6. Appendix

6.1. Derivation for Classifier-free Guidance with Three Conditionings

We introduce separate guidance scales like InstructPix2Pix to enable separately trading off the strength of each conditioning. The modified score estimate for our model is derived as follows. Our generative model learns $P(z|c_T,\ c_{I_{\text{sty}}},\ c_{I_{\text{in}}})$, which corresponds to the probability distribution of the image latents $z = \mathcal{E}(x)$ conditioned on an input image $c_{I_{\text{in}}}$, a reference style image $c_{I_{\text{sty}}}$, and a text instruction c_T. We arrive at our particular classifier-free guidance formulation by expressing the conditional probability as follows:

$$\begin{aligned} P(z|c_T,\ c_{I_{\text{sty}}},\ c_{I_{\text{in}}}) &= \frac{P(z, c_T,\ c_{I_{\text{sty}}},\ c_{I_{\text{in}}})}{P(c_T,\ c_{I_{\text{sty}}},\ c_{I_{\text{in}}})} \\ &= \frac{P(c_T|c_{I_{\text{sty}}}, c_{I_{\text{in}}}, z)P(c_{I_{\text{sty}}}|c_{I_{\text{in}}}, z)P(c_{I_{\text{in}}}|z)P(z)}{P(c_T,\ c_{I_{\text{sty}}},\ c_{I_{\text{in}}})} \end{aligned} \tag{10}$$

Diffusion models estimate the score [30] of the data distribution, i.e. the derivative of the log probability. Taking the logarithm of the expression above yields the following:

$$\begin{aligned} \log(P(z|c_T,\ c_{I_{\text{sty}}},\ c_{I_{\text{in}}})) = &\ \log(P(c_T|c_{I_{\text{sty}}}, c_{I_{\text{in}}}, z)) \\ &+ \log(P(c_{I_{\text{sty}}}|c_{I_{\text{in}}}, z)) \\ &+ \log(P(c_{I_{\text{in}}}|z)) \\ &+ \log(P(z)) \\ &- \log(P(c_T,\ c_{I_{\text{sty}}},\ c_{I_{\text{in}}})) \end{aligned} \tag{11}$$

Taking the derivative and rearranging, we obtain:

$$\begin{aligned} \nabla_z \log(P(z|c_T,\ c_{I_{\text{sty}}},\ c_{I_{\text{in}}})) = &\ \nabla_z \log(P(z)) \\ &+ \nabla z \log(P(c_{I_{\text{in}}}|z)) \\ &+ \nabla z \log(P(c_{I_{\text{sty}}}|c_{I_{\text{in}}}, z)) \\ &+ \nabla_z \log(P(c_T|c_{I_{\text{sty}}}, c_{I_{\text{in}}}, z)) \end{aligned} \tag{12}$$

This corresponds to the following formulation of classifier-free guidance, with three classifier-free-guidance scales:

$$\begin{aligned} \tilde{e}_\theta(z_t, c_{I_{\text{in}}}, c_{I_{\text{sty}}}, c_T) = &\ e_\theta(z_t, \varnothing, \varnothing, \varnothing) \\ &+ s_{I_{\text{in}}} \cdot (e_\theta(z_t, c_{I_{\text{in}}}, \varnothing, \varnothing) - e_\theta(z_t, \varnothing, \varnothing, \varnothing)) \\ &+ s_{I_{\text{sty}}} \cdot \big(e_\theta(z_t, c_{I_{\text{in}}}, c_{I_{\text{sty}}}, \varnothing) - e_\theta(z_t, c_{I_{\text{in}}}, \varnothing, \varnothing)\big) \\ &+ s_T \cdot \big(e_\theta(z_t, c_{I_{\text{in}}}, c_{I_{\text{sty}}}, c_T) - e_\theta(z_t, c_{I_{\text{in}}}, c_{I_{\text{sty}}}, \varnothing)\big) \end{aligned} \tag{13}$$

6.2. Training Details

In training, we initialize our model from the InstructPix2Pix checkpoint. Depending on the specialization, we train as few as six steps at a resolution of 256×256, with a total batch size of 256 samples. For a fair comparison, we maintain the same RL training hyperparameters as in D3PO [80] without conducting hyperparameter optimization. Additionally, we do not optimize for the best text prompt for each edit type, as we design SPIE to leverage the visual prompt as the primary carrier of semantic information, minimizing reliance on extensive textual descriptions.

The selection of hyperparameters λ and α requires balancing multiple objectives for optimal performance. While our recommended values work well across most scenarios, these parameters can be slightly fine-tuned based on the model's zero-shot capabilities to further optimize learning convergence. This flexibility allows practitioners to adapt the framework to specific task requirements while maintaining robust performance.

