PocketSR: The Super-Resolution Expert in Your Pocket Mobiles

Haoze Sun[1]*, Linfeng Jiang[2]*, Fan Li[2], Renjing Pei[2], Zhixin Wang[2], Yong Guo[2], Jiaqi Xu[2], Haoyu Chen[4], Jin Han[2], Fenglong Song[2], Yujiu Yang[1]✉, Wenbo Li[3]✉
[1]Tsinghua University [2]Huawei [3]Joy Future Academy [4]HKUST (GZ)
shz22@tsinghua.org.cn yang.yujiu@sz.tsinghua.edu.cn fenglinglwb@gmail.com

Abstract

Real-world image super-resolution (RealSR) aims to enhance the visual quality of in-the-wild images, such as those captured by mobile phones. While existing methods leveraging large generative models demonstrate impressive results, the high computational cost and latency make them impractical for edge deployment. In this paper, we introduce PocketSR, an ultra-lightweight, single-step model that brings generative modeling capabilities to RealSR while maintaining high fidelity. To achieve this, we design LiteED, a highly efficient alternative to the original computationally intensive VAE in SD, reducing parameters by 97.5% while preserving high-quality encoding and decoding. Additionally, we propose online annealing pruning for the U-Net, which progressively shifts generative priors from heavy modules to lightweight counterparts, ensuring effective knowledge transfer and further optimizing efficiency. To mitigate the loss of prior knowledge during pruning, we incorporate a multi-layer feature distillation loss. Through an in-depth analysis of each design component, we provide valuable insights for future research. PocketSR, with a model size of 146M parameters, processes 4K images in just 0.8 seconds, achieving a remarkable speedup over previous methods. Notably, it delivers performance on par with state-of-the-art single-step and even multi-step RealSR models, making it a highly practical solution for edge-device applications.

1 Introduction

Real-world image super-resolution (RealSR) [1, 2, 3, 4, 5, 6, 7] is a fundamental task in computer vision that reconstructs high-quality images from low-quality inputs. With applications spanning smartphone photography, medical imaging, and remote sensing, RealSR has driven extensive research interest and development. Recent breakthroughs in generative modeling—particularly diffusion models [8, 9, 10, 11, 12, 13, 14]—have opened new frontiers in SR by leveraging powerful generative priors to recover intricate textures and realistic structures, significantly enhancing image quality.

Scaling up text-to-image (T2I) diffusion models (*e.g.*, from SD1.5 [10] to SD3 [15] or FLUX [16]) has been shown to markedly improve SR performance [17]. Additionally, integrating multimodal large language models to generate detailed and accurate image descriptions further unlocks these models' generative potential [18, 19, 20, 21, 22]. However, the substantial computational cost and slow inference speed of such large-scale models limit their practical deployment. Consequently, research efforts have shifted toward efficient SR diffusion methods, focusing on lightweight architectures [23] and reduced sampling steps [24, 25, 26, 27, 28]. Yet, deploying these models on resource-constrained edge devices remains a formidable challenge.

Given that low-resolution (LR) inputs provide a rich and detailed prior—far more informative than the sparse textual cues in T2I generation—it is possible to achieve competitive SR performance with

*Equal Contribution ✉ Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

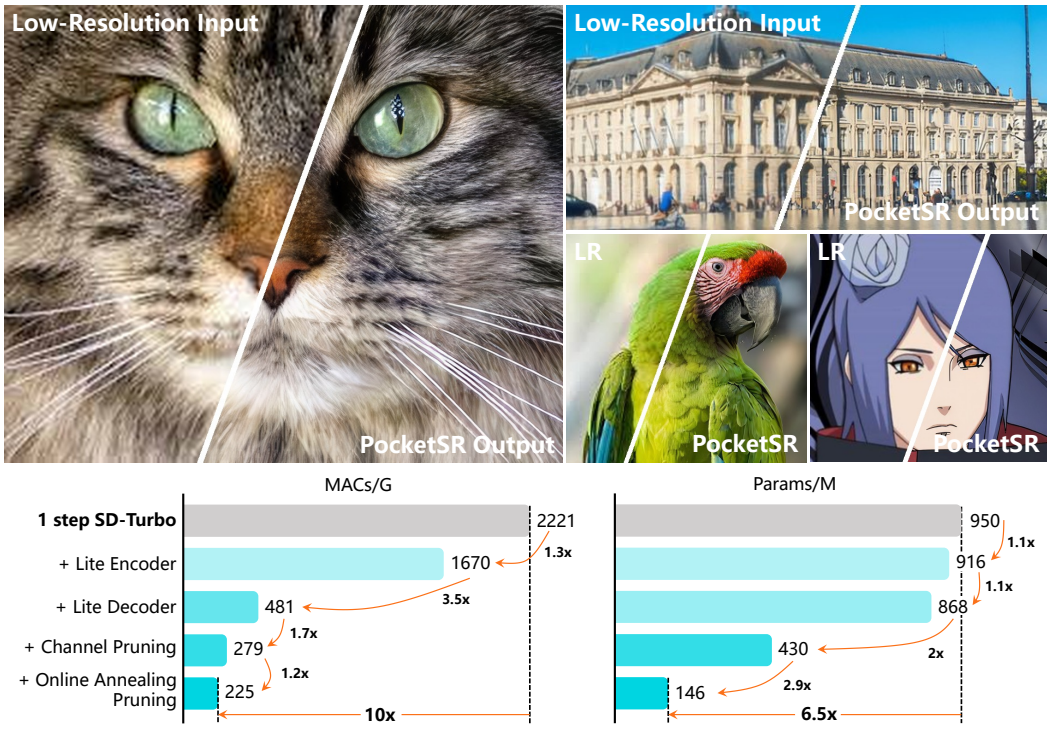

Figure 1: Visualization of the real-world image super-resolution results and efficiency of our proposed method. To enable the practical application of diffusion-based SR models, we introduce PocketSR, an ultra-lightweight, single-step solution. The top visual examples demonstrate that PocketSR achieves high-quality super-resolution across diverse scenes, preserving fine details and textures. Best viewed zoomed in. The bottom section highlights the significant reduction in parameters ($10\times$) and computational cost ($6.5\times$), allowing our model to process 4K images in just 0.8 seconds—dramatically outpacing existing methods.

a significantly compressed model. Motivated by this insight, we conduct a thorough analysis of the core components of diffusion models, including the variational autoencoder (VAE) and the U-Net architecture, and propose tailored strategies to maintain high-fidelity and realistic generation while substantially reducing the model size. Our approach is built upon three key innovations:

Lite Encoder & Decoder (LiteED). We introduce a highly compressed encoder-decoder design featuring an ultra-compact encoder for efficient LR feature extraction and conditioning, coupled with a tiny decoder for high-quality reconstruction. To mitigate potential losses in representational capacity, we incorporate a Dual-Path Feature Injection mechanism that enriches U-Net inputs with additional high-dimensional feature channels and Adaptive Skip Connections that retain critical information. The resulting LiteED contains only 2M parameters, drastically improving model efficiency while preserving image quality.

