A Implementation Details

A.1 Derivation Details in Equation 2

We propose to compute the distribution of latents $p(\boldsymbol{z}_t|\boldsymbol{y}, \boldsymbol{c}_s, \boldsymbol{c}_r)$ conditioned on degraded image $\boldsymbol{y}$, semantic prompt $\boldsymbol{c}_s$ and restoration prompt $\boldsymbol{c}_r$. Using the Bayes' decomposition similar to score-based inverse problem [58, 59], we have

$$p(\boldsymbol{z}_t|\boldsymbol{y}, \boldsymbol{c}_s, \boldsymbol{c}_r) = p(\boldsymbol{z}_t, \boldsymbol{y}, \boldsymbol{c}_s, \boldsymbol{c}_r)/p(\boldsymbol{y}, \boldsymbol{c}_s, \boldsymbol{c}_r). \tag{3}$$

Then, we compute gradients with respect to $\boldsymbol{z}_t$, and remove the gradients of input condition $\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}, \boldsymbol{c}_s, \boldsymbol{c}_r) = 0$ as:

$$\begin{aligned}
&\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t|\boldsymbol{y}, \boldsymbol{c}_s, \boldsymbol{c}_r) \\
&\quad = \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t, \boldsymbol{y}, \boldsymbol{c}_s, \boldsymbol{c}_r) && (4)\\
&\quad = \nabla_{\boldsymbol{z}_t} \log[p(\boldsymbol{c}_s) \cdot p(\boldsymbol{z}_t|\boldsymbol{c}_s) \cdot p(\boldsymbol{y}, \boldsymbol{c}_r|\boldsymbol{c}_s, \boldsymbol{z}_t)] && (5)\\
&\quad = \nabla_{\boldsymbol{z}_t} \log[p(\boldsymbol{z}_t|\boldsymbol{c}_s) \cdot p(\boldsymbol{y}, \boldsymbol{c}_r|\boldsymbol{c}_s, \boldsymbol{z}_t)] && (6)\\
&\quad = \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t|\boldsymbol{c}_s) + \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}, \boldsymbol{c}_r|\boldsymbol{c}_s, \boldsymbol{z}_t). && (7)
\end{aligned}$$

We assume $\boldsymbol{y}$ is generated through a degradation pipeline as $\boldsymbol{y} = \mathrm{Deg}(\boldsymbol{x}, \boldsymbol{c}_r)$, thus it is independent of $\boldsymbol{c}_s$ with $\boldsymbol{x}$ and $\boldsymbol{c}_r$ provided as condition. Removing redundant $\boldsymbol{c}_s$ condition, the second term in the last equation can be approximated as:

$$\begin{aligned}
&\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}, \boldsymbol{c}_r|\boldsymbol{c}_s, \boldsymbol{z}_t) \\
&\quad \approx \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}, \boldsymbol{c}_r|\boldsymbol{z}_t) && (8)\\
&\quad = \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{c}_r|\boldsymbol{z}_t) + \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\boldsymbol{z}_t, \boldsymbol{c}_r) && (9)\\
&\quad = \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\boldsymbol{z}_t, \boldsymbol{c}_r) && (10)
\end{aligned}$$

In summary of the above equations, we derive the Equation 2 in the main manuscript

$$\begin{aligned}
&\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t|\boldsymbol{y}, \boldsymbol{c}_s, \boldsymbol{c}_r) \\
&\approx \underbrace{\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t|\boldsymbol{c}_s)}_{\text{Semantic-aware (frozen)}} + \underbrace{\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\boldsymbol{z}_t, \boldsymbol{c}_r),}_{\text{Restoration-aware (learnable)}}
\end{aligned} \tag{11}$$

where $\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\boldsymbol{z}_t, \boldsymbol{c}_r)$ is synthesized using stochastic degradation pipeline $\boldsymbol{y} = \mathrm{Deg}(x, \boldsymbol{c}_r)$ to train our ControlNet.

A.2 Pseudo Code for Degradation Synthesis

To support the learning of restoration-aware term $\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{y}|\boldsymbol{z}_t, \boldsymbol{c}_r)$, we synthesize the degradation image $\boldsymbol{y}$ using clean image $\boldsymbol{x}$ with the algorithm presented in Algorithm 1. First, we randomly choose one from Real-ESRGAN pipeline and our parameterized degradation. Then the degraded image from Real-ESRGAN pipeline is paired with restoration prompt $\boldsymbol{c}_r$ =*"Remove all degradation"*. In our parameterized degradation, all processes are paired with restoration prompts $\boldsymbol{c}_r$ listed in Table 2 of the main manuscript (*e.g.*, *Deblur with sigma 3.0*).

