Not All Degradations Are Equal: A Targeted Feature Denoising Framework for Generalizable Image Super-Resolution

Supplementary Material

1. Experimental Details

Datasets and Training Setup For training, we utilize the DIV2K dataset [2], which contains 800 high-quality images with diverse content. For evaluation, we employ five standard SR benchmark datasets: Set5 [4], Set14 [39], BSD100 [27], Urban100 [17], and Manga109 [28]. We also evaluate on real-world datasets including DIV2K validation tracks (Difficult, Wild, and Mild) and the real-world DSLR dataset [6] with Canon and Nikon subsets to demonstrate generalization capabilities. During training, we employ the L1 loss function in conjunction with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The batch size is set to 16, processing low-resolution (LR) images of size 32×32 pixels. We implement a cosine annealing learning rate strategy initialized at 2×10^{-4} over 500,000 iterations. All experiments are conducted using the PyTorch framework on 4×NVIDIA A800 GPUs. For our targeted feature denoising framework, we adopt a multi-objective training scheme with loss weights set to $\lambda_{cls} = 0.1$ and $\lambda_{feat} = 0.01$. Following Theorem 3.1, training prioritizes reconstruction in early stages, deferring denoising until noise confidence surpasses 75%. This scheduling ensures stable content preservation before handling high-frequency noise.

Degradation Modeling Protocol Following recent advances in blind image restoration [19, 31, 33], we construct a comprehensive degradation pipeline to simulate diverse real-world distortions. Specifically, we adopt a *second-order* degradation process [23], which has become a standard benchmark for evaluating robustness and generalization. The following eight degradations are considered:

1. **Clean**: Bicubic downsampling only.
2. **Blur**: Gaussian blur followed by bicubic downsampling.
3. **Noise**: Additive Gaussian noise followed by bicubic downsampling.
4. **JPEG**: JPEG compression followed by bicubic downsampling.
5. **Blur+Noise**: Sequential application of Gaussian blur and additive Gaussian noise, followed by bicubic downsampling.
6. **Blur+JPEG**: Sequential application of Gaussian blur and JPEG compression, followed by bicubic downsampling.
7. **Noise+JPEG**: Sequential application of additive Gaussian noise and JPEG compression, followed by bicubic downsampling.
8. **Blur+Noise+JPEG**: Combination of all three degradations.

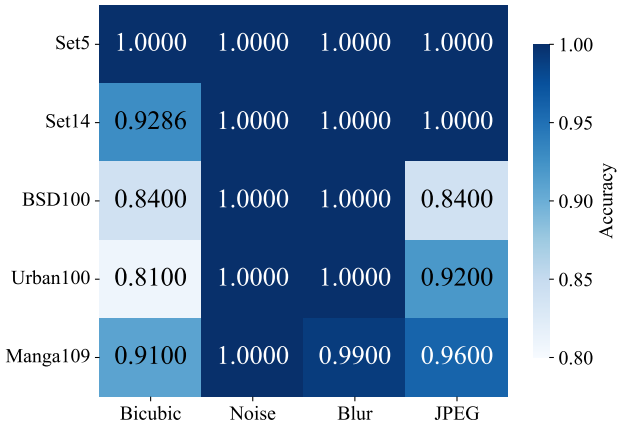

Figure 1. **Noise detection accuracy when facing unknown degradations during testing.**

Formally, for a clean high-resolution image $\mathbf{x}$, each degraded low-resolution observation $\mathbf{y}$ is synthesized as:

$$\mathbf{y} = \mathcal{D} \circ \mathcal{C} \circ \mathcal{N} \circ \mathcal{B}(\mathbf{x}), \tag{14}$$

where $\mathcal{B}$ denotes blurring, $\mathcal{N}$ denotes noise injection, $\mathcal{C}$ denotes JPEG compression, and $\mathcal{D}$ denotes downsampling. Depending on the configuration, certain operators are replaced by the identity map.

