Supplementary for Advances in 3D Neural Stylization: A Survey

1 Evaluation Implementation Details

1.1 Style Guidance Selection

The 120 evaluation style images for the neural field artistic style transfer are frequently used ones from WikiArt (Phillips & Mackintosh, 2011) and cover different brush strokes and different color hues. We further select six style images from the dataset for evaluating single-style optimization methods, including images with indexes of 3, 14, 17, 19, 71, and 122.

For neural field-based stylization methods with text guidance, we list our test prompts here. We use four scenes from InstructN2N (Haque et al., 2023) including "person-small", "fangzhou-small" "farm-small", and "campsite-small". For the scene "person-small" and "fangzhou-small", we use prompts "Make it look like a Fauvism painting", "Make the man Pixar 3D animation style", "Make the man a marble statue" and "Make it look like a Medieval painting". For the scene "farm-small" and "campsite-small", we use prompts "Make it look like the Namibian desert", "Make it look like it just snowed", "Make it midnight" and "Make it sunset".

For mesh-based stylization methods with image guidance, we use rendered views of the original textured meshes from Objaverse (Deitke et al., 2023) as style references, which are like Figure 1.

For mesh-based stylization methods with text guidance, we construct the prompts following the official guideline of each method respectively, emphasizing the object name.

1.2 Metric Implementations

Clip Score. The CLIP score is a measure of similarity between an image and a text prompt (or two images) based on the CLIP (Contrastive Language-Image Pretraining) model. CLIP computes image embeddings by mapping images into a shared space with text embeddings. The score is calculated by measuring the cosine similarity between the semantic features extracted from the model. We use "openai/clip-vit-base-patch32" as our evaluation CLIP model. Given two inputs I_1 and I_2, their embeddings are denoted as $\mathbf{f}_1$ and $\mathbf{f}_2$, respectively. The CLIP score $S(I_1, I_2)$ is computed as the cosine similarity between these embeddings:

$$S(I_1, I_2) = \frac{\mathbf{f}_1 \cdot \mathbf{f}_2}{\|\mathbf{f}_1\| \|\mathbf{f}_2\|}$$

where $\mathbf{f}_1, \mathbf{f}_2$ are the image embeddings in the shared space, $\|\mathbf{f}_1\|$ and $\|\mathbf{f}_2\|$ are the L2 norms of the respective embeddings, $\cdot$ represents the dot product.

A higher CLIP score between the style reference and the result indicates that they are more similar in terms of semantics, style, etc., as they

Fig. 1: An example rendered view of object "babby buggy" from Objaverse dataset.

have more aligned feature representations in the shared space.

For each generated textured object, we uniformly render four views (front, back, left and right view while slightly above the object) and calculate their average clip score with the corresponding style reference image/text prompt. The overall average clip score of all the results generated by one method is recorded as the clip score of the method. Especially for image-guided mesh stylization methods, we further calculate their in-domain and out-of-domain clip scores, respectively. In-domain texture transfer refers to results that take rendered images of objects within the same category as style references. In contrast, out-of-domain texture transfer involves results that use images of objects from different categories, which typically have significantly different shapes and 3D characteristics.

Clip Variance. Clip variance is proposed in Li et al. (2025) that takes the minimum value of cosine similarity between CLIP features of uniformly sampled views as a metric, which derives from the idea that images of the same object from multiple views have the same semantics. A higher CLIP variance indicates that the generated textures are more consistent. We use the same views rendered for calculating the clip score for each generated textured mesh. The overall average clip variance of all the results generated by one method is recorded as the clip variance of the method.

Temporal consistency. We compute masked warping MSE and LPIPS for short-range and long-range multi-views for evaluating temporal consistency. "Short range" indicates adjacent frame pairs while "long range" indicates frame pairs with 10 intervals except for NeRF-Synthetic object data with 3 intervals. Testing video is default testing sets of 360° videos for NeRF-Synthetic, the spherical trajectory for LLFF, and the short video trajectory for Tanks and Temples. We use the off-the-shelf optical flow estimator RAFT (Teed & Deng, 2020) to estimate optical flow between views, and calculate the according occlusion map. We follow the same formulation of the Warp Error in Lai et al. (2018) as our masked warping MSE. For each frame pair, we compute MSE on non-occluded warped pixels and LPIPS on the masked warped image, and compute average metrics over all sequential frames. Only pixels or features located in non-occluded mask areas are taken into account. Finally, we average per scene class and all scenes as final scores.