6.3. Real Image Experiment Details

Our evaluations focus on the ability to perform localized edits in complex scenes containing multi-level information, including local objects, global layout, and background environments that must remain unchanged unless explicitly instructed. We use two datasets for our experiments.

With the Oxford RobotCar Dataset [42], we train SPIE to perform seven types of edits: adding dense snow on the road, adding sparse snow on the road, adding rain on the road, adding sand on the road, changing the road to gold, changing the road to wood, and changing the entire scene to nighttime (using a full-frame mask).

With the Places Dataset [90], we train our model on six additional edit types: changing water to gold, changing a bed to leather, changing a building facade to steel, changing a telephone booth to stone, changing a lighthouse to terracotta, changing a train to wood.

When selecting edit types for our experiments, we aimed to cover a diverse range of materials, textures, and environmental modifications to demonstrate the versatility and robustness of our approach. This diverse selection allows us to assess our method's performance across both realistic modifications (weather changes) and more stylistic transformations (material changes), while testing its ability to maintain structural coherence across different scene types, object scales, and edit complexities.

To prevent overfitting on spurious cues, we alternate between five conditioning style images related to the same text prompt during training. Our evaluations cover 29,500 images at a resolution of 512×512 across both datasets and edit types (2,500 images for each of the seven Oxford RobotCar edits and 2,000 images for each of the six Places edits).

For baseline comparisons, we evaluate the InstructPix2Pix checkpoint from which we initialize our model, and the best publicly available versions of HIVE, HQ-Edit, and MagicBrush based on StableDiffusion v1.5. Some evaluation results on standardized benchmarks were taken directly from the ones reported in the original MagicBrush [84] and Emu Edit [63] papers.

While our model is trained at 256×256 resolution, we find it generalizes well to 512×512 resolution at inference time. We generate qualitative results at 512×512 resolution with 100 denoising steps using an Euler ancestral sampler with denoising variance schedule proposed by Karras et al. [31]. Editing an image with our model takes approximately 9 seconds on an A100 GPU.

6.4. Sim-to-real Experiment Details

Google Robot Evaluation Setup	Policy	Open / Close Drawer		
		Open	Close	Average
Real Eval	RT-1 (Converged)	0.815	0.926	0.870
	RT-1 (15%)	0.704	0.889	0.796
	RT-1-X	0.519	0.741	0.630
	RT-1 (Begin)	0.000	0.000	0.000
SIMPLER Eval (Variant Aggregation)	RT-1 (Converged)	0.270	0.376	0.323
	RT-1 (15%)	0.212	0.323	0.267
	RT-1-X	0.069	0.519	0.294
	RT-1 (Begin)	0.005	0.132	0.069
	MMRV↓	0.000	0.130	0.083
	Pearson r↑	0.915	0.756	0.964
SIMPLER Eval (Visual Matching)	RT-1 (Converged)	0.601	0.861	0.730
	RT-1 (15%)	0.463	0.667	0.565
	RT-1-X	0.296	0.891	0.597
	RT-1 (Begin)	0.000	0.278	0.139
	MMRV↓	0.000	0.130	0.083
	Pearson r↑	0.987	0.891	0.972
SPIE	RT-1 (Converged)	0.471	0.810	0.640
	RT-1 (15%)	0.259	0.619	0.439
	RT-1-X	0.180	0.608	0.394
	RT-1 (Begin)	0.021	0.058	0.040
	MMRV↓	0.000	0.000	0.000
	Pearson r↑	0.917	0.978	0.966

Table 4. Real-world, standard SIMPLER environment, and our visually-edited environment evaluation results across different policies on the Google Robot "(open/close) (top/middle/bottom) drawer" task. We present success rates for the 'Variant Aggregation' and 'Visual Matching' approaches, as well as our novel visually-edited environments with seven material styles. We calculate the Mean Maximum Rank Violation ('MMRV', lower is better) and the Pearson correlation coefficient ('Pearson r', higher is better) to assess the alignment between simulation and real-world relative performances across different policies.

In this section, we provide detailed descriptions of our robotics experiments using the SIMPLER environments. We follow the same evaluation protocol as established in SIMPLER [37], focusing on a language-conditioned drawer manipulation task where the robot must "(open/close) (top/middle/bottom) drawer." The robot is positioned in front of a cabinet with three drawers and must manipulate the specified drawer according to the instruction. For our simulation experiments, we also place the robot at 9 different grid positions within a rectangular area on the floor, resulting in $9 \times 3 \times 2 = 54$ total trials.