Online Annealing Pruning for U-Net. Recognizing the extensive generative priors embedded within U-Net architectures, we go beyond straightforward channel pruning [23] by introducing an online annealing pruning strategy. In this approach, lightweight modules are gradually integrated alongside existing components (*e.g.*, residual blocks, self-attention layers, and feed-forward networks), while the contributions of original modules are progressively annealed to zero. This smooth transition ensures a stable knowledge transfer to the pruned architecture. Additionally, we conduct a comprehensive ablation study to determine optimal pruning positions, ensuring that our lightweight model retains strong generative capabilities.

Multi-layer Feature Distillation. Prior studies [23, 29] have demonstrated that feature-space distillation offers a more stable optimization process compared to distillation in the image domain. Motivated by this observation, we adopt a multi-layer, multi-scale distillation scheme to facilitate more reliable knowledge transfer. When combined with our online annealing pruning strategy, this approach substantially reduces computational cost while effectively preserving generative priors.

Through these architectural optimizations, we further integrate adversarial training to develop PocketSR, an ultra-lightweight, one-step diffusion-based SR model with only 146M parameters—just 10.4% of the size of StableSR [3] and 8.2% of OSEDiff [27]. Despite its compactness, PocketSR achieves performance on par with state-of-the-art approaches while significantly reducing computational complexity and inference time, as illustrated in Sect. 4.2. Notably, it processes a 4K image in just 0.8 seconds—7 times faster than OSEDiff [27]. Our findings demonstrate that with carefully designed modifications, diffusion-based SR models can be both lightweight and powerful, paving the way for practical deployment in real-world applications.

2 Related Work

2.1 Real-world Image Super-Resolution

Real-world image super-resolution (SR) aims to recover realistic details from degraded inputs while maintaining overall fidelity. Early image super-resolution (ISR) methods [30, 31, 32, 33] often converge to the statistical mean of plausible solutions, resulting in overly smoothed outputs and loss of fine details in real-world scenarios. To overcome this, several works [34, 2, 1, 35, 36, 37] have explored generative adversarial networks (GANs) for texture enhancement. However, due to inherent limitations such as mode collapse and training instability [38, 39], GAN-based methods still struggle to produce realistic textures. More recently, diffusion models [8, 9, 10, 11, 12, 40] have shown strong generative capabilities in synthesizing fine-grained details and realistic textures, making them a promising alternative for real-world SR.

2.2 Diffusion-based ISR

Recent advances in diffusion models, especially in text-to-image (T2I) synthesis (e.g., SD3 [15], FLUX [16]), have led to the development of pre-trained diffusion-based SR methods such as StableSR [3], DiffBIR [4], PASD [5], CoSeR [41], SeeSR [19], and SUPIR [18]. These approaches benefit from the strong image priors of T2I models, achieving impressive results. However, their reliance on multi-step inference introduces high latency and substantial computational cost.

To address this, recent works aim to reduce inference time by distilling multi-step models into one- or few-step variants. ResShift [6] accelerates inference by learning residual transitions from LR to HR images via a Markov chain. Building on this, Wang et al.[24] condense multi-step capabilities into single-step networks. OSEDiff[27] further leverages VSD [42] to incorporate T2I knowledge efficiently. PiSA-SR [43] decouples structure restoration and texture enhancement using dual LoRA sets. Despite these advances, single-step SR methods still remain computationally heavier than traditional GAN-based models, limiting their deployment on edge devices and highlighting the need for better efficiency–quality trade-offs.

2.3 Efficient Diffusion Models

Recent efforts on efficient diffusion models [44, 45, 46, 47, 48, 49, 29] focus on architectural optimization to reduce redundancy in large-scale models. SnapFusion [49] disentangles the contributions of individual modules to balance efficiency and accuracy. MobileDiff [47] improves efficiency by relocating Transformer blocks to lower-resolution stages. SnapGen [29] cuts computation and model size by removing high-resolution attention and replacing standard convolutions with depthwise separable ones. AdcSR [23] further explores one-step SR model efficiency. In Sect. 4.3, we compare our design with AdcSR and show that our approach achieves better performance with higher compression.

3 Method

Driven by deployment considerations, our PocketSR framework utilizes SD-Turbo[2] as the backbone, which achieves extreme model compression through a two-stage training pipeline. In the first stage, we replace the original Stable Diffusion's (SD) variational autoencoder with the Lite Encoder-Decoder (LiteED), with all parameters set to be trainable. In the second stage, we freeze LiteED and apply our pruning strategies to the U-Net, gradually removing redundant components.

[2]https://huggingface.co/stabilityai/sd-turbo

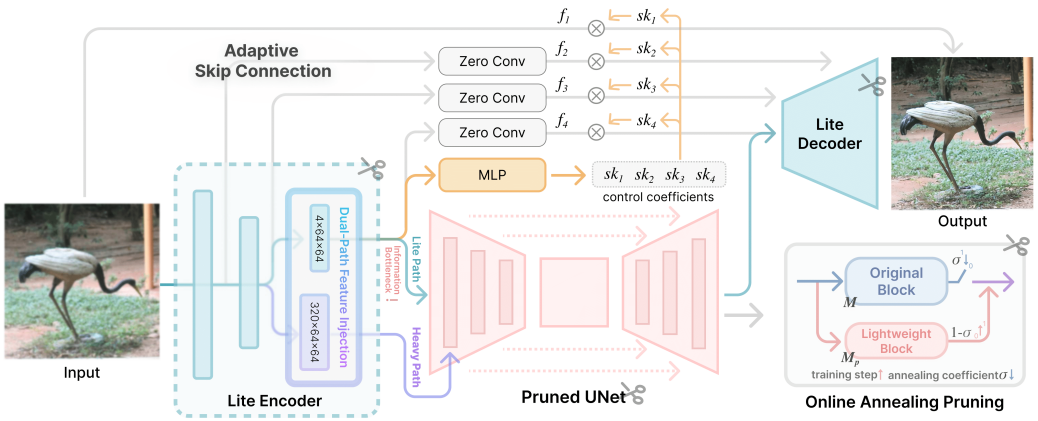

Figure 2: Overview of PocketSR framework. We replace the original Stable Diffusion variational autoencoder with LiteED, and apply online annealing pruning and multi-layer feature distillation strategies to the diffusion U-Net, effectively reducing model parameters and computational complexity while maintaining excellent super-resolution performance.

