

## A. Method details

### A.1. Empirical Intervention Optimization (EIO)

Following the idea of searching for optimal intervention parameters on a pre-defined dataset [4], we **propose Empirical Intervention Optimization (EIO)** to find the parameter vector that maximises a given objective. Because evaluating the objective over the *entire* dataset is computationally expensive, we adopt **Bayesian Optimisation (BO)** [5, 49] to guide the search.

**Posterior prediction and acquisition.** At each BO iteration we first compute the Gaussian-process *posterior mean*  $\mu_t(\mathbf{K})$  and *variance*  $\sigma_t^2(\mathbf{K})$ :

$$\mu_t(\mathbf{K}) = \mathbf{g}_t(\mathbf{K})^\top (\mathbf{G}_t + \sigma_n^2 \mathbf{I})^{-1} \mathbf{f}_t, \quad (5)$$

$$\sigma_t^2(\mathbf{K}) = \kappa(\mathbf{K}, \mathbf{K}) - \mathbf{g}_t(\mathbf{K})^\top (\mathbf{G}_t + \sigma_n^2 \mathbf{I})^{-1} \mathbf{g}_t(\mathbf{K}). \quad (6)$$

Here  $\mathbf{G}_t \in \mathbb{R}^{N_t \times N_t}$  is the Gram matrix formed by the kernel  $\kappa$ ,  $\mathbf{g}_t(\mathbf{K}) \in \mathbb{R}^{N_t}$  is the cross-covariance vector between the candidate  $\mathbf{K}$  and the observed points,  $\sigma_n^2$  denotes the learned observation-noise variance, and  $\mathbf{f}_t$  stores the EFI scores obtained so far.

We then define the *standardised improvement*  $\xi_t(\mathbf{K}) = [\mu_t(\mathbf{K}) - y_{\text{best}}] / \sigma_t(\mathbf{K})$ , and obtain the closed-form *Expected Improvement* (EI)

$$\text{EI}_t(\mathbf{K}) = (\mu_t(\mathbf{K}) - y_{\text{best}}) \Phi(\xi_t(\mathbf{K})) + \sigma_t(\mathbf{K}) \phi(\xi_t(\mathbf{K})),$$

which trades off **exploitation** (large  $\mu_t$ ) against **exploration** (large  $\sigma_t$ ). The point that maximises EI is chosen as the next intervention to evaluate. The complete procedure is summarised in Algorithm 2.

### A.2. Surrogate Intervention Optimization (SIO)

Record that the computation of  $\delta^e$ :

$$\begin{aligned} \delta^e(k) &= \mathbb{E}_{(x,m,p) \sim \mathbb{D}} [e(G_\theta^v(x, m, p; v(k)), x, m, p)] \\ &\quad - \mathbb{E}_{(x,m,p) \sim \mathbb{D}} [e(G_\theta^v(x, m, p; v(0)), x, m, p)] \end{aligned}$$

which is expensive due to not only the time-consuming image generation process but also linearly increase with the number of evaluation metrics  $|e|$ . Therefore changing the optimization objective  $f$  and  $w_e$  become costly. As an alternative we want to replace the computation of  $\delta^e$  to light-weight regression process from either MLP based or GP based regression model.

## B. Discussion details

### B.1. IOU metrics

Observing the limitations of existing evaluation metrics in distinguishing between prompt-adherence and mask-following, increasing the size of the generated object also

---

#### Algorithm 1: Image Latent LPF

---

**Input:** Image latent  $z \in \mathbb{R}^{H \times W \times 4}$ , LPF threshold  $\gamma$

**Output:** Filtered RGB Image  $I_{\text{filtered}}$

**foreach** channel  $c \in \{0, 1, 2, 3\}$  **do**

    Perform 2D FFT on channel  $c$ :  $\mathcal{F}_c = \text{FFT}(z_c)$

    Create a low-pass filter mask

$M_{\text{LPF}} \in \{0, 1\}^{H \times W}$  such that:

$$M_{\text{LPF}}(u, v) = \begin{cases} 1 & \text{if } \sqrt{u^2 + v^2} \leq \gamma \\ 0 & \text{otherwise} \end{cases}$$

    Apply the low-pass filter:  $\mathcal{F}_c^{\text{filtered}} = \mathcal{F}_c \odot M_{\text{LPF}}$

    Perform the inverse FFT:

$z_c^{\text{filtered}} = \text{IFFT}(\mathcal{F}_c^{\text{filtered}})$

**end**

Combine filtered channels to form the output:

$$z_{\text{filtered}} = \text{Concat}(z_0^{\text{filtered}}, z_1^{\text{filtered}}, z_2^{\text{filtered}}, z_3^{\text{filtered}})$$

**return**  $z_{\text{filtered}}$

---



---

#### Algorithm 2: Empirical Intervention Optimization (EIO)

---

**Input:** Parameter space  $\mathcal{K}$ ; initial observations

$\mathcal{D}_0 = \{(\mathbf{K}_i, f(\mathbf{K}_i; G_\theta^{f_c}))\}_{i=1}^n$ ; representative dataset  $\mathbb{D}$ ; maximum iterations  $T$ ; metric set  $e$  with weights  $\{w_e\}_{e \in e}$ .

**Output:** Optimal parameter vector  $\hat{\mathbf{K}}$ .

**for**  $t \leftarrow 0$  **to**  $T - 1$  **do**

    Fit GP posterior:  $(\mu_t, \sigma_t) \leftarrow \mathcal{D}_t$ ;

$$\xi_t(\mathbf{K}) = \frac{\mu_t(\mathbf{K}) - y_{\text{best}}}{\sigma_t(\mathbf{K}) + \epsilon};$$

$$\text{EI}_t(\mathbf{K}) = (\mu_t(\mathbf{K}) - y_{\text{best}}) \Phi(\xi_t(\mathbf{K})) + \sigma_t(\mathbf{K}) \phi(\xi_t(\mathbf{K}));$$

    Select next point;

$$\mathbf{K}_{t+1} = \arg \max_{\mathbf{K} \in \mathcal{K}} \text{EI}_t(\mathbf{K});$$

    Evaluate objective ( $\delta^e$ );

$$f_{t+1} = \sum_{e \in e} w_e \delta^e(\mathbf{K}_{t+1});$$

    Augment data;

$$\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\mathbf{K}_{t+1}, f_{t+1})\};$$

**return**  $\hat{\mathbf{K}} = \arg \max_{(\mathbf{K}_i, f_i) \in \mathcal{D}_T} f_i$

---

boosts the CLIP Score, even when the prompt is not followed precisely. Therefore, CLIP Score is not a reliable metric for measuring prompt-adherence alone, as it is also influenced by object size and other factors, as discussed in [11].

To disentangle the effect of object size from prompt-adherence, we introduce the IoU metric, calculated using auto-segmentation masks generated by SAM. This ap-

**Algorithm 3:** Pre-training a Unified Surrogate  $\phi$  for the Vector  $\delta = (\delta^e)_{e \in \mathbf{e}}$

**Input:** Metric set  $\mathbf{e}$ ; parameter space  $\mathcal{K}$ ; expensive generator  $G_\theta^v$ ; sampling budget  $N_{\text{rand}}$ ; regression type MODEL  $\in \{\text{GP}, \text{MLP}, \text{Poly}\}$ .

**Output:** Trained surrogate  $\phi : \mathbf{K} \mapsto \hat{\delta} \in \mathbb{R}^{|\mathbf{e}|}$ .

**Phase 1: data collection;**

**for**  $i \leftarrow 1$  **to**  $N_{\text{rand}}$  **do**

    Draw  $\mathbf{K}_i \sim \text{Uniform}(\mathcal{K})$ ;

**foreach** metric  $e \in \mathbf{e}$  **do**

$\delta_i^e \leftarrow$   
         $\mathbb{E}_{(x,m,p) \sim \mathbb{D}}[e(G_\theta^v(x,m,p;v(\mathbf{K}_i)), x,m,p)] -$   
         $\mathbb{E}_{(x,m,p) \sim \mathbb{D}}[e(G_\theta^v(x,m,p;v(\mathbf{0})), x,m,p)]$ ;

    Form vector  $\delta_i = (\delta_i^e)_{e \in \mathbf{e}}$ ;

    Store pair  $(\mathbf{K}_i, \delta_i)$  in dataset  $\mathcal{S}$ ;

**Phase 2: model fitting;**

**switch** MODEL **do**

**case** GP **do**

        Fit multiple GP for each  $e \in \mathbf{e}$  on  $\mathcal{S}$ ;

**case** MLP **do**

        Initialise an MLP with  $|\mathbf{e}|$  output units;

        Train on  $\mathcal{S}$  using Adam;

**case** Poly **do**

        Choose polynomial degree  $d$  by cross-validation;

        Fit multivariate ridge regression mapping

$\mathbf{K} \mapsto \delta$ ;

Set  $\phi(\cdot)$  to the trained predictor (GP mean, MLP forward pass, or polynomial mapping);

**return**  $\phi$ ;

	Not masked	Partially masked	Fully masked
Ratio	48.05%	34.22%	17.73%

Table 5. Table of masked ratio.

proach allows us to isolate prompt-adherence. For instance, if two outputs share the same IoU score, we infer that the output with the higher CLIP Score exhibits better prompt-adherence.