Algorithm 1 `SPIRE` Degradation Pipeline in Training

Inputs: $\boldsymbol{x}$: Clean image
Outputs: $\boldsymbol{y}$: Degraded image; $\boldsymbol{c}_r$: Restoration prompt

type ← RandChoice(Real-ESRGAN, Param)
if type = Real-ESRGAN **then** // Real-ESRGAN degradation
  $\boldsymbol{y} \leftarrow \boldsymbol{x}$
  Deg ← Random(Real-ESRGAN-Degradation)
  for Process **in** Deg **do**:
    $\boldsymbol{y} \leftarrow$ Process($\boldsymbol{y}$)
  end for
  $\boldsymbol{c}_r \leftarrow$ *"Remove all degradation"*
else// Parameterized degradation
  $\boldsymbol{c}_r \leftarrow \emptyset$
  $\boldsymbol{y} \leftarrow \boldsymbol{x}$
  Deg ← Random(Parametrized-Degradation)
  for Process, $\boldsymbol{c}_{rp}$ in Deg **do**:
    $\boldsymbol{y} \leftarrow$ Process($\boldsymbol{y}$, $\boldsymbol{c}_{rp}$)
    $\boldsymbol{c}_r \leftarrow$ Concat($\boldsymbol{c}_r, \boldsymbol{c}_{rp}$)
  end for
end if
return $\boldsymbol{y}$, $\boldsymbol{c}_r$

Fig. 8: More semantic prompting for images with multiple objects.

B More Ablation Study

Tab. 6 provides more comprehensive ablations of text prompts by providing different information to our image-to-image baseline. Semantic prompts significantly improve image quality as shown in better FID and CLIP-Image, but reduce the similarity with ground truth image. Restoration types and parameters embedded in the restoration prompts both improve image quality and fidelity. Tab. 7 presents a comparison of our skip feature modulation f_{skip} with that in StableSR [67] which modulates both skip feature f_{skip} from encoder and upsampling feature f_{up} from decoder. We observe that modulating f_{up} does not bring obvious improvements. One possible reason is that γ and β of the middle layer adapts to the feature in the upsampling layers.

C Multiple Objects Semantic Prompting

Besides single semantic restoration, real applications may involve multiple objects with different semantic categories (*e.g.* Fig. 8). In each column, we guide

Method	Sem	Res Type	Res Param	FID↓	LPIPS↓	PSNR↑	SSIM↑	CLIP_{im}↑	CLIP_{tx}↑
Ours	✗	✗	✗	13.60	0.221	23.65	0.664	0.939	0.300
Ours	✓	✗	✗	11.71	0.226	23.55	0.663	0.941	0.305
Ours	✓	✓	✗	11.58	0.223	23.61	0.665	0.942	0.305
Ours	✓	✓	✓	**11.34**	**0.219**	23.61	0.665	**0.943**	**0.306**

Table 6: Ablation of prompts provided during both training and testing. We use an image-to-image model with our modulation fusion layer as our baseline. Providing semantic prompts significantly increases the image quality (1.9 lower FID) and semantic similarity (0.002 CLIP-Image), but results in worse pixel-level similarity. In contrast, degradation type information embedded in restoration prompts improves both pixel-level fidelity and image quality. Utilizing degradation parameters in the restoration instructions further improves these metrics.

Method	Modulate f_{skip}	Modulate f_{up}	Relative Param	FID↓	LPIPS↓
Ours w/ prompts	✗	✗	1	12.14	0.223
Ours w/ prompts	✓	✓	1.06	11.21	0.219
Ours w/ prompts	✓	✗	1.03	11.34	0.219

Table 7: Ablation of the architecture. Modulating the skip feature f_{skip} improves the fidelity of the restored image with 3% extra parameters in the adaptor, while further modulating the backbone features f_{up} does not bring obvious advantage.

the upper part of the image with *peppers*, *bananas* or *leaves*, while the lower part can be restored as *potatoes* or *stones*. Thanks to the cross attention mechanism, multiple semantics can be spatially decoupled and recombined following the user's prompts, thus yielding better restoration for both objects.