3.1 Lite Encoder & Decoder

We propose LiteED, an ultra-lightweight encoder-decoder architecture designed for edge-side super-resolution. To reduce complexity, we simplify the SD encoder to several convolutional layers, while maintaining effective feature extraction. Additionally, we replace the original decoder with a lightweight alternative that offers substantial improvements in efficiency while preserving comparable image fidelity (detailed in the supplementary materials). Notably, the decoder in LiteED is modular and can be substituted with a more powerful variant to enhance reconstruction quality, albeit at the cost of efficiency. The decoder is initialized from an open-source model[3], while the encoder is randomly initialized.

To supplement the ultra-light encoder and enrich the representational capacity, we propose an adaptive skip connection mechanism in LiteED, as depicted in Figure 2. Specifically, four control coefficients are generated from the encoder output via an MLP to modulate the multi-scale skip connections. This adaptive mechanism allows the model to selectively integrate input features during decoding, effectively mitigating the loss of information due to encoder compression. Additionally, to stabilize training, we incorporate zero-convolution into the skip connections.

Dual-path Feature Injection. We identify an information bottleneck between the original SD encoder and the U-Net. For a 512×512 image, the output feature size of the SD encoder's last ResBlock is $[N, 512, 64, 64]$, but it is compressed to $[N, 4, 64, 64]$ in the final convolutional layer. This 128× reduction in feature dimensionality may lead to substantial information loss, negatively impacting super-resolution performance. To address this issue and improve the fidelity of the result, we propose a dual-path feature injection mechanism. Specifically, in our encoder, an additional high-dimensional feature of size $[N, 320, 64, 64]$ is extracted after the second convolutional layer of LiteED and injected into the U-Net following the initial block. This strategy enhances the information flow, allowing for more robust feature extraction. As for feature injection, we employ cross normalization [50], which has been shown to facilitate fast and stable convergence.

Despite its lightweight design, LiteED proves to be more effective in practice than networks with several times higher computational complexity. Moreover, the decoder in LiteED can be replaced with a larger-capacity network, which leads to better performance as demonstrated in our experiments.

3.2 U-Net Pruning

3.2.1 Online Annealing Pruning

Pruning is a straight-forward and effective model compression technique, which is widely used in efficient image generation [45, 48, 47, 49, 29, 51, 52]. In this paper, we propose online annealing

[3]https://github.com/madebyollin/taesd

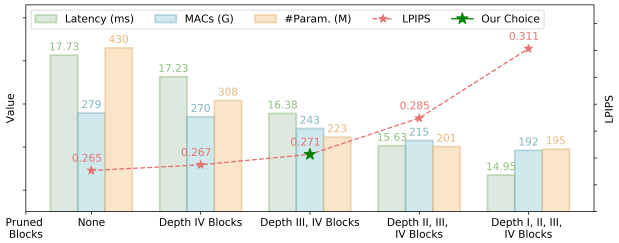

Figure 3: Analysis of the impact of pruning residual blocks at different depths using the widely adopted RealSR [53] test set, with performance measured by the LPIPS [54] metric. The inference resolution is 512×512, and we report the maximum inference speed on an A100 GPU.

Table 1: Comparison of the computational efficiency of the SD network and its modules before and after our streamlining. We highlight the computational efficiency of the original SD and the improvements brought by PocketSR.

Network	Computational Efficiency		
	Time (ms)	MACs (G)	#Param. (M)
Lite Encoder	↓99% 0.2	↓99% 8.1	↓98% 0.8
SD Encoder	22.6	559.5	34.2
Pruned U-Net	↓53% 12.5	↓64% 146.1	↓83% 144.2
SD U-Net	26.7	401.9	866
Lite Decoder	↓90% 3.0	↓94% 70.7	↓98% 1.2
SD Decoder	30.3	1259.7	49.4
PocketSR	↓80% 15.7	↓90% 224.9	↓85% 146.2
Original SD	79.7	2221.2	949.6

pruning, a stable and efficient pruning strategy specifically designed for SR. Existing Diffusion U-Net pruning methods typically simply discard [47, 49] or replace pruned modules [48, 29], which leads to a significant loss of prior information. Our pruning strategy preserves the prior information through online knowledge transfer. As shown in Figure 2, we connect the original module M in parallel with a lightweight replacement module M_P. During the training process, we continuously increase the contribution of the lightweight module, while gradually annealing the contribution of the original module to zero:

$$\mathbf{y} = \sigma \cdot M(\mathbf{x}) + (1 - \sigma) \cdot M_P(\mathbf{x}) \,, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ represent the module input and output, respectively. The annealing coefficient is defined as $\sigma = \min\left(0, (T - t)/T\right)$, where T is the total number of annealing steps, and t denotes the current training step. The parameters of the original module are frozen during training. Once training is complete, the original module is replaced with the lightweight module. The pruned modules are permanently removed at inference time, yielding a highly compact and efficient model. Our experiments show that this online pruning approach can better preserve the diffusion prior and generate more realistic textures (see Sect. 4.3).

3.2.2 Pruning Implementation

Diffusion U-Net comprises four computationally expensive module types: residual blocks, cross-attention layers, self-attention layers, and feed-forward networks (FFNs). To reduce the number of parameters and computational cost while preserving performance, we replace each module with a carefully chosen lightweight counterpart. Lightweight layers (e.g., normalization, activation) remain unchanged. For residual blocks, we replace all convolutions with depthwise separable convolutions [55], significantly reducing the number of parameters. Since text input is entirely discarded, cross-attention layers are replaced with MLPs of two linear layers. Self-attention layers are approximated using linear attention [56] to reduce computational cost. Finally, the hidden dimension of the FFN is reduced to one-fourth of its original size.

We also observe that the pruning location has a significant impact on model performance, with its effect varying according to the module's depth within the network. Specifically, *shallower modules exert a greater influence on super-resolution quality, while deeper blocks can be pruned with minimal impact.* Taking SD's U-Net as an example, we label the depths as {I, II, III, and IV}, from shallow to deep. Figure 3 illustrates the trade-off between performance and efficiency when pruning residual blocks at different depths. Our results show that pruning at deeper levels (III and IV) significantly reduces parameter count and computational cost while having negligible impact on performance. While pruning shallower modules yields greater efficiency gains, it leads to a more noticeable drop in final performance.