Following SIMPLER, we conduct experiments on RT-1 checkpoints at various training stages: RT-1-X, RT-1 trained to convergence (RT-1 Converged), RT-1 at 15% of training steps (RT-1 15%), and RT-1 at the beginning of training (RT-1 Begin).

We train our model to modify the cabinet's material using seven different styles: gold, leather, white marble, black marble, steel, stone, and wood. Importantly, we only modify the visual appearance of the cabinet without changing any physical properties such as friction coefficients, material density, center of mass, or static and dynamic friction. Since our method involves a non-deterministic diffusion process, we extend the SIMPLER protocol by averaging success rates across three different random seeds and across the seven different edit styles to produce lower-variance performance estimates.

For the VisMatch, VarAgg and Real evaluation results presented in Table 4, we directly reference the values reported in SIMPLER.

6.5. Additional Qualitative Results

This section of the appendix provides supplementary qualitative results, including a comparative evaluation against baselines (Figures 8 and 9), demonstrating SPIE's superior performance in structural preservation, semantic alignment, and realism. We also provided a extended visualization of the impact of different visual prompts on generated samples (Figure 10), showcasing how SPIE captures semantic nuances beyond text prompts. Additionally, we present examples of simulated scenes edited with enhanced realism (Figure 11), and highlight the flexibility of our approach in replicating visual prompts across diverse scenes (Figure 12). Finally, we display an extensive array of realistic edits generated by our method (Figures 13 and 14), illustrating precise structural preservation and semantic alignment across various scenes and styles.

