Stochastic Conditional Diffusion Models for Robust Semantic Image Synthesis

Juyeon Ko [*1] **Inho Kong** [*1] **Dogyun Park** [1] **Hyunwoo J. Kim** [1]

Abstract

Semantic image synthesis (SIS) is a task to generate realistic images corresponding to semantic maps (labels). However, in real-world applications, SIS often encounters noisy user inputs. To address this, we propose Stochastic Conditional Diffusion Model (SCDM), which is a robust conditional diffusion model that features novel forward and generation processes tailored for SIS with noisy labels. It enhances robustness by stochastically perturbing the semantic label maps through Label Diffusion, which diffuses the labels with discrete diffusion. Through the diffusion of labels, the noisy and clean semantic maps become similar as the timestep increases, eventually becoming identical at $t = T$. This facilitates the generation of an image close to a clean image, enabling robust generation. Furthermore, we propose a class-wise noise schedule to differentially diffuse the labels depending on the class. We demonstrate that the proposed method generates high-quality samples through extensive experiments and analyses on benchmark datasets, including a novel experimental setup simulating human errors during real-world applications. Code is available at https://github.com/mlvlab/SCDM.

1. Introduction

Semantic image synthesis (SIS) is a type of image translation that converts a given semantic map (label) into a photo-realistic image, which is the inverse of semantic segmentation. It is also one of the conditional image generation tasks with semantic label maps serving as the input conditions. The problem is formulated as approximating the conditional distribution $q(X|Y)$ where X and Y are the random variables denoting the image and the semantic map, respectively. SIS is addressed by adopting conditional generative models (Yang et al., 2019; Tang et al., 2020a; Ntavelis et al., 2020; Tan et al., 2021a) such as conditional GANs (Goodfellow et al., 2014; Mirza & Osindero, 2014). Given a semantic label map $\mathbf{y}$, these works sample a new image $\hat{\mathbf{x}}$ from the learned conditional distribution $p_\theta(X|Y = \mathbf{y})$. As diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) have gained significant attention on various generation tasks (Ramesh et al., 2021; Dhariwal & Nichol, 2021; Couairon et al., 2023), diffusion models for SIS have been recently studied by a few works (Wang et al., 2022a; Xue et al., 2023). Specifically, SDM (Wang et al., 2022b) embeds the input conditions similarly to SPADE (Park et al., 2019) and integrates diffusion models into the context of SIS. LDM (Rombach et al., 2022) learns a diffusion model on latent vectors with condition encoders.

SIS has a wide range of real-world applications such as photo editing or content creation (Chen & Koltun, 2017; Park et al., 2019; Zhu et al., 2020a; Tang et al., 2020b). In practice, SIS often involves noisy input $\tilde{\mathbf{y}}$ from users. For instance, users mark specific areas as the classes they wish to synthesize, and the marks come with errors such as jagged edges and incompletely marked areas. Even labels by professional annotators in benchmark datasets occasionally contain mistakes and show inconsistency between annotators, and the input from end users would inevitably entail noise. This poses the gap between the label distributions for training and inference. Usually, models are trained with clean labels $\mathbf{y}$ in benchmark datasets, whereas generation is performed with noisy labels $\tilde{\mathbf{y}}$. In the case of diffusion models, the model is exposed to the erroneous guidance throughout T timesteps, *i.e.*, $t = T$ to $t = 1$, generating the corresponding noisy image.

To minimize the gap, we generate samples with stochastically perturbed labels for both training and inference. Specifically, we propose to diffuse the semantic label map $\mathbf{y}_0$ to $\mathbf{y}_1, \ldots, \mathbf{y}_T$ and use them throughout the generation process. Assume there exists a clean semantic map $\mathbf{y}$ corresponding to the noisy one $\tilde{\mathbf{y}}$. Then, utilizing discrete diffusion with an absorbing state allows us to make the intermediate noisy map $\tilde{\mathbf{y}}_t$ and clean map $\mathbf{y}_t$ gradually become similar, as they are masked and eventually become identical at $t = T$. Since the trajectories $\mathbf{y}_{1:T}$ and $\tilde{\mathbf{y}}_{1:T}$ provided to the model during the generation process are similar, *i.e.*, $\mathbf{y}_t$ and $\tilde{\mathbf{y}}_t$ get closer than $\mathbf{y}$ and $\tilde{\mathbf{y}}$, the generated image is close to the clean

[*]Equal contribution [1]Department of Computer Science, Korea University, Republic of Korea. Correspondence to: Hyunwoo J. Kim <hyunwoojkim@korea.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

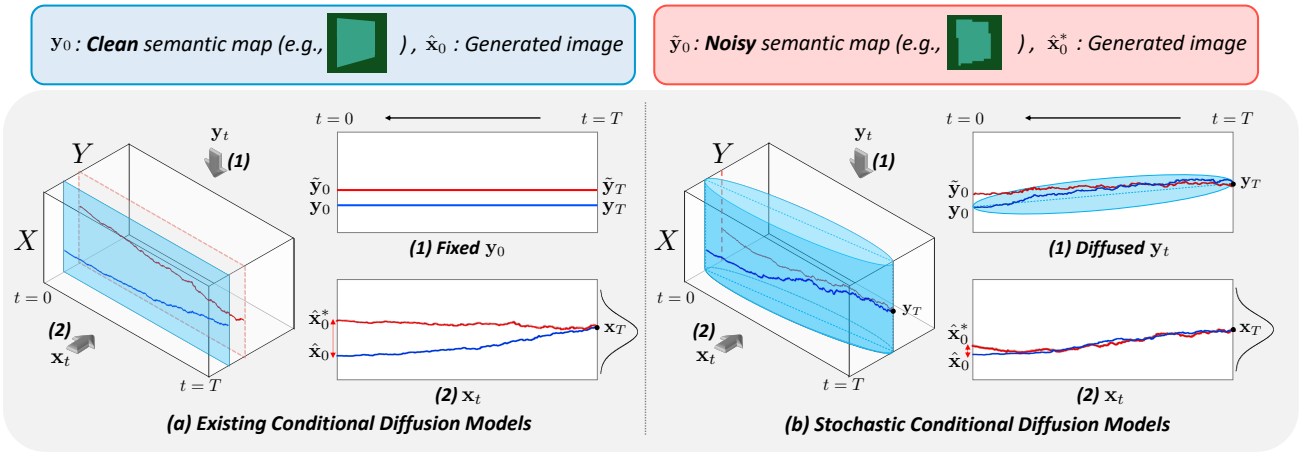


Figure 1. **Visualization of conditional generation.** Each colored trajectory represents a sampling trajectory conditioned on a noisy semantic map $\tilde{\mathbf{y}}_0$ (Red) and the corresponding clean semantic map $\mathbf{y}_0$ (Blue). They are projected onto the **(1)** semantic map space and the **(2)** image space, sharing the same $\mathbf{x}_T$. **(a)** Existing conditional diffusion models (baseline) use a fixed condition $\mathbf{y}_0$ over the generation process, and the gap between $\tilde{\mathbf{y}}_t$ and $\mathbf{y}_t$ yields erroneous conditional score estimation at each timestep t. **(b)** In contrast, our method stochastically perturbs the condition with masking, resulting in a trajectory $\mathbf{y}_{1:T}$ following a probability distribution $q(\mathbf{y}_{1:T}|\mathbf{y}_0)$, as depicted with blue shaded areas around the $\mathbf{y}_t$ trajectory. This makes the intermediate trajectories, *i.e.*, $\mathbf{y}_{1:T}|\mathbf{y}_0$ and $\tilde{\mathbf{y}}_{1:T}|\tilde{\mathbf{y}}_0$, close to each other, enhancing the robustness against the noisy labels.

image, as illustrated in Figure 1.

In this paper, we introduce **Stochastic Conditional Diffusion Model (SCDM)**, a novel conditional diffusion model specifically designed to enhance robustness on noisy labels. Our SCDM stochastically perturbs the semantic maps with *Label Diffusion* and conditions image generation on the *diffused* labels. We also incorporate label statistics and develop a new class-wise noise schedule for labels to enhance the generation quality of small and rare classes. Moreover, the generation process of SCDM entails two heterogeneous diffusion processes: a *discrete forward process* for labels and a *continuous reverse process* for images. We empirically demonstrate that SCDM can approximate $q(X|Y)$ and present theoretical analysis. Additionally, we introduce a new noisy SIS benchmark and prove the robustness of SCDM under noisy conditions.

To summarize, our **contributions** are as follows:

- We propose Stochastic Conditional Diffusion Model (SCDM), a robust conditional diffusion model for SIS that incorporates Label Diffusion, a discrete diffusion process for labels that enables differential conditioning on semantic labels.
- We provide theoretical analyses of our class-wise schedule and the relationship between the class guidances (implicit classifier gradients) induced by fixed labels and label diffusion.
- We introduce a new SIS benchmark designed to assess generation performance under noisy conditions, simulating human errors that can occur during real-world applications.
- We conduct extensive experiments and analyses on benchmark datasets and achieve competitive results.

2. Related Works

Semantic Image Synthesis. Since Pix2pix (Isola et al., 2017) have established a general framework for SIS, conditional Generative Adversarial Networks (GANs) are widely used in SIS (Yang et al., 2019; Zhu et al., 2017; Ntavelis et al., 2020; Tang et al., 2020b; Tan et al., 2021a;b; Shi et al., 2022). SPADE (Park et al., 2019) proposes spatially-adaptive normalization and successfully preserves semantic information. Since SPADE, many normalization-based approaches have been presented (Tan et al., 2021a; Lv et al., 2022). For instance, CLADE (Tan et al., 2021b) adopts class-adaptive normalization and RESAIL (Shi et al., 2022) proposes retrieval-based spatially adaptive normalization, and OASIS (Sushko et al., 2020) designs the discriminator as a semantic segmentation network. INADE (Tan et al., 2021a) utilizes class-level conditional modulation, and SAFM (Lv et al., 2022) proposes shape-aware position descriptors to modulate the features.

Recently, diffusion models (DMs) also have been proposed for SIS. SDM (Wang et al., 2022b) encodes the semantic label map with SPADE. LDM (Rombach et al., 2022) leverages a latent space for the conditions including the semantic

maps. PITI (Wang et al., 2022a) pre-trains the semantic latent space and finetunes it with the RGB-preprocessed semantic mask images, rather than using the maps with class indexes like most studies including ours. FLIS (Xue et al., 2023) incorporates not only semantic label maps but also additional text inputs. Most of these works have recognized the applicability of SIS in real-world scenarios. However, to the best of our knowledge, our method is the first DM-based model to address the issue of noisy user inputs in SIS.

Conditional Diffusion Models. By modifying the U-Net (Ronneberger et al., 2015) architecture to incorporate the conditions into the network, conditional diffusion models are prevalently leveraged for conditional generations. ADM (Dhariwal & Nichol, 2021), for example, encodes the class embedding into the network with AdaGN and utilizes classifier guidance. LDM (Rombach et al., 2022) extracts features from the various conditions and encodes them into the network through concatenation or cross-attention. DDMI (Park et al., 2024) also generates data with latent diffusion but in a domain-agnostic manner, by linking data to a continuous function within a shared latent space. Meanwhile, SDM (Wang et al., 2022b) replaces GroupNorm of U-Net decoder with SPADE to embed the labels into the network in a spatially adaptive manner. UNIT-DDPM (Sasaki et al., 2021) uses two different diffusion models and a domain translation function for the training and sampling of an image-to-image translation model. Recently, T2I diffusion models such as FLIS (Xue et al., 2023), ControlNet (Zhang et al., 2023), and GLIGEN (Li et al., 2023) have paved the way for adding spatial controls to large pretrained diffusion models. Our method diffuses labels with a carefully designed discrete diffusion process and generates images conditioned on the diffused labels, formulating a novel conditional diffusion model.

3. Preliminaries

In this work, we consider conditional diffusion model (DM) $p_\theta(\mathbf{x}_0|\mathbf{y}_0)$ for SIS task. We briefly introduce existing DM-based models that learn the conditional distribution $q(\mathbf{x}_0|\mathbf{y}_0)$ by estimating a reverse process given a fixed label $\mathbf{y}_0$ that approximates an unconditional forward process $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$. Then, as our method perturbs labels $\mathbf{y}_0$ using a discrete diffusion, we summarize basic concepts of a discrete state space diffusion process.

Semantic Image Synthesis with Diffusion Models. In previous methods, the forward process is defined as an unconditional diffusion model that adds Gaussian noise to the image as follows:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\mathbf{x}_{t-1}, \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)\mathbf{I}\right), \tag{1}$$

where the decreasing sequence $\alpha_{1:T}$ defines the noise level with strictly positive α_t. For the forward process, we have a closed-form sampling step of $\mathbf{x}_t$ at an arbitrary timestep t, $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I})$, as it is defined as a Markov chain. The reverse process $p_\theta(\mathbf{x}_{0:T}|\mathbf{y}_0)$ is also defined as a Markov chain with learned transitions starting from $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ given as:

$$p_\theta(\mathbf{x}_{0:T}|\mathbf{y}_0) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_0). \tag{2}$$

It is usually learned by a deep neural network parameterized by θ that represents Gaussian transitions given as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_0) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, \mathbf{y}_0, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, \mathbf{y}_0, t)). \tag{3}$$

The conditional DM can be trained with the hybrid loss from (Nichol & Dhariwal, 2021) as:

$$\begin{aligned} \mathcal{L}_{\text{hybrid}} &= \mathcal{L}_{\text{simple}} + \lambda\mathcal{L}_{\text{vlb}}, \qquad (4)\\ \mathcal{L}_{\text{simple}} &= \mathbb{E}_{t,\mathbf{x}_0,\mathbf{y}_0,\epsilon}\left[||\epsilon - \epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon, \mathbf{y}_0, t)||_2^2\right], \\ \mathcal{L}_{\text{vlb}} &= D_{\text{KL}}\left(p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_0)||q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\right), \end{aligned}$$

where λ is a balancing hyperparamer and ϵ_θ is a noise prediction model.

Discrete State Space Diffusion Process. For a discrete categorical random variable $z \in \{1, ..., C\}$ with C categories, DMs for discrete state spaces (Hoogeboom et al., 2021; Austin et al., 2021) are defined with transition matrices where $[\boldsymbol{Q}_t]_{ij} = q(z_t = i|z_{t-1} = j)$ and $\boldsymbol{Q}_t \in \mathbb{R}^{C\times C}$. The forward process is then defined as follows:

$$q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) := \text{Cat}(\boldsymbol{z}_t; \boldsymbol{p} = \boldsymbol{Q}_t\boldsymbol{z}_{t-1}), \tag{5}$$

where $\boldsymbol{z}$ is the one-hot column vector ($\mathbf{e}_z$) and $\text{Cat}(\boldsymbol{z}; \boldsymbol{p})$ is a categorical distribution parameterized by $\boldsymbol{p}$. We can sample $\boldsymbol{z}_t$ at an arbitrary timestep t from the following marginal starting from $\boldsymbol{z}_0$ in closed form, due to the Markov property:

$$\begin{aligned} q(\boldsymbol{z}_t|\boldsymbol{z}_0) &:= \text{Cat}(\boldsymbol{z}_t; \boldsymbol{p} = \overline{\boldsymbol{Q}}_t\boldsymbol{z}_0), \\ \text{where } \overline{\boldsymbol{Q}}_t &= \boldsymbol{Q}_t\boldsymbol{Q}_{t-1}...\boldsymbol{Q}_1. \end{aligned} \tag{6}$$

As the design choice of the transition matrix $\boldsymbol{Q}_t$ determines the diffusion process, one has to choose the matrix carefully. For instance, D3PM (Austin et al., 2021) controlled the data corruption by designing the matrix with domain knowledge or structure such as text token embedding distance.

4. Stochastic Conditional Diffusion Model

We propose Stochastic Conditional Diffusion Model (SCDM), a robust and novel conditional diffusion model for semantic image synthesis. In this section, we introduce our forward and generation processes of SCDM (Section 4.1), discrete label diffusion process (Section 4.2), and the training and sampling schemes of our method (Section 4.3).

