Supplementary of LucidDreamer: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching

Yixun Liang*[1] Xin Yang*[1,2] Jiantao Lin[1] Haodong Li[1] Xiaogang Xu[3,4] Yingcong Chen**[1,2]
[1] HKUST (GZ) [2] HKUST [3] CUHK [4] Zhejiang University
yliang982@connect.hkust-gz.edu.cn xin.yang@connect.ust.hk jlin695@hkust-gz.edu.cn
hli736@connect.hkust-gz.edu.cn xiaogangxu00@gmail.com yingcongchen@ust.hk

1. Implementation details

In our LucidDreamer framework, we adopt an explicit 3D representation, the 3D Gaussian Splatting (3DGS) [5], for 3D distillation with our proposed Interval Score Matching (ISM) objective. To optimize 3DGS towards the pseudo-ground-truth (pseudo-GT) generated by diffusion models, we follow most training hyperparameters from the original 3DGS paper. Specifically, we implement a strategy of densifying and pruning the Gaussian at every 300 iteration interval until a total of 3000 iterations. As our ISM provides precise gradients, we observe a significantly high coverage speed. Consequently, we streamline our training process to consist of around 5000 iterations, substantially less than the original 10,000 iterations required in previous works [9]. In terms of the initialization of 3DGS, we utilize the pretrained Point-E [8] checkpoint. Also, for some asymmetrical objects, we adopt camera-dependent prompts during the training following Perp-Neg [1] to reduce the Janus problems further.

LucidDreamer with negative prompts Also, we find that negative prompts would further improve the generation quality, thus, we use the negative prompts from [4] in some cases. Denoting y and y_n as the positive and negative prompts, we predict the text-conditional score of the noisy latent x_t following the classifier-free guidance [3]:

$$\boldsymbol{\epsilon}_\phi(x_t,t,y) = \boldsymbol{\epsilon}_\phi(x_t,t,y_n) + gs * (\boldsymbol{\epsilon}_\phi(x_t,t,y) - \boldsymbol{\epsilon}_\phi(x_t,t,y_n)), \quad (1)$$

where gs is the guidance scale of prompt y.

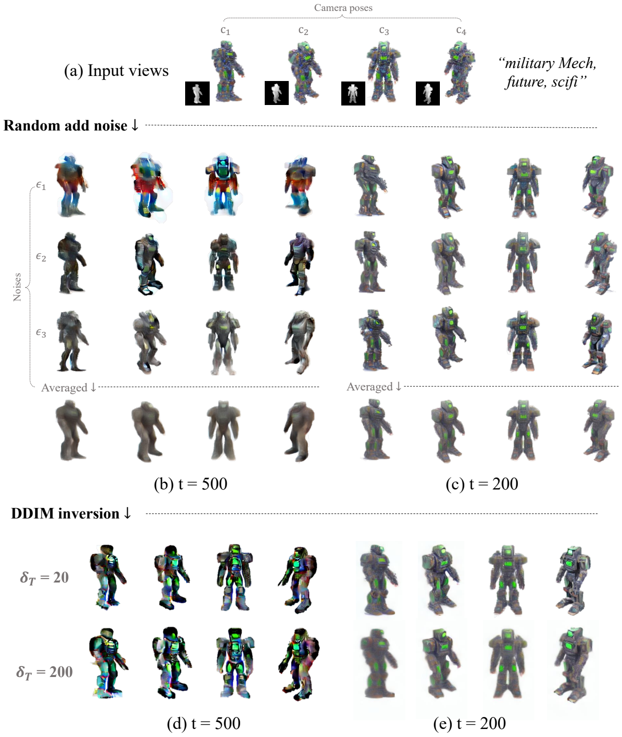


Figure 1. **(a)**: The rendered x_0 from 3D representation with camera poses $c = \{c_1, ..., c_4\}$. **(b) and (c)**: pseudo-GTs $\hat{x}_0^t$ generated via randomly add noise $\boldsymbol{\epsilon} = \{\boldsymbol{\epsilon}_1, ...\boldsymbol{\epsilon}_3\}$ to x_0 at timestep $t = \{500, 200\}$. **(e) and (f)**: pseudo-GTs $\hat{x}_0^t$ generated via DDIM inversion with step size of $\delta_T = \{20, 200\}$ at timestep $t = \{500, 200\}$. Please zoom in for details.