We interpret this phenomenon as deeper layers in the U-Net of a generative diffusion model primarily process high-level information [57, 58], such as layout and style, which contributes less to the SR task. This characteristic makes a free lunch for pruning. Our findings extend beyond residual blocks to other network modules, *e.g.*, self-attention, cross-attention, and FFNs, indicating a general pruning strategy for SR networks. Balancing efficiency and effectiveness, we prune residual blocks and FFNs at depths III and IV, while self-attention layers are pruned at depth IV. Additionally, we prune

all cross-attention layers, as their impact on super-resolution quality is minimal. Detailed ablation experiments on all network module pruning locations are provided in the supplementary material.

3.2.3 Multi-layer Feature Distillation

To better preserve generative priors during pruning, we introduce a feature-level knowledge distillation strategy. Due to the architectural mismatch between the models before and after pruning, single-layer distillation may result in training instability. Instead, we adopt a multi-layer global distillation scheme to enhance robustness and enable stable knowledge transfer:

$$\mathcal{L}_{\text{distill}} = \mathbb{E}\left[\sum_{l} \left\| f^{l}_{\text{pre-pruning}} - \phi\left(f^{l}_{\text{post-pruning}}\right)\right\|_2^2\right], \tag{2}$$

where $f^{l}_{\text{pre-pruning}}$ and $f^{l}_{\text{post-pruning}}$ denote the feature representations from the l-th layer of the models before and after pruning, respectively. Following [59], we implement the mapping function $\phi(\cdot)$ as a lightweight, trainable projection module composed of a single convolutional layer.

3.3 Training Details

Beyond the aforementioned lightweight optimizations, we further investigate the impact of channel reduction on model performance, which is compatible with our online annealing pruning strategy. Notably, direct channel reduction can substantially weaken the model's learned priors. However, this adverse effect is markedly mitigated when integrated with the proposed multi-layer feature distillation strategy. Comprehensive experimental results are provided in the supplementary materials. To strike a balance between efficiency and performance, we reduce the channel width in the U-Net to 70% of its original size. Table 1 presents the final latency, computational cost, and parameter count of our model in comparison to the original SD-Turbo.

Building on recent progress in one-step diffusion-based generation [60, 61], we introduce adversarial loss during training to improve texture fidelity. In the first stage, the model is trained using a combination of MSE loss, LPIPS [62], and adversarial loss. In the second stage, we incorporate the multi-layer feature distillation loss to retain knowledge from the full-capacity model during pruning.

In the first training phase, we train the unpruned SD U-Net equipped with LiteED for 80,000 steps. In the second phase, we first apply channel pruning over 80,000 steps, followed by module-wise online annealing pruning for an additional 8,000 steps. The total number of annealing steps is set to $T = 8000$. A fixed batch size of 64 is used throughout the entire training process. We employ the AdamW optimizer with a learning rate of 1×10^{-4}, and the timestep is fixed at $t = 999$ for one-step diffusion. Additionally, the original text embedding is replaced with a learnable embedding vector.

4 Experiments

4.1 Experimental Settings

The training dataset comprises approximately 500K high-quality images from LSDIR [63] and 10K images from FFHQ [64]. During training, images are randomly cropped into 512×512 patches. Low-quality counterparts are synthesized using the widely adopted degradation pipeline from Real-ESRGAN [1]. For evaluation, we follow the protocols in [23, 27] and report results on the DRealSR [65] and RealSR [53] benchmarks.

We compare the proposed PocketSR with four state-of-the-art multi-step methods—StableSR [3], DiffBIR [4], SeeSR [19], and ResShift [6]—as well as three leading single-step methods: SinSR [24], AdcSR [23], and OSEDiff [27]. We also compare our model with GAN-based SR methods in the supplementary material. All models are evaluated using a comprehensive set of metrics, including perceptual similarity metrics (LPIPS [62], DISTS [66]), fidelity metrics (PSNR, SSIM [67]), and no-reference quality metrics (NIQE [68], MUSIQ [69]).

4.2 Comparison with State-of-the-Arts

Quantitative and Efficiency Comparison. Table 2 reports quantitative results on RealSR [53] and DRealSR [65], with efficiency metrics listed in the last five rows. The results show that our

Table 2: Quantitative and efficiency comparisons with state-of-the-art diffusion-based methods on real-world datasets are presented. The best results for each metric are highlighted in bold. PocketSR delivers real-time inference at over 60 FPS on an A100 GPU for 512×512 inputs, while maintaining competitive quantitative performance.

Datasets	Metrics	StableSR [3]	DiffBIR [4]	SeeSR [19]	ResShift [6]	SinSR [24]	OSEDiff [27]	AdcSR [23]	**PocketSR**
RealSR [53]	LPIPS↓	0.3018	0.3636	0.3009	0.3460	0.3188	0.2921	0.2885	**0.2713**
	DISTS↓	0.2288	0.2312	0.2223	0.2498	0.2353	0.2128	0.2129	**0.2094**
	PSNR↑	24.70	24.75	25.18	**26.31**	26.28	25.15	25.47	25.47
	SSIM↑	0.7085	0.6567	0.7216	**0.7421**	0.7347	0.7341	0.7301	0.7330
	NIQE↓	5.912	5.535	5.408	7.264	6.287	5.648	5.350	**5.067**
	MUSIQ↑	65.78	64.98	69.77	58.43	60.80	69.09	**69.90**	67.07
DRealSR [65]	LPIPS↓	0.3284	0.4557	0.3189	0.4006	0.3665	0.2968	0.3046	**0.2962**
	DISTS↓	0.2269	0.2748	0.2315	0.2656	0.2485	0.2165	0.2200	**0.2139**
	PSNR↑	28.03	26.71	28.17	**28.46**	28.36	27.92	28.10	28.05
	SSIM↑	0.7536	0.6571	0.7691	0.7673	0.7515	**0.7835**	0.7726	0.7675
	NIQE↓	6.524	6.312	6.397	8.125	6.991	6.490	6.450	**5.809**
	MUSIQ↑	58.51	61.07	64.93	50.60	55.33	64.65	**66.26**	63.85
Parameters (M)		1410	1717	2524	**119**	**119**	1775	456	146
MACs (G)		79940	24234	65857	5491	2649	2265	496	**225**
Sampling Steps		200	50	50	15	1	1	1	1
Inference Time (s)		11.5	2.7	4.3	0.71	0.13	0.11	0.03	**0.016**
FPS		0.09	0.37	0.23	1.4	7.7	9.1	33.3	**62.5**

