Dynamic Attention-Guided Diffusion for Image Super-Resolution

Brian B. Moser[1,2,3] Stanislav Frolov[1,2,3] Federico Raue[1] Sebastian Palacio[1] Andreas Dengel[1,2]
[1]German Research Center for Artificial Intelligence, Germany
[2]RPTU Kaiserslautern-Landau, Germany
[3]Equal Contribution
first.last@dfki.de

Abstract

Diffusion models in image Super-Resolution (SR) treat all image regions uniformly, which risks compromising the overall image quality by potentially introducing artifacts during denoising of less-complex regions. To address this, we propose "You Only Diffuse Areas" (YODA), a dynamic attention-guided diffusion process for image SR. YODA selectively focuses on spatial regions defined by attention maps derived from the low-resolution images and the current denoising time step. This time-dependent targeting enables a more efficient conversion to high-resolution outputs by focusing on areas that benefit the most from the iterative refinement process, i.e., detail-rich objects. We empirically validate YODA by extending leading diffusion-based methods SR3, DiffBIR, and SRDiff. Our experiments demonstrate new state-of-the-art performances in face and general SR tasks across PSNR, SSIM, and LPIPS metrics. As a side effect, we find that YODA reduces color shift issues and stabilizes training with small batches.

1. Introduction

The goal of image Super-Resolution (SR) is to enhance Low-Resolution (LR) into High-Resolution (HR) images [29]. Improvements to this field significantly impact many applications, like medical imaging, remote sensing, and consumer electronics [12,27,37]. Despite its long history, image SR remains a fascinating yet challenging domain due to its inherently ill-posed nature: any LR image can lead to several valid HR images, and vice versa [2,34]. Thanks to deep learning, SR has made significant progress [10]. Initial regression-based methods, such as early convolutional neural networks, work great at low magnification ratios [5, 22, 38]. However, they fail to produce high-frequency details at high magnification ratios (≥ 4) and generate over-smoothed results [28]. Such scale ratios require models capable of hallucinating realistic details that fit the overall image.

Recently, generative diffusion models have emerged with better human-rated quality compared to regression-based methods, but they also introduced new challenges [7, 16, 32, 40]. Their indiscriminate processing of image regions leads to computational redundancies and suboptimal enhancements. Some recent methods address the first issue and reduce computational demands by working in latent space like LDMs [31], by exploiting the relationship between LR and HR latents like PartDiff [43], or by starting with a better-initialized forward diffusion instead of pure noise like in CCDF [8]. Yet, strategies to adapt model capacity based on spatial importance remain underexplored.

This paper takes the first step toward addressing the second issue and challenges the common approach of SR diffusion models by asking: Do we need to update the entire image at every time step? We hypothesize that not all image regions require the same level of detail enhancement. For instance, a face in the foreground may need more refinement than a simple, monochromatic background. Recognizing this variability in the need for detail enhancement underscores a critical inefficiency in traditional diffusion methods. Treating all image regions uniformly risks compromising the overall image quality by introducing artifacts and shifts to low-complex regions. Unlike methods that target computational efficiency, we aim to boost image quality by minimizing distortions across different low-complex regions.

In response, we introduce a diffusion mechanism focusing on detail-rich areas using time-dependent and attention-guided masking. Our method, coined "You Only Diffuse Areas" (YODA), starts by obtaining an attention map that highlights regions that need more refinement. After identification, YODA systematically replaces highlighted regions with SR predictions during the denoising process. In particular, regions with high attention values (detail-rich & salient) are refined more often. Our approach is analogous to inpainting methods like RePaint [25], where only a pre-defined region is updated to generate complementing content. In YODA's case, however, the selected regions are time-dependent. To that end, we design a dynamic approach that creates expanding masks, starting from detail-rich regions and converging

toward the overall image.

A key advantage of YODA is its compatibility with existing diffusion models, allowing for a plug&play application. We integrate YODA with three models: SR3 [32] and DiffBIR [23] for face SR and SRDiff [20] for general SR. Interestingly, YODA achieves notable image quality improvements and also improves the training process. When training with smaller batch sizes, SR3 suffers from color shifts [6, 36] while YODA produces faithful color distributions. In summary, our work:

- introduces YODA, an attention-guided diffusion approach that emphasizes image areas through masked refinement. Thus, it refines detail-rich areas more often, which leads to higher image quality.
- demonstrates that attention-guided diffusion results in better training conditions, accurate color reproduction, and competitive perceptual quality results.
- empirically shows that YODA outperforms leading diffusion models in face and general SR tasks.
- reveals that YODA improves the training performance when using smaller batch sizes, which is crucial in limited hardware scenarios.

2. Background

Our method uses attention maps for attention-guided diffusion. We leverage the self-supervised DINO framework [4] to extract attention maps. Thus, this section introduces the main components: DDPMs [16] and the DINO [4]. We refer to the supplementary materials for a discussion of related methods, such as other diffusion approaches and spatial-selection SR methods.

2.1. DDPMs

Denoising Diffusion Probabilistic Models (DDPMs) employ two distinct Markov chains [16]: the first models the forward diffusion process q transitioning from an input $\mathbf{x}$ to a pre-defined prior distribution with intermediate states $\mathbf{z}_t$, $0 < t \leq T$, while the second models the backward diffusion process p, reverting from the prior distribution back to the intended target distribution $p\left(\mathbf{z}_0 \mid \mathbf{z}_T, \mathbf{x}\right)$. In image SR, we designate $\mathbf{x}$ as the LR image and the target $\mathbf{z}_0$ as the desired HR image. The prior distribution is usually Gaussian noise.