Noise Detection Accuracy in Unseen Degradations. The noise detection module demonstrates robust discriminative capability across various benchmark datasets and degradation types, as illustrated in Figure 1. Particularly noteworthy is the perfect detection accuracy (1.0) observed for noise degradation across all evaluated datasets, which validates our hypothesis regarding the distinctive spectral characteristics of noise-induced corruption. While the module maintains high accuracy for blur degradation ($\geq$ 0.99), we observe marginally lower accuracy for JPEG artifacts in BSD100 (0.84) and bicubic degradation in Urban100 (0.81). This performance disparity aligns with our frequency-domain analysis, which revealed that noise exhibits uniform spectral distribution, making it more distinctively identifiable compared to other degradations that manifest primarily in specific frequency bands. The module's consistent performance across diverse datasets (Set5, Set14, BSD100, Urban100, Manga109) further substantiates the generalizability of our approach to real-world super-resolution scenarios involving complex degradation patterns.

Figure 2. **Visual comparison of different super-resolution methods on BSD100 dataset with bicubic_noise20 degradation.**

Table 1. **Noise detection accuracy before and after denoising.**

Dataset	SRResNet	RRDB	HAT	SwinIR
	Before → After	Before → After	Before → After	Before → After
Set5	1.00 → 0.00	1.00 → 0.01	0.99 → 0.00	0.99 → 0.00
Set14	1.00 → 0.00	1.00 → 0.00	0.98 → 0.01	0.99 → 0.01
BSD100	1.00 → 0.00	0.99 → 0.01	0.98 → 0.01	0.98 → 0.01
Urban100	0.98 → 0.02	0.97 → 0.03	0.96 → 0.03	0.95 → 0.04
Manga109	0.99 → 0.01	0.98 → 0.02	0.97 → 0.02	0.97 → 0.02

Noise Detection Accuracy Before and After Denoising. Table 1 demonstrates the efficacy of our feature denoising framework by comparing noise detection rates before and after applying the denoising module across multiple super-resolution architectures and benchmark datasets. The results reveal a remarkable transition in detection rates, with pre-denoising values approaching perfect classification (0.95-1.00) across all model-dataset combinations, indicating consistent identification of noise-corrupted features. After applying our denoising module, detection rates plummet dramatically to near-zero values (0.00-0.04), providing compelling evidence that our approach effectively eliminates noise characteristics from the feature representations. This pronounced before-after contrast is particularly evident in the SRResNet architecture, where the Set5, Set14, and BSD100 datasets exhibit a complete reversal from 1.00 to 0.00 detection rates. The consistently low post-denoising detection rates across architectures (SRResNet, RRDB, HAT, and SwinIR) and datasets substantiate the architecture-agnostic nature of our method. The marginally higher post-denoising rates observed in Urban100 (0.02-0.04) likely reflect the dataset's complex structural patterns, which present greater challenges for discriminat-

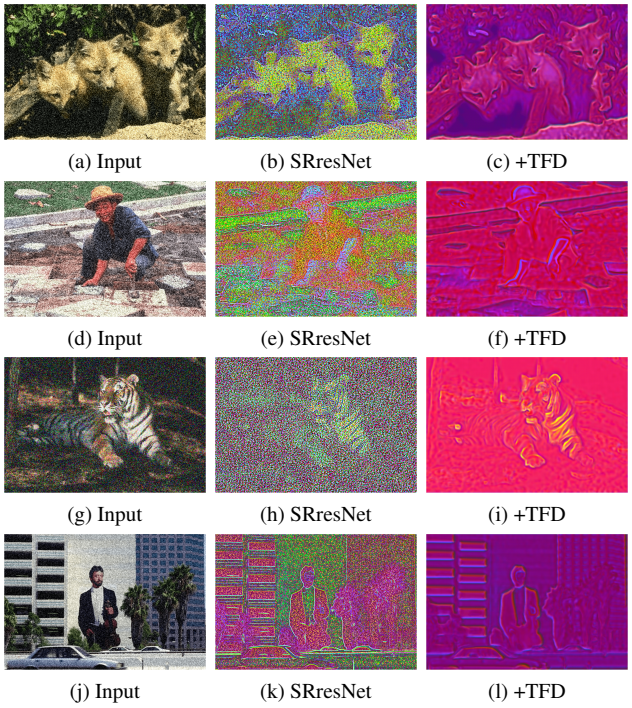

Figure 3. **Visualization of Feature Denoising Efficacy Across Diverse Visual Domains.**