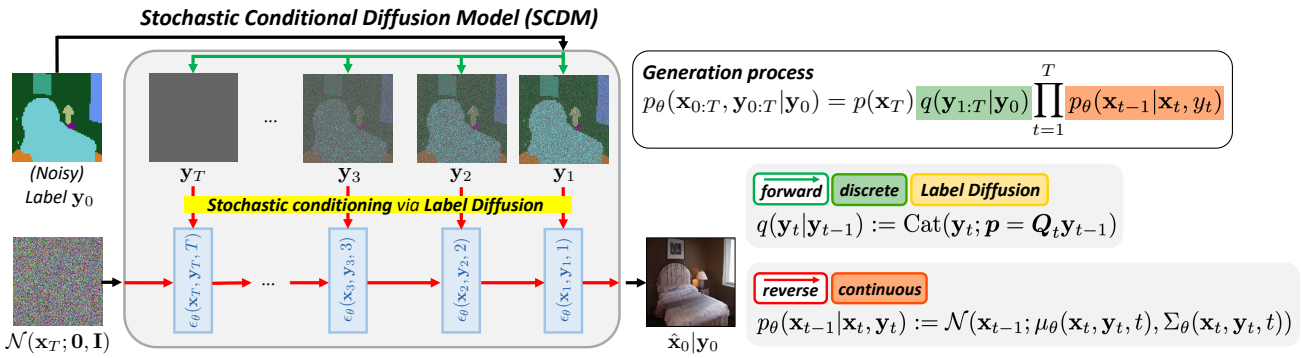


Figure 2. **Generation process of SCDM.** The Stochastic Conditional Diffusion Model (SCDM) is a robust conditional diffusion model for semantic image synthesis. SCDM consists of a *discrete forward process* for labels and a *continuous reverse process* for images. It improves the robustness to noisy semantic labels as well as generation performance on clean semantic labels. $\boldsymbol{z}_0$ denotes the i-th pixel of the semantic map, *i.e.*, $\boldsymbol{z}_0 = \mathbf{y}_0^i$ where $\mathbf{y}_0 = \{\mathbf{y}_0^1, ..., \mathbf{y}_0^{H \times W}\}$.

4.1. Definitions

Stochastic Conditional Diffusion Model (SCDM) is a class of conditional diffusion models approximating the conditional distribution $q(\mathbf{x}_0|\mathbf{y}_0)$. It conditions on diffused labels, *i.e.*, $\mathbf{y}_{1:T}$, given $\mathbf{y}_0$.

Our SCDM is defined as follows:

$$p_\theta(\mathbf{x}_0|\mathbf{y}_0) := \int\int p_\theta(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}|\mathbf{y}_0) \mathrm{d}\mathbf{x}_{1:T} \mathrm{d}\mathbf{y}_{1:T}, \quad (7)$$

where $\mathbf{x}_{1:T}$ and $\mathbf{y}_{1:T}$ are latents with the same dimensionality as $\mathbf{x}_0$ and $\mathbf{y}_0$ respectively, and $(\mathbf{x}_0, \mathbf{y}_0) \sim q(\mathbf{x}_0, \mathbf{y}_0)$.

Forward process. SCDM consists of two diffusion processes: a *continuous* diffusion process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ for images as in Eq. (1) and a *discrete* diffusion process $q(\mathbf{y}_t|\mathbf{y}_{t-1})$ for categorical semantic labels as in Eq. (5). We name the discrete diffusion process Label Diffusion. Then, the forward process of SCDM is defined as follows:

$$q(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}|\mathbf{x}_0, \mathbf{y}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t, \mathbf{y}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}), \quad (8)$$

$$q(\mathbf{x}_t, \mathbf{y}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) := q(\mathbf{x}_t|\mathbf{x}_{t-1}) q(\mathbf{y}_t|\mathbf{y}_{t-1}). \quad (9)$$

For simplicity, we employed the same t for both the image $\mathbf{x}$ and the label $\mathbf{y}$, but the noise levels are separately controlled. We manually designed a different noise schedule for the labels, referred to as the class-wise noise schedule, which will be discussed in the following section. Consequently, the diffusion processes for $\mathbf{x}$ and $\mathbf{y}$ are not synchronized regarding noise levels, and the synchronization between them is not necessary.

Note that although we diffuse images and labels independently, $\mathbf{x}_t$ and $\mathbf{y}_t$ are still correlated as $\mathbf{x}_0$ and $\mathbf{y}_0$ are dependent (*i.e.*, $\mathbf{y}_0$ is deterministically decided given $\mathbf{x}_0$).

Generation process. We define the joint distribution for our generation process as:

$$p_\theta(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}|\mathbf{y}_0) := p(\mathbf{x}_T) q(\mathbf{y}_{1:T}|\mathbf{y}_0) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_t), \quad (10)$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, \mathbf{y}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, \mathbf{y}_t, t)), \quad (11)$$

where $q(\mathbf{y}_{1:T}|\mathbf{y}_0)$ is the discrete Label Diffusion forward process and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_t)$ is the continuous reverse process. As $\mathbf{y}_0$ is given in SIS, we only have to sample the image $\hat{\mathbf{x}}_0$ and we do not need a reverse process for the label. Therefore, we define our generation process with the forward process $q(\mathbf{y}_{1:T}|\mathbf{y}_0)$ and the intermediate $\mathbf{y}_{1:T}$ are obtained from $\mathbf{y}_0$ without any neural network evaluations.

4.2. Label Diffusion

We introduce our Label Diffusion, a new discrete diffusion with *label masking* and *class-wise noise scheduling*.

4.2.1. Transition Matrix for Label Masking

To gradually erase the information of the semantic map and increase similarity among different maps as $t = 0 \rightarrow t = T$, we designed our Label Diffusion process by progressively masking labels. In other words, the original semantic labels are converted into the absorbing state (*i.e.*, `[mask]`) with some probability at each timestep. Consequently, all semantic maps eventually become identical at $t = T$, each filled with `[mask]` at every pixel. In addition, as the semantic labels are discrete variables, *i.e.*, classes, it is natural to adopt the discrete diffusion.

Given semantic classes C and absorbing state `[mask]`, we define the transition matrix $\boldsymbol{Q}_t \in \mathbb{R}^{(C+1)\times(C+1)}$ at a

timestep t as:

$$[\boldsymbol{Q}_t]_{ij} = \begin{cases} 1-\beta_{t,c} & \text{if} \quad i=j=c, \\ \beta_{t,c} & \text{if} \quad i=C+1, j=c, \\ 1 & \text{if} \quad i=j=C+1, \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

where the absorbing state is added as $(C+1)$-th class and $\beta_{t,c}$ is the probability of a label of class c to be masked. Note that class-wise defined probability $\beta_{t,c}$ enables class-wise *noise scheduling*, which will be further discussed in the following section. We assume that $\boldsymbol{Q}_t$ is applied to each pixel of the semantic map independently. Since `[mask]` is the absorbing state, once the state is in $C+1$ then the next state is always $C+1$.

In addition, the probability $q(\boldsymbol{z}_t|\boldsymbol{z}_0)$ at t starting from $\boldsymbol{z}_0$, which is one-hot vector with 1 at c-th entry as in Eq. (6), is:

$$q(\boldsymbol{z}_t|\boldsymbol{z}_0) = \text{Cat}\left(\boldsymbol{z}_t; \boldsymbol{p} = \overline{\boldsymbol{Q}}_t \boldsymbol{z}_0 = (1-\gamma_{t,c})\boldsymbol{z}_0 + \gamma_{t,c}\mathbf{e}_{C+1}\right), \tag{13}$$

where $\gamma_{t,c} = 1 - \prod_{i=1}^{t}(1-\beta_{i,c})$ denotes the probability that a semantic label c has been assigned to the absorbing state until timestep t and $\mathbf{e}_{C+1}$ is the one-hot column vector where $(C+1)$-th entry is 1. Utilizing $\gamma_{t,c}$ not only simplifies the implementation of the transition kernel but also reduces memory consumption in the generation process. The entire trajectory $\mathbf{y}_{1:T}|\mathbf{y}_0$ can be efficiently represented with a single $\mathbb{R}^{H\times W}$ matrix. For more details, see Appendix C.4.

4.2.2. Noise Scheduling

We observe that the semantic information of small objects in an image is prone to be lost at a relatively early stage of diffusion compared to larger objects. Moreover, for a rare class of objects in the dataset, it would be hard to learn their semantics if their labels are masked quickly. Thus, we propose a **class-wise noise schedule** to differentially transform semantic labels depending on the class. We designed $\gamma_{t,c}$ to ensure that labels occupying smaller areas and rarely appearing in the dataset are transitioned into the absorbing state more slowly and at a later time. The class-wise schedule improves generation quality for small and rare objects. For more details, see Section 6.2 and Appendix G.4.

For a given class c, we introduce ψ_c as defined in Eq. (15) and ϕ_c in Eq. (16). These terms take into account the area ($\approx$ object size) and frequency of the class c, respectively, for noise scheduling. We estimated ψ and ϕ with training data. Using ψ and ϕ, we define $\gamma_{t,c}$ for the class-wise noise schedule as:

$$\gamma_{t,c} := \frac{(\psi_c\phi_c)^{\eta\frac{t}{T}} - 1}{(\psi_c\phi_c)^{\eta} - 1}, \tag{14}$$

$$\psi_c = \mathbb{E}_{\mathbf{x}\in\mathcal{X}_c}\left[\mathbf{Pr}(y_{ij}=c|\mathbf{x})\right]^{-1}, \tag{15}$$

$$\phi_c = \log\left(\mathbf{Pr}(\mathbf{x}\in\mathcal{X}_c)^{-1}\right), \tag{16}$$

where $\mathcal{X}_c$ is the set of images containing class c, y_{ij} is the class label of the semantic map $\mathbf{y}$ at (i,j), and η is a hyperparameter. This properly slows down the label diffusion of small and rare objects. We provide a visual aid for $\gamma_{t,c}$ with different $\psi_c\phi_c$ values in Appendix D.

Proposition 1. *For $\gamma_{t,c}$ in Eq. (14) with $\psi_c\phi_c > 1$ for all c and $t < T$,*

$$\lim_{\eta\to 0}\gamma_{t,c} = \frac{t}{T} \quad \textit{and} \quad \lim_{\eta\to\infty}\gamma_{t,c} = 0.$$

Interestingly, our class-wise schedule generalizes the linear and uniform schedule, and no Label Diffusion. As $\eta \to 0$, the class-wise schedule defined in (14) converges to a linear and uniform noise schedule, *i.e.*, $\frac{t}{T}$. Labels across all classes have the same probability to be masked, and the marginal probability (13) linearly increases. This schedule is the same as the one leveraged in absorbing-state D3PM (Austin et al., 2021). Also, as $\eta \to \infty$, the masking probability approaches zero. Formally, this property is summarized in Proposition 1, and its proof is provided in Appendix A.2.

4.3. Training and Sampling

We train our network with the following loss function:

$$\mathcal{L} = \mathcal{L}_{\text{simple}} + \lambda\mathcal{L}_{\text{vlb}}, \tag{17}$$

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{y}_t,\epsilon}\left[||\epsilon - \epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon, \mathbf{y}_t, t)||_2^2\right], \tag{18}$$

$$\mathcal{L}_{\text{vlb}} = D_{\text{KL}}\left(p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)||q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\right), \tag{19}$$

where α_t determines the noise level for input image $\mathbf{x}$ at timestep t, $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ is a Gaussian noise, and λ is a balancing hyperparameter. This is similar to the hybrid loss (Nichol & Dhariwal, 2021), with a slight adaptation of using $\mathbf{y}_t$. See Appendix A.1 for a detailed derivation of the objective function. Note that Label Diffusion does not significantly impact the training cost of the main diffusion.

To generate a sample, by the definition of the generation process, we first sample $\mathbf{y}_{1:T}$ from the semantic map $\mathbf{y}_0$ and feed them sequentially to the process. In addition, we formulate the classifier-free guidance (Ho & Salimans, 2021) in our model:

$$\tilde{\epsilon}_\theta(\mathbf{x}_t|\mathbf{y}_t) = \epsilon_\theta(\mathbf{x}_t|\mathbf{y}_t) + s(\epsilon_\theta(\mathbf{x}_t|\mathbf{y}_t) - \epsilon_\theta(\mathbf{x}_t|\emptyset)), \tag{20}$$

where s is the guidance scale. This is similar to a line of works (Nichol et al., 2022; Rombach et al., 2022; Wang et al., 2022b) adopting the guidance, but we use the perturbed labels $\mathbf{y}_t$ by our discrete forward process instead of the clean and fixed label $\mathbf{y}_0$ for all steps.

Extrapolation. Inspired by (Lu et al., 2022), we give additional guidance in $\mathbf{x}_0$ space, as opposed to directly

sampling $\mathbf{x}_{t-1}$. We first compute $\mathbf{x}_0^{(t)}$ through $\tilde{\epsilon}_\theta(\mathbf{x}_t|\mathbf{y}_t)$. This value is then extrapolated from the preceding time-step prediction, $\tilde{\mathbf{x}}_0^{(t+1)}$, using the formula:

$$\tilde{\mathbf{x}}_0^{(t)} = \mathbf{x}_0^{(t)} + w\left(\mathbf{x}_0^{(t)} - \tilde{\mathbf{x}}_0^{(t+1)}\right), \tag{21}$$

where the constant extrapolation scale is denoted by w. Following extrapolation, we apply dynamic thresholding (Saharia et al., 2022) and subsequently randomly sample $\mathbf{x}_{t-1}$ utilizing $\mathbf{x}_t$ and $\tilde{\mathbf{x}}_0^{(t)}$.

The complete training and sampling algorithm is in Appendix B. Note that the overall algorithms remain mostly unchanged and the only modification involves incorporating `[mask]` into labels through Label Diffusion. This introduces minimal computational overhead.

4.4. Discussion

We here theoretically analyze the generative processes of SCDM to discuss an interesting relationship with the *fixed* conditional diffusion model (baseline). SCDM approximates the following conditional score:

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{y}_t) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t), \tag{22}$$

while the baseline approximates the following:

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{y}_0) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t). \tag{23}$$

As the unconditional score $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$ is identical for both models, we analyze the relationship between class guidance (gradients of implicit classifiers (Ho & Salimans, 2021; Dhariwal & Nichol, 2021)), *i.e.*, $\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t)$ and $\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t)$ in the following proposition.

Proposition 2. *Suppose there exists a differentiable function f_i such that $q(\mathbf{y}_0^i|\mathbf{x}_t) = Cat(\mathbf{y}_0^i; p = f_i(\mathbf{x}_t))$ and $\mathbf{y}_0^1|\mathbf{x}_t, ..., \mathbf{y}_0^{H\times W}|\mathbf{x}_t$ are independent, where $i \in \{1, ..., H \times W\}$ denotes the index of a pixel in semantic map and $\mathbf{y}_0^i \in \mathbb{R}^{C+1}$ is a one-hot vector. With $q(\mathbf{y}_t^i|\mathbf{y}_0^i)$ from Eq. (13) and $\gamma_{t,c} = \gamma_t$ for any c, we have the following relationship;*

$$\mathbb{E}_{q(\mathbf{y}_t|\mathbf{y}_0)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t)] = (1-\gamma_t)\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t).$$

The proof is available in Appendix G.5. Proposition 2 implies that the expectation of implicit classifier gradients in our method over all possible $\mathbf{y}_t$ given $\mathbf{y}_0$ is equivalent to $\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t)$ after time-dependent scaling. First, the scaling factor $(1-\gamma_t)$ starts from 0 when $t = T$ and is set to 1 when $t = 0$. In other words, our method acts like an unconditional generation at the beginning of the reverse process and gets stronger guidance as t goes to 0. Second, the expectation of the classifier gradient in our method is the same direction as the one in the baseline with fixed class labels. Note that this does *not* mean that our method has the same guidance as the baseline with fixed labels and time-dependent scaling. For more discussion, see Appendix G.5.

5. Experiments

5.1. Noisy SIS benchmark

We evaluate our method based on ADE20K (Zhou et al., 2017) dataset. ADE20K contains 20K images for training and 2K images for test annotated with 151 classes including the 'unlabeled' class. Additionally, we introduce three new experimental setups to assess generation performance under noisy conditions using the ADE20K dataset as follows:

[DS] This setup employs *downsampled* semantic maps that are resized by nearest-neighbor interpolation. This setup simulates human errors such as jagged edges and coarse/low-resolution user inputs. We downsample the semantic maps to 64×64 and then upsample them to 256×256. Consequently, the label maps contain jagged edges.

[Edge] This setup masks the *edges* of instances with an unlabeled class. This setup imitates incomplete annotations around edges, especially between instances. We observed that human annotators occasionally leave the pixels on the boundaries of instances as 'unlabeled' due to their inherent ambiguity, see Appendix F for examples. Assuming a pixel with a different class compared to its neighbor is the edge of the instance, we detect edges using the label map. Then, we fill the semantic map pixels with a distance of 2 or less from the edges with the unlabeled class.

[Random] This setup *randomly* adds an unlabeled class to the semantic maps (10%). This setup mimics unintended user errors and extreme random noise.

More experimental results with other benchmark datasets (*e.g.*, CelebAMask-HQ (Lee et al., 2020) and COCO-Stuff (Caesar et al., 2018)) are in Appendix G.

