

# Scale Your Instructions: Enhance the Instruction-Following Fidelity of Unified Image Generation Model by Self-Adaptive Attention Scaling

## Supplementary Material

Chao Zhou<sup>1</sup>, Tianyi Wei<sup>2,✉</sup>, Nenghai Yu<sup>1</sup>

<sup>1</sup>University of Science and Technology of China <sup>2</sup>Nanyang Technological University

{chaozhou@mail., ynh}@ustc.edu.cn , tianyi.wei@ntu.edu.sg

### 1. Generalizability of SaaS

The core principle of our Self-Adaptive Attention Scaling (SaaS) method, adaptively rescaling attention activations between image and instruction tokens, is theoretically model-agnostic, suggesting it should be compatible with various unified image generation architectures. To verify this generalizability, we integrated SaaS into the recently open-sourced MIGE model, a multimodal editing framework distinct from the one used in our main paper. The results in Fig. 1 show a significant improvement in instruction following. While the baseline MIGE model struggles with multi-part prompts (e.g., failing to add “graffiti” or render a “snowman”), the SaaS-augmented version successfully executes all sub-tasks. This confirms that SaaS is not over-fitted but serves as a versatile module for enhancing instruction fidelity across different multimodal editing architectures.

### 2. Computational Overhead of SaaS

To verify the practicality of our method, we analyzed the computational overhead introduced by SaaS. We benchmarked inference latency and peak VRAM usage on an NVIDIA RTX A6000 GPU, comparing the baseline OmniGen model with our SaaS-integrated version. As detailed in Tab. 1, the findings show that SaaS is remarkably lightweight, adding a mere 0.3 seconds to latency (1.03% increase) and only 2MB to VRAM consumption (0.02% increase). This negligible overhead confirms that the significant improvements in instruction-following fidelity are achieved with virtually no additional computational cost, making SaaS a highly efficient and practical solution.

### 3. Similar Regions Editing

Editing visually similar regions is a challenging task requiring precise spatial control. As demonstrated in Figure 2, our

✉ Tianyi Wei is the corresponding author.

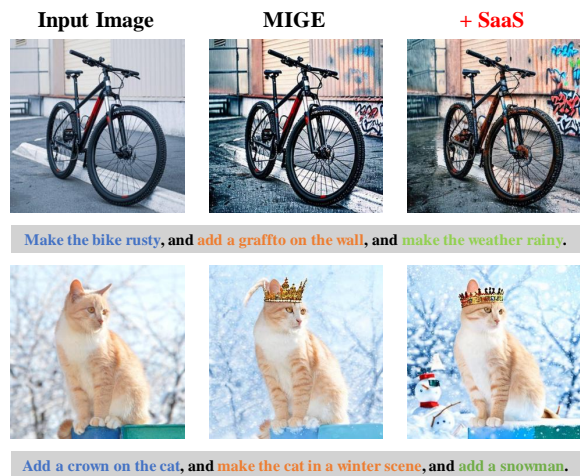


Figure 1. Cases of SaaS on MIGE. Zoom in for better visualization.

	OmniGen	+SaaS	IEP (%)
Latency (s)	29.1	29.4	1.03
VRAM (MB)	9988	9990	0.02

Table 1. IEP means Incremental Expense Proportion.

SaaS method successfully navigates this challenge. It accurately applies a targeted edit to one of two similar objects (left) and, on the same image, executes a complex prompt with eight sub-instructions (right). This performance highlights SaaS’s dual capability in both precise localization and complex instruction following.

### 4. Additional Ablation Study

**Mask Threshold.** In our SaaS framework, the choice of threshold is not critical due to the method’s inherent robustness. We provide the Otsu method for automatic threshold



Figure 2. Demonstration of SaaS on challenging editing tasks. Left: Accurately editing one of two similar regions. Right: Successfully executing a complex prompt with eight sub-instructions on the same input image.

selection, and as demonstrated in the first row of Fig. 3 and in Tab. 2, different threshold values have minimal impact on the outcome. Furthermore, as an empirical guideline, lower thresholds work better for global editing, while higher thresholds suit local editing. As illustrated in the second and third rows of Fig. 3, a threshold that is too low for local editing can result in an unrealistic appearance, while a threshold that is too high for global editing may cause the edit to fail.