To compute the segmentation masks, we use SAM by inputting k-medoids points derived from the input mask. However, when the generated object is too small to accurately capture an IoU mask, we reject the segmentation mask if its area exceeds 1.5 times the input mask, setting the IoU score to zero.

## B.2. Training data distribution

**Random Masking Strategy** As previously mentioned, the SDI model is typically trained on data processing using a random mask strategy, where 25% of an image’s area is randomly masked out. Although we cannot directly access the original training data, we employed the same random

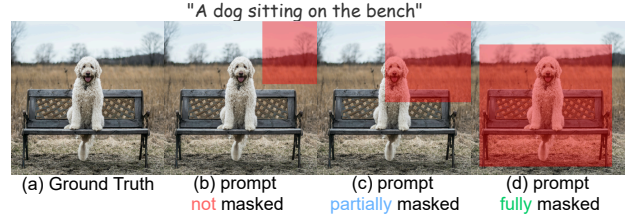


Figure 7. An illustration of possible training data generated by random mask strategy.

mask method on COCO val2017 dataset [31], which contains 4952 images with segmentation mask, and conducted a statistical analysis to examine its effects, as shown in Fig. 8.

We first calculated the average proportion of area covered by object masks within images in the dataset, which was found to be 32.2%. Consequently, when applying a random mask covering only 25% of the image, it is more likely to produce results similar to those shown in Fig. 7(b) and (c). Furthermore, we computed the mean Intersection over Union (mIoU) between the object mask and the random mask, obtaining a value of 0.2018, which further supports our hypothesis. Lastly, we analyzed the ratio of instances where objects were not fully masked versus fully masked, as shown in Tab. 5.

**Segmentation Masking Strategy** The random mask strategy in the original SDI pipeline has been identified as a limitation, reducing its mask-fitting ability. To address this, methods such as SmartBrush [55], BrushNet [22], and PowerPoint [65] leverage the auto-segmentation tool SAM to facilitate the collection of segmentation mask data, as shown in Fig. 9. By exclusively utilizing segmentation data, these models learn a direct correlation between objects and prompt instructions.

However, a potential concern arises from the inherent biases of auto-segmentation tools. Since the segmented masks predominantly focus on objects, the models may struggle to generalize their use of background information. Furthermore, the fine-tuning process could potentially hinder the generalizability of the original model backbone.

## C. Dataset details

### C.1. FreeCond Inpainting Benchmark (FCIBench)

As the quality of text-to-image (T2I) generation continues to advance, there is a growing demand for robust T2I inpainting models. However, current evaluations of these models remain relatively simplistic, often limited to object-level inpainting or reconstruction tasks. This approach overlooks the complexity of more challenging inpainting scenarios, which are critical for assessing model performance. To address this gap, we introduce FCIBench, a benchmark

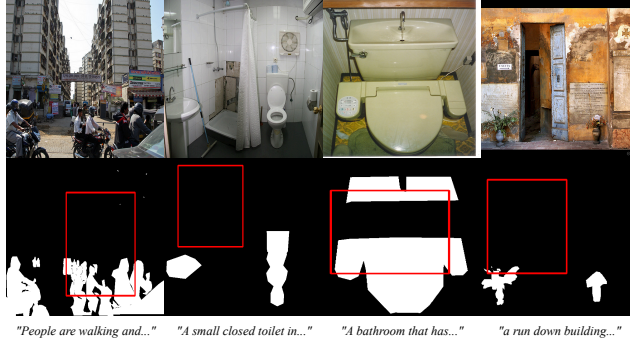


Figure 8. The result of applying the random mask strategy on the COCO dataset [31]. The white areas represent the object masks, while the red-boxed regions indicate the random masks, covering 25% of the image area.