5.2. Experimental Setup

We adopt Fréchet Inception Distance (FID) (Heusel et al., 2017) to evaluate generation quality and mean Intersection-over-Union (mIoU) to assess the alignment of the synthesis results with ground truth semantic maps. We compare our model with GAN-based methods and DM-based methods. All the baselines and ours are trained with clean benchmark datasets, and tested on the noisy SIS benchmark. We present our results on noisy labels sampled over 25 steps in Section 5.3. More information on baselines and implementation details are provided in Appendix C.2 and C.3, respectively.

Table 1. **SIS with noisy labels.** F is FID and lower is better. M is mIoU and higher is better. The bottom three rows are DM-based, and the others are GAN-based methods. The best results among the diffusion-based approaches are boldfaced, and the best results overall are underlined.

Category	Methods	DS		Edge		Random	
		F(↓)	M(↑)	F(↓)	M(↑)	F(↓)	M(↑)
GAN	SPADE	36.6	41.0	40.2	39.7	92.7	18.5
	CC-FPSE	42.0	40.4	38.7	41.7	141.3	16.2
	DAGAN	37.6	41.3	42.2	39.0	109.1	16.4
	GroupDNet	45.9	28.9	49.9	28.0	76.0	21.2
	OASIS	34.5	48.1	37.6	47.4	54.2	42.1
	CLADE	37.4	41.8	41.9	40.1	71.8	29.9
	INADE	36.1	38.3	40.3	35.6	61.3	30.2
	SCGAN	38.1	44.0	63.9	43.3	40.9	38.6
	SAFM*	36.7	50.1	40.0	44.9	80.6	34.1
DM	SDM	35.5	43.8	39.4	39.4	141.9	11.8
	LDM	38.9	28.1	39.5	26.0	36.3	27.1
	Ours	**32.4**	**44.7**	**31.2**	**40.1**	**28.1**	**45.2**

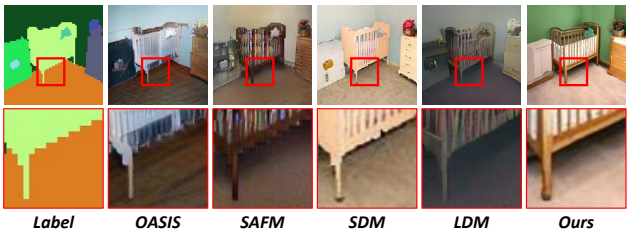
5.3. SIS with Noisy Labels

Table 1 shows the performances of our method and baselines under the three settings. SCDM demonstrates its superior robustness to all three types of noise in generation quality measured by FID, compared to other baselines. Notably, in the three setups, the performance gaps between the best baseline scores and our method are +2.1, +6.4, and +8.2, respectively. We also evaluate the semantic correspondence between the clean ground-truth label maps and the generation results and report mIoU scores. Our method achieved the best mIoU performance among DM-based models in all three settings. Compared to strong GAN-based baselines, including SAFM (Lv et al., 2022) (denoted with '*' in Table 1) that leverages extra ground truth instance maps during sampling, our results show comparable results in correspondence.

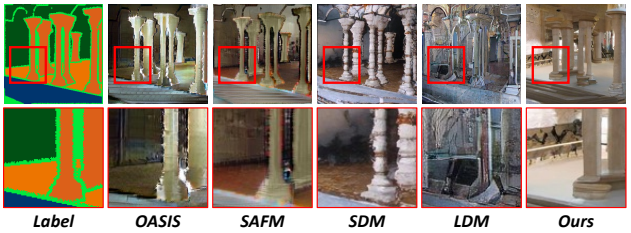
We present the qualitative comparisons in Figure 3. While the baselines synthesized the jagged or unnatural crib images given the low-resolution semantic label, ours produced clean edges and generated a realistic image, as shown in Figure 3(a). In Figure 3(b), our approach naturally fills in the unlabeled edge areas, whereas the baselines fail to generate realistic images, especially on the 'unlabeled' edges. As shown in Figure 3(c), ours successfully generates when conditioned on randomly corrupted masks, while others fail and synthesize artifacts.

5.4. SIS with Clean Labels (Standard SIS)

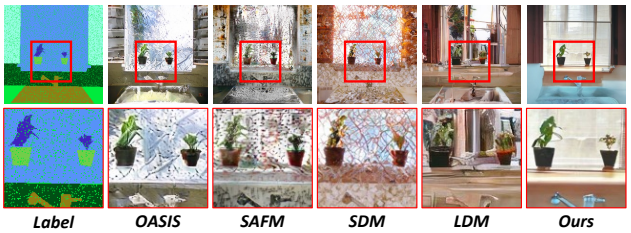
We also evaluate our method in a standard SIS setting with clean labels. In this experiment, we additionally adopt LPIPS (Zhang et al., 2018) as a diversity metric. Table 2 shows that SCDM achieves the best performance in all three



(a) Masks with jagged edges (DS)



(b) Incomplete masks (Edge), limegreen areas denote 'unlabeled'.



(c) Corrupted masks (Random)

Figure 3. **Generation results on noisy labels.**

metrics (*e.g.*, FID, LPIPS, mIoU) compared to recent DM-based baselines in all datasets. Also, including GAN-based models, the proposed method shows comparable performances. Specifically, our method surpasses all baselines on CelebAMask-HQ in all three metrics with a significant gain of +1.1 (FID) compared to the state-of-the-art method. Qualitative comparisons and more detailed analysis are in Appendix G.1.

6. Analysis

In this section, we analyze our method to understand (1) the efficacy of Label diffusion and extrapolation and (2) the effect of the class-wise noise schedule.

6.1. Ablation Study

To further demonstrate SCDM's enhanced robustness against noisy labels, we first show the effect of our components in the original (clean) benchmark. Subsequently, we show that our samples generated with noisy labels closely resemble the results obtained with clean labels, supporting our motivation in Figure 1.

Table 2. **Quantitative performance comparison on generation quality.** The baseline methods are categorized into Generative Adversarial Networks (GAN) and Diffusion Models (DM). For FID, lower is better. For LPIPS and mIoU, higher is better. The best results among the diffusion-based approaches are boldfaced, and the best results overall are underlined. '-' indicates that the method did not report the metric or train the dataset, or the checkpoint or samples are not publicly available. 'Seg' denotes that the method leverages a pre-trained segmentation network during training. '†' denotes a one-shot method.

	Methods	Seg	CelebAMask-HQ			ADE20K			COCO-Stuff		
			FID(↓)	LPIPS(↑)	mIoU(↑)	FID(↓)	LPIPS(↑)	mIoU(↑)	FID(↓)	LPIPS(↑)	mIoU(↑)
GAN	RESAIL (Shi et al., 2022)	✓	-	-	-	30.2	-	49.3*	18.3	-	44.7
	SAFM (Lv et al., 2022)	✓	-	-	-	32.8	-	52.6	24.6	-	43.3
	ECGAN (Tang et al., 2023a)	✓	-	-	-	25.8	0.52	50.6*	15.7	-	46.3
	ECGAN++ (Tang et al., 2023b)	✓	-	-	-	24.7	0.54	52.7*	14.9	-	47.9
	pix2pixHD (Wang et al., 2018)	✗	38.5	0	76.1	81.8	0	20.3*	111.5	0	14.6
	SPADE (Park et al., 2019)	✗	29.2	0	75.2	33.9	0	44.5	33.9	0	36.9
	CC-FPSE (Liu et al., 2019)	✗	-	-	-	31.7	0.078	47.3	19.2	0.098	40.8
	DAGAN (Tang et al., 2020a)	✗	29.1	0	76.6	31.9	0	45.5	-	-	-
	GroupDNet (Zhu et al., 2020b)	✗	25.9	0.365	76.1	41.7	0.230	33.7	-	-	-
	OASIS (Sushko et al., 2020)	✗	-	-	-	28.3	0.286	50.9	17.0	0.328	44.2
	INADE (Tan et al., 2021a)	✗	21.5	0.415	74.1	35.2	0.459	41.4	-	-	-
	SCGAN (Wang et al., 2021)	✗	20.8	0	75.5	29.3	0	50.0	18.1	0	41.7
	CLADE (Tan et al., 2021b)	✗	30.6	0	75.4	35.4	0	44.7	29.2	0	36.9
	Unconditional Generator† (Chae et al., 2024)	✗	18.5	-	53.1	-	-	-	-	-	-
DM	SDM (Wang et al., 2022b)	✗	18.8	0.404	77.0	27.5	0.524	48.7	15.9	0.518	34.9
	LDM (Rombach et al., 2022)	✗	21.5	0.315	74.6	36.5	0.417	23.2	-	-	-
	PITI (Wang et al., 2022a)	✗	-	-	-	27.3	-	-	15.8	0.489	32.2
	Ours	✗	**17.4**	**0.418**	**77.2**	**26.9**	**0.530**	**49.4**	**15.3**	**0.519**	**38.1**

Table 3. **Ablation study on *ADE20K*.** For FID (F), lower is better. For LPIPS (L) and mIoU (M), higher is better.

	Method	25 steps			50 steps			100 steps			250 steps			1000 steps		
		F(↓)	L(↑)	M(↑)	F(↓)	L(↑)	M(↑)	F(↓)	L(↑)	M(↑)	F(↓)	L(↑)	M(↑)	F(↓)	L(↑)	M(↑)
(a)	Base	44.6	0.471	33.8	35.8	0.489	47.1	31.9	0.500	48.2	29.3	0.506	48.6	28.1	0.508	48.6
(b)	+ Label Diffusion	39.5	0.492	45.5	33.6	0.513	47.3	29.8	**0.522**	48.6	27.7	0.528	48.7	26.9	0.530	48.8
(c)	+ Extrapolation	**27.7**	**0.518**	**48.7**	**27.0**	**0.525**	**48.7**	**26.7**	**0.522**	**49.8**	**26.7**	**0.530**	**49.6**	**26.8**	**0.531**	**49.9**

6.1.1. Does SCDM successfully estimate $q(X|Y)$?

The effect of two components of SCDM (Label Diffusion and extrapolation) is analyzed by an ablation study in Table 3. In this analysis, we adopt LPIPS (Zhang et al., 2018) to additionally compare the generation diversity and measure the average distance between multi-modal synthesis results. We compare the following; (a) Base generates images conditioned on original fixed semantic maps $\mathbf{y}_0$, (b) +Label Diffusion generates images conditioned on perturbed labels by our Label Diffusion, and (c) +Extrapolation uses Eq. (21) instead of direct sampling of $\mathbf{x}_{t-1}$ on top of (b). The results are reported for the few-step (25, 50, 100, and 250 steps) and the full-step (1000 steps) settings.

By eliminating all of our components, (c) → (a), the performance of all three metrics significantly declined for all sampling steps. This empirically shows that our SCDM successfully estimates the conditional distribution $q(X|Y)$. The degradation became more substantial when omitting extrapolation, (b) → (a), in fewer steps, highlighting its significance in the few-step generation. Surprisingly, full SCDM (c) with only 25 steps (FID of 27.7) outperforms the baseline (a) with 1000 steps (FID of 28.1). This supports the effectiveness of the proposed method.

6.1.2. Does Label Diffusion contribute to robustness?

Furthermore, we examine the effect of Label Diffusion in the noisy SIS setting to verify the robustness of our method, as depicted in Figure 1. Specifically, we demonstrate that the generation results given clean and noisy labels are similar. To rigorously analyze the effects, we fix the random seeds and use the same $\mathbf{x}_T$. Then we compare the generation results of our method and the baseline, conditioned on clean (original dataset) semantic maps with results of noisy (DS, Edge, and Random) semantic maps, respectively. For a quantitative comparison, we adopt the following metrics:

Table 4. **Ablation study on noisy SIS.** Generation results with and without Label Diffusion are compared, where samples generated from each noisy dataset are compared with those from the clean dataset. For LPIPS and FID, lower is better. For SSIM and PSNR, higher is better.

Dataset	Method	Metric			
		LPIPS(↓)	SSIM(↑)	PSNR(↑)	FID(↓)
DS	Baseline	0.221	0.823	30.4	24.2
	Ours	**0.180**	**0.865**	**31.3**	**19.0**
Edge	Baseline	0.248	0.771	29.7	32.7
	Ours	**0.223**	**0.825**	**30.2**	**20.0**
Random	Baseline	0.560	0.427	28.1	145.6
	Ours	**0.076**	**0.944**	**32.9**	**10.0**

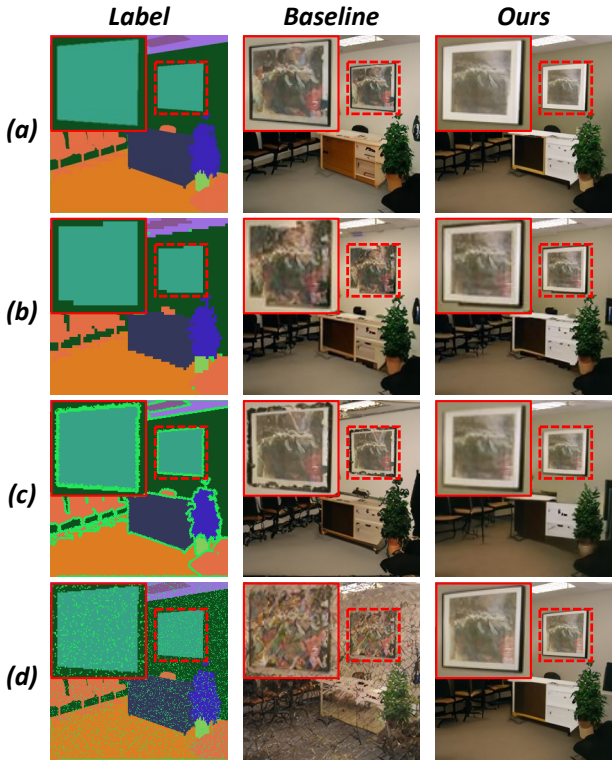


Figure 4. **Generation results with and without Label Diffusion.** The results are sampled with the fixed random seeds and the same $\mathbf{x}_T$, and generated with (a) clean labels, (b) DS, (c) Edge, and (d) Random setup noisy labels, respectively.

LPIPS (perceptual similarity), SSIM (structural similarity), PSNR (peak signal-to-noise ratio), and FID, and details are in Appendix C.1. LPIPS, SSIM, and PSNR are calculated *sample*-wise, comparing each pair of generated samples of the clean and noisy semantic maps, while FID compares the distribution of the generated *set* of images.

Table 5. **mIoU per each group on *ADE20K*.** The classes are grouped based on their $\psi_c\phi_c$ scores.

Noise Schedule	All	Frequent	Common	Rare
Linear & uniform	43.0	56.1	40.8	32.2
Class-wise	**49.4** (+6.4)	**60.3** (+4.2)	**47.7** (+6.9)	**38.4** (+8.1)

Results in Table 4 indicate that Label Diffusion significantly contributes to robust generation, with our samples exhibiting better similarity in all four metrics. Particularly, ours resulted in +0.041, +0.025, and +0.484 of LPIPS gain over baseline in DS, Edge, and Random settings, respectively. Figure 4 presents a qualitative comparison supporting our intuition behind SCDM. Without Label Diffusion (Baseline in Figure 4), the generated results show inconsistency, whereas samples conform to the result of clean labels when Label Diffusion is employed (Ours in Figure 4).

6.2. Effect of Class-wise Noise Schedule

In this section, we elucidate the effect of our class-wise noise schedule by comparing the generation results of two different models using the class-wise schedule and linear and uniform schedule, *i.e.*, $\eta \to 0$. We observe that the class-wise schedule clearly improves the image quality, especially in terms of semantic correspondence (mIoU (↑) of **49.4** (class-wise) $>$ 43.0 (uniform) on ADE20K). The quantitative and qualitative results are presented in the Appendix G.4. Furthermore, the class-wise schedule exhibits superiority in small and rare class synthesis. By organizing the classes into three groups - frequent, common, and rare - based on their $\psi_c\phi_c$ scores, we compare mIoU per each group on ADE20K and report the performance in Table 5. Notably, our class-wise schedule exhibited mIoU gain in all the groups, with the highest performance improvement of +8.1 in the rare group.

7. Conclusion

This paper introduces SCDM, a novel and robust conditional diffusion model for semantic image synthesis. The discrete diffusion for labels, which we name Label Diffusion, is designed with label masking. Label Diffusion ensures that the intermediate labels along the generation process become similar and eventually identical at $t = T$. Additionally, the class-wise noise schedule improves the generation quality for small and rare objects. We define the generation process with a discrete forward process of labels and a continuous reverse process of images, as the labels are given in SIS. SCDM demonstrates its robustness in noisy SIS setups which we designed to simulate human errors in real-world applications.

Acknowledgements

This work was partly supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT)(IITP-2024-2020-0-01819, 10%), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(NRF-2023R1A2C2005373, 30%), the Virtual Engineering Platform Project (Grant No. P0022336, 30%), funded by the Ministry of Trade, Industry & Energy (MoTIE, South Korea), and the National Supercomputing Center with supercomputing resources including technical support (KSC-2022-CRE-0261, 30%).