method achieves strong super-resolution performance with excellent computational efficiency. First, the proposed single-step model, PocketSR, has only 146M parameters, achieves the lowest MACs among all methods, and processes a 512×512 image in 0.016 seconds on an A100 GPU—nearly twice as fast as AdcSR. This low latency and lightweight design make it ideal for real-time and edge deployment. PocketSR achieves an inference time of only 140ms on a newly released smartphone model from 2025, representing an over 80% reduction compared to the original, non-lightweight backbone. Second, PocketSR attains the best LPIPS, DISTS, and NIQE scores, demonstrating superior perceptual quality. Notably, on the DRealSR dataset, PocketSR surpasses the second-best method by **10%** in NIQE, indicating clearer texture restoration. Third, it delivers competitive fidelity, achieving PSNR and SSIM on par with AdcSR on RealSR and clearly outperforming multi-step models such as StableSR, DiffBIR, and SeeSR. Although ResShift and SinSR perform better than PocketSR in PSNR and SSIM, their texture are not as realistic as PocketSR.

Qualitative Comparison. Figure 4 shows qualitative comparisons, demonstrating that our method consistently maintains high fidelity and excels in detail reconstruction. In the first row, only DiffBIR, ResShift, SinSR, and PocketSR successfully recover the diagonal striped texture in the upper-right corner. However, the textures produced by DiffBIR and SinSR appear visually unrealistic. Among single-step methods, only PocketSR reconstructs an accurate and perceptually convincing pattern, highlighting its strength in preserving fine details. The second row further shows that PocketSR is the only method able to restore the building's structural details, reinforcing its high fidelity and detail recovery capabilities. Figure 1 showcases results across a broader range of natural scenes, including animals and animes. These results further verify that PocketSR not only achieves faithful reconstructions but also delivers rich, fine-grained textures across diverse visual contexts.

4.3 Ablation Study

We carefully analyze the effects of the proposed lite encoder and decoder, online annealing pruning, and multi-layer feature distillation, demonstrating an excellent trade-off between super-resolution quality and efficiency through extensive experiments. The experiments in this section follow the first-stage training in Section 3.3, where the unpruned SD U-Net is jointly trained with various encoder and decoder architectures for 80,000 steps. All settings share the same model and training configuration, differing only in encoder/decoder architecture. RealSR is used as the test dataset.

Effect of Adaptive Skip Connection and Dual-path Feature Injection. The proposed lite encoder simplifies the original SD encoder to improve efficiency, which inevitably introduces information

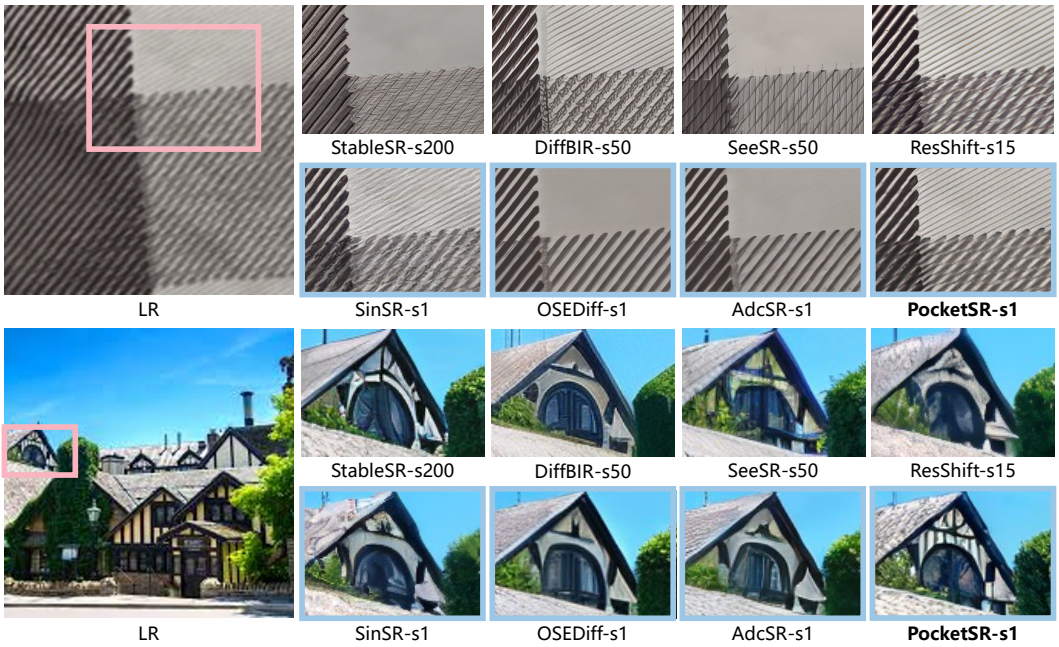

Figure 4: Qualitative results on real-world images. Single-step methods are highlighted for clarity. PocketSR delivers competitive performance, generating well-preserved structures and fine-grained textures, even when compared to multi-step models.

Table 3: We conducted ablation studies on the core architectural components of LiteED, including the Adaptive Skip Connection (ASC) and Dual-path Feature Injection (DFI). Additionally, we compared the proposed lightweight encoder with alternative designs to validate the effectiveness of our architecture. Finally, we replaced the decoder to assess the structural robustness of LiteED.

Decoder	Encoder	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	NIQE↓	Time (ms)	MACs (G)	Param. (M)
Lite Dec. (Ours)	**Lite Enc. (Ours)**	**25.61**	**0.7431**	**0.2474**	**0.1911**	5.200	31.4	481.6	868.3
	Lite Enc. w/o ASC	25.42	0.7426	0.2737	0.2050	5.329	30.7	480.8	868.3
	Lite Enc. w/o DFI	25.35	0.7427	0.2580	0.1940	5.136	31.1	478.6	**867.6**
	PixelUnshuffle	25.23	0.7417	0.2697	0.2028	5.552	**30.5**	**474.8**	867.7
	SD Enc.	25.19	0.7259	0.2685	0.1981	**5.092**	51.7	1032.2	901.4
SD Dec.	**Lite Enc. (Ours)**	25.95	0.7503	**0.2390**	**0.1843**	6.899	**58.1**	**1669.7**	**916.2**
	SD Enc.	**26.00**	**0.7548**	0.2445	0.1892	**6.898**	79.7	2221.2	949.6

loss. To address this, we introduce the Adaptive Skip Connection (ASC), which flexibly transfers essential features to the decoder during encoding. This not only eases the training of the diffusion U-Net but also improves both fidelity and perceptual quality in the super-resolution results.

