Taming Mode Collapse in Score Distillation for Text-to-3D Generation

Supplementary Material

A. Deferred Theory

We present deferred proofs and derivations in this section. In the beginning, we justify several claimed properties of J_{KL} (Eq. 4). Then we formally derive ESD (Eq. 8) via our proposed objective J_{Ent} (Eq. 7). Lastly, we prove that Classifier-Free Guidance trick (CFG) (Eq. 9) can be used to implement ESD.

A.1. Justification of Vanilla KL Divergence J_{KL}

Let us consider KL divergence objective restated from Eq. 4:

$$J_{KL}(\boldsymbol{\theta}) = \mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\mathcal{D}_{\mathrm{KL}}(q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})\|p_t(\boldsymbol{x}_t|\boldsymbol{y}))\right], \tag{12}$$

where we recall the notations: $\alpha_t, \sigma_t \in \mathbb{R}_+$ are time-dependent diffusion coefficients, $\boldsymbol{c} \sim p_c(\boldsymbol{c})$ is a camera pose drawn from a prior distribution over $\mathbb{SO}(3)\times\mathbb{R}^3$, and $g(\boldsymbol{\theta},\boldsymbol{c})$ renders an image at viewpoint $\boldsymbol{c}$ from the 3D representation $\boldsymbol{\theta}$. $p_t(\boldsymbol{x}_t|\boldsymbol{y})$ is the Gaussian diffused image distribution denoted as below:

$$p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \int p_0(\boldsymbol{x}_0|\boldsymbol{y})\mathcal{N}(\boldsymbol{x}_t|\alpha_t\boldsymbol{x}_0,\sigma_t^2\boldsymbol{I})d\boldsymbol{x}_0, \tag{13}$$

where $p_0(\boldsymbol{x}_0|\boldsymbol{y})$ is the text-conditioned distribution of clean images. We also define $q_t(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})$ as the Gaussian diffused distribution of rendered images:

$$q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y}) = \int q_0^{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{c})\mathcal{N}(\boldsymbol{x}_t|\alpha_t\boldsymbol{x}_0,\sigma_t^2\boldsymbol{I})d\boldsymbol{x}_0, \tag{14}$$

where we assume $\boldsymbol{x}_0$ is independent of text prompt $\boldsymbol{y}$ given the camera pose and underlying 3D representation. Furthermore, we assume the rendering process has no randomness, thus $q_0^{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{c}) = \delta(\boldsymbol{x}_0 - g(\boldsymbol{\theta},\boldsymbol{c}))$ can be written as a Dirac distribution.

Now, we can derive the gradient of $J_{KL}(\boldsymbol{\theta})$, as summarized in the following lemma:

Lemma 1 (Gradient of J_{KL})**.** *For any $\boldsymbol{\theta}$, we have:*

$$\nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta}) = -\mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c}),\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\omega(t)\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial\boldsymbol{\theta}}\sigma_t\nabla\log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right], \tag{15}$$

where $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$, and $\boldsymbol{x}_0 = g(\boldsymbol{\theta},\boldsymbol{c})$.

Proof. Due to the linearity of expectation, we have:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\mathcal{D}_{\mathrm{KL}}(q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})\|p_t(\boldsymbol{x}_t|\boldsymbol{y}))\right] \tag{16}$$

$$= \mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\nabla_\theta\mathcal{D}_{\mathrm{KL}}(q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})\|p_t(\boldsymbol{x}_t|\boldsymbol{y}))\right] \tag{17}$$

$$= \mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}\left[\log\frac{q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y})}\right]\right] \tag{18}$$

Fixing t and $\boldsymbol{c}$, we apply reparameterization trick:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}\left[\log\frac{q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y})}\right] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\underbrace{\nabla_{\boldsymbol{\theta}}\log q_t^{\boldsymbol{\theta}}(\alpha_t g(\boldsymbol{\theta},\boldsymbol{c})+\sigma_t\boldsymbol{\epsilon}|\boldsymbol{c},\boldsymbol{y})}_{(a)} - \underbrace{\nabla_{\boldsymbol{\theta}}\log p_t(\alpha_t g(\boldsymbol{\theta},\boldsymbol{c})+\sigma_t\boldsymbol{\epsilon}|\boldsymbol{y})}_{(b)}\right]. \tag{19}$$

Notice that $q_t^{\boldsymbol{\theta}}(\alpha_t g(\boldsymbol{\theta}, \boldsymbol{c}) + \sigma_t \boldsymbol{\epsilon} | \boldsymbol{c}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\epsilon} | \mathbf{0}, \boldsymbol{I})$ by substituting to Eq. 14, which is independent of $\boldsymbol{\theta}$. Thus $(a) = \mathbf{0}$. For term (b), by chain rule, we have:

$$\nabla_{\boldsymbol{\theta}} \log p_t(\alpha_t g(\boldsymbol{\theta}, \boldsymbol{c}) + \sigma_t \boldsymbol{\epsilon} | \boldsymbol{y}) = \alpha_t \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \nabla \log p_t(\alpha_t g(\boldsymbol{\theta}, \boldsymbol{c}) + \sigma_t \boldsymbol{\epsilon} | \boldsymbol{y}). \tag{20}$$