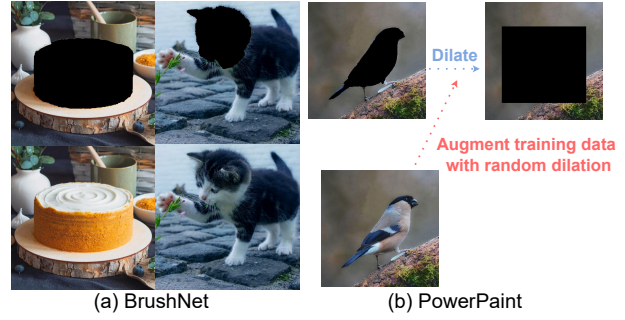


Figure 9. The illustration of segmentation-based training strategy.

designed to evaluate T2I inpainting models in three key scenarios that require a sophisticated understanding of both background conditions and prompt instructions:

- **Coarse Mask:** The model must seamlessly integrate prompt instructions within a masked region, even when the mask boundaries are imprecise, ensuring a harmonious blend with the surrounding background.
- **Non-object Prompt:** The model is tasked with modifying background-like regions to align with prompt instructions, rather than merely inpainting discrete objects.
- **Interaction Prompt:** The model must ensure coherent interaction between the generated content and the existing background, maintaining logical and visual consistency.

The overall generation pipelines are illustrated in Fig. 10.

**FCIBench: Coarse Mask** The main challenge in this setting is to seamlessly blend the instructed object into an ambiguous inpainting mask. The T2I model must not only generate the instructed object but also ensure that the filled-in content is contextually coherent. To address this, we incorporate human-annotated masks into the automated image generation pipeline.

**FCIBench: Non-Object Prompt** When the goal is to modify only a part of an existing object, manual annotation is unnecessary. Instead, we employ a combination of automatic segmentation tools—specifically, Grounding SAM<sup>3</sup>—followed by morphological operations (5 iterations of erosion with a  $3 \times 3$  kernel and random spatial shifts ranging from 0 to 50 pixels).

**FCIBench: Interaction Prompt** For background-related instructions during the inpainting process, we leverage TF-

T2I<sup>4</sup> to generate instruction sketch conditioned on background image and additional instruction prompts. Although minor artifacts may occur, this approach provides rough spatial guidance for the instructed region. We then apply Grounding SAM for mask localization, followed by 5 iterations of dilation using a  $3 \times 3$  kernel and random shifts between 0 and 50 pixels.

## D. Experiment details

For all included baselines, we adopt the official configurations provided in their respective GitHub repositories. We use 50 diffusion steps, a classifier-free guidance scale of 15, and a consistent random seed across all experiments. (For FLUX, we follow the default setting of 3.5 for classifier-free guidance.) The EIO-optimized FreeCond coefficients, along with detailed evaluation results, are presented in Tab. 6. The optimization objective is  $f(\cdot) = 100 \cdot e^{\text{HPS}} + 2 \cdot e^{\text{CLIP}} - 100 \cdot e^{\text{LPIPS}}$