Forward Diffusion: In forward diffusion, an HR image $\mathbf{z}_0$ is incrementally modified by adding Gaussian noise over a series of time steps. This process can be mathematically represented as:

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t \mid \sqrt{1-\alpha_t}\,\mathbf{z}_{t-1}, \alpha_t \mathbf{I}) \tag{1}$$

The hyperparameters $0 < \alpha_{1:T} < 1$ represent the noise variance injected at each time step. It is possible to sample from any point in the noise sequence without needing to generate all previous steps through the following simplification [33]:

$$q(\mathbf{z}_t \mid \mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t \mid \sqrt{\gamma_t}\,\mathbf{z}_0, (1-\gamma_t)\mathbf{I}), \tag{2}$$

where $\gamma_t = \prod_{i=1}^{t}(1-\alpha_i)$. The intermediate step $\mathbf{z}_t$ is

$$\mathbf{z}_t = \sqrt{\gamma_t} \cdot \mathbf{z}_0 + \sqrt{1-\gamma_t} \cdot \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right) \tag{3}$$

Backward Diffusion: The backward diffusion process is where the model learns to denoise, effectively reversing the forward diffusion to recover the HR image. In image SR, the reverse process is conditioned on the LR image to guide the generation of the HR image:

$$p_\theta\left(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{x}\right) = \mathcal{N}\left(\mathbf{z}_{t-1} \mid \mu_\theta(\mathbf{z}_t, \mathbf{x}, \gamma_t), \Sigma_\theta(\mathbf{z}_t, \mathbf{x}, \gamma_t)\right) \tag{4}$$

The mean μ_θ depends on a parameterized denoising function f_θ, which can either predict the added noise ε_t or the underlying HR image $\mathbf{z}_0$. Following the standard approach of Ho et al. [16], we focus on predicting the noise. Hence, the mean is:

$$\mu_\theta(\mathbf{x}, \mathbf{z}_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta\left(\mathbf{x}, \mathbf{z}_t, \gamma_t\right)\right) \tag{5}$$

Following Saharia et al. [32], setting the variance of $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$ to $(1-\alpha_t)$ yields the subsequent refining step with $\varepsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\mathbf{z}_{t-1} \leftarrow \mu_\theta(\mathbf{x}, \mathbf{z}_t, \gamma_t) + \sqrt{1-\alpha_t}\varepsilon_t \tag{6}$$

Optimization: The optimization goal for DDPMs is to train the parameterized model to accurately predict the noise added during the diffusion process. The loss function used to measure the accuracy of the noise prediction is:

$$\mathcal{L}\left(\theta\right) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{z}_0)} \mathop{\mathbb{E}}_{t} \left\| \varepsilon_t - f_\theta\left(\mathbf{x}, \mathbf{z}_t, \gamma_t\right) \right\|_1 \tag{7}$$

2.2. DINO

DINO is a self-supervised learning approach for feature extractors on unlabeled data [4]. It employs a teacher and a student network, where the student learns to imitate the features learned by the teacher. The student gets only local views of the image (i.e., 96×96), whereas the teacher receives global views (i.e., 224×224). This setup encourages the student to learn "local-to-global" correspondences. The features learned through self-supervision are directly accessible in the self-attention modules. These self-attention maps provide information on the scene layout and object boundaries. We leverage the generality, availability, and robustness of these attention maps as a measure of an image's saliency to guide the diffusion process for significantly improved image quality. In another context, a similar approach has been applied to image compression, demonstrating its ability to capture essential image content in the attention maps [3].

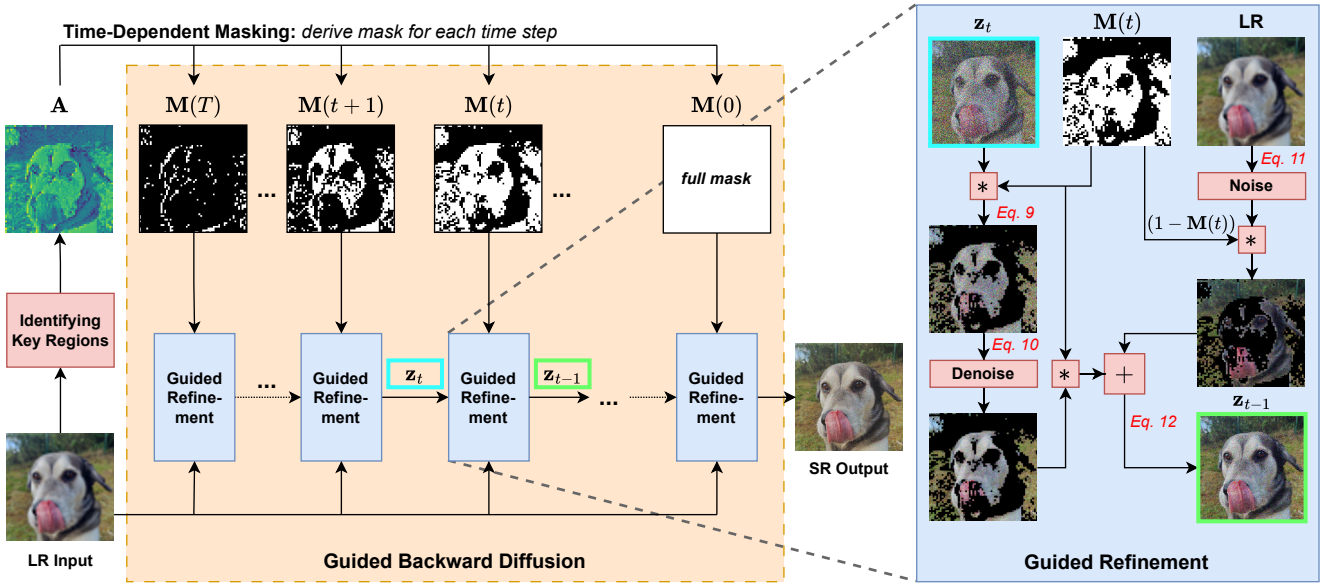


Figure 1. Overview of YODA. First, extract an attention map $\mathbf{A}$ from the LR input. Next, use the values of $\mathbf{A}$ to produce a time-dependent masking $\mathbf{M}(t)$. For $t : T \rightarrow 0$, the area of selected pixels expands from detail-rich regions to the whole image. Our diffusion process uses these masks for dynamic and attention-guided refinement, emphasizing regions differently. More specifically, it starts with masked areas that need refinement (derived from $\mathbf{z}_t$ and $\mathbf{M}(t)$) and LR regions, which retain the noise level needed for the next time step. Finally, the SR and LR areas are combined to form a whole image with no masked-out regions for the next iteration.