Plugging back to Eq. 18, we obtain:

$$\nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta}) = -\mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \cdot \alpha_t \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \nabla \log p_t(\boldsymbol{x}_t | \boldsymbol{y}) \right] \tag{21}$$

$$= -\mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \sigma_t \nabla \log p_t(\boldsymbol{x}_t | \boldsymbol{y}) \right], \tag{22}$$

where $\boldsymbol{x}_t = \alpha_t g(\boldsymbol{\theta}, \boldsymbol{c}) + \sigma_t \boldsymbol{\epsilon}$. □

Below we reproduce two results, which state both SDS (Eq. 1) and VSD (Eq. 2) optimize for J_{KL}.

Lemma 2 (SDS minimizes J_{KL} [31]). *For any $\boldsymbol{\theta}$, we have $J_{SDS}(\boldsymbol{\theta}) = J_{KL}(\boldsymbol{\theta}) + const$.*

Proof. It is sufficient to show $\nabla_{\boldsymbol{\theta}} J_{SDS}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta})$. By expansion:

$$\nabla_{\boldsymbol{\theta}} J_{SDS}(\boldsymbol{\theta}) = -\mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \left(\sigma_t \nabla \log p_t(\boldsymbol{x}_t | \boldsymbol{y}) - \boldsymbol{\epsilon} \right) \right] \tag{23}$$

$$= \underbrace{-\mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \sigma_t \nabla \log p_t(\boldsymbol{x}_t | \boldsymbol{y}) \right]}_{\nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta})} + \underbrace{\mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\omega(t) \sigma_t \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \boldsymbol{\epsilon} \right]}_{=\mathbf{0}}, \tag{24}$$

where the second term equals $\mathbf{0}$ because $\boldsymbol{\epsilon}$ is zero mean and sampled independently. □

Lemma 3 (Single-particle VSD minimizes J_{KL} [56]). *For any $\boldsymbol{\theta}$, we have $J_{VSD}(\boldsymbol{\theta}) = J_{KL}(\boldsymbol{\theta}) + const$.*

Proof. It is sufficient to show $\nabla_{\boldsymbol{\theta}} J_{VSD}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta})$. By a similar expansion:

$$\nabla_{\boldsymbol{\theta}} J_{VSD}(\boldsymbol{\theta}) = -\mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \left(\sigma_t \nabla \log p_t(\boldsymbol{x}_t | \boldsymbol{y}) - \sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{c}, \boldsymbol{y}) \right) \right] \tag{25}$$

$$= \underbrace{-\mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \sigma_t \nabla \log p_t(\boldsymbol{x}_t | \boldsymbol{y}) \right]}_{\nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta})} \tag{26}$$

$$+ \underbrace{\mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{c}, \boldsymbol{y}) \right]}_{=(a)} \tag{27}$$

Then we conclude the proof by showing $(a) = \mathbf{0}$ due to the fact that the first-order moment of score functions equals zero:

$$(a) = \mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \mathbb{E}_{\boldsymbol{x}_t \sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x} | \boldsymbol{c}, \boldsymbol{y})} \left[\alpha_t \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{c}, \boldsymbol{y}) \right] \right] \tag{28}$$

$$= \mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \mathbb{E}_{\boldsymbol{x}_t \sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x} | \boldsymbol{c}, \boldsymbol{y})} \left[\nabla_{\boldsymbol{\theta}} \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{c}, \boldsymbol{y}) \right] \right] \tag{29}$$

$$= \mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \int \frac{\nabla_{\boldsymbol{\theta}} q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{c}, \boldsymbol{y})}{q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{c}, \boldsymbol{y})} q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{c}, \boldsymbol{y}) d\boldsymbol{x}_t \right] \tag{30}$$

$$= \mathbb{E}_{t \sim \mathcal{U}[0,T], \boldsymbol{c} \sim p_c(\boldsymbol{c})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \nabla_{\boldsymbol{\theta}} \int q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t | \boldsymbol{c}, \boldsymbol{y}) d\boldsymbol{x}_t \right] = \mathbf{0}, \tag{31}$$

where we use change of variables by reversing the chain rule in Eq. 29, and the last step is because the integral equals one, which is independent of $\boldsymbol{\theta}$. □

Remark 1. For multi-particle VSD, Lemma 3 may not hold. This is because the reverse chain rule in Eq. 29 is no longer applicable as $q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c}, \boldsymbol{y})$ also becomes a function of $\boldsymbol{\theta}$.