2. Inconsistency in SDS pseudo-GT

In our main paper, we discussed the inconsistency issue regards the pseudo-GTs produced by SDS [9] in our revisiting of SDS. Specifically, it raised our concerns when we spotted significant inconsistency among the pseudo-GTs. Our investigation points out that such inconsistency is mainly caused by the following properties of the SDS algorithm:

** Corresponding author.
*The first two authors contributed equally to this work.
* *Conceptualization*: **Yixun Liang**: 60%, **Xin Yang**: 40%,
Methodology: **Xin Yang**: 60%, **Yixun Liang**: 40%.

(1) randomness in timestep t; (2) randomness in the noise component ϵ of x_t; (3) randomness in camera pose c.

To better explain the issue, we conducted a quantitative experiment on the inconsistency of pseudo-GTs with the aforementioned properties. In Fig. 1 (a), we visualize the input views of 4 camera poses and the pseudo-GTs produced by SDS at different timesteps (Fig. 1 (b) and (c)) and with different noise ϵ (row 2 to 3). It can be seen that even with the noise fixed, the SDS pseudo-GTs tend to be inconsistent over different camera poses and timesteps and eventually lead to feature-averaged results, which is inevitable under the SDS distillation scheme.

Also, In Fig. 2 of our main paper, we visualize the “feature-averaging" problem with a batch size of 4 to ease our explanation. Howver, in the original DreamFusion, the batch size are set as 1. Thus, in Fig. 2, we visualize the process of a fixed input view being updated in a sequential manner with SDS loss to explain that the “feature-averaging" problem also happened in batch size 1. Notably, since the noise and timestep are randomly selected in each iteration, the style of the pseudo-GTs still fluctuates significantly during the training, and the input view eventually goes smoother, which also fits our discussion and conclusion in the main paper.

3. Complementary Experiments of ISM

3.1. Benefits of DDIM inversion

In the previous section, we visualize the inconsistency issue of SDS pseudo-GTs. In the methodology section of our main paper, we propose to mitigate such a problem by introducing DDIM inversion for noisy latent estimation. Hence, we further examine the effect of replacing the vanilla add noise function for $x_0 \to x_t$ with DDIM inversion in Fig. 1 (d) and (e). It can be seen that, the pseudo-GTs that incorporate with DDIM inversion are more similar to the input views in Fig. 1 (a). Therefore, they are significantly more consistent feature and style-wise between different views and timesteps compared to Fig. 1 (b) and (c). Meanwhile, such a property holds when we increase δ_T from 20 to 200. Notably, DDIM inversion doesn’t necessarily handle the quality problem of the pseudo-GTs generated with a single-step prediction with diffusion models. We will delve deeper into this problem in Sec. 3.2.

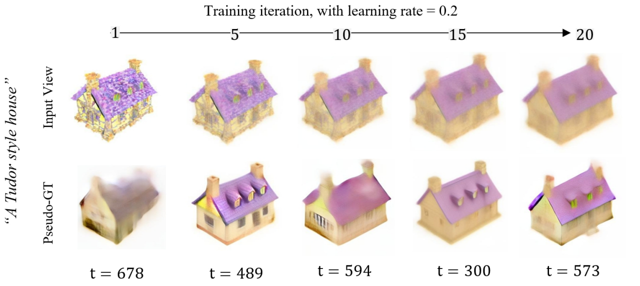


Figure 2. **"feature averaging" effect with sequentially updating.** The input view is updated sequentially with SDS loss. Which shows that the proposed problem also happen in batch size 1