3. Methodology

Our proposed method, coined "You Only Diffuse Areas" (YODA), has three major phases:

- **Identifying Key Regions:** Estimate the weighting of pixel positions in a LR image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ with an attention map $\mathbf{A} \in \mathbb{R}^{H \times W}$.
- **Time-Dependent Masking:** Use $\mathbf{A}$ to define a time-dependent, binary mask generator function $\mathbf{M} : \mathbb{N}_0 \rightarrow \{0, 1\}^{H \times W}$. The generated masks identify salient areas at time step $t \in \mathbb{N}_0$ of the diffusion process.
- **Guided Backward Diffusion:** Concentrate the diffusion process on the regions identified by the time-dependent masking $\mathbf{M}(t)$ and generate a partially enhanced image by combining the prediction with complementing LR areas.

3.1. Identifying Key Regions

YODA starts by prioritizing areas in the input. This is achieved by generating an attention map $\mathbf{A}$ with $0 \leq \mathbf{A}_{i,j} \leq 1$ from the LR image $\mathbf{x}$. The greater the value of $\mathbf{A}_{i,j}$, the more refinements it receives. Note that extracting $\mathbf{A}$ is computationally efficient as it has to be generated only once for each image. For generating $\mathbf{A}$, we evaluated several approaches, including innate methods (i.e., not-learnable) and learnable methods, i.e., ResNet [15] and Transformer architectures [11]. For the latter, we leverage the DINO framework for its robustness in self-supervised learning, extracting refined attention maps directly from LR images without necessitating extra annotated data [4]. This choice is motivated by DINO's demonstrated efficacy in highlighting essential features within images using pre-existing models, e.g., for image compression [3].

Note that it is challenging to define important regions in SR because there is no clear definition. However, we observe that foreground objects are typically critical to human perception, while background areas seem less significant. Therefore, we choose to consider self-supervised methods to extract attention maps, which serve as an unbiased proxy for identifying objects and weightings in an image. This choice is based on the observation that self-supervised methods like DINO naturally focus on regions that generally capture human attention [4]. Our experiments with YODA confirm that emphasizing these areas improves performance, as later results will demonstrate. An additional overview of how DINO and the attention maps are used is shown in the supplementary materials. Next, we describe the process of creating time-dependently masks for the backward diffusion by utilizing the attention map $\mathbf{A}$.

3.2. Time-Dependent Masking

Given the LR input image $\mathbf{x}$ and the attention map $\mathbf{A}$, we introduce a novel strategy to dynamically focus the diffusion process on salient areas. Even though $\mathbf{A}$ is fixed, we

will use it to refocus the diffusion model during the backward diffusion process dynamically. Thus, we can leverage it to influence the number of refinement steps for each position. Therefore, for two positions (i, j) and (i', j') with $\mathbf{A}_{i,j} > \mathbf{A}_{i',j'}$, YODA applies more refinement steps to the location (i, j) than to (i', j'). Since $0 \leq \mathbf{A}_{i,j} \leq 1$, the number of diffusion steps employed to a specific position (i, j) is determined as a proportion of the maximum time steps, T. For instance, $\mathbf{A}_{i,j} = 0.7$ means (i, j) is refined during 70% of all diffusion steps. In addition, we introduce a lower bound hyperparameter $0 < l < 1$, ensuring that every region undergoes a minimum amount of refinements. In other words, the hyperparameter l reliably guarantees that every spatial position is refined at least $l \cdot T$ times. As the backward diffusion process progresses from time step T to 0, we can define the time-dependent masking process for any time step $T \geq t \geq 0$ approaching $t = 0$ as:

$$\mathbf{M}(t)_{i,j} = \begin{cases} 1, \text{ if } T \cdot (\mathbf{A}_{i,j} + l) \geq t \\ 0, \text{ otherwise} \end{cases} \tag{8}$$

Equation 8 ensures that the diffusion process gets applied a variable number of times for different regions, allowing the salient areas to diffuse over a longer time span. It is important to highlight that once a spatial position is marked for refinement, it continues to undergo refinement across all subsequent steps: $\mathbf{M}(t)_{i,j} \geq \mathbf{M}(t-k)_{i,j} \, \forall k > 0$. Figure 1 shows an example of our time-dependent masking. For each time step t, we can determine with $\mathbf{M}(t)_{i,j} = 1$ whether a given spatial position (i, j) should be refined or not.

3.3. Guided Backward Diffusion

YODA's guided diffusion process iteratively refines the image from a noisy state $\mathbf{z}_T$ to a HR state $\mathbf{z}_0$. This phase involves selectively refining areas based on the current time step's mask, $\mathbf{M}(t)$, and blending these refined areas with the unrefined, remaining LR regions, $(1 - \mathbf{M}(t))$. YODA ensures a seamless transition between refined and unrefined areas, improving image quality with a focus on key regions.