Finally, we show that optimizing J_{KL} is equivalent to optimizing J_{MLE} (Eq. 5). First, recall that:

$$J_{MLE}(\boldsymbol{\theta}) = -\mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}\left[\log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right]\right]. \tag{32}$$

Then we state the following lemma:

Lemma 4 (J_{KL} is equivalent to maximal likelihood estimation)**.** *For any $\boldsymbol{\theta}$, we have $J_{MLE}(\boldsymbol{\theta}) = J_{KL}(\boldsymbol{\theta}) + const$.*

Proof. Again, we show $\nabla_{\boldsymbol{\theta}} J_{MLE}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta})$:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} J_{MLE}(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} - \mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}\left[\log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right]\right] &(33)\\
&= -\mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}\left[\nabla_{\boldsymbol{\theta}}\log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right]\right] &(34)\\
&= -\mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}\left[\alpha_t\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial\boldsymbol{\theta}}\nabla\log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right]\right] &(35)\\
&= -\mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c}),\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\omega(t)\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial\boldsymbol{\theta}}\sigma_t\nabla\log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right], &(36)
\end{aligned}$$

where the last step is basic reparameterization of $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$, and $\boldsymbol{x}_0 = g(\boldsymbol{\theta},\boldsymbol{c})$. □

As we argue in Sec. 3 (Eq. 5), the root reason J_{KL} degenerates to J_{MLE} is because the entropy term in J_{KL} becomes a constant independent of $\boldsymbol{\theta}$.

A.2. Derivation of Entropic Score Distillation

In this section, we derive the gradient for our entropy regularized objective (Eq. 8). We restate the entropy regularized objective (Eq. 7) below:

$$J_{Ent}(\boldsymbol{\theta},\lambda) = -\mathbb{E}_{t\sim\mathcal{U}[0,T],\boldsymbol{c}\sim p_c(\boldsymbol{c})}\left[\omega(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})}\log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right] - \lambda\,\mathbb{E}_{t\sim\mathcal{U}[0,T]}\left[\omega(t)\frac{\sigma_t}{\alpha_t}H[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})]\right], \tag{37}$$

where the entropy term $H[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})]$ is defined as:

$$H\left[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right] = -\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})}\left[\log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right], \tag{38}$$

and distribution $q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})$ is defined as:

$$q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}) = \int q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})p_c(\boldsymbol{c})d\boldsymbol{c}. \tag{39}$$

Notice that $J_{Ent}(\boldsymbol{\theta},\lambda) = J_{MLE}(\boldsymbol{\theta}) - \lambda\,\mathbb{E}_{t\sim\mathcal{U}[0,T]}\left[\omega(t)\frac{\sigma_t}{\alpha_t}H[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})]\right]$, therefore, to derive Eq. 8, we simply need the gradient of the entropy term:

Lemma 5 (Gradient of entropy)**.** *It holds that:*

$$\nabla_{\boldsymbol{\theta}} H\left[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right] = -\mathbb{E}_{\boldsymbol{c}\sim p_c(\boldsymbol{c}),\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\alpha_t\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial\boldsymbol{\theta}}\nabla\log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right]. \tag{40}$$

Proof. We expand entropy by reparameterization of $q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})$ as sampling two independent variables $\boldsymbol{c}, \boldsymbol{\epsilon}$:

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}} H\left[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right] = \nabla_{\boldsymbol{\theta}}\,\mathbb{E}_{\boldsymbol{c}\sim p_c(\boldsymbol{c}),\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[-\log q_t^{\boldsymbol{\theta}}(\alpha_t g(\boldsymbol{\theta},\boldsymbol{c}) + \sigma_t\boldsymbol{\epsilon}|\boldsymbol{y})\right] &(41)\\
&= -\mathbb{E}_{\boldsymbol{c}\sim p_c(\boldsymbol{c}),\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\nabla_{\boldsymbol{\theta}}\log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}) + \alpha_t\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial\boldsymbol{\theta}}\nabla_{\boldsymbol{x}_t}\log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right]\Bigg|_{\boldsymbol{x}_t=\alpha_t g(\boldsymbol{\theta},\boldsymbol{c})+\sigma_t\boldsymbol{\epsilon}} &(42)\\
&= -\underbrace{\mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})}\left[\nabla_{\boldsymbol{\theta}}\log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right]}_{=(a)} - \mathbb{E}_{\boldsymbol{c}\sim p_c(\boldsymbol{c}),\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\alpha_t\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial\boldsymbol{\theta}}\nabla_{\boldsymbol{x}_t}\log q_t^{\boldsymbol{\theta}}(\alpha_t g(\boldsymbol{\theta},\boldsymbol{c}) + \sigma_t\boldsymbol{\epsilon}|\boldsymbol{\theta})\right], &(43)
\end{aligned}$$

where it is noteworthy that $\nabla_{\boldsymbol{x}_t} \log q_t^{\boldsymbol{\theta}}$ simply denotes the score function of $q_t^{\boldsymbol{\theta}}$ by explicitly indicating the derivative is taken in terms of $\boldsymbol{x}_t$. Eq. 42 is obtained by path derivative. It remains to show $(a) = \mathbf{0}$. We recall that the first-order moment of a score function equals to zero:

$$\begin{aligned}(a) &= \int \nabla_{\boldsymbol{\theta}} \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}) q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}) d\boldsymbol{x}_t = \int \frac{\nabla_{\boldsymbol{\theta}} q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})}{q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})} q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}) d\boldsymbol{x}_t & (44)\\ &= \nabla_{\boldsymbol{\theta}} \int q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}) d\boldsymbol{x}_t & (45)\\ &= \mathbf{0}, & (46)\end{aligned}$$

where the last step involves a change of variable and the integral turns out to be independent of $\boldsymbol{\theta}$. □

As a consequence, we can conclude the update rule yielded by Eq. 7 in the following theorem:

Theorem 2 (Entropic Score Distillation)**.** *For any $\boldsymbol{\theta}$ and $\lambda \in \mathbb{R}$, the following holds:*

$$\nabla_{\boldsymbol{\theta}} J_{Ent}(\boldsymbol{\theta}, \lambda) = -\mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \left(\sigma_t \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) - \lambda \sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right)\right], \tag{47}$$

where $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$, and $\boldsymbol{x}_0 = g(\boldsymbol{\theta}, \boldsymbol{c})$.