Impact Statement

This paper focuses on robust generation against noisy user input, a crucial aspect in real-world applications. Since clean labels and benchmark datasets inherently contain biases, our model might reinforce these biases when its samples are widely used across the Internet. Nevertheless, our research opens new avenues for future work in image generation for real-world problems, potentially expanding the application of diffusion models in practical scenarios.

References

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, 2021.

Caesar, H., Uijlings, J., and Ferrari, V. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.

Chae, J., Cho, H., Go, S., Choi, K., and Uh, Y. Semantic image synthesis with unconditional generator. In *NeurIPS*, 2024.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

Chen, Q. and Koltun, V. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017.

Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., and Qiao, Y. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

Couairon, G., Verbeek, J., Schwenk, H., and Cord, M. Diffedit: Diffusion-based semantic image editing with mask guidance. In *ICLR*, 2023.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *NeurIPS*, 2014.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPSW*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. In *NeurIPS*, 2021.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

Lee, C.-H., Liu, Z., Wu, L., and Luo, P. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020.

Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., and Lee, Y. J. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023.

Liu, X., Yin, G., Shao, J., Wang, X., et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *NeurIPS*, 2019.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *ICLR*, 2019.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

Lv, Z., Li, X., Niu, Z., Cao, B., and Zuo, W. Semantic-shape adaptive feature modulation for semantic image synthesis. In *CVPR*, 2022.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *ICML*, 2021.

Ntavelis, E., Romero, A., Kastanis, I., Gool, L. V., and Timofte, R. SESAME: semantic editing of scenes by adding, manipulating or erasing objects. In *ECCV*, 2020.

Park, D., Kim, S., Lee, S., and Kim, H. J. Ddmi: Domain-agnostic latent diffusion models for synthesizing high-quality implicit neural representations. In *ICLR*, 2024.

Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *ICML*, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

Sasaki, H., Willcocks, C. G., and Breckon, T. P. Unit-ddpm: Unpaired image translation with denoising diffusion probabilistic models. *arXiv preprint arXiv:2104.05358*, 2021.

Shi, Y., Liu, X., Wei, Y., Wu, Z., and Zuo, W. Retrieval-based spatially adaptive normalization for semantic image synthesis. In *CVPR*, 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Sushko, V., Schönfeld, E., Zhang, D., Gall, J., Schiele, B., and Khoreva, A. You only need adversarial supervision for semantic image synthesis. In *ICLR*, 2020.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

Tan, Z., Chai, M., Chen, D., Liao, J., Chu, Q., Liu, B., Hua, G., and Yu, N. Diverse semantic image synthesis via probability distribution modeling. In *CVPR*, 2021a.

Tan, Z., Chen, D., Chu, Q., Chai, M., Liao, J., He, M., Yuan, L., Hua, G., and Yu, N. Efficient semantic image synthesis via class-adaptive normalization. *TPAMI*, 2021b.

Tang, H., Bai, S., and Sebe, N. Dual attention gans for semantic image synthesis. In *ACM MM*, 2020a.

Tang, H., Xu, D., Yan, Y., Torr, P. H., and Sebe, N. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *CVPR*, 2020b.

Tang, H., Qi, X., Xu, D., Torr, P. H., and Sebe, N. Edge guided gans with semantic preserving for semantic image synthesis. In *ICLR*, 2023a.

Tang, H., Sun, G., Sebe, N., and Van Gool, L. Edge guided gans with multi-scale contrastive learning for semantic image synthesis. In *TPAMI*, 2023b.

Wang, T., Zhang, T., Zhang, B., Ouyang, H., Chen, D., Chen, Q., and Wen, F. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022a.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

Wang, W., Bao, J., Zhou, W., Chen, D., Chen, D., Yuan, L., and Li, H. Semantic image synthesis via diffusion models. *arXiv preprint arXiv:2207.00050*, 2022b.

Wang, Y., Qi, L., Chen, Y.-C., Zhang, X., and Jia, J. Image synthesis via semantic composition. In *ICCV*, 2021.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 2004.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., and Sun, J. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

Xue, H., Huang, Z., Sun, Q., Song, L., and Zhang, W. Freestyle layout-to-image synthesis. In *CVPR*, 2023.

Yang, D., Hong, S., Jang, Y., Zhao, T., and Lee, H. Diversity-sensitive conditional generative adversarial networks. In *ICLR*, 2019.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *CVPR*, 2017.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., and Shechtman, E. Toward multimodal image-to-image translation. In *NeurIPS*, 2017.

Zhu, P., Abdal, R., Qin, Y., and Wonka, P. Sean: Image synthesis with semantic region-adaptive normalization. In *CVPR*, 2020a.

Zhu, Z., Xu, Z., You, A., and Bai, X. Semantically multi-modal image synthesis. In *CVPR*, 2020b.

The appendix is organized into the following sections.

- Appendix A: Derivations
  - A.1 Variational Lower Bound for Stochastic Conditional Diffusion Model
  - A.2 Proof of Proposition 1
  - A.3 Proof of Proposition 2
- Appendix B: Algorithms
- Appendix C: Experimental Setup
  - C.1 Metrics
  - C.2 Baselines
  - C.3 Implementation Details
  - C.4 Efficient Trajectory Representation
- Appendix D: Detailed Explanations on ψ_c and ϕ_c
- Appendix F: ADE20K Dataset Annotation Examples
- Appendix E: Limitations
- Appendix G: Additional Experimental Results
  - G.1 Standard SIS Setting
  - G.2 Extrapolation Hyperparameter Search
  - G.3 Visualization of Label Diffusion
  - G.4 Effect of Class-wise Noise Schedule
  - G.5 Further Discussion and Analysis on Class Guidance
  - G.6 Validation of SCDM Generation Process
  - G.7 More Qualitative Results - Multimodal Generation
  - G.8 More Qualitative Results - SIS with Noisy Labels
  - G.9 More Qualitative Results - Standard SIS Setting

A. Derivations

A.1. Variational Lower Bound for Stochastic Conditional Diffusion Model

In this section, we provide a detailed derivation of the objective function (Eq. (17)) discussed in Section 4.3 of the main paper. We start by defining our Stochastic Conditional Diffusion Model as:

$$p_\theta(\mathbf{x}_0|\mathbf{y}_0) := \int\int p_\theta(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}|\mathbf{y}_0)\mathrm{d}\mathbf{x}_{1:T}\mathrm{d}\mathbf{y}_{1:T}, \tag{24}$$

where the generation process is defined as follows:

$$p_\theta(\mathbf{x}_{0:T}, \mathbf{y}_{1:T}|\mathbf{y}_0) := p(\mathbf{x}_T)q(\mathbf{y}_{1:T}|\mathbf{y}_0)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_t), \tag{25}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, \mathbf{y}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, \mathbf{y}_t, t)). \tag{26}$$

The diffusion forward process in which SCDM approximates is:

$$q(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}|\mathbf{x}_0, \mathbf{y}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t, \mathbf{y}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}), \tag{27}$$

$$q(\mathbf{x}_t, \mathbf{y}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) := q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{y}_t|\mathbf{y}_{t-1}). \tag{28}$$

Variational bound on negative log-likelihood of SCDM can be derived as follows:

$$\mathbb{E}_{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}\left[-\log p_\theta(\mathbf{x}_0|\mathbf{y}_0)\right] \tag{29}$$
$$= \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}{p_\theta(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}\right] \tag{30}$$
$$= \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}{p_\theta(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}\frac{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}\right] \tag{31}$$
$$= \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}\frac{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}{p_\theta(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}\right] \tag{32}$$
$$= \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}\right] + \mathbb{E}_q\left[-\log \frac{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}{p_\theta(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}\right] \tag{33}$$
$$= \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}\right] - \underbrace{D_{\mathrm{KL}}\left(q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)||p_\theta(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{y}_0)\right)}_{D_{\mathrm{KL}}\geq 0} \tag{34}$$
$$\leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}\right], \tag{35}$$

$$\mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T},\mathbf{y}_{1:T}|\mathbf{y}_0)}{q(\mathbf{x}_{1:T},\mathbf{y}_{1:T}|\mathbf{x}_0,\mathbf{y}_0)}\right] \tag{36}$$
$$= \mathbb{E}_q\left[-\log \frac{p(\mathbf{x}_T)q(\mathbf{y}_{1:T}|\mathbf{y}_0)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{\prod_{t=1}^T q(\mathbf{x}_t,\mathbf{y}_t|\mathbf{x}_{t-1},\mathbf{y}_{t-1})}\right] \tag{37}$$
$$= \mathbb{E}_q\left[-\log \frac{p(\mathbf{x}_T)q(\mathbf{y}_{1:T}|\mathbf{y}_0)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{y}_t|\mathbf{y}_{t-1})}\right] \tag{38}$$
$$= \mathbb{E}_q\left[-\log \frac{p(\mathbf{x}_T)q(\mathbf{y}_{1:T}|\mathbf{y}_0)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})\prod_{t=1}^T q(\mathbf{y}_t|\mathbf{y}_{t-1})}\right] \tag{39}$$
$$= \mathbb{E}_q\left[-\log \frac{p(\mathbf{x}_T)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] \tag{40}$$
$$= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t=1}^T \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L \tag{41}$$

Also, we can derive the reduced variance variational bound for SCDM as follows:

$$L = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t=1}^T \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] \tag{42}$$
$$= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t=2}^T \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1,\mathbf{y}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\right] \tag{43}$$
$$= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t=2}^T \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1,\mathbf{y}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\right] \tag{44}$$
$$= \mathbb{E}_q\left[-\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} - \sum_{t=2}^T \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1,\mathbf{y}_1)\right] \tag{45}$$
$$= \mathbb{E}_q\left[D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)) + \sum_{t=2}^T D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{y}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1,\mathbf{y}_1)\right]. \tag{46}$$

The only difference between the typical bound of a conditional DM and our bound is the substitution of $\mathbf{y}_0$ for $\mathbf{y}_t$ in $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_t)$. Therefore, we can use the DDPM simple loss ($\mathcal{L}_{\text{simple}}$) (Ho et al., 2020) or the hybrid loss ($\mathcal{L}_{\text{simple}} + \lambda\mathcal{L}_{\text{vlb}}$) (Nichol & Dhariwal, 2021) for training our Stochastic Conditional Diffusion Model.

A.2. Proof of Proposition 1

We provide the proof of Proposition 1, which shows that our class-wise noise schedule generalizes the linear and uniform schedule and no Label Diffusion, *i.e.*, a typical conditional DM described in Section 3. Proposition 1 and the following proof hold under two assumptions of $\psi_c\phi_c > 1$ for all c and $t < T$.

Proposition 1. *For $\gamma_{t,c}$ in Eq. (14) with $\psi_c\phi_c > 1$ for all c and $t < T$,*

$$\lim_{\eta\to 0} \gamma_{t,c} = \frac{t}{T} \quad \textit{and} \quad \lim_{\eta\to\infty} \gamma_{t,c} = 0.$$

Proof. First, when η converges to 0, the linear and uniform noise schedule can be derived as follows:

$$\begin{aligned}
\lim_{\eta\to 0} \gamma_{t,c} &= \lim_{\eta\to 0} \frac{(\psi_c\phi_c)^{\eta\frac{t}{T}} - 1}{(\psi_c\phi_c)^{\eta} - 1} && (47)\\
&= \lim_{\eta\to 0} \frac{(\psi_c\phi_c)^{\eta\frac{t}{T}} - (\psi_c\phi_c)^{0\frac{t}{T}}}{(\psi_c\phi_c)^{\eta} - (\psi_c\phi_c)^{0}} && (48)\\
&= \lim_{\eta\to 0} \frac{\frac{(\psi_c\phi_c)^{\eta\frac{t}{T}} - (\psi_c\phi_c)^{0\frac{t}{T}}}{\eta}}{\frac{(\psi_c\phi_c)^{\eta} - (\psi_c\phi_c)^{0}}{\eta}} && (49)\\
&= \lim_{\eta\to 0} \frac{\ln(\psi_c\phi_c)\frac{t}{T}(\psi_c\phi_c)^{\eta\frac{t}{T}}}{\ln(\psi_c\phi_c)(\psi_c\phi_c)^{\eta}} && (50)\\
&= \lim_{\eta\to 0} \frac{t}{T}(\psi_c\phi_c)^{\eta(\frac{t}{T}-1)} && (51)\\
&= \frac{t}{T}. && (52)
\end{aligned}$$

When η explodes to $+\infty$, the noise schedule without Label Diffusion can be derived as follows:

$$\begin{aligned}
\lim_{\eta\to +\infty} \gamma_{t,c} &= \lim_{\eta\to +\infty} \frac{(\psi_c\phi_c)^{\eta\frac{t}{T}} - 1}{(\psi_c\phi_c)^{\eta} - 1} && (53)\\
&= \lim_{\eta\to +\infty} \frac{(\psi_c\phi_c)^{\eta(\frac{t}{T}-1)} - (\psi_c\phi_c)^{-\eta}}{1 - (\psi_c\phi_c)^{-\eta}} && (54)\\
&= 0. \qquad \because \frac{t}{T} - 1 < 0 && (55)
\end{aligned}$$

□

These two derivations show that our class-wise noise schedule generalizes previous works. Although we set $\eta = 1$ for our experiments, different η can be searched and employed for controlling the noise schedules, which we leave as future work.

A.3. Proof of Proposition 2

We provide the proof of Proposition 2.

Proposition 2. *Suppose there exists a differentiable function f_i such that $q(\mathbf{y}_0^i|\mathbf{x}_t) = Cat(\mathbf{y}_0^i; p = f_i(\mathbf{x}_t))$ and $\mathbf{y}_0^1|\mathbf{x}_t, ..., \mathbf{y}_0^{H\times W}|\mathbf{x}_t$ are independent, where $i \in \{1, ..., H \times W\}$ denotes the index of a pixel in semantic map and $\mathbf{y}_0^i \in \mathbb{R}^{C+1}$ is a one-hot vector. With $q(\mathbf{y}_t^i|\mathbf{y}_0^i)$ from Eq. (13) and $\gamma_{t,c} = \gamma_t$ for any c, we have the following relationship;*

$$\mathbb{E}_{q(\mathbf{y}_t|\mathbf{y}_0)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t)] = (1 - \gamma_t)\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t).$$

Proof.

$$q(\mathbf{y}_0|\mathbf{x}_t) = q(\mathbf{y}_0^1, \mathbf{y}_0^2, ..., \mathbf{y}_0^{H\times W}|\mathbf{x}_t) \tag{56}$$

$$= \prod_{i=1}^{H\times W} q(\mathbf{y}_0^i|\mathbf{x}_t), \tag{57}$$

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log \prod_{i=1}^{H\times W} q(\mathbf{y}_0^i|\mathbf{x}_t) \tag{58}$$

$$= \sum_{i=1}^{H\times W} \nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0^i|\mathbf{x}_t). \tag{59}$$

Since $\mathbf{y}_0^i|\mathbf{x}_t$ are independent and our Label Diffusion forward process is applied to each pixel of $\mathbf{y}_0$ independently, $\mathbf{y}_t^i|\mathbf{x}_t$ are independent. Therefore,

$$q(\mathbf{y}_t|\mathbf{x}_t) = q(\mathbf{y}_t^1, \mathbf{y}_t^2, ..., \mathbf{y}_t^{H\times W}|\mathbf{x}_t) \tag{60}$$

$$= \prod_{i=1}^{H\times W} q(\mathbf{y}_t^i|\mathbf{x}_t), \tag{61}$$

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log \prod_{i=1}^{H\times W} q(\mathbf{y}_t^i|\mathbf{x}_t) \tag{62}$$

$$= \sum_{i=1}^{H\times W} \nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t^i|\mathbf{x}_t), \tag{63}$$

$$\mathbb{E}_{q(\mathbf{y}_t|\mathbf{y}_0)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t)] = \sum_{i=1}^{H\times W} \mathbb{E}_{q(\mathbf{y}_t|\mathbf{y}_0)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t^i|\mathbf{x}_t)] \tag{64}$$

$$= \sum_{i=1}^{H\times W} \mathbb{E}_{q(\mathbf{y}_t^i|\mathbf{y}_0)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t^i|\mathbf{x}_t)] \tag{65}$$

$$= \sum_{i=1}^{H\times W} \mathbb{E}_{q(\mathbf{y}_t^i|\mathbf{y}_0^i)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t^i|\mathbf{x}_t)]. \tag{66}$$

Thus, our proof can be substituted for proving the following statement:

$$\mathbb{E}_{q(\mathbf{y}_t^i|\mathbf{y}_0^i)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t^i|\mathbf{x}_t)] = (1-\gamma_t)\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0^i|\mathbf{x}_t). \tag{67}$$