Additionally, we design the Dual-path Feature Injection (DFI) module to alleviate the information bottleneck between the encoder and U-Net (see Sect. 3.1). In our Dual-path Feature Injection (DFI) design, the lite path provides compressed, information-dense features that offer global structural guidance and align well with the VAE feature distribution in SD, making them easier for the pretrained U-Net to utilize. In contrast, the heavy path contains richer details but lower information density, making it harder for generative models to use directly.

As shown in Table 3, both ASC and DFI consistently improve performance across nearly all metrics with minimal overhead. Figure 5 further proves: removing either module causes visible degradation, particularly in the word "Sausage" (first row) and the plastic chair edges (second row).

Comparison between the Lite Encoder and Alternative Encoding Strategies. We compare our lite encoder with the PixelUnshuffle approach and the original SD encoder (SD Enc.). PixelUnshuffle, introduced by AdcSR [23], is a lightweight method for injecting low-resolution (LR) features. Despite similar computational cost, our encoder consistently outperforms PixelUnshuffle across all metrics, with up to **8.3%** improvement in LPIPS. Moreover, our design achieves superior performance on

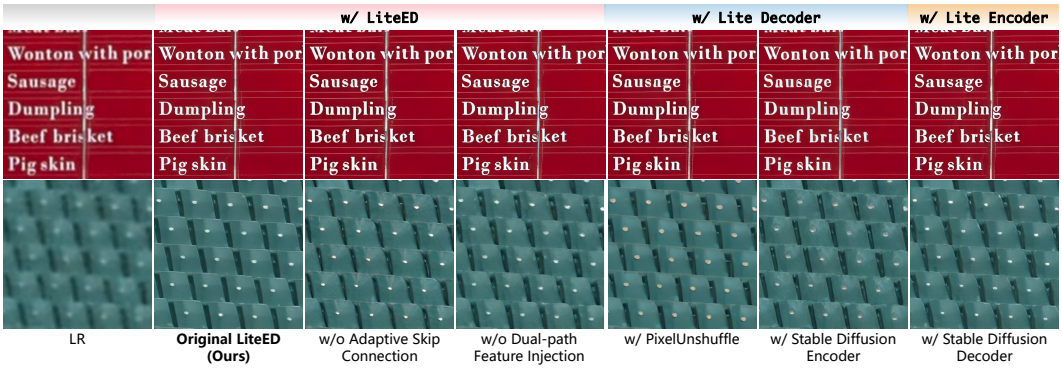

Figure 5: Ablation study of the LiteED design on "Canon 003" from RealSR (top) and "DSC 1286" from DRealSR (bottom).

Figure 6: Ablation study of our pruning strategy on "Nikon 004" and "Nikon 015" images from RealSR.

Table 4: Ablation study on pruning strategies. We conduct ablation studies on the online annealing strategy and the multi-layer feature distillation loss employed during pruning. Additionally, we compare the effectiveness of multi-layer distillation against single-layer distillation applied solely to the output layer of the U-Net.

Strategy	Online Annealing	Multi-layer Distillation	LPIPS↓	DISTS↓	NIQE↓
Ours	✓	✓	**0.2713**	**0.2094**	**5.067**
(1)	✗	✓	0.2732	0.2120	5.126
(2)	✓	✗	0.2816	0.2162	5.097
(3)	✓	Single-layer	0.2762	0.2116	5.077

reference-based metrics than the SD encoder, with only **46.7%** of its computational cost. We attribute this to the absence of skip connections and the potential bottleneck in the SD encoder, which may result in the loss of fidelity-critical information. As shown in Figure 5, PixelUnshuffle often causes fine detail loss (first row) and oversmoothing (second row), while the SD encoder tends to produce unnatural textures. These results highlight that a well-designed lightweight encoder can outperform complex counterparts in SR tasks, offering valuable insights for efficient model design.

Compatibility with more powerful image decoders. Prior studies [70, 23] have shown that decoder complexity often has a greater impact on performance than encoder complexity. Building on this, we investigate the decoder flexibility of LiteED by replacing its decoder with the original SD decoder, while keeping the lite encoder (w/ ASC & DFI) unchanged. This substitution leads to improved SR quality, albeit with a **247%** increase in computational overhead. These results underscore the generalizability of LiteED, enabling flexible decoder adjustment to balance efficiency and performance. We also compare our design with the original SD VAE (w/o ASC & DFI). The variant using the SD decoder and lite encoder achieves comparable performance with a **24.8%** reduction in computational cost, further validating the effectiveness of the LiteED architecture.

Effect of Online Annealing Pruning. In Table 4, we compare the proposed online annealing pruning strategy with direct lightweight module replacement, denoted as Strategy (1). Our approach consistently outperforms the baseline across all metrics, especially the no-reference NIQE score, thanks to its progressive knowledge transfer that better preserves the SD model's generative prior. As shown in Figure 6, while direct pruning fails to recover clear brick patterns and wheat textures, our method yields visually superior and high-quality results.

Effect of Multi-layer Feature Distillation. Multi-layer feature distillation is introduced to enhance the stability and robustness of knowledge transfer during pruning. As shown in Table 4, removing it (Strategy 2) degrades performance across all metrics, with LPIPS showing the largest drop (**3.8%**). We also evaluate single-layer distillation at the U-Net output (Strategy 3), which slightly outperforms Strategy 2 but still lags behind the full multi-layer setup. As illustrated in Figure 6, removing multi-layer distillation results in noticeable noise and artifacts due to unstable training dynamics.

5 Conclusion and limitation

We present PocketSR, a highly efficient single-step model for real-world image super-resolution. By replacing the heavy VAE in SD with LiteED, PocketSR reduces parameters and latency while preserving fidelity. Our proposed pruning strategy further enhances efficiency by progressively transferring generative priors to lightweight modules, optimizing performance without compromising quality. Experiments show that it achieves significant speedup and performance comparable to state-of-the-art RealSR models, making it practical for broad deployment.

One limitation of PocketSR is that its detail generation capability under severe degradations remains to be improved. Moreover, the current framework has not yet been optimized in conjunction with edge hardware, which we identify as a promising avenue for future research.