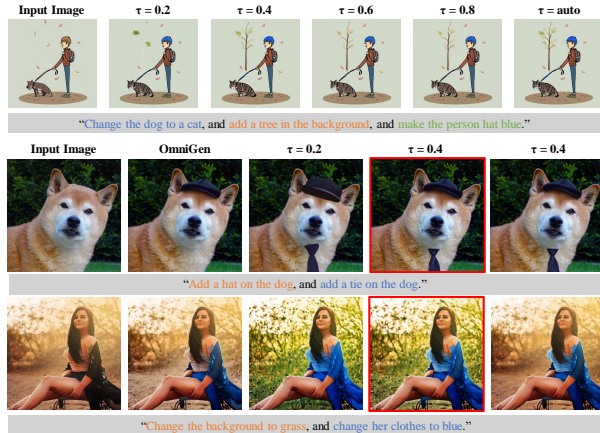


Figure 3. Visual comparison of editing results under different mask thresholds  $\tau$ . Zoom in for better visualization.

Threshold	0.2	0.4	0.6	0.8	auto
PickScore	0.195	0.201	0.200	0.200	0.203

Table 2. PickScore values of various thresholds

**Denoising Steps and Attention Layers.** Regarding denoising steps, SaaS is more effective when applied in the early stages. As shown in Fig. 4, executing SaaS in the early steps achieves similar results to applying it throughout all steps, whereas applying it in the later steps has little to no effect. Regarding attention layers, SaaS is more effective when applied to deeper layers, yielding results comparable to executing it across all layers. While applying SaaS to

shallower layers still has some impact, its effectiveness is noticeably lower than in deeper layers.

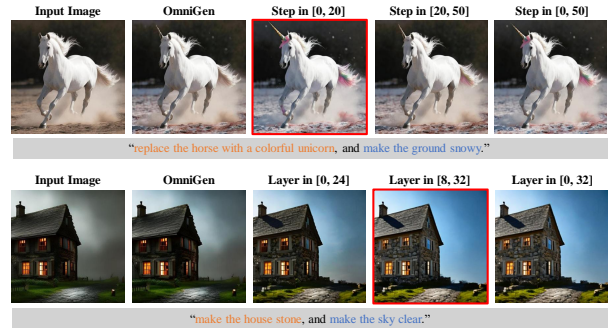


Figure 4. Visual comparisons between various steps and layers. Zoom in for better visualization.

## 5. Additional Comparison

**Instruction-based Image Editing.** In Fig. 5, we provide more qualitative comparison results of our method with other current state-of-the-art methods on the instruction-based image editing task. As can be seen, our method outperforms others in terms of instruction-following fidelity.

Furthermore, we provide a quantitative comparison against several methods: UltraEdit, ACE++, and a simple baseline of increasing the guidance scale (Increase Guidance). As shown in Tab. 3, our method outperforms these approaches, achieving state-of-the-art (SOTA) results on metrics including CLIP-T and PickScore.

Edit Task	Method	CLIP-I $\uparrow$	DINO-v2 $\uparrow$	CLIP-T $\uparrow$	PickScore $\uparrow$
Single Instruction	UltraEdit	0.876	0.750	0.266	0.228
	ACE++	0.941	0.855	0.249	0.152
	Increase Guidance	0.879	0.732	0.262	0.228
	SaaS (ours)	0.900	0.835	<b>0.282</b>	<b>0.397</b>
Multiple Sub-instruction	UltraEdit	0.835	0.552	0.284	0.197
	ACE++	0.950	0.860	0.240	0.150
	Increase Guidance	0.862	0.740	0.282	0.181
	SaaS (ours)	0.892	0.786	<b>0.315</b>	<b>0.469</b>

Table 3. Quantitative comparison on more baselines.