With a slight abuse of notation, we use $\mathbf{y}_0$ to denote $\mathbf{y}_0^i$, $\mathbf{y}_t$ to denote $\mathbf{y}_t^i$, and $f(\mathbf{x}_t)$ to denote $f_i(\mathbf{x}_t)$ for the rest of the proof. Then, $\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t)$ can be derived as follows:

$$q(\mathbf{y}_0|\mathbf{x}_t) = \mathbf{y}_0^T f(\mathbf{x}_t), \tag{68}$$

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t) = \frac{1}{q(\mathbf{y}_0|\mathbf{x}_t)} \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \mathbf{y}_0 \tag{69}$$

$$= \frac{1}{\mathbf{y}_0^T f(\mathbf{x}_t)} \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \mathbf{y}_0. \tag{70}$$

Also, $\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t)$ can be derived as follows:

$$q(\mathbf{y}_t|\mathbf{x}_t) = \sum_{\mathbf{y}_0} q(\mathbf{y}_t, \mathbf{y}_0|\mathbf{x}_t) = \sum_{\mathbf{y}_0} q(\mathbf{y}_t|\mathbf{y}_0, \mathbf{x}_t) q(\mathbf{y}_0|\mathbf{x}_t) \tag{71}$$

$$= \sum_{\mathbf{y}_0} q(\mathbf{y}_t|\mathbf{y}_0) q(\mathbf{y}_0|\mathbf{x}_t) = \sum_{c=1}^{C+1} \mathbf{y}_t^T \overline{\boldsymbol{Q}}_t \mathbf{e}_c \mathbf{e}_c^T f(\mathbf{x}_t) \tag{72}$$

$$= \mathbf{y}_t^T \overline{\boldsymbol{Q}}_t (\sum_{c=1}^{C+1} \mathbf{e}_c \mathbf{e}_c^T) f(\mathbf{x}_t) \tag{73}$$

$$= \mathbf{y}_t^T \overline{\boldsymbol{Q}}_t f(\mathbf{x}_t), \tag{74}$$

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t) = \frac{1}{q(\mathbf{y}_t|\mathbf{x}_t)} \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \overline{\boldsymbol{Q}}_t^T \mathbf{y}_t \tag{75}$$

$$= \frac{1}{\mathbf{y}_t^T \overline{\boldsymbol{Q}}_t f(\mathbf{x}_t)} \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \overline{\boldsymbol{Q}}_t^T \mathbf{y}_t. \tag{76}$$

Therefore, $\mathbb{E}_{q(\mathbf{y}_t|\mathbf{y}_0)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t)]$ can be derived as follows:

$$\mathbb{E}_{q(\mathbf{y}_t|\mathbf{y}_0)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t)] = \sum_{\mathbf{y}_t} q(\mathbf{y}_t|\mathbf{y}_0) \frac{1}{\mathbf{y}_t^T \overline{\boldsymbol{Q}}_t f(\mathbf{x}_t)} \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \overline{\boldsymbol{Q}}_t^T \mathbf{y}_t \tag{77}$$

$$= \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \sum_{\mathbf{y}_t} q(\mathbf{y}_t|\mathbf{y}_0) \frac{1}{\mathbf{y}_t^T \overline{\boldsymbol{Q}}_t f(\mathbf{x}_t)} \overline{\boldsymbol{Q}}_t^T \mathbf{y}_t \tag{78}$$

$$= \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \sum_{\mathbf{y}_t} \mathbf{y}_t^T \overline{\boldsymbol{Q}}_t \mathbf{y}_0 \frac{1}{\mathbf{y}_t^T \overline{\boldsymbol{Q}}_t f(\mathbf{x}_t)} \overline{\boldsymbol{Q}}_t^T \mathbf{y}_t \tag{79}$$

$$= \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \sum_{\mathbf{y}_t} \overline{\boldsymbol{Q}}_t^T \frac{1}{\mathbf{y}_t^T \overline{\boldsymbol{Q}}_t f(\mathbf{x}_t)} \mathbf{y}_t \mathbf{y}_t^T \overline{\boldsymbol{Q}}_t \mathbf{y}_0 \tag{80}$$

$$= \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \overline{\boldsymbol{Q}}_t^T \left(\sum_{\mathbf{y}_t} \frac{1}{\mathbf{y}_t^T \overline{\boldsymbol{Q}}_t f(\mathbf{x}_t)} \mathbf{y}_t \mathbf{y}_t^T \right) \overline{\boldsymbol{Q}}_t \mathbf{y}_0 \tag{81}$$

$$= \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \overline{\boldsymbol{Q}}_t^T \left(\sum_{c=1}^{C+1} \frac{1}{\mathbf{e}_c^T \overline{\boldsymbol{Q}}_t f(\mathbf{x}_t)} \mathbf{e}_c \mathbf{e}_c^T \right) \overline{\boldsymbol{Q}}_t \mathbf{y}_0 \tag{82}$$

$$= \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \overline{\boldsymbol{Q}}_t^T \ \mathbf{D} \ \overline{\boldsymbol{Q}}_t \mathbf{y}_0, \tag{83}$$

where $\mathbf{D}$ is a diagonal matrix. Since the class corresponding to the absorbing state, *i.e.*, class $C+1$ does not exist in the original dataset, we have $q(\mathbf{y}_0 = \mathbf{e}_{C+1}) = 0 \Leftrightarrow [f(\mathbf{x}_t)]_{C+1} = 0$. Therefore, $[\mathbf{D}]_{cc} = ((1-\gamma_t)[f(\mathbf{x}_t)]_c)^{-1}$ when $c \neq C+1$, otherwise, γ_t^{-1}. As $\mathbf{y}_0$ is a one-hot vector,

$$\mathbb{E}_{q(\mathbf{y}_t|\mathbf{y}_0)}[\nabla_{\mathbf{x}_t} \log q(\mathbf{y}_t|\mathbf{x}_t)] = \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \left((1-\gamma_t) \frac{1}{\mathbf{y}_0^T f(\mathbf{x}_t)} \mathbf{y}_0 + \gamma_t (\mathbf{1} + \frac{1-\gamma_t}{\gamma_t} \mathbf{e}_{C+1}) \right) \tag{84}$$

$$= (1-\gamma_t) \frac{1}{\mathbf{y}_0^T f(\mathbf{x}_t)} \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} \mathbf{y}_0 + \gamma_t \frac{\partial f(\mathbf{x}_t)}{\partial \mathbf{x}_t} (\mathbf{1} + \frac{1-\gamma_t}{\gamma_t} \mathbf{e}_{C+1}) \tag{85}$$

$$= (1-\gamma_t) \nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t) + \gamma_t \nabla_{\mathbf{x}_t} (\mathbf{1} + \frac{1-\gamma_t}{\gamma_t} \mathbf{e}_{C+1})^T f(\mathbf{x}_t) \tag{86}$$

$$= (1-\gamma_t) \nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t) + \gamma_t \nabla_{\mathbf{x}_t} 1 \tag{87}$$

$$= (1-\gamma_t) \nabla_{\mathbf{x}_t} \log q(\mathbf{y}_0|\mathbf{x}_t). \tag{88}$$

□

B. Algorithms

Algorithm 1 and 2 summarize the general training and sampling process of our SCDM, respectively. While $\mathbf{m}$ has to be $\mathbf{e}_{C+1} \in \mathbb{R}^{C+1}$, we implemented the absorbing vector with a zero vector $\mathbf{0} \in \mathbb{R}^{C}$ to minimize the modification of the pretrained SIS model.

Algorithm 1 Training

Require: image $\mathbf{x}_0 \in \mathbb{R}^{H \times W \times 3}$, label $\mathbf{y}_0 \in \mathbb{R}^{H \times W \times C}$, noise schedule $\alpha_{1:T}, \gamma_{1:T}$, absorbing vector $\mathbf{m} \in \mathbb{R}^{C}$

while not converged **do**
- $\epsilon_{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ $\{// \ \epsilon_{\mathbf{x}} \in \mathbb{R}^{H \times W \times 3}\}$
- $\forall_{i,j} [\epsilon_{\mathbf{y}}]_{ij} \sim \text{Uniform}(0, 1)$ $\{// \ \epsilon_{\mathbf{y}} \in \mathbb{R}^{H \times W}\}$
- $t \sim \text{Uniform}(\{1, ..., T\})$
- $\mathbf{x}_t \leftarrow \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon_{\mathbf{x}}$
- $\forall_{i,j} [\mathbf{y}_t]_{ij} \leftarrow [\mathbf{y}_0]_{ij}$ if $[\epsilon_{\mathbf{y}}]_{ij} \geq \gamma_t$ else $\mathbf{m}$
- Take a gradient descent step on $\mathcal{L}_{\text{hybrid}}\left(\epsilon_{\mathbf{x}}, \epsilon_\theta(\mathbf{x}_t, \mathbf{y}_t, t), \Sigma_\theta(\mathbf{x}_t, \mathbf{y}_t, t)\right)$

end while

Algorithm 2 Sampling

Require: label $\mathbf{y}_0 \in \mathbb{R}^{H \times W \times C}$, noise schedule $\alpha_{1:T}, \gamma_{1:T}$, guidance scale s, absorbing vector $\mathbf{m} \in \mathbb{R}^{C}$, extrapolation scale w

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

for $t \leftarrow T, ..., 1$ **do**
- $\forall_{i,j} [\epsilon_{\mathbf{y}}]_{ij} \sim \text{Uniform}(0, 1)$ $\{// \ \epsilon_{\mathbf{y}} \in \mathbb{R}^{H \times W}\}$
- $\forall_{i,j} [\mathbf{y}_t]_{ij} \leftarrow [\mathbf{y}_0]_{ij}$ if $[\epsilon_{\mathbf{y}}]_{ij} \geq \gamma_t$ else $\mathbf{m}$
- $\epsilon_{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t \neq 1$ else $\mathbf{0}$ $\{// \ \epsilon_{\mathbf{x}} \in \mathbb{R}^{H \times W \times 3}\}$
- $\tilde{\epsilon}_\theta \leftarrow \epsilon_\theta(\mathbf{x}_t, \mathbf{y}_t, t) + s(\epsilon_\theta(\mathbf{x}_t, \mathbf{y}_t, t) - \epsilon_\theta(\mathbf{x}_t, \mathbf{0}, t))$ {// Classifier-free guidance}
- $\mathbf{x}_0^{(t)} \leftarrow \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\tilde{\epsilon}_\theta}{\sqrt{\alpha_t}}$ {// Reparameterize ϵ-pred model to predict $\mathbf{x}_0$}
- $\mathbf{x}_0^{(t)} \leftarrow \text{dynamic_thresholding}(\mathbf{x}_0^{(t)})$ {// Dynamic thresholding}
- $\tilde{\mathbf{x}}_0^{(t)} \leftarrow \mathbf{x}_0^{(t)} + w\left(\mathbf{x}_0^{(t)} - \tilde{\mathbf{x}}_0^{(t+1)}\right)$ if $t \neq T$ else $\mathbf{x}_0^{(t)}$ {// Extrapolation}
- $\mu_t = \frac{1}{\sqrt{1-\alpha_t}}\left(\sqrt{\alpha_{t-1}}(1 - \frac{\alpha_t}{\alpha_{t-1}})\tilde{\mathbf{x}}_0^{(t)} + \sqrt{\frac{\alpha_t}{\alpha_{t-1}}}(1 - \alpha_{t-1})\mathbf{x}_t\right)$
- $\mathbf{x}_{t-1} \leftarrow \mu_t + \Sigma_\theta(\mathbf{x}_t, \mathbf{y}_t, t)^{\frac{1}{2}}\epsilon_{\mathbf{x}}$ {// Sample $\mathbf{x}_{t-1}$}

end for

Return $\mathbf{x}_0$

C. Experimental Setup

C.1. Metrics

C.1.1. Experiments in Section 5 and 6.1.1

FID (fidelity). To quantitatively measure generation quality, we adopt Fréchet Inception Distance (FID) (Heusel et al., 2017) as our evaluation metrics. FID captures the image's visual quality by comparing the distribution between real and generated images on the inception network's (Szegedy et al., 2017) feature space.

mIoU (semantic correspondence). Additionally, following previous works, we assess the alignment of the synthesis results with ground truth semantic maps by using off-the-shelf pretrained segmentation networks and report mean Intersection-over-Union (mIoU). We feed the sampled images to the pre-trained, off-the-shelf semantic segmentation networks: U-Net (Lee et al., 2020; Ronneberger et al., 2015) for CelebAMask-HQ, ViT-Adapter-S (Chen et al., 2022) with UperNet (Xiao et al., 2018) for ADE20K, and DeepLabV2 (Chen et al., 2015) for COCO-Stuff. For a fair comparison, we tried to measure and reproduce the mIoU of all baseline samples. For the standard SIS setting, when samples or checkpoints were not publicly

available, we report the results from (Tang et al., 2023a) and denote them with '*' in the table.

LPIPS (diversity). In addition to FID and mIoU, we adopt Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) as our generation diversity metrics in Section G.1 and compare the average distance between multi-modal synthesis results. Specifically, following (Tan et al., 2021a), we sample 10 different images for each semantic map, compute the pairwise LPIPS distance, and average over the label maps.

C.1.2. Experiments in Section 6.1.2

LPIPS. In Section 6.1.2, we adopt LPIPS to compare similarity, as LPIPS essentially measures the similarity of two different images. Therefore, lower LPIPS is better in our experimental setting. We calculate LPIPS by comparing images generated with clean and noisy labels, and averaging the results.

PSNR. Peak Signal-to-Noise Ratio (PSNR) is a metric that evaluates the reconstruction quality, *i.e.*, higher PSNR is better. We calculate PSNR by comparing images generated with clean and noisy labels, and averaging the results.

SSIM. Structural Similarity Index Measure (SSIM) (Wang et al., 2004) is a metric to evaluate the similarity of a pair of images, *i.e.*, higher SSIM is better. We calculate SSIM by comparing images generated with clean and noisy labels, and averaging the results.

FID. In Section 6.1.2, we use FID to compare the distribution between the set of generated images conditioned on clean labels and the set of generated images conditioned on noisy labels. In our experimental setting, lower FID indicates that the two distributions are closer, which is better.

C.2. Baselines

We compare our model with GAN-based methods such as pix2pixHD (Wang et al., 2018), SPADE (Park et al., 2019), DAGAN (Tang et al., 2020a), SCGAN (Wang et al., 2021), CLADE (Tan et al., 2021b), CC-FPSE (Liu et al., 2019), GroupDNet (Zhu et al., 2020b), INADE (Tan et al., 2021a), OASIS (Sushko et al., 2020), RESAIL (Shi et al., 2022), SAFM (Lv et al., 2022), ECGAN (Tang et al., 2023a), ECGAN++ (Tang et al., 2023b), and Unconditional Generator with Semantic Mapper (Chae et al., 2024). We also evaluate our method with diffusion-based approaches, SDM (Wang et al., 2022b), LDM (Rombach et al., 2022), and PITI (Wang et al., 2022a). Although FLIS (Xue et al., 2023) and ControlNet (Zhang et al., 2023) do tackle SIS with diffusion models, they utilize extra text inputs (captions) with the semantic maps. For a fair comparison, we did not include FLIS and ControlNet as baselines.

As the pretrained weights of pix2pixHD and PITI on ADE20K are not publicly available, we excluded them from our noisy SIS experiments on ADE20K. Additionally, the pretrained weights and the full code of RESAIL, ECGAN, and Unconditional Generator are not available in public, therefore we did not include them in our noisy SIS experiments baselines as well.

For the standard SIS setting, we report all the results evaluated with the uploaded samples or checkpoints except when they are unavailable. For PITI, as the COCO-Stuff pretrained weights were only available, we sampled images with the weights and reported the LPIPS and mIoU scores. For CelebAMask-HQ results of SDM, we report the reproduced results. Although LDM did not include SIS experiments on benchmark datasets in their official paper, we trained their model with their SIS training configurations and reported the results in both noisy and standard SIS experiments, for more comparison with DM-based approaches.

C.3. Implementation Details

We followed the overall architecture of SDM (Wang et al., 2022b), a diffusion model for semantic image synthesis. Specifically, we embed the condition into the U-Net decoder with SPADE (Park et al., 2019) and constructed a similar U-Net structure, where the architecture details are publicly available in their GitHub repository. SDM trained their model with two stages: (1) pre-trained their conditional diffusion model and (2) fine-tuned the pre-trained model by randomly dropping out the semantic label maps in order to use classifier-free guidance in sampling. We trained our SCDM starting from their pre-trained weights to reduce the training time. We trained our model with 4 NVIDIA RTX A6000 GPUs for 1-2 days. Image sampling and evaluations are conducted on a server with 8 NVIDIA RTX 3090 GPUs.