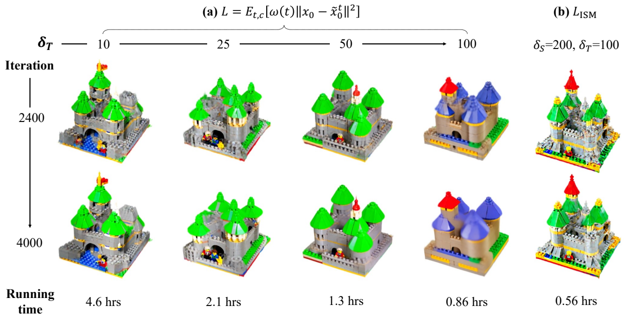


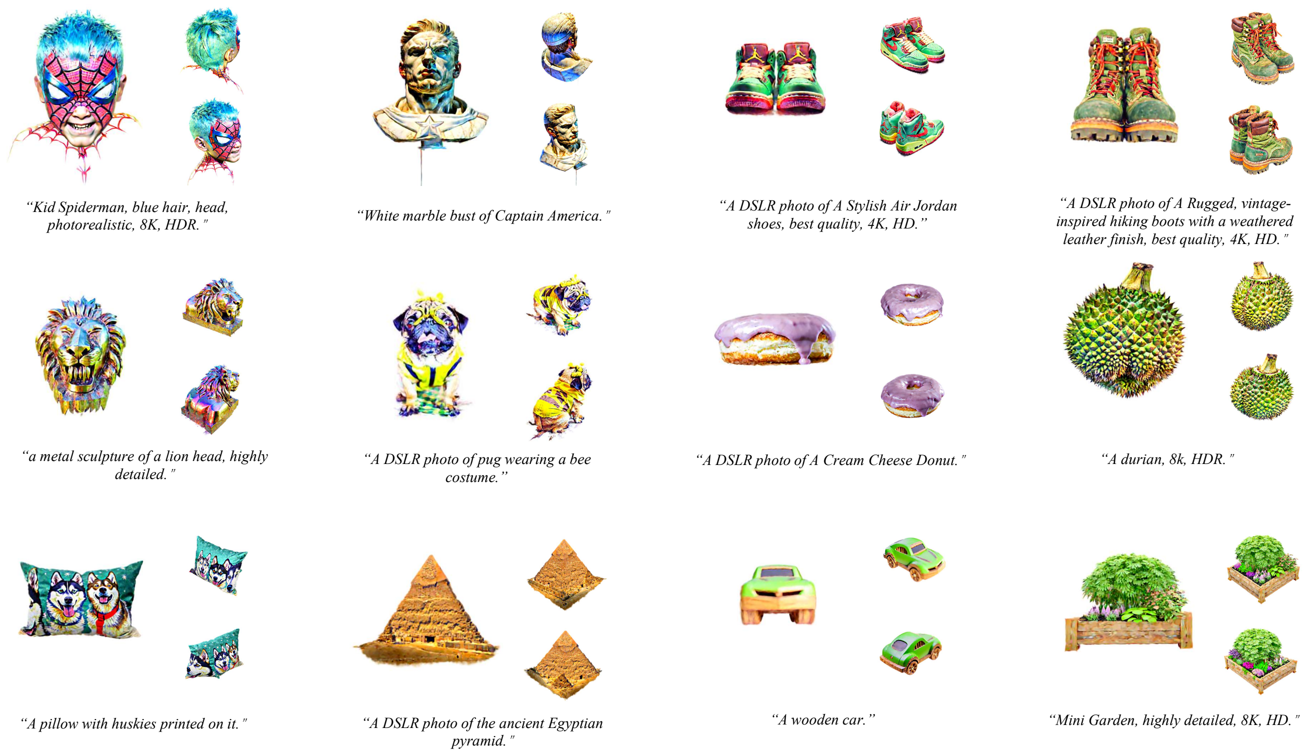
Figure 3. **Comparison of the distillation results and running time.** (a) Distillation results with the naive objective (Eq. (2)) at different $\delta_T = \{10, 25, 50, 100\}$. (b) Distillation results with our proposed ISM objective (Eq. (4)). Please zoom in for details.

3D distillation v.s. image-to-image translation As we discussed in the main paper, ISM follows the basic intuition of SDS which generates pseudo-GTs with 2D diffusion models by referencing x_0. Intuitively, such a process is quite similar to the diffusion-based image-to-image translation tasks that have been discussed in some previous works [7, 10] that intend to alter the input image towards the given condition in a similar manner. In such a perspective, since SDS perturbs the clean sample x_0 with random noises, it encounters the same problem with SDEdit [7] that it struggles to find an ideal timestep t which ensures both the editability of the algorithm while maintaining the basic structure of the input image.

Instead, our ISM adopts DDIM inversion to estimate x_t from x_0 and thus share more common senses with DDIB [10] which mitigates the aforementioned problem. In essence, the DDIB proposes to edit images in a first “DDIM inversion” then “DDIM denoising” paradigm, which can be viewed as building two concatenated Schrödinger bridges [?] that are intrinsically entropy-regularized optimal transport. Similarly, our proposed ISM can be seen as first bridging the distribution of rendered images $q(x_0)$ to the latent space $p_\phi(x_t)$ of pretrained diffusion models ϕ via DDIM inversion, then, we bridge $p_\phi(x_t)$ to the target distribution ($p_\phi(x_0|y)$) via DDIM denoising. Then, we optimize $q(x_0)$ towards $p_\phi(x_0|y)$ along these bridges, which makes our ISM also an entropy-regularized optimal transport objective that is discussed in DDIB [10]. Consequently, our ISM is able to provide better pseudo-GTs for 3D distillation, which elucidates its superior performance over SDS.