<sup>3</sup><https://github.com/IDEA-Research/Grounded-Segment-Anything>

<sup>4</sup><https://github.com/BlueDyee/TF-TI2I>

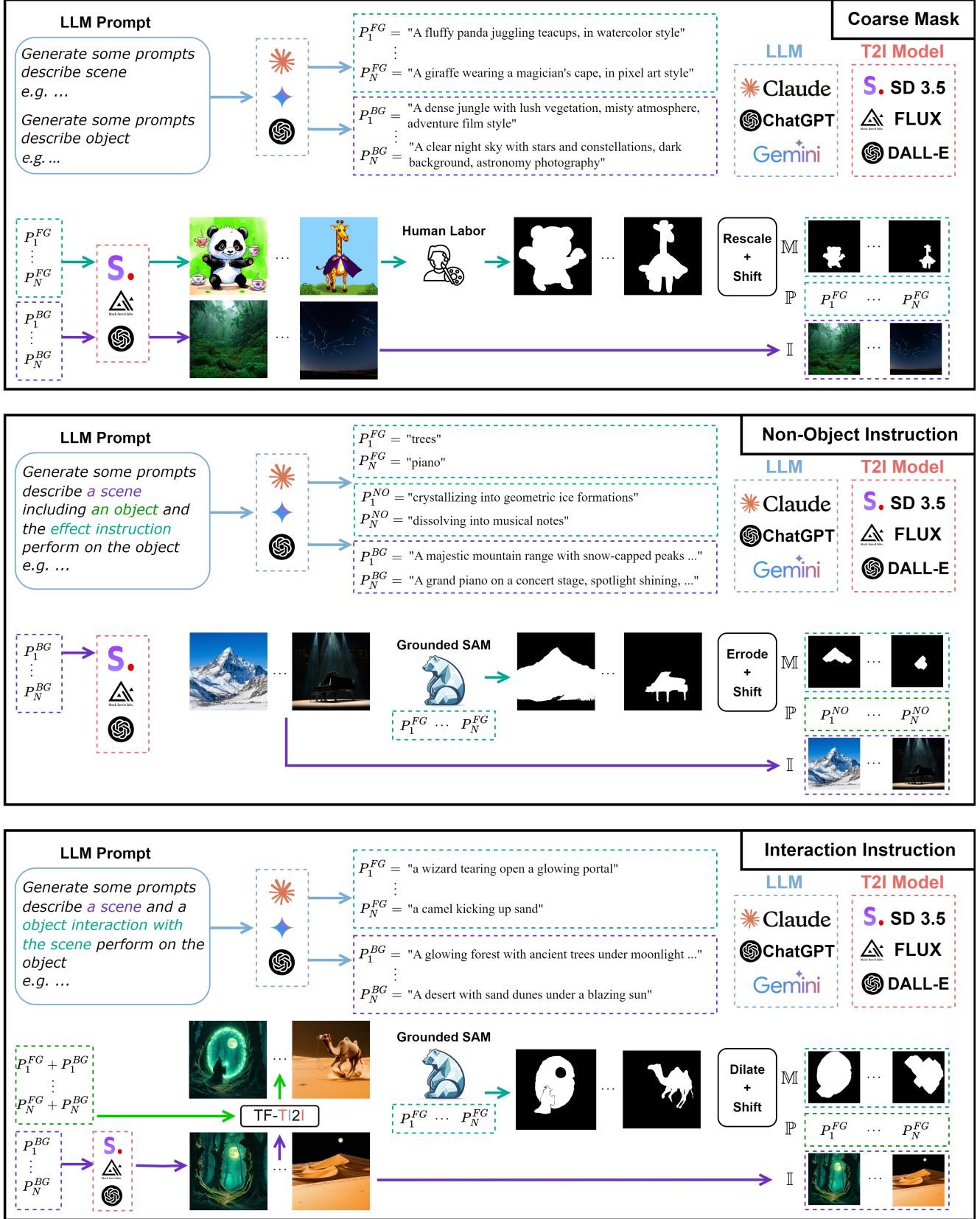


Figure 10. Illustration of the FCIBench generation pipelines. The generation process of FCIBench involve diverse effort of both human and non-human resources (LLMs, T2I models, Grounding SAM, morphological tools, TF-TI2I).

	$[k_\alpha, k_\beta, k_\gamma]$	CLIP	IoU	ImageReward	HPS	DINO	LPIPS	PSNR
SDI	$[0, 0, 0]$	$16.7 \pm 0.1$	$17.0 \pm 0.2$	$-10.2 \pm 0.1$	$22.1 \pm 0.2$	$97.2 \pm 0.1$	$53.5 \pm 0.2$	$25.9 \pm 0.1$
SDI+F	$[0.3, 0.6, 0.8]$	$20.2 \pm 0.1$	$38.6 \pm 0.3$	$-4.7 \pm 0.1$	$23.9 \pm 0.0$	$94.7 \pm 0.1$	$60.9 \pm 0.2$	$24.8 \pm 0.1$
SDXL	$[0, 0, 0]$	$18.4 \pm 0.1$	$18.0 \pm 0.1$	$-7.6 \pm 0.1$	$23.5 \pm 0.1$	$97.6 \pm 0.1$	$32.1 \pm 0.1$	$30.9 \pm 0.1$
SDLX+F	$[2.5, 0.4, 0.7]$	$22.2 \pm 0.1$	$58.0 \pm 0.1$	$-2.2 \pm 0.1$	$25.0 \pm 0.1$	$93.8 \pm 0.1$	$32.1 \pm 0.1$	$30.5 \pm 0.1$
FLUX	$[0, 0, 0]$	$18.6 \pm 0.2$	$28.8 \pm 0.1$	$-6.1 \pm 0.2$	$22.2 \pm 0.1$	$96.5 \pm 0.1$	$35.3 \pm 0.1$	$30.7 \pm 0.1$
FLUX+F	$[0, 0, 0.1]$	$19.2 \pm 0.1$	$66.4 \pm 0.6$	$-7.1 \pm 0.1$	$22.1 \pm 0.1$	$94.2 \pm 0.1$	$39.0 \pm 0.1$	$29.9 \pm 0.1$

Table 6. Quantitative results of FreeCond. The parameters  $[k_\alpha, k_\beta, k_\gamma]$  are optimized using EIO, and the reported metrics are computed based on outputs generated from five different random seeds.