More specifically, at each time step t, the areas that will be refined when transitioning from t to $(t-1)$are determined based on the current iteration $\mathbf{z}_t$ and the current mask $\mathbf{M}(t)$:

$$\widetilde{\mathbf{z}}_t \leftarrow \mathbf{M}(t) \odot \mathbf{z}_t \tag{9}$$

Next, we divide the current image$\mathbf{z}_t$ into two components that will later be combined as $\mathbf{z}_{t-1}$ for the next time step: $\mathbf{z}_{t-1}^{SR}$, which is the refined image prediction, and $\mathbf{z}_{t-1}^{LR}$, the complementary LR image. The state $\mathbf{z}_{t-1}^{LR}$ represents unchanged LR areas by using $\mathbf{x}$ as the mean. Both components acquire the same noise level $\Sigma_\theta(\widetilde{\mathbf{z}}_t, \mathbf{x}, \gamma_t)$, and can be described by:

$$\mathbf{z}_{t-1}^{SR} \sim \mathcal{N}\left(\mu_\theta(\widetilde{\mathbf{z}}_t, \mathbf{x}, \gamma_t), \Sigma_\theta(\widetilde{\mathbf{z}}_t, \mathbf{x}, \gamma_t)\right) \tag{10}$$

$$\mathbf{z}_{t-1}^{LR} \sim \mathcal{N}\left(\mathbf{x}, \Sigma_\theta(\widetilde{\mathbf{z}}_t, \mathbf{x}, \gamma_t)\right) \tag{11}$$

Finally, YODA combines the complementing and non-overlapping image regions into a full image[1]:

$$\mathbf{z}_{t-1} \leftarrow \mathbf{M}(t) \odot \mathbf{z}_{t-1}^{SR} + (1 - \mathbf{M}(t)) \odot \mathbf{z}_{t-1}^{LR} \tag{12}$$

Consequently, the areas refined by $\mathbf{z}_t^{SR}$ expand as $t \to 0$, whereas the areas described by $\mathbf{z}_t^{LR}$ shrink in size. The new state, $\mathbf{z}_{t-1}$, now contains both SR and LR areas and, importantly, does not have any masked-out regions. As a result, $\mathbf{z}_{t-1}$ can be used in the next iteration step. This guided refinement is depicted on the right part of Figure 1.

Note that we use $\widetilde{\mathbf{z}}_t$ instead of $\mathbf{z}_t$ in Equation 10 and Equation 11. In our initial experiments, masking before the noise prediction (i.e., $\widetilde{\mathbf{z}}_t$) produced a marginal improvement in comparison to the full intermediate state $\mathbf{z}_t$ (around 0.1-0.2 dB in PSNR). We theorize that it is connected to the optimization target explained next. With $\widetilde{\mathbf{z}}_t$, we force the model inherently to focus locally, which, due to our selective loss function, would otherwise have to be learned.

3.4. Optimization

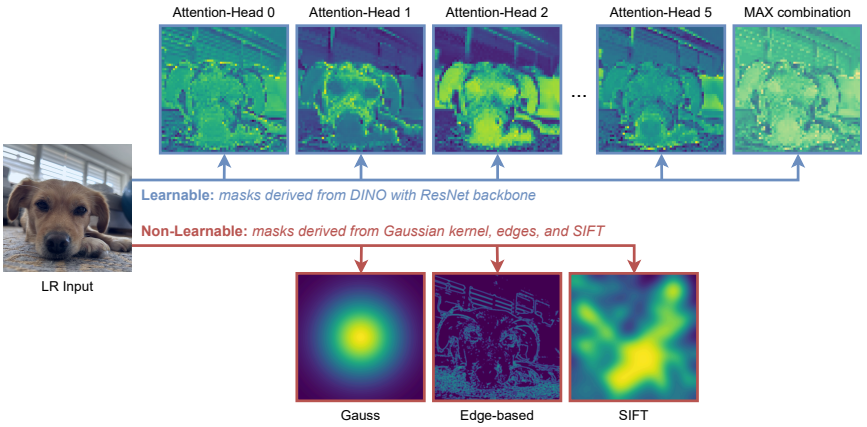
To confine the backward diffusion process to specific image regions as determined by the current time step $0 \leq t \leq T$ and the corresponding mask $\mathbf{M}(t)$, we adapt the training objective from Equation 7 as follows to focus on regions within the mask $\mathbf{M}(t)$. Thus, YODA optimizes only areas described by $\mathbf{M}(t)$:

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})} \mathop{\mathbb{E}}_{t} \left\| \mathbf{M}(t) \odot \left[\varepsilon_t - f_\theta\left(\mathbf{x}, \mathbf{z}_t, \gamma_t\right)\right] \right\|_1 \tag{13}$$

4. Experiments

We start by analyzing different methods for obtaining attention maps for YODA. Then, we evaluate YODA's performance in tandem with SR3 [32] and DiffBIR [23] for face, as well as SRDiff [20] for general SR. We chose SR3, DiffBIR, and SRDiff because they are the most prominent representative diffusion models for image SR in the respective tasks, where YODA can be integrated straightforwardly. However, YODA can be theoretically applied to any existing method. We present quantitative and qualitative results for both tasks, demonstrating YODA's high-quality results compared to the baselines using standard metrics such as PSNR, SSIM, and LPIPS [28]. All experiments were run on a single NVIDIA A100-80GB GPU. In the supplementary materials, we discuss the complexity of YODA and explore its potential synergies with other diffusion models. Also, we used a lower bound hyperparameter (see Section 3.2) of $l = 0.2$ in all experiments and were inspired by the rate-distortion trade-off presented by Ho et al. [16] that reaches the semantic compression stage at roughly $t = T - 0.2 \cdot T$.

[1] Equation 12 is similar to RePaint [25], a diffusion-based inpainting method While RePaint uses a constant mask for all time steps, YODA has time-dependent masks to dynamically control the updated image regions.