Table 2. **Ablation study on different feature fusion strategies.**

Model	Fusion Method	Test Sets (PSNR)				
		Set5	Set14	BSD100	Urban100	Manga109
SRResNet	Addition	26.27	23.76	23.45	21.89	18.86
	Concatenation	26.44	23.92	23.62	22.05	19.01
	Multiplication	**26.71**	**24.10**	**23.87**	**22.28**	**19.22**
	Baseline	24.85	23.25	23.06	21.24	18.42
SwinIR	Addition	26.53	24.62	23.83	22.28	19.12
	Concatenation	26.68	24.78	23.97	22.46	19.25
	Multiplication	**26.92**	**24.96**	**24.14**	**22.63**	**19.34**
	Baseline	26.25	24.53	23.91	22.18	19.10

ing between residual noise and high-frequency content details. These quantitative results corroborate our qualitative observations and theoretical analysis, confirming that our approach effectively addresses the noise overfitting phenomenon by selectively suppressing noise-related features while preserving content-relevant information.

Feature Visualization Analysis. Figure 3 compares feature maps across degraded inputs, SRResNet, and our TFD-enhanced outputs, revealing how TFD reshapes feature representation under noise. In a wildlife scene with fox cubs, SRResNet's features are dominated by chaotic color noise, masking structural details, while TFD recovers clear object boundaries and preserves the cubs' morphology. For a human subject outdoors, SRResNet features are corrupted by irregular activations, weakening the semantic consistency of facial and body contours, whereas TFD suppresses noise and enhances structural clarity. In a challenging case of a tiger in natural habitat, SRResNet's features dissolve into noise, making the striped pattern almost unrecognizable, while TFD restores both texture and shape with remarkable fidelity. In an urban scene, SRResNet struggles to maintain geometric regularity, fragmenting building edges and human outlines, while TFD reconstructs rectilinear structures and preserves human silhouettes. These consistent improvements across diverse cases demonstrate that TFD selectively suppresses noise-induced distortions while safeguarding content-relevant details, offering a robust and generalizable solution to feature corruption in degraded image super-resolution.

Comparison with Existing Regulation Strategies. The quantitative results presented in Tables 3 provide a systematic evaluation of TFD against existing regularization techniques across five benchmark datasets. For CNN-based architectures (SRResNet, RRDB), TFD consistently outperforms both Dropout and Feature Alignment methods across all degradation types, with particularly substantial gains on noise-corrupted images. Specifically, on SRResNet, TFD achieves average PSNR improvements of 0.78dB over the baseline and 0.42dB over the next best method (Alignment) on Set5. This performance advantage extends to transformer-based architectures (HAT, SwinIR), where TFD maintains its superiority despite their inherently stronger baseline performance. The improvement pattern is consistent across datasets of varying complexity - from the simpler Set5 to the challenging Urban100 and content-specialized Manga109. Notably, TFD's efficacy becomes more pronounced under complex degradation scenarios (e.g., Blur+Noise+JPEG), suggesting its robust generalization capability. These results empirically validate our hypothesis that targeted noise suppression, rather than uniform regularization, is crucial for enhancing cross-degradation generalization in image super-resolution.

Comparison with Different Fusion Strategies. The results in Table 2 demonstrate the effectiveness of different cross-domain feature integration strategies within our framework. When comparing various integration methods, we observe that Multiplication consistently outperforms alternative strategies across all benchmark datasets. For SRResNet, adaptive modulation delivers significant improvements over element-wise addition and channel concatenation. This pattern holds for SwinIR as well, though with smaller margins due to its stronger baseline performance. The superiority of Multiplication can be attributed to its dynamic nature—the frequency-derived attention mask selectively modulates spatial features based on noise concentration, effectively preserving structural details while suppressing noise artifacts. In contrast, element-wise addition treats all features equally, while concatenation merely combines rather than filters information.