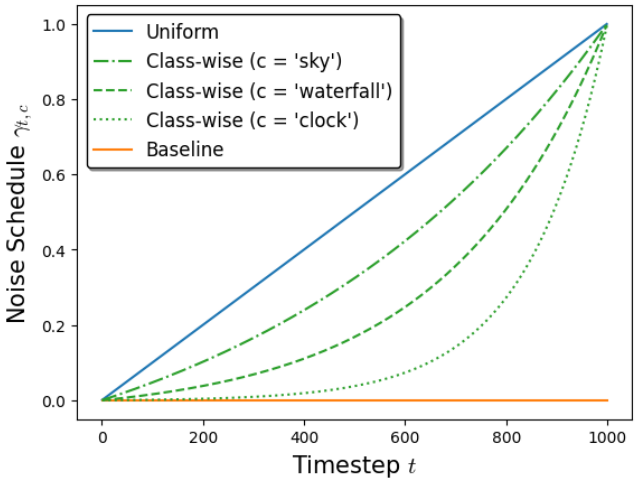


Figure 5. **Visualization of $\gamma_{t,c}$ throughout diffusion in the baseline, linear and uniform, and class-wise noise schedule.** $\gamma_{t,c}$ indicates the probability of a label c has transitioned to the absorbing state until timestep t. In our class-wise schedule, small and rare objects tend to be intact relatively longer in the diffusion process, *e.g.*, clock.

For the hyperparameters, we used $\lambda = 0.001$ for our hybrid loss (Nichol & Dhariwal, 2021), classifier-free guidance (Ho & Salimans, 2021) scale $s = 0.5$, 20% of drop rate for the SIS experiments on three datasets, noise schedule hyperparameter $\eta = 1$, dynamic thresholding (Saharia et al., 2022) percentile of 0.95, and the extrapolation scale of $w = 0.8$. Except for the ablation study (in Table 3), we didn't use extrapolation for experiments with 1000 sampling steps, as extrapolation is designed to enhance small-step generation. We also utilized exponential moving average (EMA) with 0.9999 decay, and AdamW (Loshchilov & Hutter, 2019) optimizer. We employed instance labels of CelebAMask-HQ and COCO-Stuff to produce instance edge maps and used them as additional input following SDM.

Furthermore, for the class-wise noise schedule, we clamped ϕ_c to be at least 1, to down the diffusion of rare classes without speeding up the transition of frequent classes. We set the class-wise noise schedules of the 'unlabeled' class of ADE20K and COCO-Stuff to be the uniform schedule, *i.e.*, $\gamma_{t,c} = t/T$ for $c =$'unlabeled'. In addition, we implemented the calculation of $\gamma_{t,c}$ using $t-1$ instead of t in the code to ensure that the assumption of Lemma 1 is satisfied.

For our experiments on ADE20K, we modified the calculation of $\psi_c\phi_c$ values of the class-wise noise schedule with an empirically chosen scale factor. This is because ADE20K has numerous classes and high diversity in the dataset compared to the number of training data, while CelebAMask-HQ has relatively low diversity and COCO-Stuff has a large number of training data. Specifically, we modified $\psi_c\phi_c$ as:

$$\psi_c\phi_c := \lambda\, \mathbf{Pr}(\mathbf{y}_{ij} = c)^{-1} \log\left(\mathbf{Pr}(\mathbf{x} \in \mathcal{X}_c)^{-1}\right), \tag{89}$$

where λ is the scale factor (≈ 0.278) making the smallest value of $\psi_c\phi_c$ converge to 1, *i.e.*, $\gamma_{t,c} = t/T$.

C.4. Efficient Trajectory Representation

The generation process of SCDM utilizes $\mathbf{y}_T, ..., \mathbf{y}_1$ at each timestep $t = T, ..., 1$. As we have not defined the reverse process of the labels, we use the label forward process $q(\mathbf{y}_{1:T}|\mathbf{y}_0)$, which sequentially constructs $\mathbf{y}_1, ..., \mathbf{y}_T$ from $\mathbf{y}_0$. Consequently, $\mathbf{y}_{1:T}$ needs to be *cached* during sampling in order to be accessed from $t = T$ to $t = 1$. However, this results in a significant memory consumption, posing a challenge to our sampling process. To address this issue, we leverage $\gamma_{t,c}$ from Eq (13). Specifically, we represent the entire trajectory $\mathbf{y}_{1:T}$ with a single matrix $\mathbf{U} \in \mathbb{R}^{H\times W}$, where $\mathbf{U}_{ij}$ denotes the 'timestep' that (i, j)-th pixel is masked. Suppose that (i, j)-th pixel corresponds to class c. Since $\gamma_{t,c}$ can be expressed as a strictly monotonic function $\gamma_c(t)$, we can easily sample $\mathbf{U}_{ij}$ using the inverse CDF method on $\gamma_c^{-1}(u)$, where $u \sim \text{Uniform}(0, 1)$.

D. Detailed Explanations on ψ_c and ϕ_c

In this section, we elaborate on ψ_c and ϕ_c as introduced in Section 4.2 and provide some visual aid for $\gamma_{t,c}$. ψ_c accounts for the area ($\approx$ object size) of class c. Eq. (15) defines ψ_c, which we estimated with training data. ψ_c ensures that classes with smaller average areas transition to the masked state later. Specifically, the term is defined as the inverse of the ratio of average total image pixels to the average class pixels. As the average area of class c decreases, the ψ_c value increases. On the other hand, ϕ_c ensures a slow diffusion (to the masked state) of classes that rarely appear in the dataset. It is defined in Eq. (16), and it can be regarded as the inverse frequency of the class in training images. A larger ϕ_c value corresponds to a class that rarely appears in the dataset. In Figure 5, we visualize $\gamma_{t,c}$ with different noise schedules. The class-wise schedule of class 'clock' ensures that the class is diffused at a relatively later time, as its $\psi_c\phi_c$ value is large.

E. Limitations

One possible limitation of this work is that as the method aims to generate clean images from the noisy labels, the user's intention might be ignored, *i.e.*, the user might actually want to generate noisy images. Controlling faithfulness to the semantic maps could be one possible future direction, which we leave as future work. Another limitation could be that our method cannot dynamically learn the optimal noise schedules for labels. If the proposed method can learn the optimal noise schedule, it will improve the flexibility and power of semantic image synthesis.

F. ADE20K Dataset Annotation Examples

In this section, we provide examples of erroneous/inconsistent annotations.

Jagged edges (DS). We observe that some images have jagged edges as label maps given in Figure 12. This could be the result of erroneous annotations but also could occur due to the coarse and low resolution of the images. ADE20K dataset contains images of various sizes. Similarly, the size of user inputs can be diverse in real-world applications.

Incomplete masks (Edge). Figure 13 shows some examples of incomplete masks, where the pixels on the boundaries of instances were left as 'unlabeled'. This usually occurs due to their inherent ambiguity.

Inconsistent annotation (*e.g.*, umbrella). Similar to Figure 9, Figure 11 shows training images (first row) and their annotations (second row). Some umbrellas are correctly annotated with the shafts (Figure 11(a)), while others are annotated without the shafts and handles (Figure 11(b)). Trained with these inconsistent labels, baselines did not synthesize the shaft for the parasol (class 'umbrella') in Figure 9, whereas our SCDM successfully generated the shafts leveraging our Label Diffusion.

G. Additional Experimental Results

G.1. Standard SIS Setting

G.1.1. Generation quality comparison.

Figure 6 presents the qualitative comparison with other SIS models and our method on CelebAMask-HQ. Remarkably, our method generates realistic images even for an intricate semantic map with fluttering hair over the face, while others struggle with generating the occluded eye naturally.

As Figure 7 illustrates the qualitative results on ADE20K, our method demonstrates its advantage over previous methods in generating details, and convincing images within the given semantics. Especially, our SCDM generates a more realistic and clear image of the waterfall than the baselines. In addition, our approach more naturally depicts the rocks visible through the water, compared to the baselines. For ADE20K, we used slightly modified $\psi_c\phi_c$ with some scale factors, and the details are in Appendix C.3.

Lastly, on COCO-Stuff our method outperforms the state-of-the-art methods in FID and LPIPS. The qualitative comparison with others and ours is given in Figure 8. Ours synthesized a more realistic and clearer image of a clock tower, by successfully generating the numbers and hands of the clock. Furthermore, ours generated a more natural image of the tree branches over the clock tower.

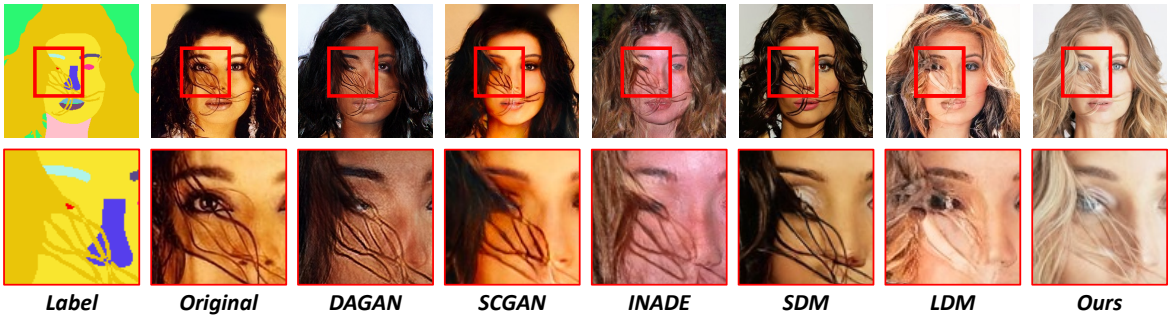


Figure 6. **Qualitative results on CelebAMask-HQ.** Ours successfully catches the fine-grained details and generates realistic images conditioning on challenging semantic labels with highly occluded objects (*e.g.*, eye covered by hairs).

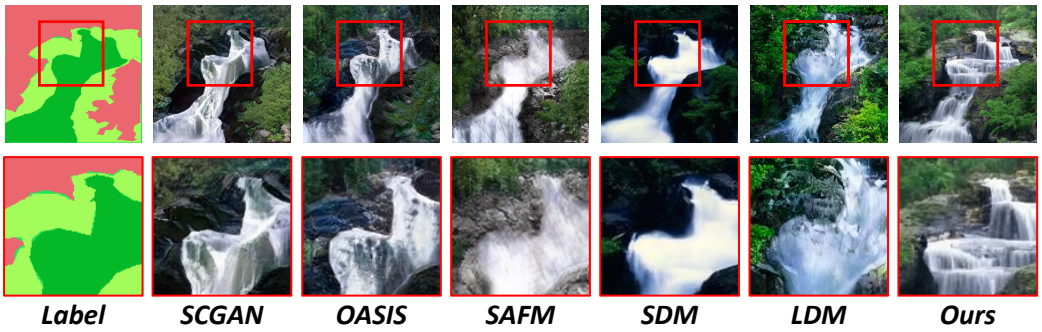


Figure 7. **Qualitative results on ADE20K.** Ours is capable of synthesizing realistic and clear images, as our generated images show more depth and clearer results of the waterfall.

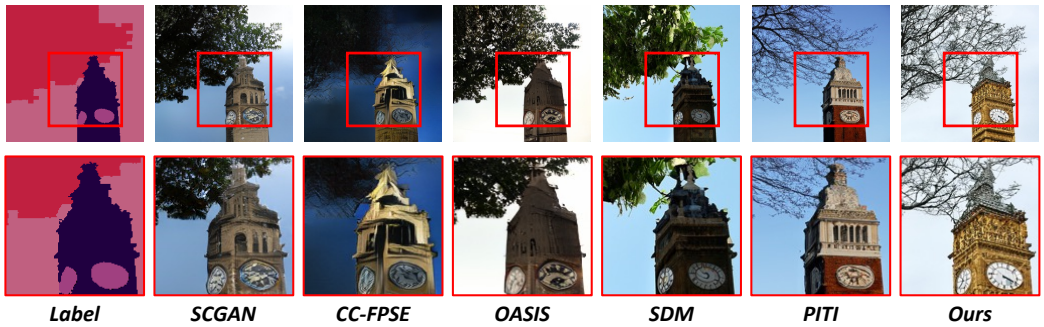


Figure 8. **Qualitative results on COCO.** Our SCDM clearly captures the details (*e.g.*, the hands on the clock) and naturally depicts the branches over the clock tower.

G.1.2. Analysis on semantic correspondence.

We evaluate the semantic correspondence using pre-trained segmentation networks and compare our method with baselines on mIoU, and the results are in Table 2. SCDM shows better performance than the state-of-the-art on CelebAMask-HQ, while for the other two datasets, diffusion-based approaches including ours show comparatively weaker performance than adversarial methods such as SAFM or ECGAN. We believe that one of the reasons for this could be the erroneous annotation of images. For example, 'umbrella' is often annotated without a shaft and a handle (examples are given in Appendix F). In Figure 9, baselines trained with these images did not synthesize the shaft for the given semantic map of a parasol (class 'umbrella') without a shaft, while ours did. Although ours is more realistic, this can result in lower mIoU scores. We also suspect that the performance of the off-the-shelf segmentation models used for mIoU evaluation is not very robust. We ran the segmentation model on ground-truth (*i.e.*, real and clean) images and it shows a poor mIoU with semantic labels (43.1

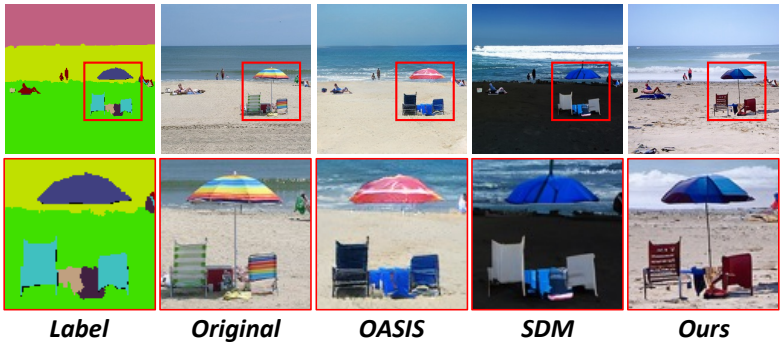


Figure 9. **Robustness to inconsistent semantic labels.** Our SCDM exhibits robustness to inconsistent semantic labels. Although the parasol (class 'umbrella') in the original image has a shaft, it is not accurately annotated in the semantic map. Interestingly, ours synthesized a natural image with the missing shaft whereas all baselines failed to generate it.

Table 6. **Extrapolation hyperparameter search on *ADE20K*.** The samples are generated with 25 sampling steps.

w	**FID (↓)**
0.4	33.0
0.5	31.6
0.6	30.3
0.7	29.2
0.8	**27.7**
0.9	30.9

on ADE20K and 34.7 on COCO). Additionally, it is noteworthy that baselines with high mIoU scores such as RESAIL, SAFM, and ECGAN leverage 'pretrained' segmentation models in their semantic alignment losses (denoted by 'Seg' in Table 2). In contrast, ours does not rely on such explicit guidance for correspondence.

G.2. Extrapolation Hyperparameter Search

For the extrapolation hyperparameter w, we searched for values in [0.4, 0.5, 0.6, 0.7, 0.8, 0.9], following a similar protocol of (Ho & Salimans, 2021). The search results are in Table 6, and we chose the value $w = 0.8$ in terms of the generation quality (FID).

G.3. Visualization of Label Diffusion

We show the intermediate steps of our Label Diffusion as a visual aid for understanding the two different noise schedules. Figure 15(a) and 15(b) illustrate the intermediate results of the diffused labels in the Label Diffusion process using the linear and uniform noise schedule and the class-wise schedule, respectively. The first row of each subfigure shows the diffused semantic map given to the model during the discrete diffusion process, while the second row magnifies and displays the label of a clock on a cabinet from the first row. The $\psi_c \phi_c$ values of the 'clock' and 'cabinet' classes are 651.3 and 17.3, respectively. As the class 'clock' occupies relatively small areas in the dataset, the class-wise schedule makes the clock diffused slowly, and denoised fastly during the generation process. Meanwhile, in the uniform noise schedule, the 'clock' class is perturbed faster than in the class-wise schedule (Figure 15(b)), because the uniform schedule evenly diffuses all class labels in the label map.

Table 7. **Quantitative noise schedule comparison on semantic correspondence (mIoU (↑)).** The results are sampled over 1000 sampling steps.

Noise Schedule	CelebAMask-HQ	ADE20K	COCO-Stuff
Linear & Uniform	76.8	43.0	36.8
Class-wise	**77.2**	**49.4**	**38.1**

Table 8. **Sampling results with different noise schedules for generation process on *ADE20K*.** The results are sampled from (a) SCDM trained with a class-wise noise schedule ($\eta = 1$), and (b) baseline (trained without Label Diffusion, *i.e.*, $\eta = +\infty$).