	Attention Maps	PSNR ↑	SSIM ↑	LPIPS ↓
no masks	PULSE	16.88	0.440	n.a.
	FSRGAN	23.01	0.620	n.a.
	SR3 Reported	23.04	0.650	n.a.
	SR3 Reproduced	22.35	0.646	0.082
innate	Gaussian	22.13	0.602	0.260
	Edge-based	22.93	0.648	0.151
	SIFT	22.84	0.678	0.095
DINO with ViT-S/8	Attention-Head 0	22.91	0.650	0.105
	Attention-Head 1	22.43	0.616	0.130
	Attention-Head 2	22.55	0.633	0.111
	Attention-Head 3	22.73	0.641	0.110
	Attention-Head 4	22.85	0.645	0.097
	Attention-Head 5	22.86	0.648	0.101
	Attention-AVG	23.25	0.663	0.122
	Attention-MAX	23.46	0.683	0.103
DINO with ResNet-50	Attention-Head 0	22.82	0.649	0.115
	Attention-Head 1	22.54	0.627	0.117
	Attention-Head 2	22.84	0.650	0.107
	Attention-Head 3	22.78	0.645	0.105
	Attention-Head 4	22.38	0.620	0.127
	Attention-Head 5	22.50	0.630	0.119
	Attention-AVG	23.55	0.682	0.093
	Attention-MAX	**23.84**	**0.695**	**0.072**

Figure 2. **(Left)** Comparison of various methods to extract attention maps used for our method (blue = low attention; yellow = high attention). Top row denotes maps derived from ResNet-50 using DINO. It shows various attention head outputs and the max aggregation of all attention maps (MAX). Bottom row denotes non-learnable methods, namely Gaussian, Edge-based, and using SIFT's points of interest. **(Right)** Comparison of different attention maps with SR3+YODA for $16 \rightarrow 128$ on CelebA-HQ. Aggregating the attention maps extracted with DINO and ResNet-50 backbone under the MAX strategy performs best. The attention maps are then used for dynamic binary masking.

4.1. Choosing Good Attention Maps

YODA relies on attention maps. Thus, we thoroughly evaluated different choices. We considered the pre-trained attention heads from the last layer of DINO with the respective backbone model, i.e., ResNet [15] and ViT [11]. For ResNet-50, we used a dedicated method to extract the attention maps from its weights [14]. A qualitative comparison of attention maps generated with DINO and ResNet-50 is shown in Figure 2 (left), demonstrating that YODA highlights perceptually essential areas (more visual results are in the appendix). Additionally, we test non-learnable methods to extract attention maps, also shown in Figure 2 (left):

- **Gaussian:** Placing a simple 2D Gaussian pattern at the center of the image provides a straightforward approach, which relies on the assumption that the essential parts of an image are centered.
- **Edge-based:** Using the Canny edge detector, the attention maps are defined by the edges of the image, where close edges are connected and blurred.
- **Scale-Invariant Feature Transform (SIFT):** Through Gaussian differences, SIFT [24] provides an attention map characterized by scale invariance. It produces an attention map by applying 2D Gaussian patterns around the points of interest.

DINO masks perform best: Figure 2 (right) presents our study on several baselines and masking variants. The straightforward Gaussian approach performs worst as it does not adapt to image features. The edge-based segmentation and SIFT methods improve the performance over the reproduced baseline using a small batch size. However, they underperform relative to the reported SR3 results, which used a larger batch size. In comparison, using DINO's attention maps for YODA shows significant improvements. We tested individual attention heads (0 to 5) independently, along with combination strategies that include averaging (AVG) and selecting the maximum value (MAX). The MAX combination achieved the best results compared to individual heads or the AVG combination.

ResNet produces more sensitive masks with more pixel updates: Figure 3 (left) investigates the ratio of diffused pixels using our time-dependent masking. The upper bound is 100%, where diffusion is applied across all locations in every time step (standard diffusion). Any result under 100% shows that not all pixels are diffused during all time steps. As can be seen, DINO with ResNet-50 produces higher attention values, resulting in more total pixel updates. Also, the high variance indicates a high adaptability. It can employ 100% for some samples, a characteristic not observed with ViT-S/8. Nevertheless, the ViT-S/8's improved performance compared to non-learnable methods and its low ratio make it attractive for future work on optimized inference speed based on sparser diffusion, e.g., LazyDiffusion [30].

ResNet progresses faster towards the whole image: Figure 3 (right) shows the ratio of diffused pixels depending on time steps with the MAX aggregation for ResNet-50 and ViT-S/8. As the backward diffusion progresses from T to 0, ResNet-50 initiates and incorporates the refinement of the whole image areas more quickly than ViT-S/8. For the

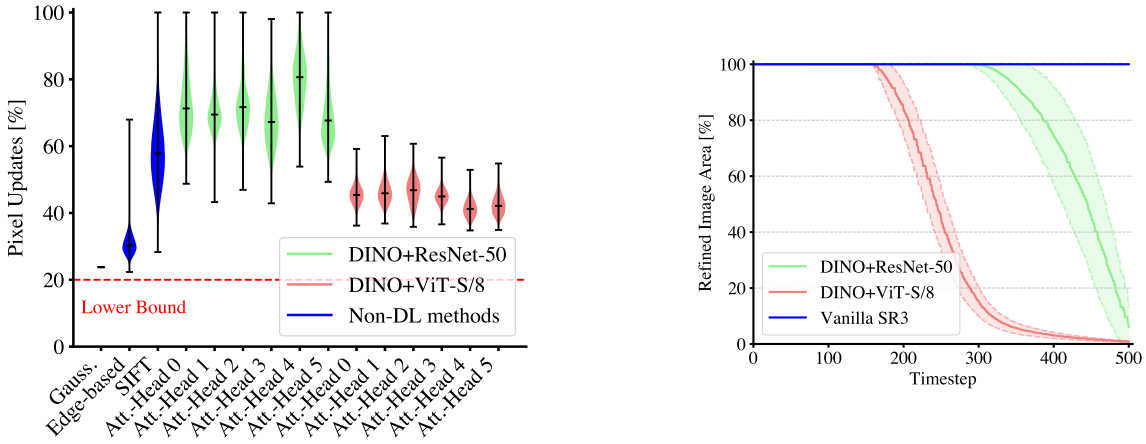


Figure 3. **(Left)** Ratio comparison between diffused pixels using our time-dependent masking approach and the total number of pixel updates in standard diffusion. On average, DINO with a ResNet-50 backbone leads to more pixel updates than the VIT-S/8 backbone. The lower bound, defined by l, is a threshold to eliminate areas that would never undergo diffusion. **(Right)** Refined image area in percentage across time steps for the MAX combination. Note that the sampling process goes from $T = 500$ to $T = 0$. ResNet-50 initiates the refinement process much earlier, advances more rapidly toward refining the entire image, and has a higher standard deviation.

first 200 time steps, the attention map derived by ViT-S/8 addresses less than 20% of the image area, whereas ResNet-50 has already developed to 100%. Therefore, we assume that the ResNet-50 backbone's superior performance is attributed to its faster progression toward refining the whole image. Intermediate diffusion results and error maps can be found in the supplementary materials.