Table 3. **Average PSNR of different methods in $\times 4$ blind SR on five benchmarks with eight types of degradations.**

Data	Method	Clean	Blur	Noise	JPEG	Blur+Noise	Blur+JPEG	Noise+JPEG	Blur+Noise+JPEG	Average
Set5	SRResNet [20]	24.85	24.73	23.69	23.69	23.25	23.41	23.10	22.68	23.68
	+Dropout ($p = 0.7$)	25.63	25.23	23.82	24.05	23.47	23.64	23.46	23.01	24.04
	+Alignment	25.93	25.62	24.15	24.38	23.79	23.86	23.71	23.19	24.33
	+TFD	**26.71**	**26.20**	**24.31**	**24.49**	**23.39**	**23.92**	**23.67**	**22.99**	**24.46**
	RRDB [32]	25.18	25.12	22.92	23.82	23.44	23.45	23.32	22.81	23.76
	+Dropout ($p = 0.5$)	26.02	26.07	23.23	24.15	23.73	23.88	23.68	23.18	24.24
	+Alignment	26.78	26.55	24.02	24.70	24.12	24.14	23.93	23.26	24.69
	+TFD	**26.83**	**26.59**	**24.96**	**24.56**	**24.07**	**24.04**	**23.81**	**23.14**	**24.75**
	SwinIR [23]	26.25	26.03	24.15	24.37	23.80	23.84	23.67	22.99	24.39
	+Dropout ($p = 0.5$)	26.32	26.08	24.21	24.41	24.00	23.93	23.65	23.09	24.46
	+Alignment	26.49	26.23	24.61	24.68	24.13	24.17	23.89	23.09	24.66
	+TFD	**26.92**	**26.43**	**24.76**	**24.56**	**23.90**	**24.03**	**23.77**	**23.09**	**24.68**
Set14	SRResNet [20]	23.25	23.05	22.50	22.36	22.23	22.10	22.06	21.77	22.41
	+Dropout ($p = 0.7$)	23.73	23.45	22.53	22.62	22.28	22.39	22.28	21.98	22.66
	+Alignment	24.12	23.80	22.68	22.99	22.65	22.63	22.55	22.16	22.95
	+TFD	**24.54**	**24.15**	**23.02**	**23.11**	**22.43**	**22.79**	**22.60**	**22.13**	**23.10**
	RRDB [32]	23.74	23.36	22.33	22.59	22.47	22.17	22.29	21.95	22.61
	+Dropout ($p = 0.5$)	24.02	23.87	22.54	22.83	22.58	22.59	22.45	22.10	22.87
	+Alignment	24.70	24.35	22.91	23.21	22.80	22.76	22.71	22.21	23.21
	+TFD	**24.72**	**24.37**	**23.49**	**23.27**	**22.85**	**22.89**	**22.75**	**22.25**	**23.32**
	SwinIR [23]	24.53	24.25	23.46	23.14	22.53	22.73	22.59	22.20	23.18
	+Dropout ($p = 0.5$)	24.57	24.19	23.53	23.18	22.73	22.71	22.65	22.22	23.22
	+Alignment	24.65	24.28	23.53	23.29	22.87	22.79	22.81	22.28	23.31
	+TFD	**24.96**	**24.60**	**23.56**	**23.23**	**22.70**	**22.83**	**22.69**	**22.30**	**23.36**
BSD100	SRResNet [20]	23.06	22.99	22.45	22.48	22.26	22.34	22.22	22.05	22.48
	+Dropout ($p = 0.7$)	23.31	23.26	22.50	22.69	22.25	22.50	22.41	22.16	22.64
	+Alignment	23.83	23.64	22.77	23.04	22.53	22.79	22.62	22.32	22.94
	+TFD	**23.87**	**23.71**	**22.67**	**22.96**	**22.36**	**22.78**	**22.52**	**22.29**	**22.89**
	RRDB [32]	23.38	23.32	22.09	22.73	22.39	22.47	22.42	22.15	22.62