Acknowledgments. This work was supported by the National Key Research and Development Program of China (No. 2024YFB2808903). Thanks for the computing resources provided by Lei Ke during the rebuttal.

References

[1] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," in *ICCV*, 2021, pp. 1905–1914.

[2] K. Zhang, J. Liang, L. Van Gool, and R. Timofte, "Designing a practical degradation model for deep blind image super-resolution," in *ICCV*, 2021, pp. 4791–4800.

[3] J. Wang, Z. Yue, S. Zhou, K. C. Chan, and C. C. Loy, "Exploiting diffusion prior for real-world image super-resolution," *IJCV*, 2024.

[4] X. Lin, J. He, Z. Chen, Z. Lyu, B. Dai, F. Yu, Y. Qiao, W. Ouyang, and C. Dong, "Diffbir: Toward blind image restoration with generative diffusion prior," in *ECCV*. Springer, 2024, pp. 430–448.

[5] T. Yang, R. Wu, P. Ren, X. Xie, and L. Zhang, "Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization," in *ECCV*. Springer, 2024, pp. 74–91.

[6] Z. Yue, J. Wang, and C. C. Loy, "Resshift: Efficient diffusion model for image super-resolution by residual shifting," *Neurlps*, vol. 36, pp. 13 294–13 307, 2023.

[7] H. Chen, W. Li, J. Gu, J. Ren, H. Sun, X. Zou, Z. Zhang, Y. Yan, and L. Zhu, "Low-res leads the way: Improving generalization for super-resolution by self-supervised learning," in *CVPR*, 2024, pp. 25 857–25 867.

[8] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Neurlps*, vol. 33, pp. 6840–6851, 2020.

[9] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022, pp. 10 684–10 695.

[11] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Neurlps*, vol. 35, pp. 36 479–36 494, 2022.

[12] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu *et al.*, "Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis," *arXiv preprint arXiv:2310.00426*, 2023.

[13] J. Ren, W. Li, H. Chen, R. Pei, B. Shao, Y. Guo, L. Peng, F. Song, and L. Zhu, "Ultrapixel: Advancing ultra high-resolution image synthesis to new peaks," *Neurlps*, vol. 37, pp. 111 131–111 171, 2024.

[14] J. Ren, W. Li, Z. Wang, H. Sun, B. Liu, H. Chen, J. Xu, A. Li, S. Zhang, B. Shao *et al.*, "Turbo2k: Towards ultra-efficient and high-quality 2k video synthesis," *arXiv preprint arXiv:2504.14470*, 2025.

[15] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," in *ICML*, 2024.

[16] B. F. Labs, "Flux," https://github.com/black-forest-labs/flux, 2024.

[17] J. Li, J. Cao, Y. Guo, W. Li, and Y. Zhang, "One diffusion step to real-world super-resolution via flow trajectory distillation," *arXiv preprint arXiv:2502.01993*, 2025.

[18] F. Yu, J. Gu, Z. Li, J. Hu, X. Kong, X. Wang, J. He, Y. Qiao, and C. Dong, "Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild," in *CVPR*, 2024, pp. 25 669–25 680.

[19] R. Wu, T. Yang, L. Sun, Z. Zhang, S. Li, and L. Zhang, "Seesr: Towards semantics-aware real-world image super-resolution," in *CVPR*, 2024, pp. 25 456–25 467.

[20] H. Sun, W. Li, J. Liu, K. Zhou, Y. Chen, Y. Guo, Y. Li, R. Pei, L. Peng, and Y. Yang, "Beyond pixels: Text enhances generalization in real-world image restoration," *arXiv preprint arXiv:2412.00878*, 2024.

[21] T. Wu, K. Ma, J. Liang, Y. Yang, and L. Zhang, "A comprehensive study of multimodal large language models for image quality assessment," in *ECCV*. Springer, 2024, pp. 143–160.

[22] T. Wu, J. Zou, J. Liang, L. Zhang, and K. Ma, "Visualquality-r1: Reasoning-induced image quality assessment via reinforcement learning to rank," *arXiv preprint arXiv:2505.14460*, 2025.

[23] B. Chen, G. Li, R. Wu, X. Zhang, J. Chen, J. Zhang, and L. Zhang, "Adversarial diffusion compression for real-world image super-resolution," in *CVPR*, 2025.

[24] Y. Wang, W. Yang, X. Chen, Y. Wang, L. Guo, L.-P. Chau, Z. Liu, Y. Qiao, A. C. Kot, and B. Wen, "Sinsr: diffusion-based image super-resolution in a single step," in *CVPR*, 2024.

[25] R. Xie, C. Zhao, K. Zhang, Z. Zhang, J. Zhou, J. Yang, and Y. Tai, "Addsr: Accelerating diffusion-based blind super-resolution with adversarial diffusion distillation," *arXiv preprint arXiv:2404.01717*, 2024.

[26] L. Dong, Q. Fan, Y. Guo, Z. Wang, Q. Zhang, J. Chen, Y. Luo, and C. Zou, "Tsd-sr: One-step diffusion with target score distillation for real-world image super-resolution," *arXiv preprint arXiv:2411.18263*, 2024.

[27] R. Wu, L. Sun, Z. Ma, and L. Zhang, "One-step effective diffusion network for real-world image super-resolution," *Neurlps*, vol. 37, pp. 92 529–92 553, 2025.

[28] A. Zhang, Z. Yue, R. Pei, W. Ren, and X. Cao, "Degradation-guided one-step image super-resolution with diffusion priors," *arXiv preprint arXiv:2409.17058*, 2024.

[29] D. Hu, J. Chen, X. Huang, H. Coskun, A. Sahni, A. Gupta, A. Goyal, D. Lahiri, R. Singh, Y. Idelbayev *et al.*, "Snapgen: Taming high-resolution text-to-image models for mobile devices with efficient architectures and training," *arXiv preprint arXiv:2412.09619*, 2024.

[30] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *CVPRW*, 2017, pp. 136–144.

[31] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *ECCV*, 2018, pp. 286–301.

[32] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *ICCV*, 2021, pp. 1833–1844.

[33] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *CVPR*, 2022, pp. 5728–5739.

[34] 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 *CVPR*, 2017, pp. 4681–4690.

[35] X. Wang, Y. Li, H. Zhang, and Y. Shan, "Towards real-world blind face restoration with generative facial prior," in *CVPR*, 2021, pp. 9168–9178.