D More Restoration Prompting

Fig. 9 shows the application of restoration prompt on images with different degradations and content, including Midjourney image and real-world cartoon. Since these images are not in our training data domain, a blind enhancement with prompt *"Remove all degradation"* can not achieve satisfying results. Utilizing restoration prompting (*e.g.*, *"Upsample to 6.0x; Deblur with sigma 2.9;"*) can successfully guide our model to improve the details and color tones of the Midjourney image. In the right half, a manually designed restoration prompt also reduces the jagged effect to smooth the lines in the cartoon image.

To study whether the model follows restoration instructions, a dense walking of restoration prompt is presented in Fig. 10. From left to right, we increase the strength of denoising in the restoration prompt. From top to the bottom, the strength of deblurring gets larger. The results demonstrate that our restoration framework refines the degraded image continuously following the restoration prompts

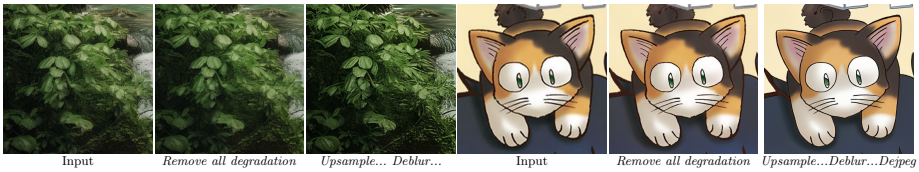

Fig. 9: Restoration prompting for images from internet.

E Real-world images

Although our model is trained on synthetic degradation, it generalize to real-world data RealPhoto60 [78], as shown in the Fig. 11. Compared to a model without semantic prompt, the synthetic semantic prompts from LLAVA [35] enhance fine-level details in Fig. 13 (*e.g.*, grass under sheep in the upper left figure, and the staircase in the mountain in lower right photo). These results demonstrate an additional potential advantage of employing language prompts in real-world restoration: the ease of leveraging the logical reasoning capabilities in pre-trained large language models.

F Limitation

Although our framework can generate high-fidelity results following semantic and restoration prompts, it is prone to occasional hallucinations. As shown in Fig. 12, the image quality is degraded and the semantic is unclear when the input prompt (*"Snow leopard"*) is misaligned with the ground truth (*"Panda bear"*). Instead of relying on user input or frozen language models, one future direction can be fine-tuning multimodal language models to automatically provide more accurate instructions, thus reducing hallucinations. In addition, we plan to scale up our model parameters and extend it to more diverse and realistic degradation types in our future work.

Fig. 12: Hallucinations when the prompt is unmatched with input image restorations.

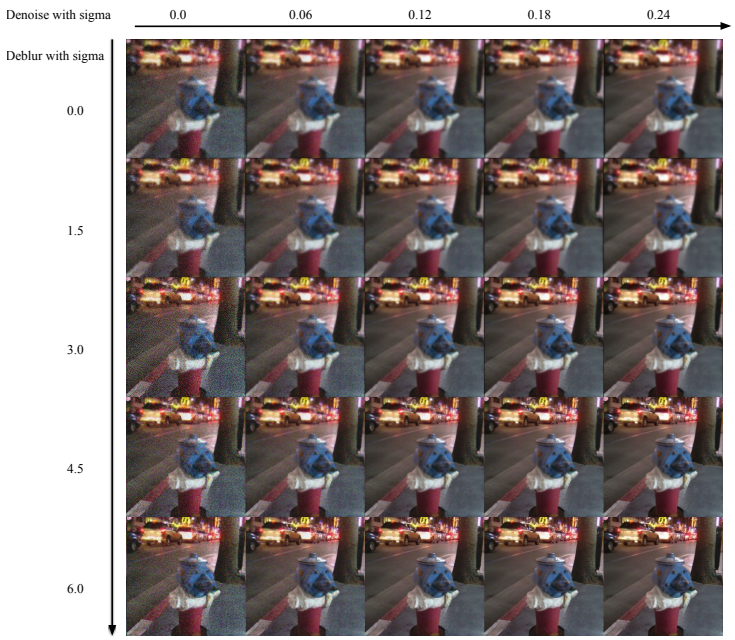

Fig. 10: Prompt space walking visualization for the restoration prompt. Given the same degraded input (upper left) and empty semantic prompt ∅, our method can decouple the restoration direction and strength via only prompting the **quantitative number in natural language**. An interesting finding is that our model learns a continuous range of restoration strength from discrete language tokens.