**Visual Conditional Image Generation.** We provide more qualitative results of visual conditional image generation in Fig. 6. On the left are images generated from the depth map, and on the right are images generated from the segmentation map. The text below each set of images corresponds to the respective instructions. As can be seen, whether generated from the depth map or segmentation map, our SaaS method demonstrates better instruction-following fidelity and also produces higher-quality images.





















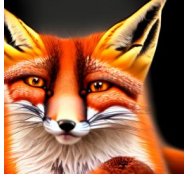

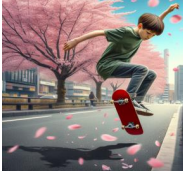
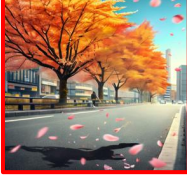

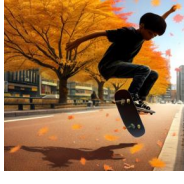
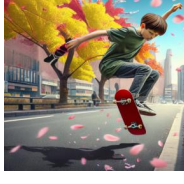
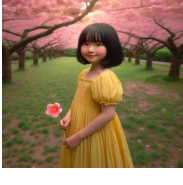

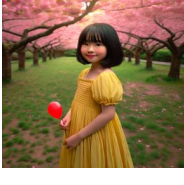
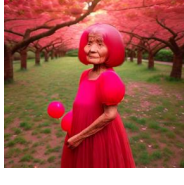
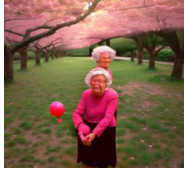





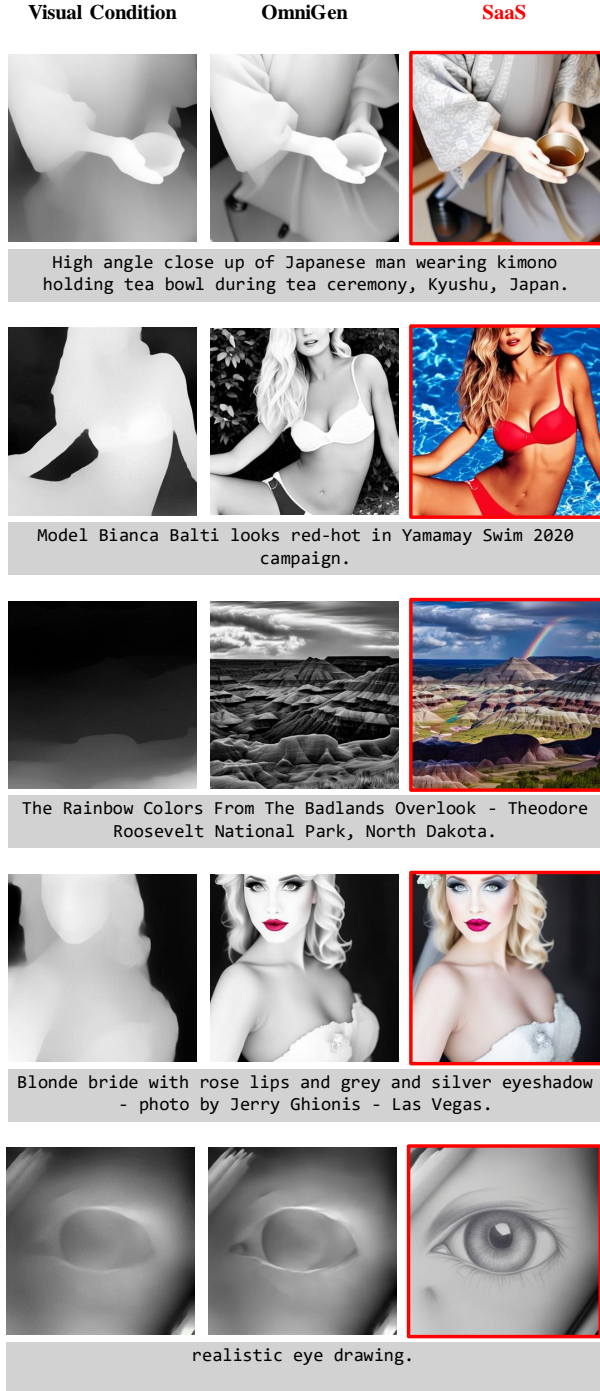
Instruction	Input Image	Ours	OmniGen	IP2P	MagicBrush
Change the background to Disney World.					
Add the phrase "AT THE ZOO NOW!".					
Make it into abstract.					
Make the cat's eye open, and change the background to black, and change the cat to fox.					
Replace the cherry blossom trees with autumn-colored maple trees, and remove the skateboarder.					
Transform the pink flower into a red balloon, and age the girl into an elderly woman.					
Change the cat to Ginger cat, and add a red collar on the cat, and put a pair of sunglasses on the cat.					