[36] T. Yang, P. Ren, X. Xie, and L. Zhang, "Gan prior embedded network for blind face restoration in the wild," in *CVPR*, 2021, pp. 672–681.

[37] L. Yang, S. Wang, S. Ma, W. Gao, C. Liu, P. Wang, and P. Ren, "Hifacegan: Face renovation via collaborative suppression and replenishment," in *ACMMM*, 2020, pp. 1551–1560.

[38] J. Liang, H. Zeng, and L. Zhang, "Details or artifacts: A locally discriminative learning approach to realistic image super-resolution," in *CVPR*, 2022, pp. 5657–5666.

[39] L. Xie, X. Wang, X. Chen, G. Li, Y. Shan, J. Zhou, and C. Dong, "Desra: Detect and delete the artifacts of gan-based real-world super-resolution models," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 38 204–38 226.

[40] G. Liu, H. Sun, J. Li, F. Yin, and Y. Yang, "Accelerating diffusion models for inverse problems through shortcut sampling," *arXiv preprint arXiv:2305.16965*, 2023.

[41] H. Sun, W. Li, J. Liu, H. Chen, R. Pei, X. Zou, Y. Yan, and Y. Yang, "Coser: Bridging image and language for cognitive super-resolution," in *CVPR*, 2024, pp. 25 868–25 878.

[42] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation," *Neurlps*, vol. 36, pp. 8406–8441, 2023.

[43] L. Sun, R. Wu, Z. Ma, S. Liu, Q. Yi, and L. Zhang, "Pixel-level and semantic-level adjustable super-resolution: A dual-lora approach," *arXiv preprint arXiv:2412.03017*, 2024.

[44] T. Castells, H.-K. Song, T. Piao, S. Choi, B.-K. Kim, H. Yim, C. Lee, J. G. Kim, and T.-H. Kim, "Edgefusion: on-device text-to-image generation," *arXiv preprint arXiv:2404.11925*, 2024.

[45] G. Fang, X. Ma, and X. Wang, "Structural pruning for diffusion models," in *Neurlps*, 2023.

[46] Y. Zhu, X. Liu, and Q. Liu, "Slimflow: Training smaller one-step diffusion models with rectified flow," in *ECCV*. Springer, 2024, pp. 342–359.

[47] Y. Zhao, Y. Xu, Z. Xiao, H. Jia, and T. Hou, "Mobilediffusion: Instant text-to-image generation on mobile devices," in *ECCV*. Springer, 2024, pp. 225–242.

[48] B.-K. Kim, H.-K. Song, T. Castells, and S. Choi, "Bk-sdm: A lightweight, fast, and cheap version of stable diffusion," in *ECCV*. Springer, 2024, pp. 381–399.

[49] Y. Li, H. Wang, Q. Jin, J. Hu, P. Chemerys, Y. Fu, Y. Wang, S. Tulyakov, and J. Ren, "Snapfusion: Text-to-image diffusion model on mobile devices within two seconds," *Neurlps*, vol. 36, pp. 20 662–20 678, 2023.

[50] B. Peng, J. Wang, Y. Zhang, W. Li, M.-C. Yang, and J. Jia, "Controlnext: Powerful and efficient control for image and video generation," *arXiv preprint arXiv:2408.06070*, 2024.

[51] J. Knodt, "Structural dropout for model width compression," *arXiv preprint arXiv:2205.06906*, 2022.

[52] A. N. Gomez, I. Zhang, S. R. Kamalakara, D. Madaan, K. Swersky, Y. Gal, and G. E. Hinton, "Learning sparse networks using targeted dropout," *arXiv preprint arXiv:1905.13678*, 2019.

[53] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, "Toward real-world single image super-resolution: A new benchmark and a new model," in *ICCV*, 2019, pp. 3086–3095.

[54] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.

[55] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[56] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *ICML*. PMLR, 2020, pp. 5156–5165.

[57] X. Ma, G. Fang, and X. Wang, "Deepcache: Accelerating diffusion models for free," in *CVPR*, 2024, pp. 15 762–15 772.

[58] C. Si, Z. Huang, Y. Jiang, and Z. Liu, "Freeu: Free lunch in diffusion u-net," in *CVPR*, 2024, pp. 4733–4743.

[59] Y. Liu, J. Cao, B. Li, W. Hu, J. Ding, and L. Li, "Cross-architecture knowledge distillation," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 3396–3411.

[60] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, "Adversarial diffusion distillation," in *ECCV*. Springer, 2024, pp. 87–103.

[61] T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and W. T. Freeman, "Improved distribution matching distillation for fast image synthesis," in *NeurIPS*, 2024.

[62] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.

[63] Y. Li, K. Zhang, J. Liang, J. Cao, C. Liu, R. Gong, Y. Zhang, H. Tang, Y. Liu, D. Demandolx *et al.*, "Lsdir: A large scale dataset for image restoration," in *CVPR*, 2023, pp. 1775–1787.

[64] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[65] P. Wei, Z. Xie, H. Lu, Z. Zhan, Q. Ye, W. Zuo, and L. Lin, "Component divide-and-conquer for real-world image super-resolution," in *ECCV*. Springer, 2020, pp. 101–117.

[66] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *TPAMI*, vol. 44, no. 5, pp. 2567–2581, 2020.

[67] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *TIP*, vol. 13, no. 4, pp. 600–612, 2004.

[68] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.

[69] J. Ke, Q. Wang, Y. Wang, P. Milanfar, and F. Yang, "Musiq: Multi-scale image quality transformer," in *ICCV*, 2021, pp. 5148–5157.

[70] T. Hu, F. Chen, H. Wang, J. Li, W. Wang, J. Sun, and Z. Li, "Complexity matters: Rethinking the latent space for generative modeling," *Neurlps*, vol. 36, pp. 29 558–29 579, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstracts and introductions truly reflect the main contribution and scope of the article.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our limitations are included in Section 5 of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our papers are experimentally oriented and do not contain theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide details of the model training details to reproduce the main results in Section 3.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We are sorry, but due to company policy, we are unable to provide open source code to the community at this time.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details of the training and test sets in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the specificity of the low-level visual task, model effects are mainly judged by IQA metrics, and error bars, confidence intervals, or statistical significance tests are less frequently used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include the comparison of computational efficiency in our experiments, including the number of parameters, computational cost, and inference time.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics strictly.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work is centered around the low-level vision and does not have a significant social impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our model does not have this risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit all the creators or original owners of assets used in the paper and mention the license and terms of use explicitly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.