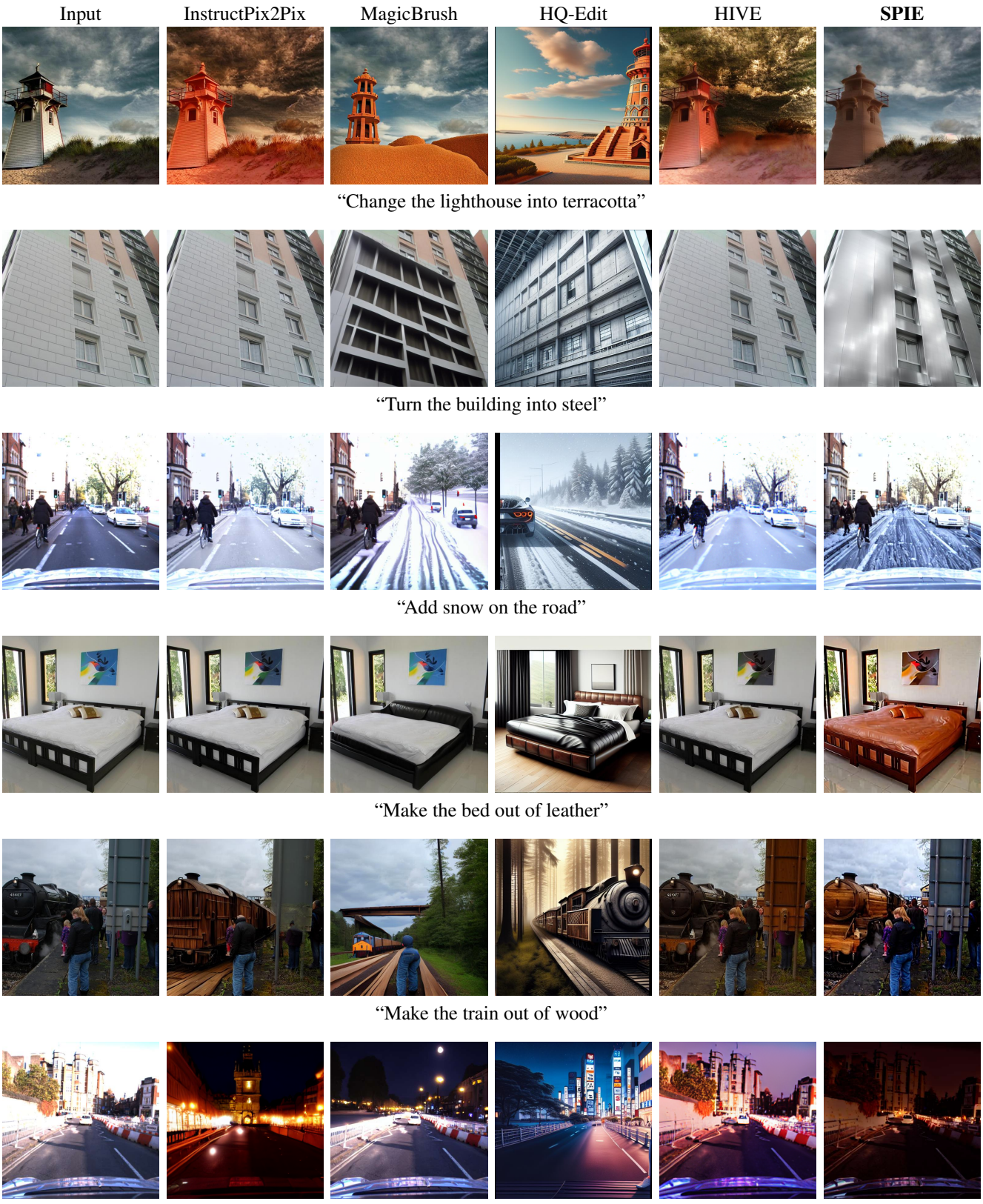

Figure 8. Comparative evaluation of our method against baselines on a diverse set of prompts and images, highlighting superior performance in structural preservation, semantic alignment, and realism.

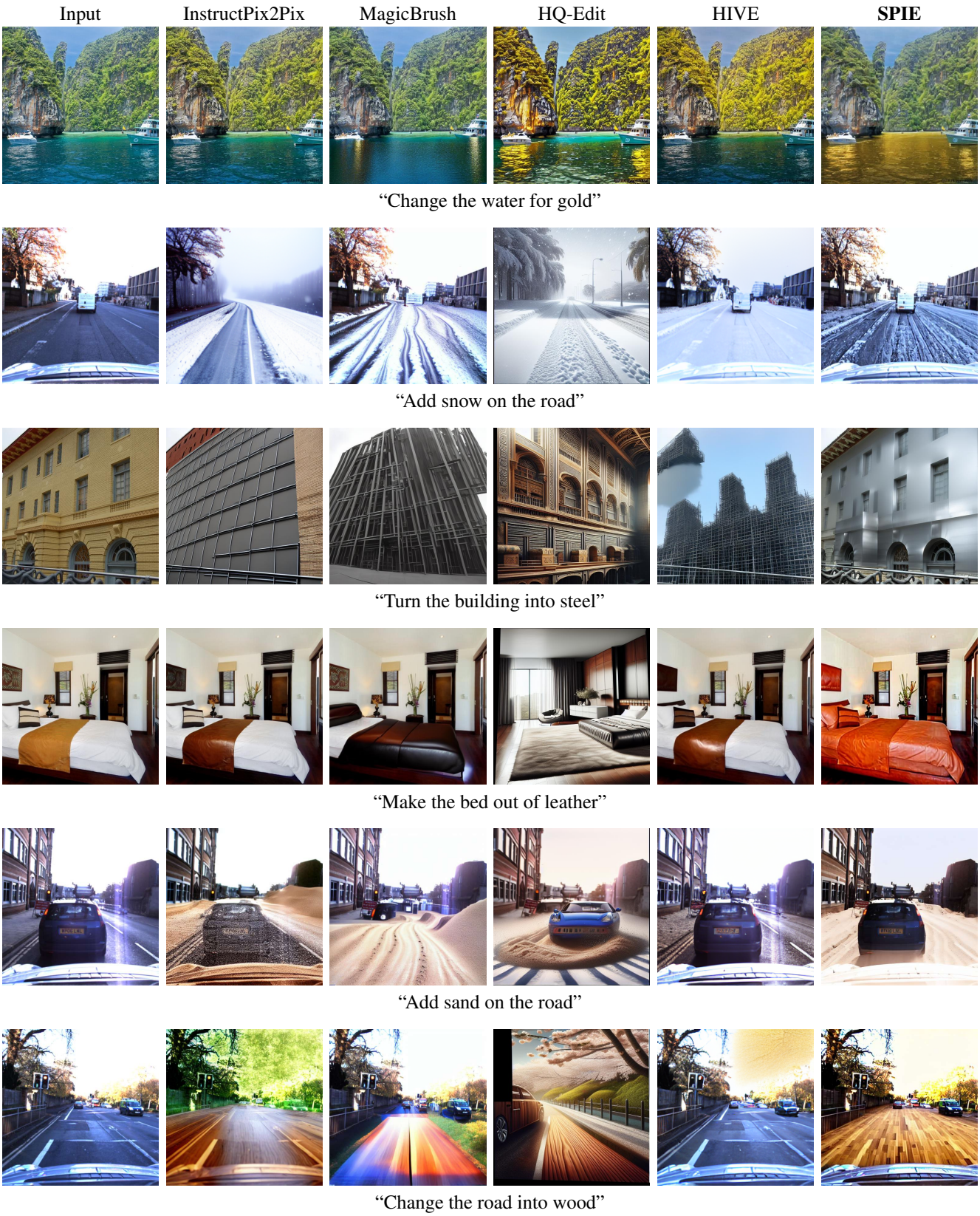

Figure 9. Comparative evaluation of our method against baselines on a diverse set of prompts and images, highlighting superior performance in structural preservation, semantic alignment, and realism.