Figure 5. **Additional qualitative comparison of instruction-based image editing.** We compare our SaaS with these state-of-the-art image editing methods. Zoom in for better visualization.

### Generation from Depth Map



### Generation from Segmentation Map

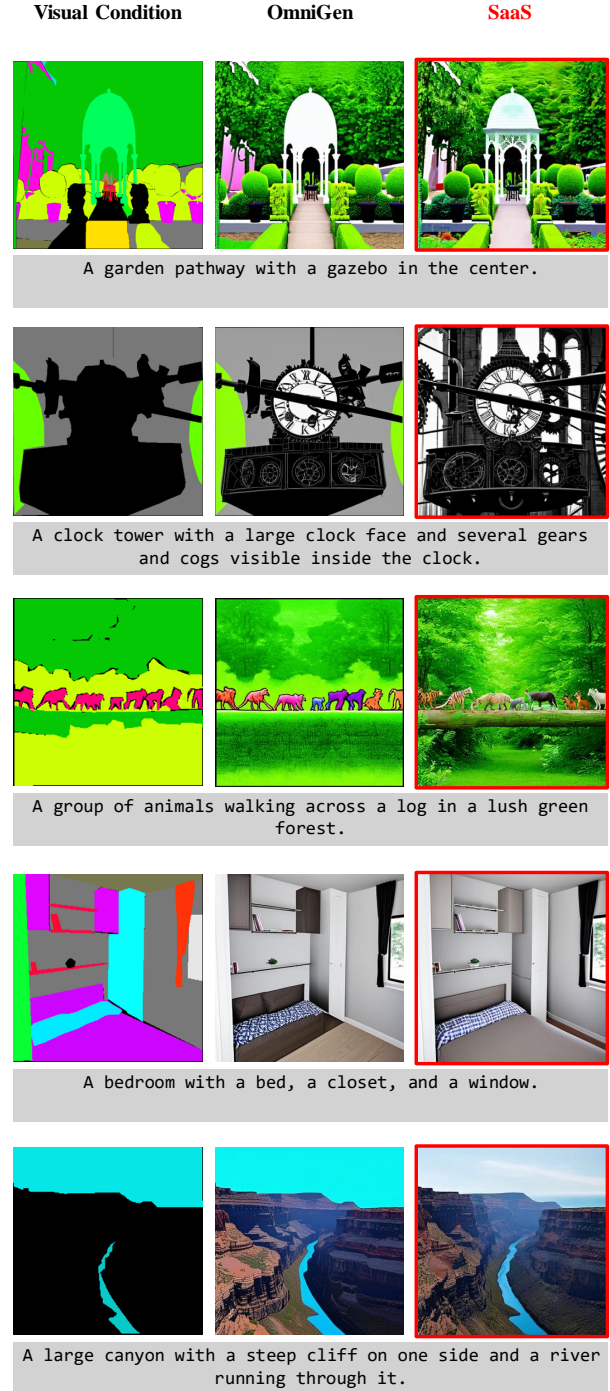


Figure 6. **Additional qualitative comparison of visual conditional image generation.** We compare our SaaS method with OmniGen in the generation tasks from the depth map and the segmentation map. The text below each image represents the corresponding instruction. For better visualization, please zoom in.