3.2. Discussion of η_t

In our main paper, we propose to replace the single-step pseudo-GT estimation adopted in SDS with a multi-step



“Kid Spiderman, blue hair, head, photorealistic, 8K, HDR.”

“White marble bust of Captain America.”

“A DSLR photo of A Stylish Air Jordan shoes, best quality, 4K, HD.”

“A DSLR photo of A Rugged, vintage-inspired hiking boots with a weathered leather finish, best quality, 4K, HD.”

“a metal sculpture of a lion head, highly detailed.”

“A DSLR photo of pug wearing a bee costume.”

“A DSLR photo of A Cream Cheese Donut.”

“A durian, 8k, HDR.”

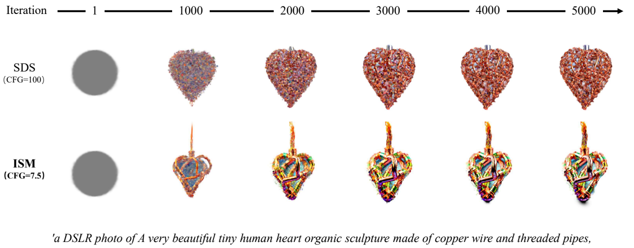
“A pillow with huskies printed on it.”

“A DSLR photo of the ancient Egyptian pyramid.”

“A wooden car.”

“Mini Garden, highly detailed, 8K, HD.”

Figure 4. More results generated by our LucidDreamer framework. Please zoom in for details.



'a DSLR photo of A very beautiful tiny human heart organic sculpture made of copper wire and threaded pipes, very intricate, curved, Studio lighting, high resolution.'

Figure 5. **Comparision of convergence speed.** Our ISM could quickly generate a clear structure (1000 iterations). While SDS failed. Please zoom in for details.

denoising operation. Then, combining the multi-step DDIM inversion with DDIM denoising with the same step size, we formulate our naive objective of 3D distillation as follows:

$$\begin{aligned}\mathcal{L}(\theta) =&\mathbb{E}_c\,[\frac{\omega(t)}{\gamma(t)}||\boldsymbol{x}_0 - \tilde{\boldsymbol{x}}_0^t||^2]\\ =&\mathbb{E}_{t,c}\,[\frac{\omega(t)}{\gamma(t)}||\gamma(t)\underbrace{[\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t,t,y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s,s,\emptyset)]}_{\text{interval scores}} + \eta_t||^2],\end{aligned} \tag{2}$$

where η_t is a bias term depending on the denoising process $x_t \to \tilde{x}_0^t$. For example, when we adopt the step size of the DDIM inversion process $x_0 \to x_t$, δ_T, as the step size of the denoising process, it leads to:

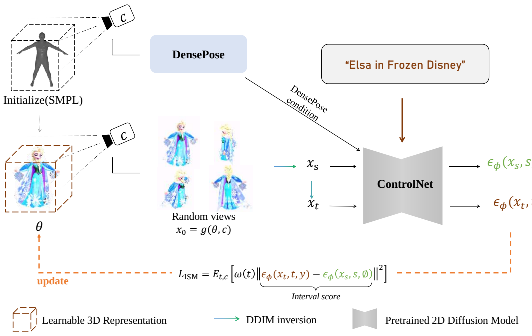

c
Initialize(SMPL)
DensePose
DensePose condition
"Elsa in Frozen Disney"
c
x_s
x_t
ControlNet
$\epsilon_\phi(x_s, s, \emptyset)$
$\epsilon_\phi(x_t, t, y)$
θ
Random views
$x_0 = g(\theta, c)$
update
$L_{\text{ISM}} = E_{t,c}\left[\omega(t)\left\|\epsilon_\phi(x_t,t,y) - \epsilon_\phi(x_s,s,\emptyset)\right\|^2\right]$
Interval score
Learnable 3D Representation
DDIM inversion
Pretrained 2D Diffusion Model