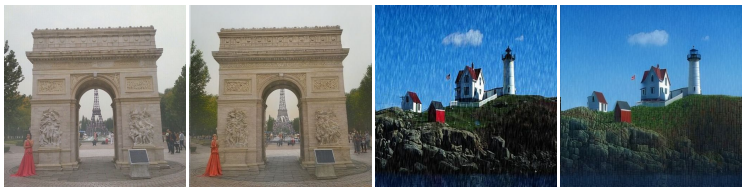

Fig. 14: Image restoration for unseen degradations.

G Mixed and universal degradation.

Our method can also restore mixed degradation (Figure 1 and Table 1 in the paper). For unseen degradations such as haze or rain, our pretrained model can still handle them properly, as shown in the figure below, since our pretrained prior and training data contains those concepts (*e.g.*, "A clear sky").

Fig. 11: Qualitative result on real-world images.

Model / Task	×4 super-resolution	denoising	de-jpeg
SwinIR (task-specific)	0.309	0.361	0.319
Ours	**0.265**	**0.305**	**0.214**

Table 8: Comparison with task-specific SwinIR.

H Comparison of inference cost

Our method takes 1.4s (50 DDIM steps) to run on TPUv4 and 1130 GFLOPS per step). Our UNet model has 1240 M (275 M trainable) parameters. The overall computation cost is comparable with 1203 M StableSR (19.3s on GPU) and 1510 M DiffBir (based on Stable Diffusion, 10.9s on GPU), and less than SUPIR (based on 2.6B SD-XL).

I Comparison with more models

We follow the test set design of task-specific SwinIR. In the table 8, our method outperforms the task-specific SwinIR and achieves lower LPIPS in evaluation. Following StableSR and Real-esrgan, our model is trained on large-scale open-domain images with ESRGAN degradations, which has a noticeable difference

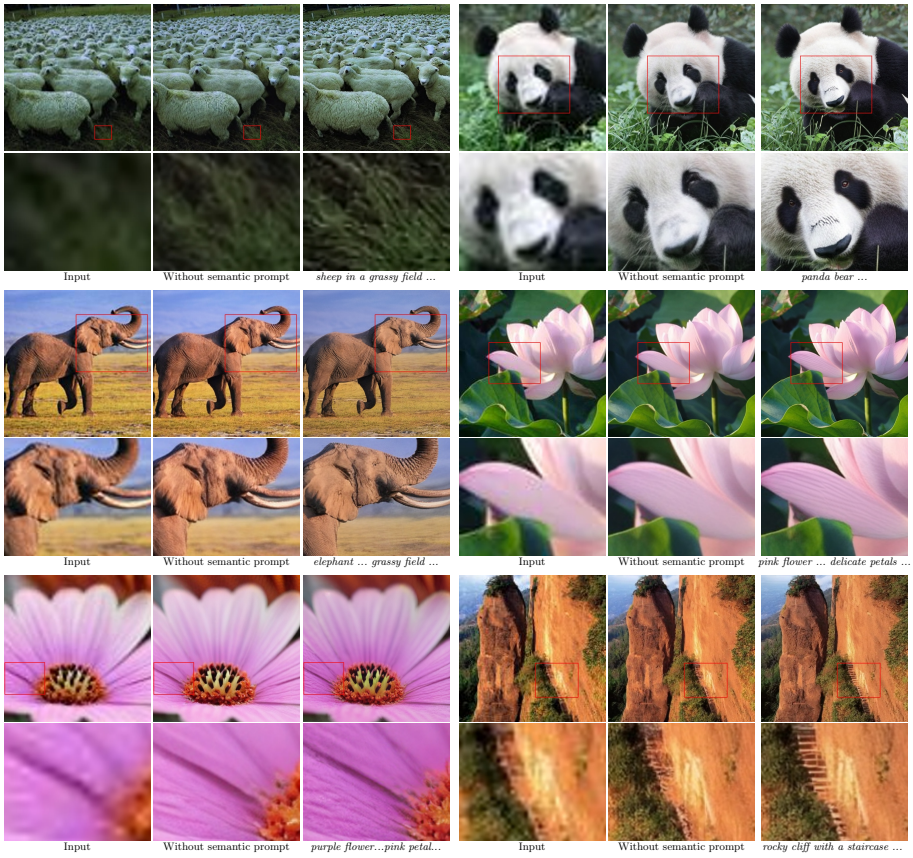

Fig. 13: Real-world examples showing the effect of semantic prompts.

with the degradation in all-in-one restoration mentioned by reviewers (*e.g.*, DA-CLIP considers raindrop but Real-esrgan does not). Thus, comparing our framework and other concurrent work (*e.g.*, DiffBir) to all-in-one restoration techniques proves difficult. To alleviate the concern, we evaluate our framework on the denoising testset of CBSD68 and our method achieves a comparable LPIPS (0.305) with DA-CLP (0.294).