Proof. Since $J_{Ent}(\boldsymbol{\theta}, \lambda) = J_{MLE}(\boldsymbol{\theta}) - \lambda\, \mathbb{E}_{t\sim\mathcal{U}[0,T]} \left[\omega(t) \frac{\sigma_t}{\alpha_t} H[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})]\right]$, by Lemma 4 and Lemma 5:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J_{Ent}(\boldsymbol{\theta}, \lambda) &= \nabla_{\boldsymbol{\theta}} J_{MLE}(\boldsymbol{\theta}) - \lambda\, \mathbb{E}_{t\sim\mathcal{U}[0,T]} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \nabla_{\boldsymbol{\theta}} H[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})]\right] & (48)\\ &= -\mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \sigma_t \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right] & (49)\\ &\quad + \mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \lambda \sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right], & (50)\end{aligned}$$

by which we conclude the proof by merging two expectations. □

A.3. Justification of Classifier-Free Guidance Trick

In this section, we first prove Theorem 1 and show that CFG trick (Eq. 9) can be utilized to implement ESD. To begin with, we define another type of KL divergence as below:

$$\overline{J_{KL}} = \mathbb{E}_{t\sim\mathcal{U}[0,T]} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \mathcal{D}_{\mathrm{KL}}(q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}) \| p_t(\boldsymbol{x}_t|\boldsymbol{y}))\right] \tag{51}$$

Then we present the following lemma, which represents the gradient of $\overline{J_{KL}}$:

Lemma 6 (Gradient of $\overline{J_{KL}}$)**.** *It holds that:*

$$\nabla_{\boldsymbol{\theta}} \overline{J_{KL}} = -\mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\omega(t) \frac{\partial g(\boldsymbol{\theta}, \boldsymbol{c})}{\partial \boldsymbol{\theta}} \left(\sigma_t \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) - \sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right)\right], \tag{52}$$

where $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$, and $\boldsymbol{x}_0 = g(\boldsymbol{\theta}, \boldsymbol{c})$.

Proof. We prove by showing $\overline{J_{KL}}$ is a special case of J_{Ent} when setting $\lambda = 1$:

$$\begin{aligned}&\overline{J_{KL}}(\boldsymbol{\theta}) = \mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \log \frac{q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})}{p_t(\boldsymbol{x}_t|\boldsymbol{y}))}\right] & (53)\\ &= -\mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}))\right] + \mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right] & (54)\\ &= J_{MLE}(\boldsymbol{\theta}) + \mathbb{E}_{t\sim\mathcal{U}[0,T]}\, \mathbb{E}_{\boldsymbol{x}_t\sim q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right] & (55)\\ &= J_{MLE}(\boldsymbol{\theta}) - \mathbb{E}_{t\sim\mathcal{U}[0,T]} \left[\omega(t) \frac{\sigma_t}{\alpha_t} H\left[q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right]\right] = J_{Ent}(\boldsymbol{\theta}, 1) & (56)\end{aligned}$$

□

Now we prove Theorem 1 using previous results:

Proof of Theorem 1. It is sufficient to show that $\nabla_{\boldsymbol{\theta}} J_{Ent}(\boldsymbol{\theta}, \lambda) = \lambda \nabla_{\boldsymbol{\theta}} \overline{J_{KL}}(\boldsymbol{\theta}) + (1-\lambda)\nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta})$. By Lemma 1, 6, as well as Theorem 2, we can obtain:

$$\lambda \nabla_{\boldsymbol{\theta}} \overline{J_{KL}}(\boldsymbol{\theta}) + (1-\lambda)\nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta}) = -\lambda\, \mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})} \left[\omega(t)\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial \boldsymbol{\theta}} \left(\sigma_t \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) - \sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right)\right] \tag{57}$$

$$- (1-\lambda)\, \mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})} \left[\omega(t)\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial \boldsymbol{\theta}} \sigma_t \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y})\right] \tag{58}$$

$$= \nabla_{\boldsymbol{\theta}} J_{Ent}(\boldsymbol{\theta}, \lambda), \tag{59}$$

by merging two expectations. ☐

Further on, our CFG trick implementation of ESD (Eq. 9) can be regarded as a corollary of Theorem 1 and Lemma 3:

Theorem 3 (Classifier-Free Guidance Trick)**.** *For any* $\boldsymbol{\theta}$ *and* $\lambda \in \mathbb{R}$, $\nabla_{\boldsymbol{\theta}} J_{Ent}(\boldsymbol{\theta}, \lambda)$ *equals to the following:*

$$- \mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})} \left[\omega(t)\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial \boldsymbol{\theta}} \left(\sigma_t \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) - \lambda\sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}) - (1-\lambda)\sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})\right)\right], \tag{60}$$

where $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$*, and* $\boldsymbol{x}_0 = g(\boldsymbol{\theta}, \boldsymbol{c})$*.*

Proof. By Theorem 1, we know that $\nabla_{\boldsymbol{\theta}} J_{Ent}(\boldsymbol{\theta}, \lambda) = \lambda \nabla_{\boldsymbol{\theta}} \overline{J_{KL}}(\boldsymbol{\theta}) + (1-\lambda)\nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta})$. Moreover, by Lemma 3, we have $\nabla_{\boldsymbol{\theta}} J_{KL}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} J_{VSD}(\boldsymbol{\theta})$. As a result, the following can be derived:

$$\nabla_{\boldsymbol{\theta}} J_{Ent}(\boldsymbol{\theta}, \lambda) = \nabla_{\boldsymbol{\theta}} \overline{J_{KL}}(\boldsymbol{\theta}) + (1-\lambda)\nabla_{\boldsymbol{\theta}} J_{VSD}(\boldsymbol{\theta}) \tag{61}$$

$$= -\lambda\, \mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})} \left[\omega(t)\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial \boldsymbol{\theta}} \left(\sigma_t \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) - \sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y})\right)\right] \tag{62}$$

$$- (1-\lambda)\, \mathbb{E}_{t\sim\mathcal{U}[0,T], \boldsymbol{c}\sim p_c(\boldsymbol{c}), \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})} \left[\omega(t)\frac{\partial g(\boldsymbol{\theta},\boldsymbol{c})}{\partial \boldsymbol{\theta}} \left(\sigma_t \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) - \sigma_t \nabla \log q_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c},\boldsymbol{y})\right)\right], \tag{63}$$

as desired after merging two expectations. ☐

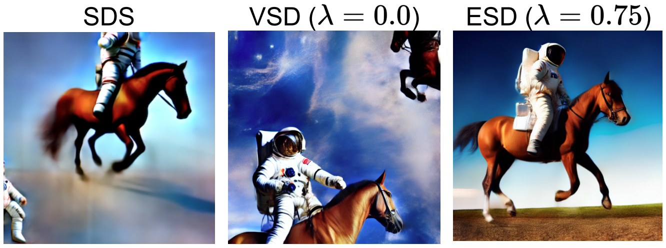
B. Illustrative Examples

Gaussian Distribution Fitting. In this section, we provide the necessary details on Fig. 3, where we fit a 2D Gaussian distribution via SDS, VSD, and ESD. Suppose the targeted Gaussian distribution is $p_0(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0|\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$, where $\boldsymbol{\mu}^* \in \mathbb{R}^D$ is the mean vector and $\boldsymbol{\Sigma}^* \in \mathbb{R}^{D\times D}$ is the positive definite covariance matrix. Define differentiable function $g(\{\boldsymbol{b}, \boldsymbol{A}\}, \boldsymbol{c}) = \boldsymbol{b} + \boldsymbol{A}\boldsymbol{c}$, where $\boldsymbol{b}$ and $\boldsymbol{A}$ are parameters to be fitted, and $\boldsymbol{c} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is a random variable sampled from a standard Gaussian distribution. It is obvious that probability density function of $g(\{\boldsymbol{b}, \boldsymbol{A}\}, \boldsymbol{c})$ is $q_0^{\boldsymbol{b},\boldsymbol{A}}(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0|\boldsymbol{b}, \boldsymbol{A}\boldsymbol{A}^\top)$. Our objective is to match p_0 and $q_0^{\boldsymbol{b},\boldsymbol{A}}$ by optimizing parameters $\boldsymbol{b}$ and $\boldsymbol{A}$ with SDS, VSD, and ESD. Notice that diffusion perturbed p_0 and $q_0^{\boldsymbol{b},\boldsymbol{A}}$ are still Gaussian distributions. And the score functions used in score distillation can be computed in the closed form:

$$\nabla \log p_t(\boldsymbol{x}_t) = (\alpha_t^2 \boldsymbol{\Sigma}^* + \sigma_t^2 \boldsymbol{I})^{-1}(\alpha_t \boldsymbol{\mu}^* - \boldsymbol{x}_t), \tag{64}$$

$$\nabla \log q_t^{\boldsymbol{b},\boldsymbol{A}}(\boldsymbol{x}_t) = (\alpha_t^2 \boldsymbol{A}\boldsymbol{A}^\top + \sigma_t^2 \boldsymbol{I})^{-1}(\alpha_t \boldsymbol{b} - \boldsymbol{x}_t), \quad \nabla \log q_t^{\boldsymbol{b},\boldsymbol{A}}(\boldsymbol{x}_t|\boldsymbol{c}) = \frac{\alpha_t(\boldsymbol{b} + \boldsymbol{A}\boldsymbol{c}) - \boldsymbol{x}_t}{\sigma_t^2}. \tag{65}$$

In our experiments, we run score distillation for 2k steps with 100 warm-up steps. The learning rate is set to 0.01. We observe that SDS and VSD have similar convergence behavior as maximal likelihood methods. When increasing λ from 0 to 1, i.e., enhancing the effect of entropy maximization, the fitted distribution gradually covers the targeted support.