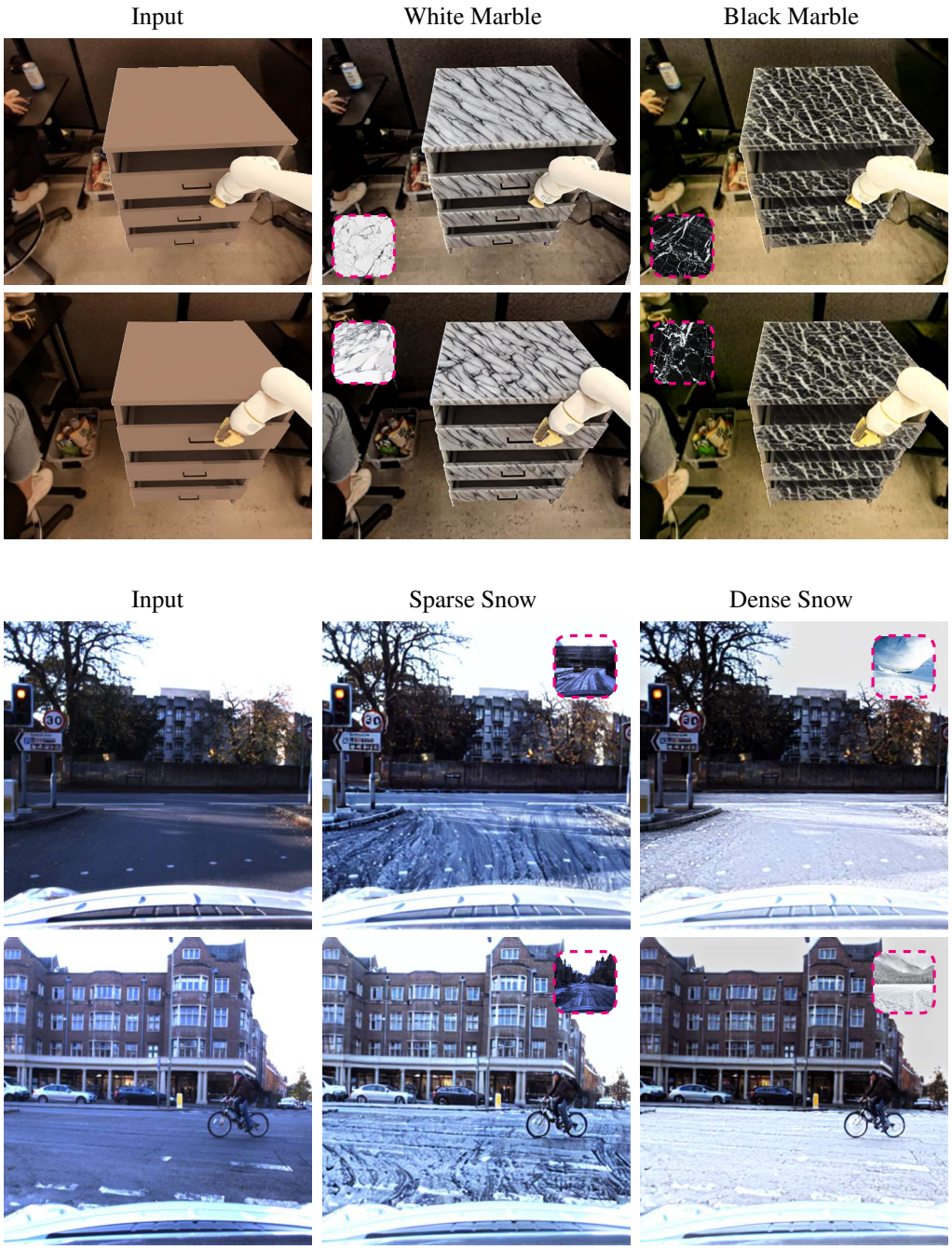

Figure 10. Visualization of the impact of different visual prompts on generated samples when provided a same text instruction. We show edits of both simulation and real-world images. Displayed within the dashed frame is one of the 5 style conditioning images relevant to this edit. Our method effectively captures semantic nuances beyond that described in the text prompt, understanding that the generated marble should be white or black, and that the generated snow should be sparse or dense.