Figure 6. Framework of zero-shot Avatar Generation. In our paper, we first initialize the 3D representation via SMPL [6]. Then, we rely on ControlNet [11] conditioned on DensePose [2] signals provied by a pretrained DensePose predictor to offer more robust supervision.

$$\begin{aligned}\eta_t =& + \gamma(s)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_s, s, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{s-\delta_T}, s-\delta_T, \emptyset)]\\ &- \gamma(s)[\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset)]\\ &+ \gamma(s-\delta_T)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_{s-\delta_T}, s-\delta_T, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{s-2\delta_T}, s-2\delta_T, \emptyset)]\\ &- \gamma(s-\delta_T)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_s, s, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{s-\delta_T}, s-\delta_T, \emptyset)]\\ &+ ...\\ &+ \gamma(\delta_T)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_{\delta_T}, \delta_T, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_0, 0, \emptyset)]\\ &- \gamma(\delta_T)[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_{2\delta_T}, 2\delta_T, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{\delta_T}, \delta_T, \emptyset)].\end{aligned} \tag{3}$$

Despite η_t containing a series of neighboring interval scores with opposite scales that are deemed to cancel each other out, it inevitably leaks interval scores such as $(\gamma(s) - \gamma(s-\delta_T))[\boldsymbol{\epsilon}_\phi(\tilde{\boldsymbol{x}}_s, s, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_{s-\delta_T}, s-\delta_T, \emptyset)]$ and etc depending on the hyperparameters.

Recap that the intuition behind Eq. (2) is to distill update directions from all timestep t. Intuitively, because our algorithm would traverse all t, it is beyond our intention to distill update directions of the other timesteps (i.e., $s, s-\delta_T, ..., \delta_T$) when we focus on t. Furthermore, it is rather time-consuming to compute $\tilde{x}_0^t$ since it requires equivalent steps of estimation for inversion and denoising.

In this paper, we propose to omit η_t from Eq. (2), which leads to our ISM objective, where:

$$\mathcal{L}_{\text{ISM}}(\theta) = \mathbb{E}_{t,c}\left[\omega(t)||\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_s, s, \emptyset)||^2\right]. \quad (4)$$

In Fig. 3, we compare the distillation results of the naive objective versus ISM (with accelerated DDIM inversion). The results indicate that distilling 3D objects with ISM, as opposed to using the naive (2), is not only markedly more efficient but also yields results with enhanced details. While the efficiency gain of ISM is anticipated, our hypothesis is that the observed improvement in details stems from the ISM objective's emphasis on updating directions solely at timestep t. This focus helps avoid the potentially inconsistent update directions at other timesteps $s, s-\delta_T, ..., \delta_T$ while we are not focusing on these timesteps. We will leave the investigation of such a problem to our future work.

3.3. The convergence speed of ISM v.s. SDS

We also compare the convergence speed of ISM and SDS. Specifically, we fixed the noise and hyperparameters and generated 3D assets using SDS and ISM, respectively. As shown in Fig. 5, our proposal (ISM) converges faster than SDS. *e.g.* Our ISM generates a clear and reasonable structure using only 1000 iterations, while SDS is quite noisy at the same stage.

4. Zero-shot Avatar Generation

Our framework is highly adaptable to pose-specific avatar generation scenarios, as depicted in Fig 6, which showcases the detailed workflow. To begin with, we utilize SMPL as an initialization step for positioning the Gaussian point cloud. Subsequently, we employ a pre-trained DensePose model to generate a segmentation map of the human body. This segmentation map serves as a conditional input for the pre-trained ControlNet, where we use an open-source controlnet-seg [11].

5. Details of User Study

In this paper, we conduct a user study to research the user's preferences on the current SoTA text-to-3D methods. In the user study, we ask the participants to compare the 360° rendered video of generated assets from 6 different methods (including our proposal). We provide 28 sets of videos generated by different prompts. We collected 50 questionnaires from the internet and summarized the users' preferences, as shown in the main paper.

6. More visual results

We show additional generated results in Fig. 4. It can be seen that our LucidDreamer could generate 3D assets with high visual quality and 3D consistency.

References

[1] Mohammadreza Armandpour, Huangjie Zheng, Ali Sadeghian, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv*, 2023. 1

[2] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 3

[3] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 1

[4] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. *arXiv*, 2023. 1

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ToG*, 2023. 1

[6] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 3

[7] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv*, 2021. 2

[8] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv*, 2022. 1

[9] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1

[10] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. *arXiv*, 2022. 2

[11] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 3, 4