Summary: As DINO with ResNet-50 and MAX aggregation performs best, we used it for all remaining main experiments.

4.2. Face Super-Resolution

We use FFHQ [18] for training and CelebA-HQ for testing [17]. All SR3 models were trained for 1M iterations as in Saharia et al. [32]. We evaluated three scenarios with bicubic degradation: $16 \rightarrow 128$, $64 \rightarrow 256$, and $64 \rightarrow 512$. Due to hardware limitations and missing quantitative results in the original publication of SR3, our experiments required a reduction from the originally used batch size of 256: we used a batch size of 4 for the $64 \rightarrow 512$, and 8 for the $64 \rightarrow 256$ scenario to fit on a single A100-80GB GPU. For blind face SR (unknown degradation between LR and HR, $64 \rightarrow 256$), we follow Lin et al. and test YODA with DiffBIR [23], which uses more complex degradation models (e.g., blurring), like introduced by Real-ESRGAN and others [38, 41].

Results: Table 1 shows significant improvements when SR3 (face SR) or DiffBIR (blind face SR) is coupled with YODA across all metrics. DiffBIR applies a diffusion process following an initial regression-based predictor and uses only 50 sampling steps. Consequently, smaller relative improvements are expected, but it shows that YODA is also efficient for more complex degradation models.

Scaling	Type	Model	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
4×	Regression	RRDB [38]	27.77	0.870	0.151	67.46
	Diffusion	SR3 [32]	17.98	0.607	0.138	80.72
	Diffusion	**SR3 + YODA**	**26.33**	**0.838**	**0.090**	**59.99**
8×	Regression	RRDB [38]	26.91	0.780	0.220	62.85
	Diffusion	PartDiff (K=25) [43]	n.a.	n.a.	0.222	n.a.
	Diffusion	PartDiff (K=50) [43]	n.a.	n.a.	0.217	n.a.
	Diffusion	SR3 [32]	17.44	0.631	0.147	66.20
	Diffusion	**SR3 + YODA**	**25.04**	**0.800**	**0.126**	**53.95**
8× (blind)	Diffusion	DiffBIR [23]	24.49	0.717	0.247	115.22
	Diffusion	**DiffBIR + YODA**	**24.56**	**0.718**	**0.245**	**111.93**

Table 1. Face SR results with 4× scaling ($64 \rightarrow 256$) and 8× scaling ($64 \rightarrow 512$) on CelebA-HQ with SR3 (non-blind) and DiffBIR (blind means unknown degradation type) standalone and combined with YODA. Note that RRDB [38] is also reported and that regression-based methods typically yield higher pixel-based scores (PSNR/SSIM) than generative approaches [32]. SR3 was trained for 1M steps and a reduced batch size of 4 and 8 instead of 256 to fit on a single A100-80GB GPU. Note that DiffBIR uses diffusion after an initial, regression-based predictor with only 50 sampling steps. Thus, smaller relative improvements are expected.

We explain the poor performance of SR3 with a phenomenon also observed by other works [6, 36]: color shifting, which we attribute to the reduced batch size necessitated by limited hardware access. Fitting SR3 on a single A100-80GB GPU required reducing the batch size from 256 to 8. Examples are shown in Figure 4 and in the appendix. Color shifting manifests in pronounced deviation in pixel-based metrics (PSNR/SSIM) but only slightly decreased perceptual quality (LPIPS). Our results suggest that YODA's role extends beyond mere performance enhancement. It actively mitigates color shifting and stabilizes training, especially when faced with hardware constraints. Due to selective refinement,

LR	SR3	SR3+YODA	HR
↑ PSNR:	11.811	18.069	
↑ SSIM:	0.3311	0.6588	
↓ LPIPS:	0.2847	0.2628	

Figure 4. SR3 and SR3+YODA reconstructions, $64 \rightarrow 256$ (4×). SR3 suffers from color shifting, as also observed by [6, 36]. YODA solves this issue and produces higher-quality reconstructions.

Scaling	Model	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
4×	SR3 [32]	28.3	0.67	0.10	74.99
	SR3 + YODA	**31.6**	**0.87**	**0.05**	**58.34**
8×	SR3 [32]	29.3	0.71	0.10	55.94
	SR3 + YODA	**31.5**	**0.84**	**0.07**	**49.02**

Table 2. Results without color-shifting on face SR by normalizing the channel-wise means to those of the HR ground-truth data for 4× scaling ($64 \rightarrow 256$) and 8× scaling ($64 \rightarrow 512$).

YODA maintains more of the LR image in less complex areas, reducing the risk of introducing artifacts/shifts during denoising while enhancing details in complex regions. With YODA, SR3 can be trained with a much smaller batch size.

Results without color-shifting: Table 2 shows the same experiments but with the channel-wise mean normalized by the ground-truth data, thus disentangling image quality improvements from color bias. While this procedure is not feasible in practice, it highlights YODA's significantly improved performance beyond reducing color shifting.