J More Visual Comparison

More visual comparisons with baselines are provided in Fig. 15.

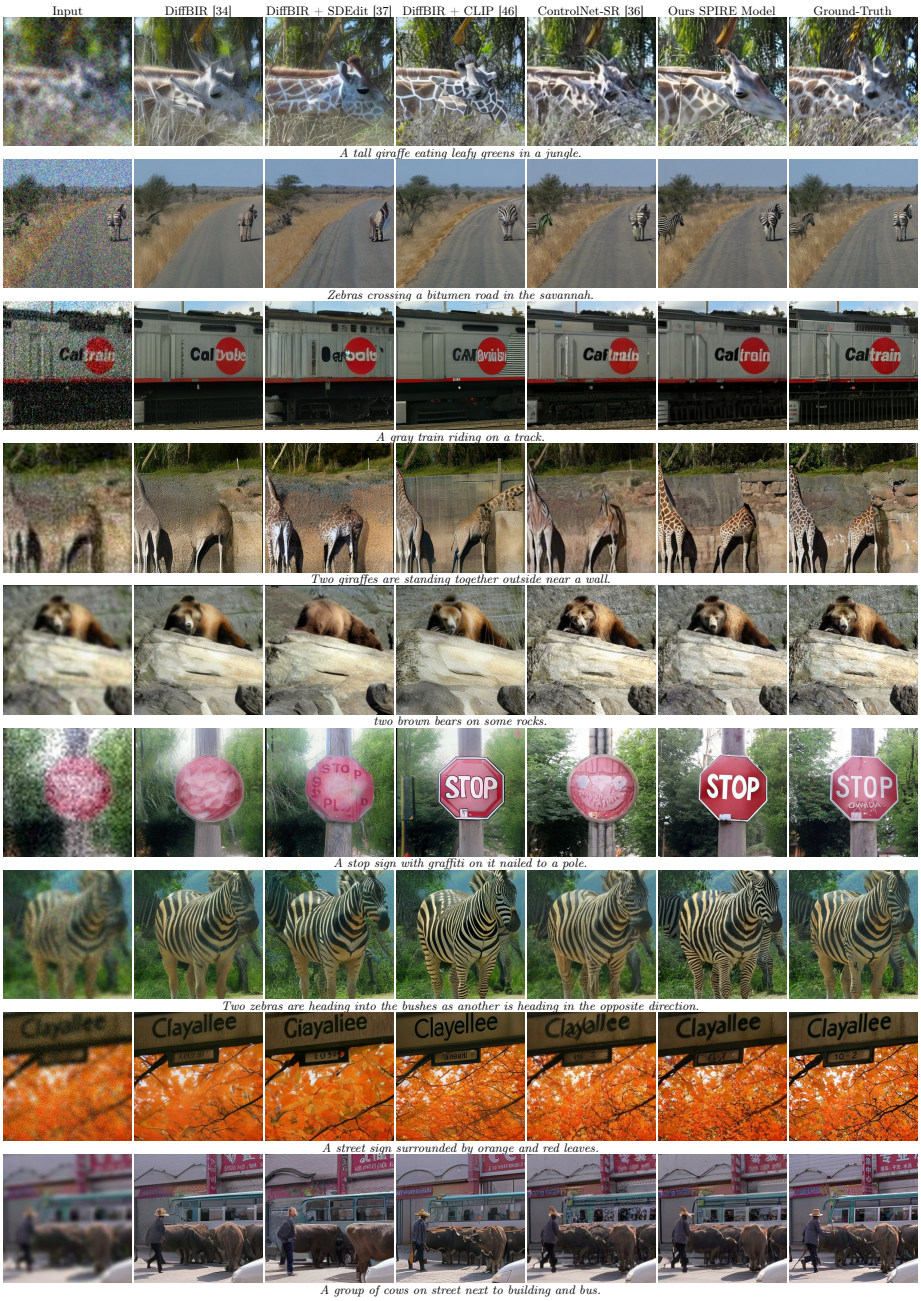

Fig. 15: Main visual comparison with baselines. (Zoom in for details)