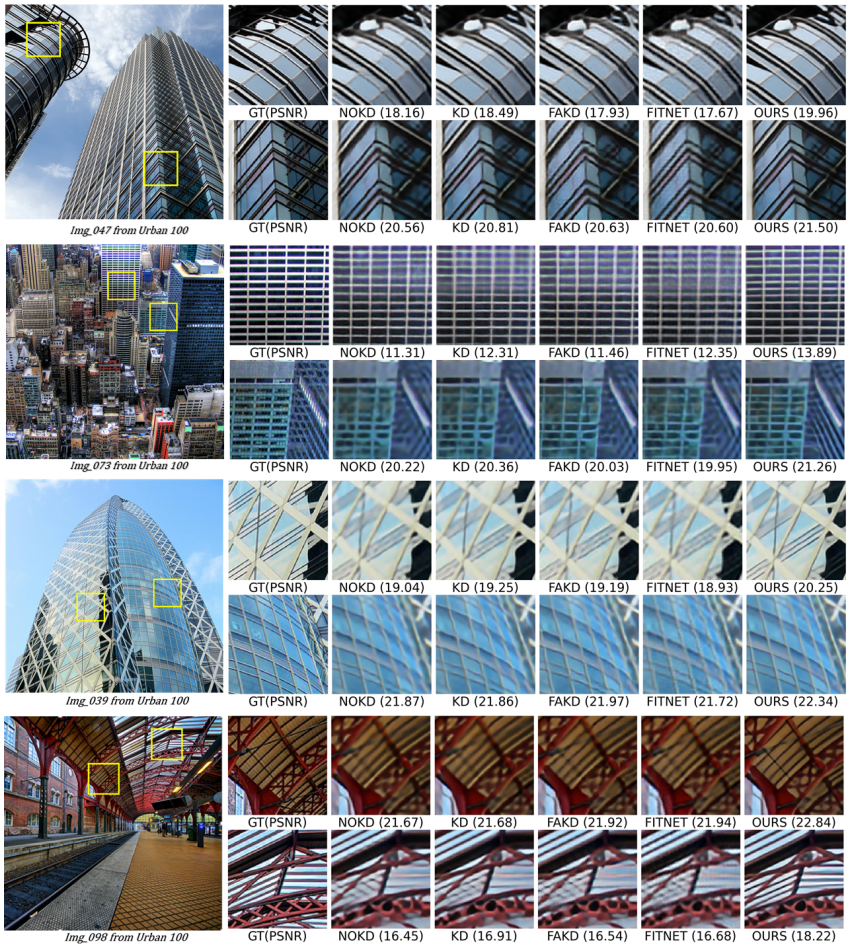


Figure 3: Visual comparison ($\times 4$) with existing SRKD methods from Urban100. The numbers in the bracket denote the PSNR of the presented patches.

Table 6: Ablation on the two prior mixers. The RCAN student model of c32b5g5 is distilled by the teacher model of c32b6g10.

Feature Prior Mixer	Block Prior Mixer	Urban100
		PSNR / SSIM
✗	✗	25.60 / 0.7700
✓	✗	25.63 / 0.7711
✗	✓	25.65 / 0.7717
✓	✓	25.69 / 0.7728

Table 7: Ablation on the encoder type in MiPKD feature mixer module without block prior mixer module.

Encoder Type	Urban100	
	PSNR	SSIM
No Encoder	24.51	0.7149
Shared Encoder	25.61	0.7704
Separate Encoder	25.63	0.7711

teacher and student models' feature maps, a shared encoder cannot effectively map them to the same latent space, leading to noisy mixtures. Assigning separate encoders to the teacher and student models yields the best results, indicating that mixing feature priors in the same latent space is necessary.

Table 8 compares the encoder and decoder of different network architectures with similar sizes. The convolutional neural network can better project the representations to the unified latent space, as the result shows that CNN exhibits better performance than the MLP encoder/decoder.

Ablation on the "auto-encoder" loss L_k^{ae}. We compared the MiPKD with and without L_k^{ae} in Table 9. The results indicate that the auxiliary "auto-encoder" loss makes the mapping between the raw feature maps' space and the latent space more accurate, leading to better student model performance.