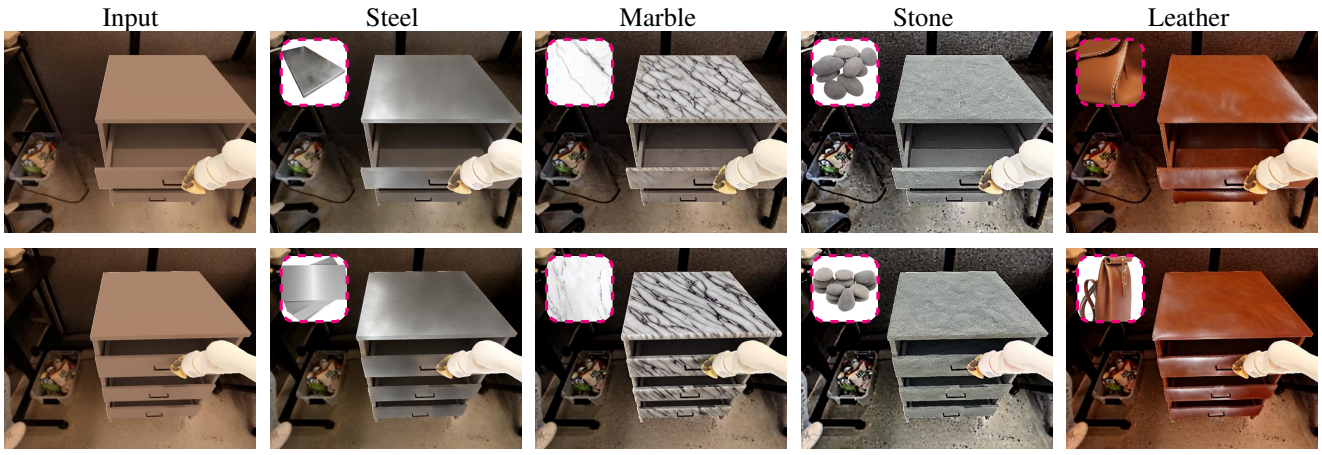

Figure 11. Examples of a simulated scene edited by our method, showcasing enhanced realism compared to the original image across various styles. When training the diffusion model across the diverse styles, we use the same text prompt format: *"make the cupboard out of [X]"*.