SDS VSD ($\lambda = 0.0$) ESD ($\lambda = 0.75$)

Figure 8. **Image Reconstruction Example.** We leverage SDS, VSD, and ESD to recover a high-resolution 2D image by matching the distribution of its random crops with a pre-trained text-conditioned diffusion model. Prompt: An astronaut riding a horse in space.

2D Image Reconstruction. We also conduct 2D image experiments to demonstrate the working mechanism of ESD. We focus on the image reconstruction task via partial observations [6, 57], where the optimized parameters represent a high-resolution image and a random small window of the image will be rendered at each iteration of score distillation. Formally, let $\boldsymbol{\theta}$ denote the high-resolution image, and $g(\boldsymbol{\theta}, \boldsymbol{m}) = \boldsymbol{\theta} \odot \boldsymbol{m}$, where $\boldsymbol{m}$ is a random binary mask. We choose $\nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$ as a pre-trained text-to-image diffusion model, and $\boldsymbol{y}$ is the text prompt, specified as "An astronaut riding a horse in space" in our experiments. Akin to [56], $\nabla \log q_t(\boldsymbol{x}_t)$ and $\nabla \log q_t(\boldsymbol{x}_t|\boldsymbol{m})$ are fitted via LoRA using cropped images. During training, we fix training steps as 10k, learning rate as 1e-2 for $\boldsymbol{\theta}$, and 1e-4 for LoRA. We also apply the cosine learning rate scheduler with 100 warm-up steps. We present qualitative results in Fig. 8. It demonstrates that SDS and VSD cause "Janus"-like problems where each image contains duplicated instances while ESD avoids such issues and generates only one target object.

C. Experiment Details

In this section, we provide more details on the implementation of ESD and the compared baseline methods. All of them are implemented under the `threestudio` framework and include three stages: coarse generation, geometry refinement, and texture refinement, following [56]. For the coarse generation stage, we adopt foreground-background disentangled hash-encoded NeRF [28] as the underlying 3D representation, and DMTet [40] for two refinement stages. All scenes are trained for 25k steps for the coarse stage, 10k steps for geometry refinement, and 30k steps for texture refinement, for the sake of fair comparison. At each iteration, we randomly render one view (i.e., batch size equals one). We progressively adjust rendering resolution: within the first 5k steps, we render at 64×64 resolution and increase to 256×256 resolution afterward.

SDS [31]. Following the original paper, we set the CFG weight to 100. Additionally, we encourage sparsity of the density field and penalize the mismatch between orientation and predicted normal maps. Lighting augmentation is also enabled for SDS. The geometry refinement step is directly borrowed from VSD: a DMTet is initialized by NeRF's density field via marching cube while the end-to-end optimization with SDS is then conducted on the geometry representation for both geometry and texture.

VSD [56]. We reuse the standard setting of VSD for all three stages. In particular, we fix the CFG coefficient to 7.5 and only use single-particle VSD, conforming with our theoretical analysis. During the geometry refinement stage, we adopt SDS guidance instead of VSD.

Debiased-SDS [12]. Our implementation of Debiased-SDS is built upon SDS. We enable both score debiasing and prompt debiasing. For score debiasing, we follow the default setting and linearly increase the absolute threshold for gradient clipping from 0.5 to 2.0. All other hyperparameters follow from SDS.

Perp-Neg [2]. Perp-Neg implementation is based on SDS as well. As suggested by the original paper, in positive prompts, we leverage weights $r_{interp} = 1 - 2|\text{azimuth}|/\pi$ for front-side prompt interpolation and $r_{interp} = 2 - 2|\text{azimuth}|/\pi$ for side-back interpolation. In negative prompts, the interpolating function is chosen as the shifted exponential function $\alpha \exp(-\beta r_{interp}) + \gamma$. Specifically, we choose $\alpha_{sf} = 1, \beta_{sf} = 0.5, \gamma_{sf} = -0.606, \alpha_{fsb} = 1, \beta_{fsb} = 0.5, \gamma_{fsb} = 0.967, \alpha_{fs} = 4, \beta_{fs} = 0.5, \gamma_{fs} = -2.426, \alpha_{sf} = 4, \beta_{sf} = 0.5, \gamma_{sf} = -2.426$. See [2] for more details on the meaning of these hyperparameters.