Model	Sampling noise schedule	η	FID	mIoU
(a) Ours	No Label Diffusion	$+\infty$	30.6	47.8
	Linear and uniform	0	34.5	28.9
	Class-wise	1	**27.7**	**48.7**
(b) Base	No Label Diffusion	$+\infty$	**28.1**	**48.6**
	Linear and uniform	0	76.2	15.7
	Class-wise	1	44.0	43.1

G.4. Effect of Class-wise Noise Schedule

In this section, we show the quantitative comparison of the class-wise schedule and the linear and uniform schedule, continuing the discussion from Section 6.2 of the main paper. The results are given in Table 7, on three (original) benchmark datasets. The results are sampled with 1000 sampling steps, without extrapolation. As demonstrated by the mIoU gains, the class-wise noise schedule increases the semantic correspondence of the generated images. Additionally, we provide the qualitative comparison on COCO-Stuff in Figure 14.

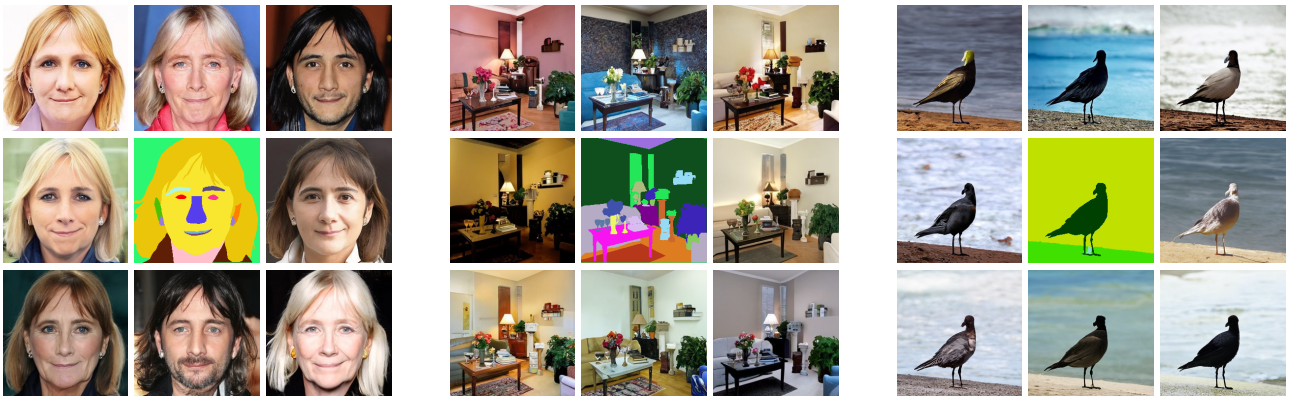
G.5. Further Discussion and Analysis on Class Guidance

In Section 4.4, we analyzed Label Diffusion regarding the conditional score that our method approximates. Although our method does not have the same guidance as the baseline with fixed labels and time-dependent scaling, we further assess the effect of scaling and compare it with our method. Specifically, we modify the fixed classifier-free guidance scale through scheduling, *i.e.*, s to $s(1 - \gamma_t)$ in Eq. (20), resulting in comparable fidelity with the baseline using fixed guidance (FID (↓) of 28.6 (CFG scale scheduling) and 28.1 (baseline)). Nevertheless, it still yields suboptimal results compared to ours (FID of 26.9). What further distinguishes our method from CFG scale scheduling is that our absorbing state explicitly informs the model that the pixel is unconditional, facilitating the natural generation of the ambiguous parts in an image. Some concrete examples of the ambiguous parts are given in Figure 6 and 8, where the eye is occluded by hairs and the tree branches are over the clock tower.

G.6. Validation of SCDM Generation Process

In this section, we validate the SCDM generation process by showing the effect of our Label Diffusion during sampling. Although our forward process diffuses images and labels independently (Eq. (9) of the main paper), our model successfully learns the joint distribution of the images and labels, and therefore generates the images dependent on the given perturbed labels.

To show this empirically, we first trained our model with the class-wise noise schedule (*i.e.*, $\eta = 1$). Then we sampled images with three different Label Diffusion noise schedules by controlling η: (1) $\eta = +\infty$, *i.e.*, no Label Diffusion and using fixed $\mathbf{y}_0$ for the entire generation process, (2) $\eta = 0$, *i.e.*, the linear and uniform noise schedule, and (3) $\eta = 1$, *i.e.*, the class-wise noise schedule that was leveraged during training. The results are reported in Table 8(a) and we used 25 sampling steps. As the FID scores show that the quality of the samples not generated with the trained noise schedule is worse than the



(a) *CelebAMask-HQ*. (b) *ADE20K*. (c) *COCO-Stuff*.

Figure 10. **Multimodal generation results.**

images sampled with $\eta = 1$, we can observe that Label Diffusion affects the generation process.

Additionally, we sampled images using the baseline model with the same noise schedules, *i.e.*, $\eta = +\infty$, $\eta = 0$, and $\eta = 1$. The results are reported in Table 8(b) and we used 1000 sampling steps (without extrapolation). The performance deteriorates when Label Diffusion is applied to the baseline (*i.e.*, trained without Label Diffusion). This also indicates that Label Diffusion contributes to the generation process, validating the SCDM generation process.

G.7. More Qualitative Results - Multimodal Generation

We provide additional generation results of our method, showing SCDM's capability of generating diverse samples. Figure 10 shows multimodal generation results on CelebAMask-HQ, ADE20K, and COCO-Stuff (standard SIS setting).

G.8. More Qualitative Results - SIS with Noisy Labels

We provide additional generation results of our method and baselines on SIS with noisy labels. The results of **[DS]**, **[Edge]**, and **[Random]** are illustrated in Figure 16, 17, and 18, respectively, and the experiments are on the ADE20K dataset.

G.9. More Qualitative Results - Standard SIS Setting

We present additional generation results of our SCDM and other baselines on the standard SIS setting. Figure 19 shows the results of our method and qualitative comparisons on CelebAMask-HQ. Figure 20 and 21 show the results and qualitative comparisons on ADE20K and COCO-Stuff, respectively.

(a) Training data examples with shaft annotated.

(b) Training data examples with shaft not annotated.

Figure 11. **Examples of inconsistent semantic labels (class 'umbrella') in the training set.**

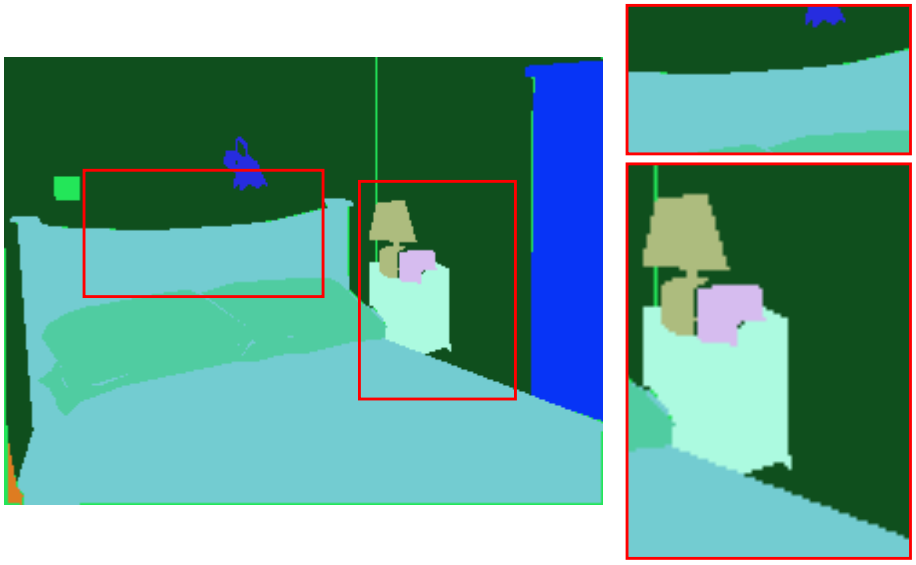


Figure 12. **Examples of jagged edges in the ADE20K dataset.**

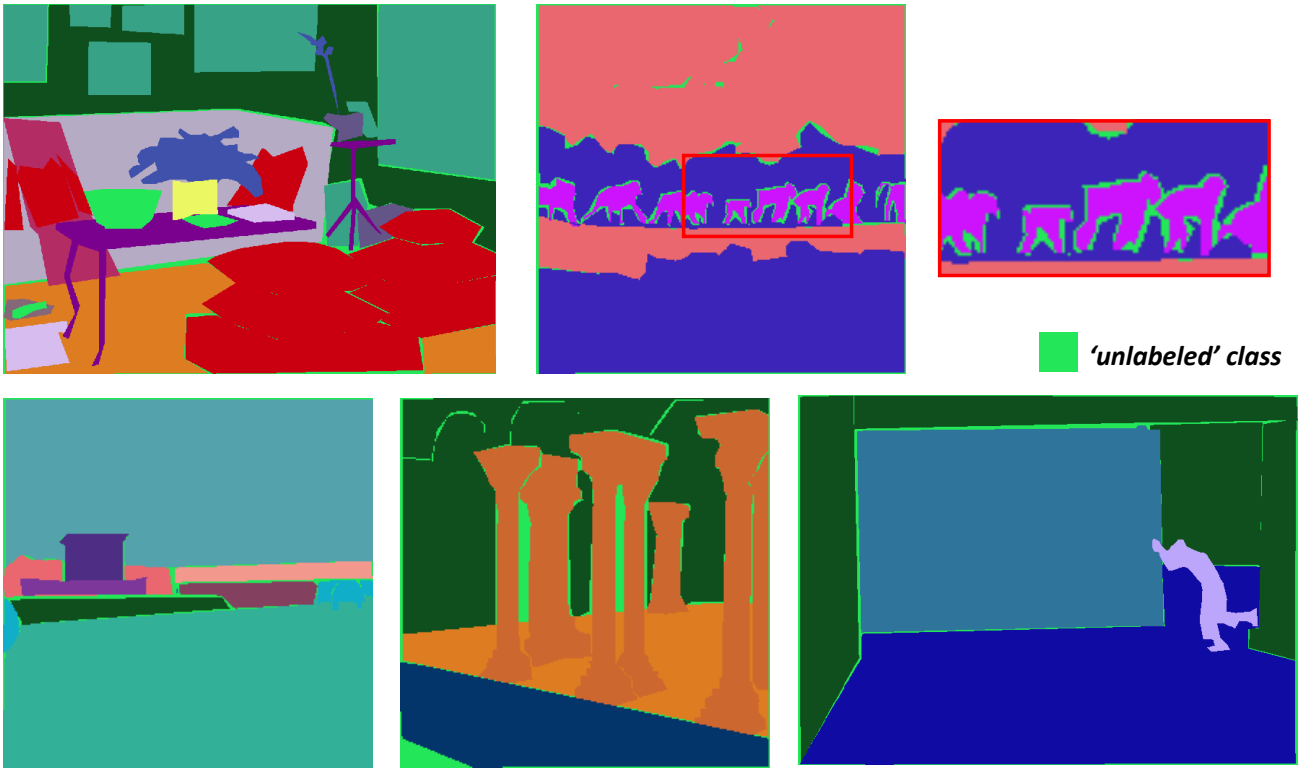


Figure 13. **Examples of incomplete masks in the *ADE20K* dataset.**

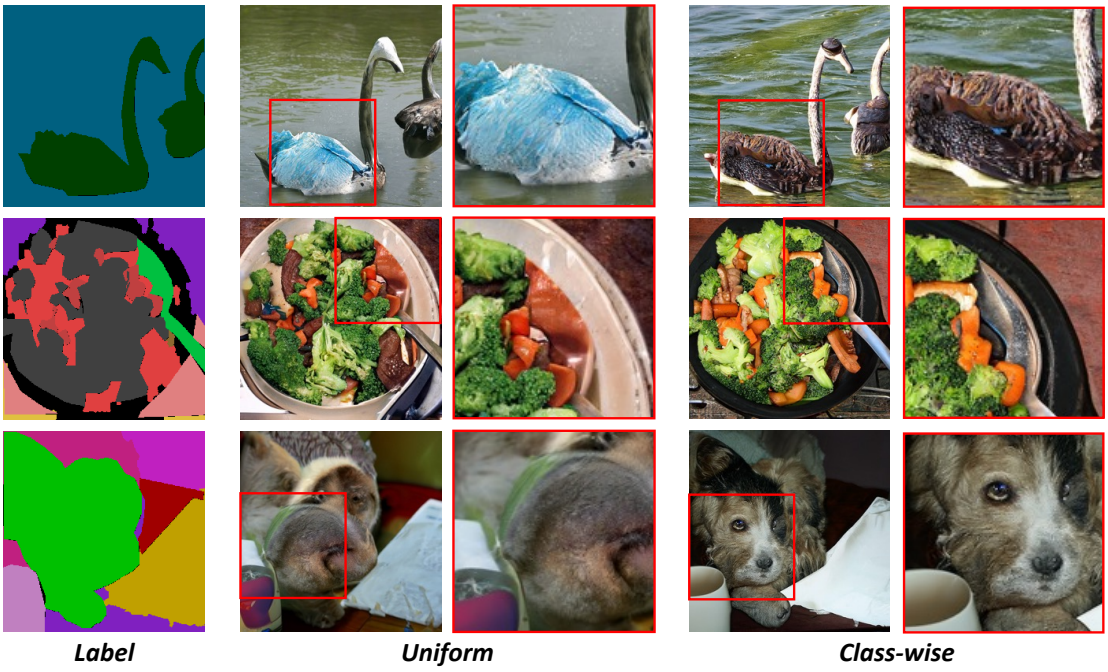
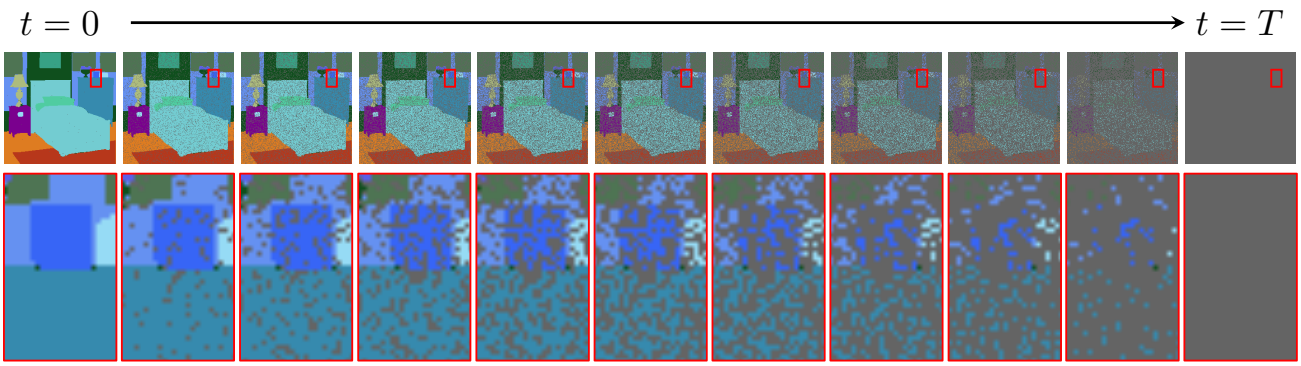
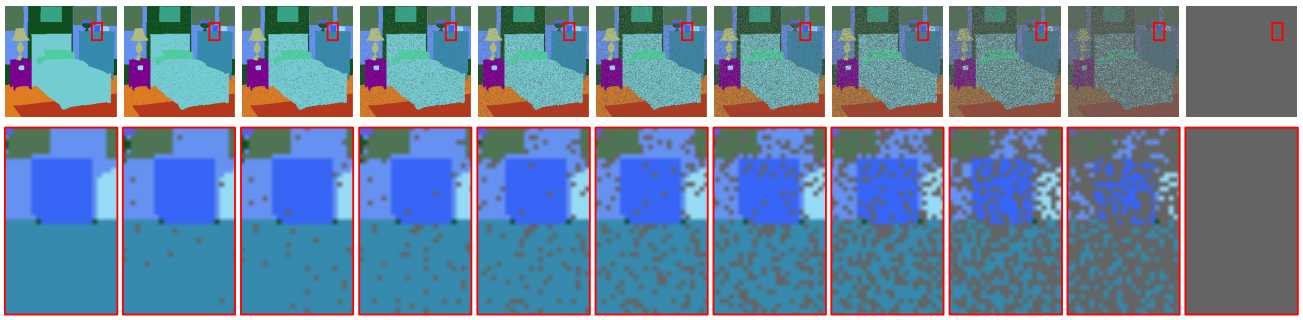


Figure 14. **Qualitative noise schedule comparison on *COCO-Stuff*.**



(a) Uniform and linear noise schedule.



(b) Class-wise noise schedule.