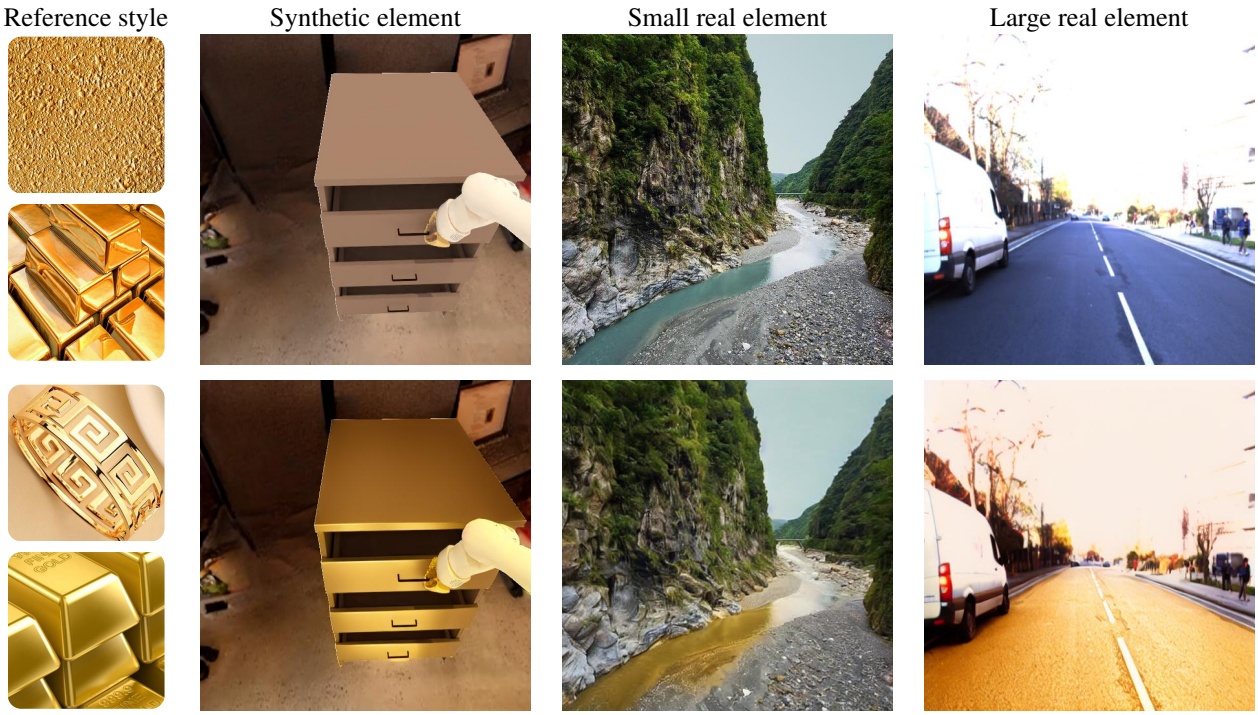

Figure 12. The visual prompts can be precisely replicated across scenes with diverse contexts and layouts, demonstrating the flexibility of our approach. We showcase variants of input images, including synthetic and real-world scenes, small and large elements, and various materials, to illustrate the model's ability to generalize effectively. Notably, our method achieves this versatility by leveraging only a few visual exemplars during training, ensuring robust performance across a wide range of inputs.

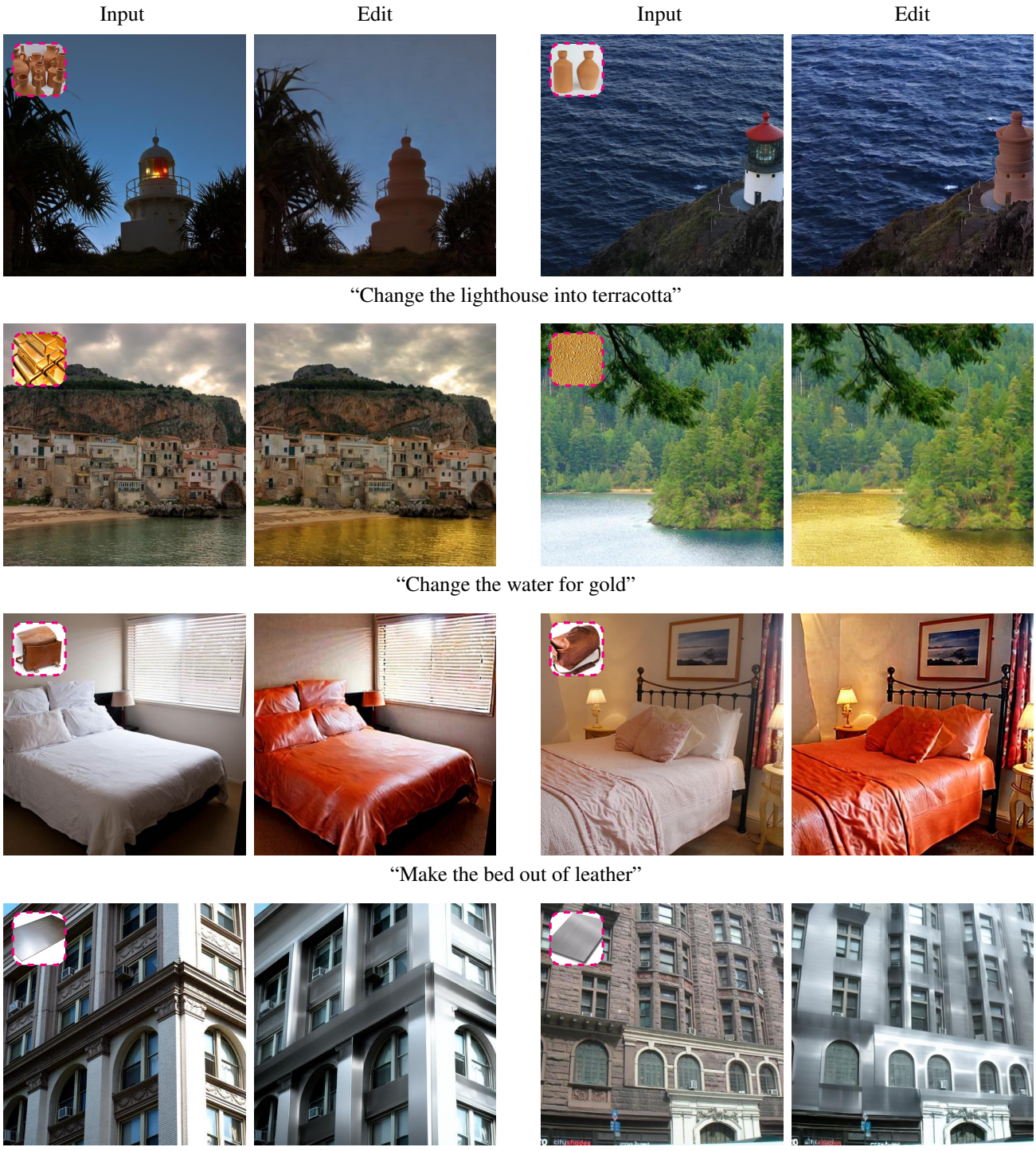

Figure 13. Diverse examples of realistic edits generated by our method, demonstrating precise structural preservation and semantic alignment across various scenes and styles. We show the input images, as well as both the text and visual prompts used to generate these edits.