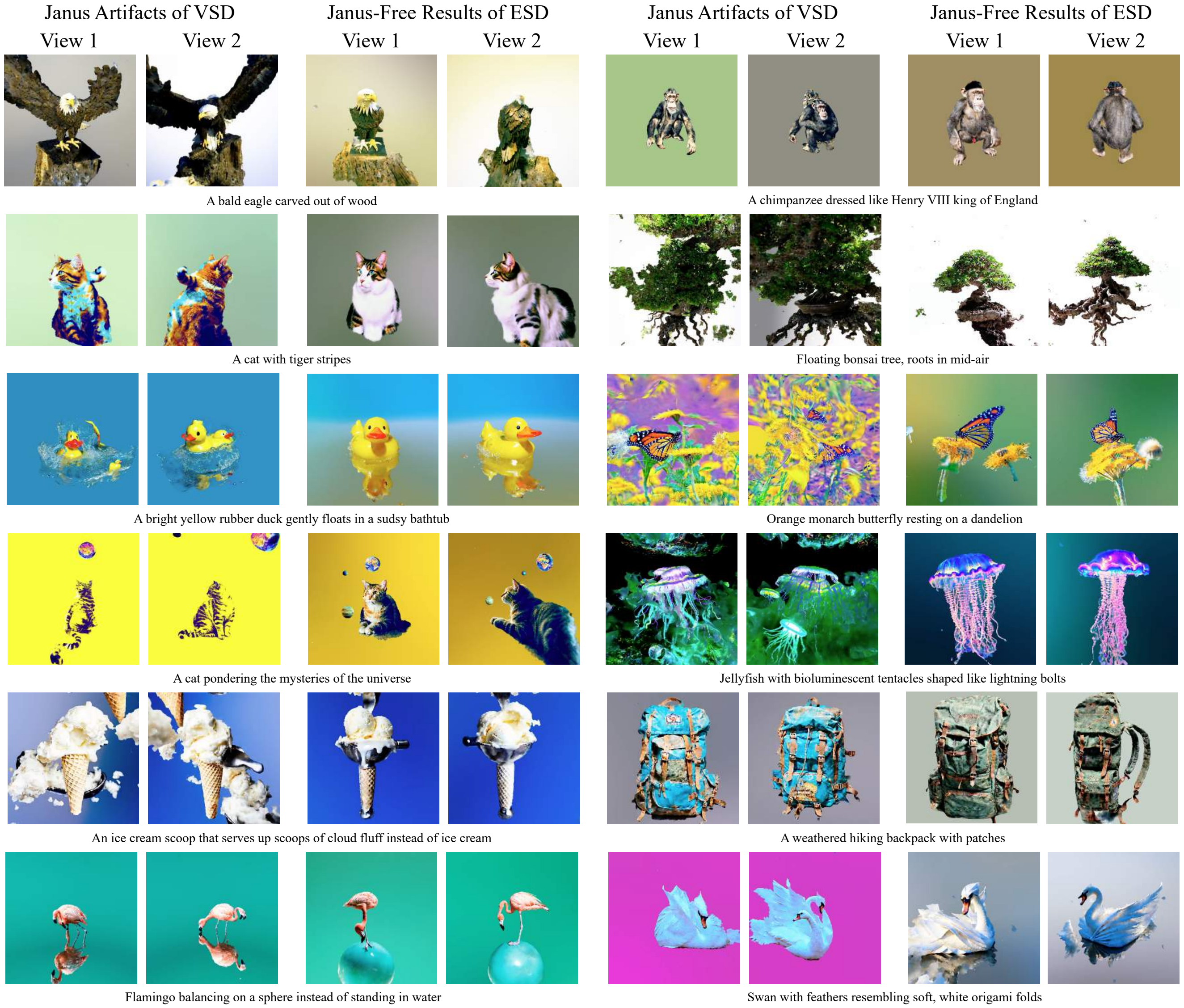

Janus Artifacts of VSD
View 1
View 2
Janus-Free Results of ESD
View 1
View 2
Janus Artifacts of VSD
View 1
View 2
Janus-Free Results of ESD
View 1
View 2
A bald eagle carved out of wood
A chimpanzee dressed like Henry VIII king of England
A cat with tiger stripes
Floating bonsai tree, roots in mid-air
A bright yellow rubber duck gently floats in a sudsy bathtub
Orange monarch butterfly resting on a dandelion
A cat pondering the mysteries of the universe
Jellyfish with bioluminescent tentacles shaped like lightning bolts
An ice cream scoop that serves up scoops of cloud fluff instead of ice cream
A weathered hiking backpack with patches
Flamingo balancing on a sphere instead of standing in water
Swan with feathers resembling soft, white origami folds

Figure 9. **More Qualitative Results.** We present the two views of each object synthesized by VSD (ProlificDreamer) and our method, respectively. Best view in an electronic copy.

ESD. Our ESD implementation is similar to VSD. We leverage the extrinsics matrix (4×4) as the camera pose embedding, and condition the diffusion model by replacing its class embedding branch. We introduce CFG trick to linearly mix camera-conditioned and unconditioned score functions fine-tuned with rendered images. We find CFG 0.5 generally yields desirable results. We also set the probability of unconditioned training to 0.5. In particular, view-dependent prompting is disabled for the fine-tuned score function.

D. More Qualitative Results

In this section, we present more qualitative results in Fig. 9. All text prompts are generated by GPT-4V and directly picked from [58]. We mainly compare ESD with VSD to highlight the influence of the entropy regularization term. The observation is consistent with our main text. The outcomes of VSD often exhibit broken geometries, duplicated objects, and multiple signature views, which contradict the inherent characteristics of the generated subjects. ESD can effectively mitigate “Janus” issues and generate more realistic contents.