Analysis across attention regions: In Figure 5, we analyze the LPIPS scores across different attention value intervals within a single attention map $\mathbf{A}$ (using the MAX aggregation, 0.01 interval size). These intervals represent varying attention levels assigned by DINO to different regions of the image. The high LPIPS scores associated with bicubic upsampling in high-attention areas underscore the effectiveness of DINO in capturing perceptually significant regions. As a result, YODA significantly improves LPIPS scores across all attention regions, particularly in regions with higher attention values (highlighted by the trend curve). Please refer to the appendix for more details.

User study: In addition to the quantitative results on $16 \rightarrow 128$ (8×) provided in Figure 2 and inspired by Saharia et al., we conducted a user study. We selected 50 random test images, asked 45 participants which SR prediction is preferred, and plotted the preferences per image (see appendix). As a result, YODA was preferred 55.7% of the time over SR3 (44.3%). More details are in the appendix.

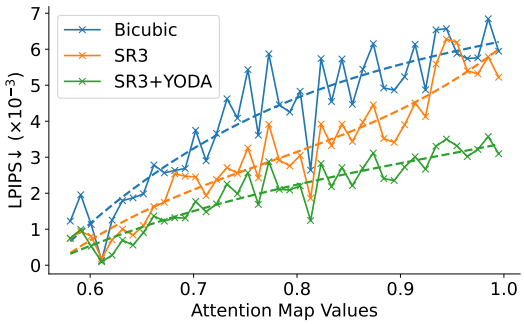


Figure 5. Regional LPIPS comparison across normalized attention values for CelebA, $64 \rightarrow 256$ (4×). We use 0.01 intervals and fit a polynomial through the means. High-attention areas are perceptually relevant and correspond to more difficult pixels (higher LPIPS). YODA reaches better scores, especially within high-attention areas. Note that dynamic masking stops around $t \approx 0.6 \cdot T$, see Figure 3.

4.3. General Super-Resolution

The experimental setup follows SRDiff [20], based on the setup of SRFlow and bicubic degradation [26]. For training, we employed 800 2K resolution images from DIV2K [1] and 2,650 from Flickr2K [35]. For testing, we used the DIV2K validation set (100 images). Moreover, we evaluated SR3, which was not originally tested on DIV2K.

Results: Table 3 shows the 4× general SR results on the DIV2K val. The reported values include regression-based methods, which typically yield higher pixel-based scores than generative models [32]. The disparity is due to PSNR/SSIM penalizing misaligned high-frequency details, a known challenge in the wider SR field [28]. We observe that results from SR3 underperform without YODA. We hypothesize that SR3 strongly depends on larger batch sizes and longer diffusion times. Combining SRDiff with YODA improves the quality in PSNR (+0.21db) and SSIM (+0.01), with a minor deterioration in LPIPS (+0.01). Nonetheless, when looking qualitatively at the predictions, one can observe that SRDiff strongly benefits from YODA, as shown in Figure 6. YODA produces much better LPIPS values for perceptually essential areas, i.e., hair and cars, but falls short in background areas, i.e., blurry grass or dark ground. The appendix contains more visual results.

Discussion: Overall, YODA's strengths are more significant when combined with SR3 than with SRDiff. A critical distinction between SR3 and SRDiff is the handling of conditional information, i.e., the LR image, which we identify as a potential contributor to the reduced perceptual score LPIPS. SRDiff employs an LR encoder that generates an embedding during the denoising phase. Meanwhile, SR3 directly uses the LR image during the backward diffusion.

Type	Methods	PSNR ↑	SSIM ↑	LPIPS ↓
Interpolation	Bicubic	26.70	0.77	0.409
Regression	EDSR [22]	28.98	0.83	0.270
	LIIF [5]	29.24	0.84	0.239
	RRDB [38]	29.44	0.84	0.253
GAN	RankSRGAN [42]	26.55	0.75	0.128
	ESRGAN [38]	26.22	0.75	0.124
Flow	SRFlow [26]	27.09	0.76	0.120
	HCFlow [21]	27.02	0.76	0.124
Flow + GAN	HCFlow++ [21]	26.61	0.74	0.110
VAE + AR	LAR-SR [13]	27.03	0.77	0.114
Diffusion	SR3 [32]	14.14	0.15	0.753
	SR3 + YODA	**27.24**	**0.77**	**0.127**
	SRDiff [20]	27.41	0.79	**0.136**
	SRDiff + YODA	**27.62**	**0.80**	0.146

Table 3. Quantitative results of 4× general SR on DIV2K val. YODA improves across pixel-centric metrics such as PSNR and SSIM, but yields a marginal decline in LPIPS.

Figure 6. Zoom-in regions of DIV2K images (first row). LPIPS is reported in the boxes (the lower, the better). YODA consistently produces better texture and more high-frequency details.

5. Conclusion

In this work, we presented "You Only Diffuse Areas" (YODA), an attention-guided diffusion-based image SR approach that emphasizes essential areas through time-dependent masking. YODA first extracts attention maps that reflect the pixel-wise saliency of each scene using a self-supervised, general-purpose vision encoder. The attention maps are then used to guide the diffusion process by focusing on key regions in each time step while providing a fusion technique to ensure that masked and non-masked image regions are correctly connected between two successive time steps. This targeting allows for a more efficient transition to high-resolution outputs, prioritizing areas that gain the most from iterative refinements, such as detail-intensive regions. Beyond better performance, YODA stabilizes training and mitigates the color shift issue when a reduced batch size constraints the underlying diffusion model. As a result, YODA consistently outperforms strong baselines like SR3, DiffBIR, and SRDiff while requiring less computational resources by using smaller batch sizes.

6. Limitations & Future Work

A notable constraint of this study is its dependence on a good saliency estimation. Even though DINO is trained to be a generic vision encoder, it has known limitations that will reflect on the quality of YODA [9]. Additionally, DINO is explicitly trained on input image resolutions such as 224×224, which may not suffice for image SR applications with much larger spatial sizes of the LR image. Meanwhile, the modularity of YODA allows for the saliency model to be switched once a better one becomes available. An ideal solution would be a scale-invariant extraction of attention maps, e.g., a more extended version of our SIFT-adapted approach. Lastly, YODA can be extended for other diffusion-based methods, e.g., latent-based methods like LDM [31] or unsupervised methods based on singular value decomposition like DDRM [19] or DDNM [39], which is orthogonal to our work (see appendix for more details).