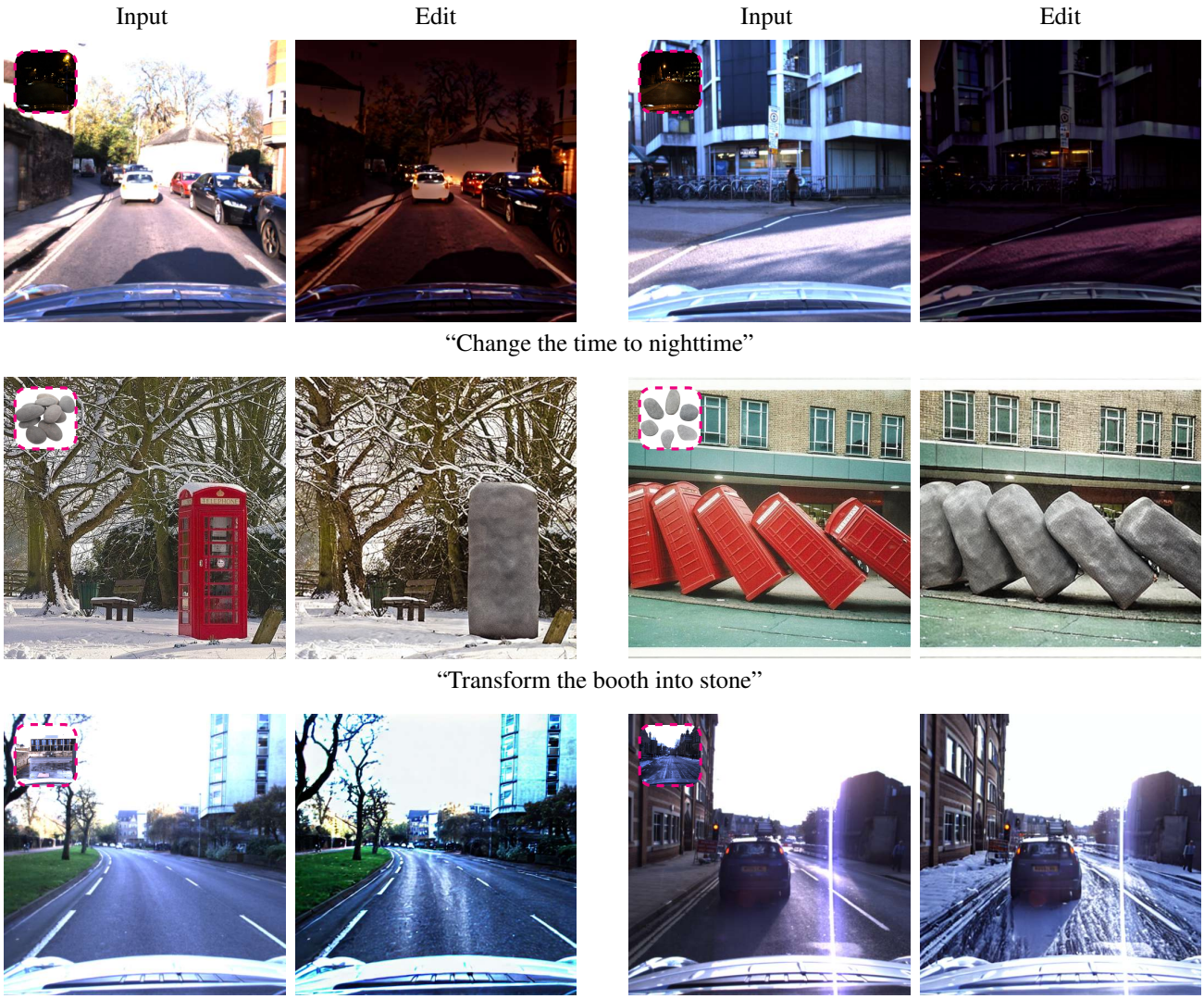

Figure 14. Diverse examples of realistic edits generated by our method, demonstrating precise structural preservation and semantic alignment across various scenes and styles. We show the input images, as well as both the text and visual prompts used to generate these edits