	+Dropout ($p = 0.5$)	23.59	23.66	22.68	22.86	22.53	22.71	22.52	22.28	22.85
	+Alignment	24.59	24.54	23.47	23.67	22.85	23.21	22.97	22.54	23.48
	+TFD	**24.11**	**24.05**	**23.13**	**23.19**	**22.74**	**22.95**	**22.69**	**22.41**	**23.15**
	SwinIR [23]	23.91	23.83	23.27	23.04	22.61	22.82	22.61	22.34	23.05
	+Dropout ($p = 0.5$)	23.90	23.87	23.30	23.08	22.68	22.80	22.64	22.33	23.08
	+Alignment	24.04	23.96	23.40	23.15	22.77	22.98	22.76	22.40	23.18
	+TFD	**24.14**	**24.06**	**23.50**	**23.27**	**22.84**	**23.05**	**22.84**	**22.57**	**23.28**
Urban100	SRResNet [20]	21.24	21.06	20.82	20.60	20.46	20.30	20.43	20.10	20.63
	+Dropout ($p = 0.7$)	21.57	21.25	20.85	20.90	20.48	20.49	20.66	20.22	20.80
	+Alignment	21.94	21.65	21.19	21.20	20.73	20.72	20.91	20.37	21.09
	+TFD	**22.28**	**21.89**	**21.20**	**21.30**	**20.52**	**20.84**	**20.87**	**20.33**	**21.15**
	RRDB [32]	21.57	21.18	19.61	20.93	20.57	20.40	20.74	20.24	20.66
	+Dropout ($p = 0.5$)	21.89	21.75	19.92	21.12	20.53	20.70	20.84	20.33	20.89
	+Alignment	22.29	21.95	20.21	21.40	20.76	20.85	21.03	20.38	21.11
	+TFD	**22.44**	**22.13**	**21.66**	**21.45**	**20.99**	**20.93**	**21.09**	**20.53**	**21.40**
	SwinIR [23]	22.18	21.90	20.56	21.32	20.89	20.79	20.98	20.45	21.13
	+Dropout ($p = 0.5$)	22.27	21.99	20.67	21.38	20.92	20.91	20.96	20.55	21.21
	+Alignment	22.34	22.07	20.69	21.48	21.02	20.98	21.12	20.53	21.28
	+TFD	**22.63**	**22.31**	**21.61**	**21.47**	**20.95**	**20.93**	**21.08**	**20.55**	**21.44**
Manga109	SRResNet [20]	18.42	18.75	18.32	18.30	18.60	18.53	18.25	18.43	18.45
	+Dropout ($p = 0.7$)	18.98	19.12	18.52	18.66	18.94	18.85	18.66	18.72	18.81
	+Alignment	19.18	19.46	19.90	19.02	19.27	19.17	18.98	19.01	19.25
	+TFD	**19.22**	**19.52**	**18.98**	**18.96**	**19.14**	**19.11**	**18.83**	**18.92**	**19.09**
	RRDB [32]	18.59	18.64	18.30	18.41	18.83	18.43	18.38	18.41	18.50
	+Dropout ($p = 0.5$)	18.73	19.03	18.72	18.60	19.15	18.81	18.59	18.71	18.79
	+Alignment	19.40	19.61	18.96	19.24	19.43	19.31	19.12	19.15	19.28
	+TFD	**19.28**	**18.64**	**19.09**	**19.05**	**19.21**	**19.09**	**18.84**	**18.91**	**19.01**
	SwinIR [23]	19.10	19.27	18.71	18.95	19.07	19.02	18.79	18.80	18.96
	+Dropout ($p = 0.5$)	19.15	19.30	18.83	19.03	19.12	18.98	18.75	18.84	19.00
	+Alignment	19.24	19.45	18.98	19.28	19.37	19.35	19.15	19.12	19.24
	+TFD	**19.20**	**19.37**	**19.34**	**18.91**	**19.17**	**19.12**	**18.89**	**18.90**	**19.34**