7. Societal Impact

YODA can significantly benefit fields like medical imaging and remote sensing by improving visual quality. Yet, high-quality SR can also be exploited maliciously by adding realism to deceptive or misleading media content. Using SR methods responsibly and promoting transparency and ethical guidelines in deployment is crucial. Also, the reliance on pre-trained models for attention maps, such as DINO, may introduce biases inherent in the training data. These biases could affect the quality and fairness of the SR results, particularly in diverse or underrepresented populations. Future work should aim to mitigate these biases.

Acknowledgment

This work was supported by the EU project SustainML (Grant 101070408) and by Carl Zeiss Foundation through the Sustainable Embedded AI project (P2021-02-009).

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshop*, 2017. 7

[2] Saeed Anwar and Nick Barnes. Densely residual laplacian super-resolution. In *IEEE TPAMI*, 2020. 1

[3] Federico Baldassarre, Alaaeldin El-Nouby, and Hervé Jégou. Variable rate allocation for vector-quantized autoencoders. In *ICASSP*, 2023. 2, 3

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 3

[5] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, 2021. 1, 8

[6] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, H Kim, and S Yoon. Perception prioritized training of diffusion models. 2022 ieee. In *CVPR*, 2022. 2, 6, 7

[7] Hyungjin Chung, Eun Sun Lee, and Jong Chul Ye. Mr image denoising and super-resolution using regularized reverse diffusion. In *IEEE Transactions on Medical Imaging*, 2022. 1

[8] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *CVPR*, 2022. 1

[9] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 8

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. In *IEEE TPAMI*, 2015. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3, 5

[12] Walid El-Shafai, Anas M Ali, Samy Abd El-Nabi, El-Sayed M El-Rabaie, and Fathi E Abd El-Samie. Single image super-resolution approaches in medical images based-deep learning: a survey. *Multimedia Tools and Applications*, pages 1–37, 2023. 1

[13] Baisong Guo, Xiaoyun Zhang, Haoning Wu, Yu Wang, Ya Zhang, and Yan-Feng Wang. Lar-sr: A local autoregressive model for image super-resolution. In *CVPR*, 2022. 8

[14] Shir Gur, Ameen Ali, and Lior Wolf. Visualization of supervised and self-supervised neural networks via attribution guided factorization. In *AAAI*, 2021. 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 5

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1, 2, 4

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 6

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 6

[19] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *NeurIPS*, 35:23593–23606, 2022. 8

[20] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. In *Neurocomputing*, 2022. 2, 4, 7, 8

[21] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *ICCV*, 2021. 8

[22] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshop*, 2017. 1, 8

[23] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Ben Fei, Bo Dai, Wanli Ouyang, Yu Qiao, and Chao Dong. Diffbir: Towards blind image restoration with generative diffusion prior. *arXiv preprint arXiv:2308.15070*, 2023. 2, 4, 6

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004. 5

[25] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 1, 4

[26] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020. 7, 8

[27] Brian B. Moser, Stanislav Frolov, Federico Raue, Sebastian Palacio, and Andreas Dengel. Dwa: Differential wavelet amplifier for image super-resolution. In Lazaros Iliadis, Antonios Papaleonidas, Plamen Angelov, and Chrisina Jayne, editors, *ICANN*, 2023. 1

[28] Brian B. Moser, Federico Raue, Stanislav Frolov, Sebastian Palacio, Jörn Hees, and Andreas Dengel. Hitchhiker's guide to super-resolution: Introduction and recent advances. In *IEEE TPAMI*, 2023. 1, 4, 7

[29] Brian B Moser, Arundhati S Shanbhag, Federico Raue, Stanislav Frolov, Sebastian Palacio, and Andreas Dengel. Diffusion models, image super-resolution and everything: A survey. *arXiv preprint arXiv:2401.00736*, 2024. 1

[30] Yotam Nitzan, Zongze Wu, Richard Zhang, Eli Shechtman, Daniel Cohen-Or, Taesung Park, and Michaël Gharbi. Lazy diffusion transformer for interactive image editing, 2024. 5

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 8

[32] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. In *IEEE TPAMI*, 2022. 1, 2, 4, 6, 7, 8

[33] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2

[34] Wanjie Sun and Zhenzhong Chen. Learned image downscaling for upscaling using content adaptive resampler. In *IEEE Transactions on Image Processing*, 2020. 1

[35] Radu Timofte, Shuhang Gu, Jiqing Wu, and Luc Van Gool. Ntire 2018 challenge on single image super-resolution: Methods and results. In *CVPR Workshop*, 2018. 7

[36] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv:2305.07015*, 2023. 2, 6, 7

[37] Xuan Wang, Jinglei Yi, Jian Guo, Yongchao Song, Jun Lyu, Jindong Xu, Weiqing Yan, Jindong Zhao, Qing Cai, and Haigen Min. A review of image super-resolution approaches based on deep learning and applications in remote sensing. *Remote Sensing*, 14(21):5423, 2022. 1

[38] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV Workshop*, 2018. 1, 6, 8

[39] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022. 8

[40] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *CVPR*, 2022. 1

[41] Jun Xiao, Rui Zhao, Shun-Cheung Lai, Wenqi Jia, and Kin-Man Lam. Deep progressive convolutional neural network for blind super-resolution with multiple degradations. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2856–2860. IEEE, 2019. 6

[42] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *ICCV*, 2019. 8

[43] Kai Zhao, Alex Ling Yu Hung, Kaifeng Pang, Haoxin Zheng, and Kyunghyun Sung. Partdiff: Image super-resolution with partial diffusion models. *arXiv:2307.11926*, 2023. 1, 6