Figure 15. **Visualization of Label Diffusion intermediate steps.**

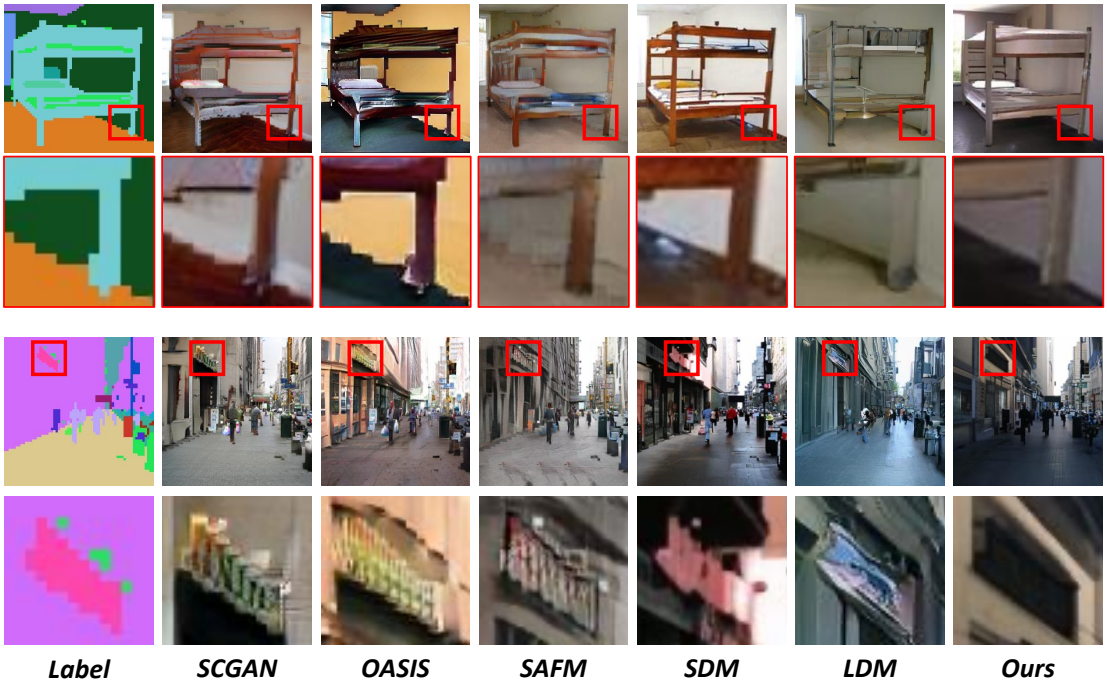


Figure 16. **More qualitative comparisons on SIS with noisy labels (DS).** The generation results are conditioned with semantic masks with jagged edges.

Figure 17. **More qualitative comparisons on SIS with noisy labels (Edge).** The generation results are conditioned with incomplete masks on the edges of instances.

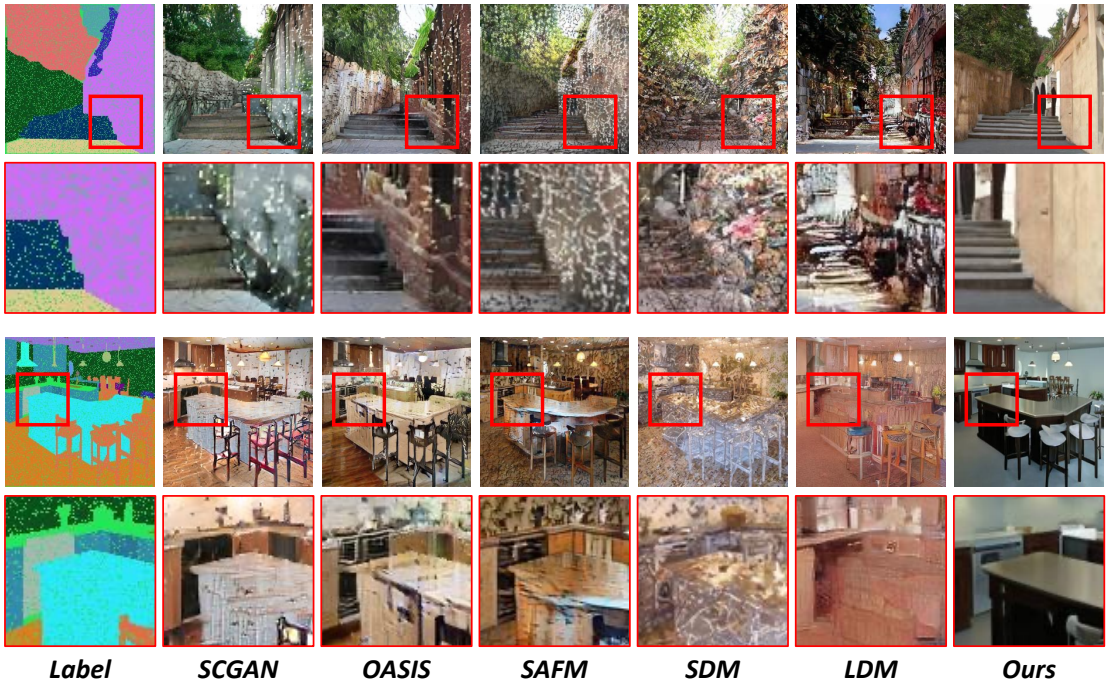


Figure 18. **More qualitative comparisons on SIS with noisy labels (Random).** The generation results are conditioned with corrupted masks.

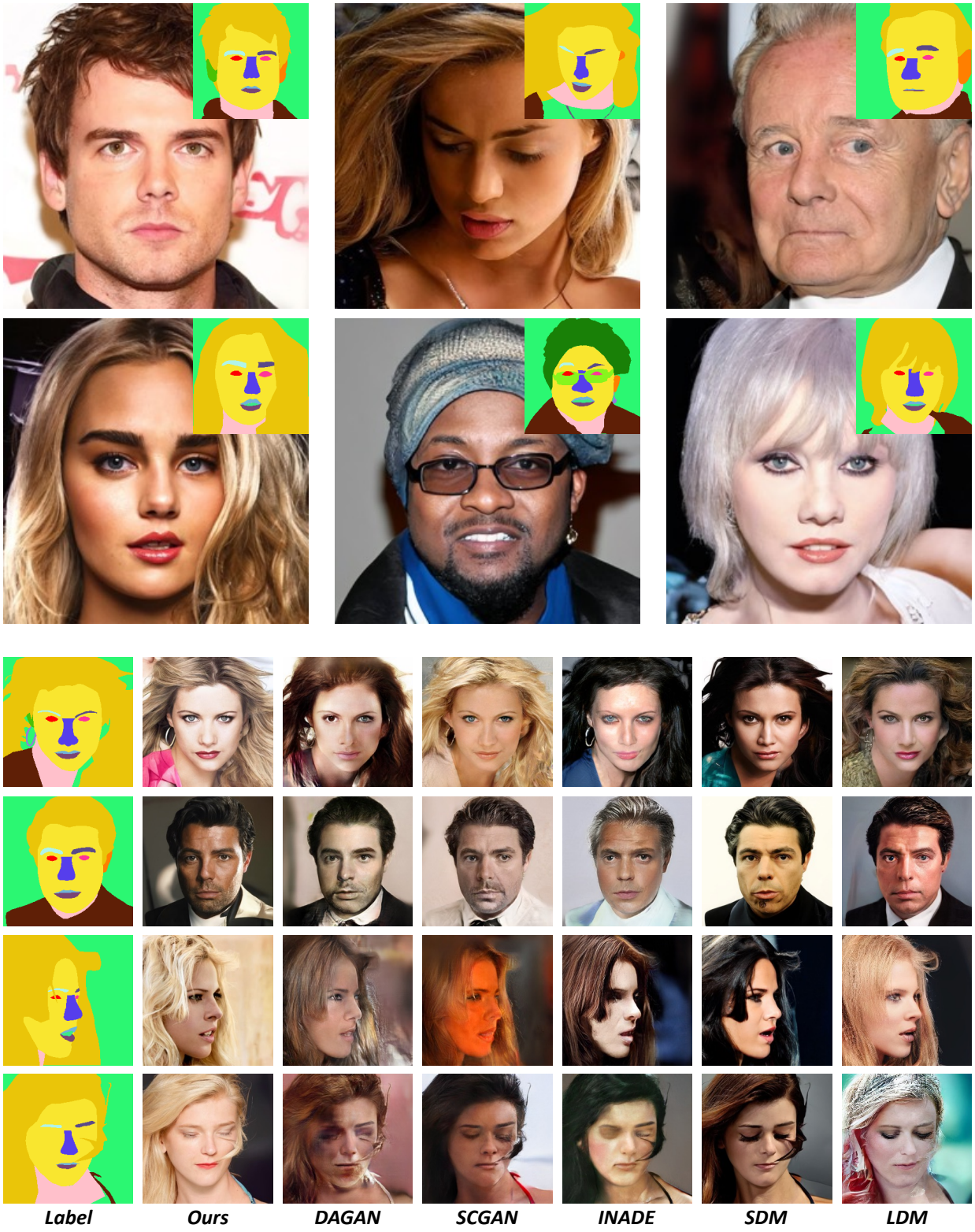


Figure 19. **More qualitative results and comparisons on *CelebAMask-HQ*.** The first two rows are the result of our model, while the other rows depict qualitative comparisons in terms of generation quality.

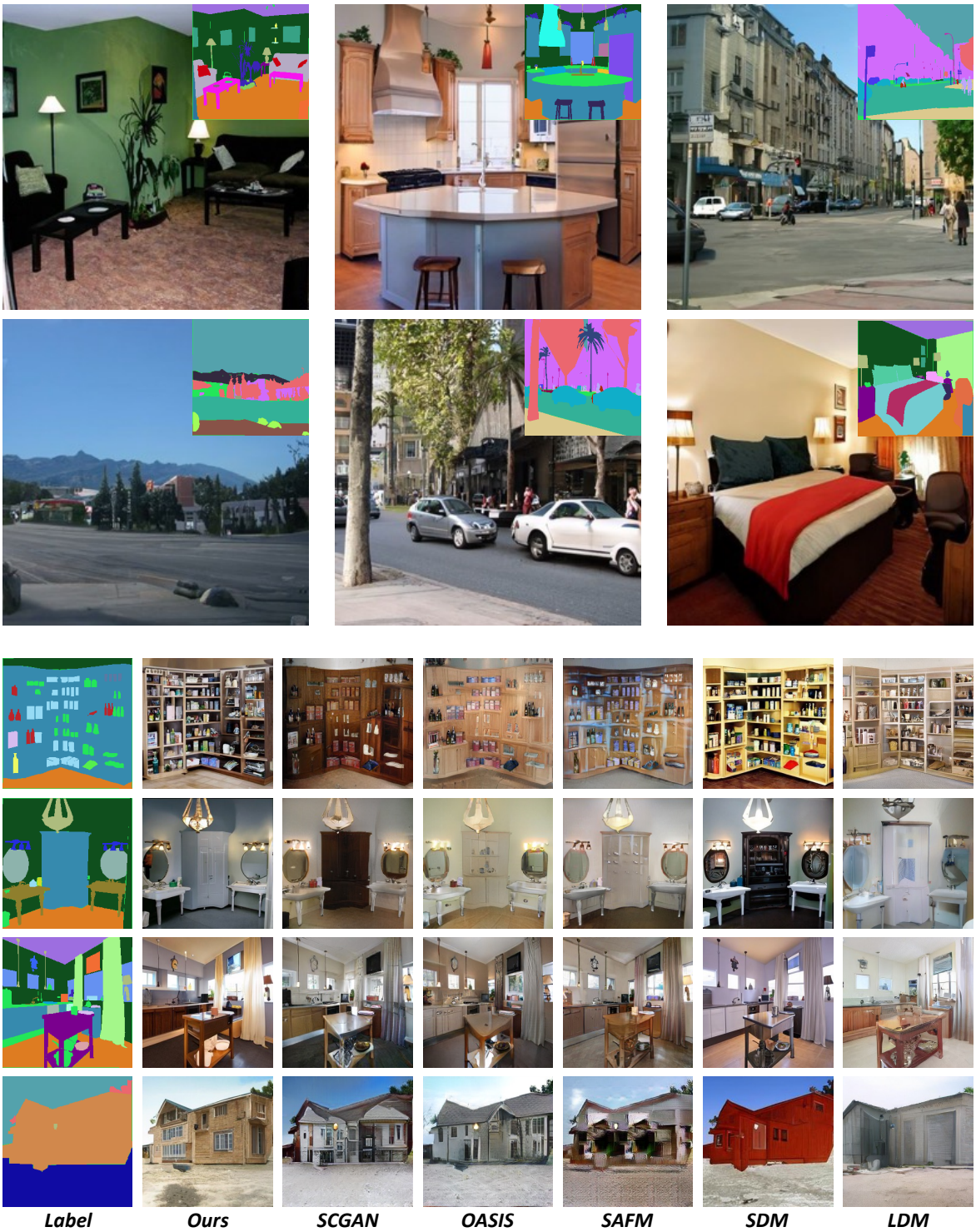


Figure 20. **More qualitative results and comparisons on *ADE20K*.** The first two rows show the result of our model, while the other rows depict qualitative comparisons in terms of generation quality.

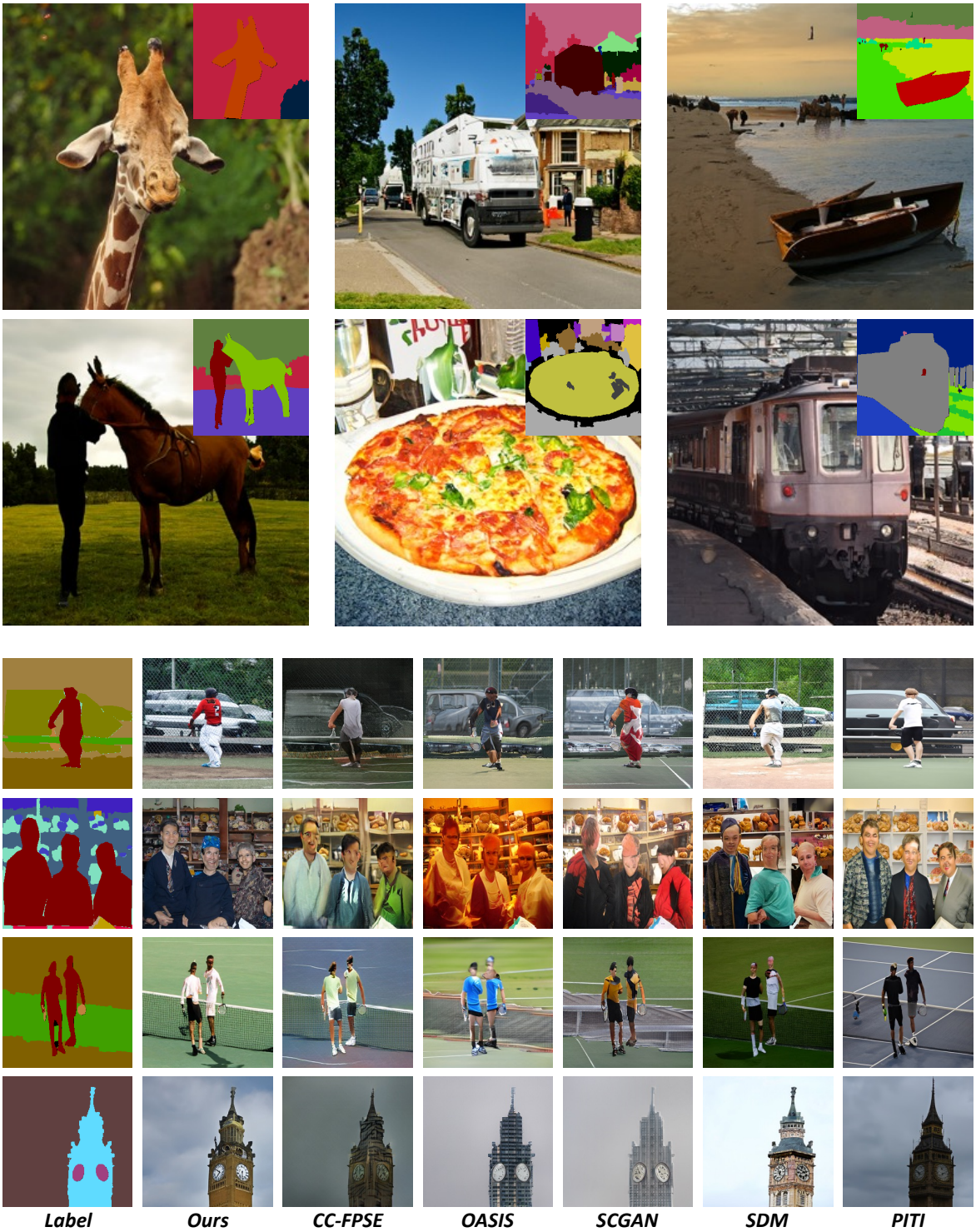


Figure 21. **More qualitative results and comparisons on *COCO-Stuff*.** The first two rows show the result of our model, while the other rows depict qualitative comparisons in terms of generation quality.