E. Numerical Evaluation

Human Evaluation Criteria. In our human evaluation of Successful generation Rate (SR), a text prompt is labeled as "successfully generated" if at least one of three random seeds yields a generation satisfying the criteria: ([2], Appendix A.2):

1. The rendered images show the requested object(s), which is positioned on the correct view.
2. The rendered images do not show hallucination including counterfactual details, for example, a panda has three ears.
3. The rendered images do not have unrealistic color or texture or massive floaters.

Extension of Tab. 1. In Tab. 2, we include standard deviations of all numerical results presented in Tab. 1. We note that ESD exhibits a smaller variance of different metrics, indicating its training might be well-regularized and more robust.

Table 2. **Quantitative Comparisons.** (↓) means the lower the better, and (↑) means the higher the better.

	CLIP (↓)	FID (↓)	IQ (↓)	IV (↑)	IG (↑)	SR (↑)
SDS [31]	0.737±0.068	291.860±61.242	4.295±0.419	**4.8552±0.342**	0.123±0.053	15.00%
VSD [56]	0.725±0.072	265.141±58.549	3.149±1.234	3.5712±1.345	0.137±0.061	19.17%
ESD (Ours)	**0.714±0.065**	**235.915±56.558**	**3.135±1.088**	4.0314±1.285	**0.327±0.185**	**55.83%**

Breakdown Table. We provide a breakdown table to present quantitative evaluations of results in Fig. 4. The numbers are reported in Tab. 3. The conclusion is consistent with our argument in Sec. 7. Our ESD consistently outperforms all the compared baseline methods, especially in FID and IG. This implies that ESD effectively boosts the view diversity and accurately matches the distribution between pre-trained image distribution and rendered image distribution.

Table 3. **Quantitative Comparisons.** (↓) means the lower the better, and (↑) means the higher the better.

	CLIP (↓)	FID (↓)	IQ (↓)	IV (↑)	IG (↑)	CLIP (↓)	FID (↓)	IQ (↓)	IV (↑)	IG (↑)
	Michelangelo style statue of dog reading news on a cellphone					A rabbit, animated movie character, high detail 3d model				
SDS [31]	0.694	365.304	4.469	**5.119**	0.145	**0.712**	200.084	4.365	**4.970**	**0.138**
VSD [56]	0.758	296.168	**2.514**	3.041	0.209	0.720	150.120	**1.083**	1.173	0.083
Debiased-SDS [12]	0.778	351.493	4.058	4.814	0.186	0.735	216.058	4.443	4.857	0.093
Perp-Neg [2]	0.793	306.918	3.970	4.572	0.151	0.727	176.279	2.453	2.665	0.086
ESD (Ours)	**0.685**	**292.716**	2.523	4.080	**0.617**	0.725	**149.763**	1.385	1.567	0.132
	A rotary telephone carved out of wood					A plush dragon toy				
SDS [31]	0.853	309.929	3.478	4.179	0.202	0.889	243.984	4.622	**5.008**	0.084
VSD [56]	0.855	305.920	3.469	4.214	0.214	0.821	273.495	**4.382**	4.728	0.078
Debiased-SDS [12]	0.927	313.893	4.098	4.201	0.025	0.878	262.474	4.827	4.954	0.026
Perp-Neg [2]	0.868	308.554	3.488	4.021	0.153	0.839	309.276	4.691	4.816	0.027
ESD (Ours)	**0.846**	**299.578**	**3.332**	**4.439**	**0.366**	**0.815**	**237.518**	4.436	4.971	**0.121**

F. Limitations and Failure Cases

We note that by Theorem 1, ESD still optimizes for a mode-seeking KL divergence. This suggests that ESD may still lead to mode collapse especially when the target image distribution is overly concentrated on one peak [36]. Careful tuning of λ is also necessary to balance the per-view sharpness/details and cross-view diversity. It also remains open whether ESD can further benefit multi-particle VSD or amortized text-to-3D training [23].

Below we present a failure case produced by our ESD in Fig. 10, where the back view of the marble still contains a mouse face while the side views exhibit duplicate ears. We point out that even though ESD can encourage diversity among views, however, it may still incline to one mode when the target image distribution is overwhelmingly concentrated at one point. The specified text prompt in Fig. 10 is in this case as we observe the majority of sampled images from a pre-trained diffusion model with the corresponding prompt are the frontal views of a marble mouse.

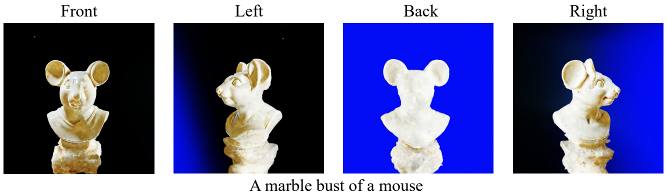


Figure 10. **Failure case.** We present four views of a failure case yielded by ESD with prompt "A marble bust of a mouse" and CFG weight $\lambda = 0.5$.