Table 8: Comparison of different encoder and decoder network settings.

Encoder/Decoder Type	Urban100 PSNR/SSIM
MLP	26.42/0.7964
Conv	26.66/0.8029

Table 9: Ablation study on the auto-encoder loss $\mathcal{L}_k^{ae}$.

Auto-encoder Loss	Urban100 PSNR/SSIM
✗	26.42/0.7971
✓	26.66/0.8029

Table 10: Ablation analysis on the masking strategy for feature prior mixture.

masking strategy	Urban100 PSNR/SSIM
Cosine	25.62/0.7711
Grid mask	25.61/0.7669
CKA	25.63/0.7713
Random	25.69/0.7728

Besides, the mask generation strategies are compared in Table 10. Compared with 1) masking according to the Cosine or CKA similarity between teacher and student models' feature maps or 2) generating the complementary pairs of feature map by fixed grid pattern, the random 3D-mask exhibits the best performance and least calculation consumption. A more flexible, generalizable strategy is applied in the prior mixer module.

Ablation on the Loss weights setting of feature and block mixers. The impact of various weights of feature mixers loss and block mixer loss is evaluated as Table 11 shown, where λ_{rec}, λ_{kd}, λ_{feat}, λ_{block} represent the weights for reconstruction loss, logits-kd loss, feature prior mixer and block prior mixer respectively. Considering the initial fluctuation caused by mixing the block from networks, λ_{block} is applied since 0.1 presented the best student performance as the Table 9 shown. In addition, the reconstruction loss of auto-encoder in the feature prior mixer is introduced in the initial stage of training. As the reconstruction ability of decoder improves, it's beneficial for the prior mixer to fuse dark knowledge and restore the enhanced feature map efficiently.

Table 11: Ablation analysis on the weights of different losses

λ_{rec}	λ_{kd}	λ_{feat}	λ_{block}	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM
1	1	1	1	32.46/0.8972	28.75/0.7851	27.68/0.7399	26.53/0.7976
1	1	0.1	1	32.34/0.8970	28.73/0.7849	27.67/0.7394	26.47/0.7960
1	1	0.1	0.1	32.42/0.8980	28.75/0.7857	27.68/0.7399	26.51/0.7988
1	1	1	0.1	32.58/0.8998	28.84/0.7875	27.75/0.7418	26.66/0.8029

6 Conclusion

In this paper, we proposed the prior mixing mechanism for KD on SR in feature and block levels. The teacher's knowledge is effectively integrated with the student's feature via the Feature Prior Mixer, and the reconstructed feature propagates stochastically by the Block Prior Mixer. The masked feature maps are fused in a unified latent space, and the mixed prior narrows the optimization space. The Block Prior Mixer propagates the reconstructed feature and re-ensembles the networks to constrain the student model. The two granularity of the prior mixtures follows the common idea of prior mixing and propagation, which effectively reduces the capacity disparity between the teacher and the student. Extensive experiments demonstrate the effectiveness and superiority of the proposed MiPKD method.

REFERENCES

Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, pp. 252–268, 2018.

Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. *BMVC*, 2012.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, pp. 12299–12310, 2021.

Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, and Qingyuan Li. Fast, accurate and lightweight super-resolution with neural architecture search. In *ICPR*, pp. 59–64. IEEE, 2021.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pp. 184–199. Springer, 2014.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2015.

Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, pp. 391–407. Springer, 2016.

Hangxiang Fang, Yongwen Long, Xinyi Hu, Yangtao Ou, Yuanjia Huang, and Haoji Hu. Dual cross knowledge distillation for image super-resolution. *Journal of Visual Communication and Image Representation*, 95:103858, 2023.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *IJCV*, 129:1789–1819, 2021.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Zibin He, Tao Dai, Jian Lu, Yong Jiang, and Shu-Tao Xia. Fakd: Feature-affinity based knowledge distillation for efficient image super-resolution. In *ICIP*, pp. 518–522. IEEE, 2020.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Cheeun Hong, Sungyong Baik, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Cadyq: Content-aware dynamic quantization for image super-resolution. In *ECCV*, 2022.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pp. 5197–5206, 2015.

Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACMMM*, pp. 2024–2032, 2019.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pp. 1646–1654, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

Wonkyung Lee, Junghyup Lee, Dohyung Kim, and Bumsub Ham. Learning with privileged information for efficient image super-resolution. In *ECCV*, pp. 465–482. Springer, 2020.

Huixia Li, Chenqian Yan, Shaohui Lin, Xiawu Zheng, Baochang Zhang, Fan Yang, and Rongrong Ji. Pams: Quantized super-resolution via parameterized max scale. In *ECCV*, pp. 564–580, 2020.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, pp. 1833–1844, 2021.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pp. 136–144, 2017.

Dongyang Liu, Meina Kan, Shiguang Shan, and Xilin Chen. Function-consistent feature distillation. *arXiv preprint arXiv:2304.11832*, 2023.

Xiaotong Luo, Qiuyuan Liang, Ding Liu, and Yanyun Qu. Boosting lightweight single image super-resolution via joint-distillation. In *ACMMM*, pp. 1535–1543, 2021.

Yinglan Ma, Hongyu Xiong, Zhe Hu, and Lizhuang Ma. Efficient super resolution using binarized neural network. In *CVPR Workshops*, pp. 0–0, 2019.

David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pp. 416–423. IEEE, 2001.

Ying Nie, Kai Han, Zhenhua Liu, An Xiao, Yiping Deng, Chunjing Xu, and Yunhe Wang. Ghostsr: Learning ghost features for efficient image super-resolution. *arXiv preprint arXiv:2101.08525*, 2021.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pp. 3967–3976, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4):1–34, 2021.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *CoRR*, abs/1412.6550, 2014.

Dehua Song, Chang Xu, Xu Jia, Yiyi Chen, Chunjing Xu, and Yunhe Wang. Efficient residual dense block search for image super-resolution. In *AAAI*, volume 34, pp. 12007–12014, 2020.

Dehua Song, Yunhe Wang, Hanting Chen, Chang Xu, Chunjing Xu, and DaCheng Tao. Addersr: Towards energy efficient image super-resolution. In *CVPR*, pp. 15648–15657, 2021.

Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, pp. 114–125, 2017.

Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *CVPR*, pp. 12965–12974, 2020.

Li Wang, Dong Li, Lu Tian, and Yi Shan. Efficient image super-resolution with collapsible linear blocks. In *CVPR*, pp. 817–823, 2022a.

Longguang Wang, Xiaoyu Dong, Yingqian Wang, Xinyi Ying, Zaiping Lin, Wei An, and Yulan Guo. Exploring sparsity in image super-resolution for efficient inference. In *CVPR*, pp. 4917–4926, 2021a.

Xintao Wang, Liangbin Xie, Ke Yu, Kelvin C.K. Chan, Chen Change Loy, and Chao Dong. BasicSR: Open source image and video restoration toolbox. `https://github.com/XPixelGroup/BasicSR`, 2022b.

Yanbo Wang, Shaohui Lin, Yanyun Qu, Haiyan Wu, Zhizhong Zhang, Yuan Xie, and Angela Yao. Towards compact single image super-resolution via contrastive self-distillation. In *IJCAI*, 2021b.

Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, pp. 17683–17693, 2022c.

Gengqi Yao, Zhan Li, Bir Bhanu, Zhiqing Kang, Ziyi Zhong, and Qingfeng Zhang. Mtkdsr: Multi-teacher knowledge distillation for super resolution image reconstruction. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 352–358. IEEE, 2022.

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

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pp. 5728–5739, 2022.

Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, pp. 711–730. Springer, 2012.

Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *ECCV*, pp. 649–667. Springer, 2022.

Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. Data-free knowledge distillation for image super-resolution. In *CVPR*, pp. 7852–7861, 2021a.

Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pp. 286–301, 2018.

Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Aligned structured sparsity learning for efficient image super-resolution. *NeurIPS*, 34:2695–2706, 2021b.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.