Gram Style loss. We use the pre-trained VGG-19 model to extract image features output by $\{relu2\_1, relue3\_1, relu4\_1, relu5\_1\}$ layers. For each layer feature $F^l \in \mathbb{R}^{H^l \times W^l \times d^l}$, $l \in \{1, 2, 3, 4, 5\}$, we get style representation by normalized Gram matrix of flattened feature $V(F^l) \in \mathbb{R}^{d^l \times H^l W^l}$:

$$\frac{V(F^l)V(F^l)^T}{H^l W^l d^l}.$$

The metric of style loss for each image cs and target style image s is calculated by:

$$\mathcal{L}_{style} = \frac{1}{5} \sum_{l \in \{1..5\}} \frac{\|V(F^l(cs))V(F^l(cs))^T - V(F^l(s))V(F^l(s))^T\|_2^2}{H^l W^l d^l}.$$

Both images are resized to 256×256 for evaluation. The final metric numbers are averaged on all evaluated views.

1.3 SNeRF Re-implementatiopn

Following the original paper's implementation, we used Plenoxels (Fridovich-Keil S. and YU A. et al., 2022) official implementation for scene reconstruction, and re-implemented neural style transfer Gatys et al. (2016) for multi-view image style optimization using Pytorch. As described in *SNeRF* (Nguyen-Phuoc et al., 2022), we used the pre-trained VGG-16, and feature outputs of layers $relu1\_1, relu2\_1, relu3\_1, relu4\_1$ for style loss (Gram matrix) and $relu4\_1$ for content loss. We set style and content hyperparameters 1,000,000 and 1, and optimize each image with content image as initialization with L-BFGS optimizer with 500 iterations. We stylized images in 512×512, and then resized stylized images to the original training resolution for scene reconstruction with fixed geometry during the 3D stylization stage. p Since style loss employs Gram matrix loss, for a fair comparison, we also used the off-the-shelf pre-trained AdaIN (Huang & Belongie, 2017) arbitrary style transfer model for multi-view stylization and Plenoxels for reconstruction. Since *SNeRF* is sensitive to multi-view content consistency, in addition to fixing the geometry branch during the 3D stylization stage, during 2D stylization we added the original content feature with the

stylization weight $\alpha = 0.5$ during 2D style transfer at inference and thus preserved more original content. At inference, the AdaIN layer becomes:

$$AdaIN(c,s) = \alpha * (\sigma(F(s))(\frac{F(c) - \mu(F(c))}{\sigma(F(c))}) + \mu(F(s))) + (1-\alpha) * F(c).$$

Likewise, we stylized images in 512×512 and then resized stylized images to the original training resolution for scene reconstruction.

References

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., ... Farhadi, A. (2023). Objaverse: A universe of annotated 3d objects. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 13142–13153).

Fridovich-Keil S. and YU A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A. (2022). Plenoxels: Radiance fields without neural networks. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition.*

Gatys, L., Ecker, A., Bethge, M. (2016). Image style transfer using convolutional neural networks. *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2414–2423).

Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A. (2023). Instruct-nerf2nerf: Editing 3d scenes with instructions. *Proceedings of the ieee/cvf international conference on computer vision.*

Huang, X., & Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. *Proceedings of the ieee international conference on computer vision* (pp. 1501–1510).

Lai, W., Huang, J.-B., Wang, O., Shechtman, E., Yumer, E., Yang, M.-H. (2018). Learning blind video temporal consistency. *Proceedings of the european conference on computer vision (eccv)* (pp. 170–185).

Li, K., Fan, Y., Wu, Y., Sun, Z., Yang, W., Ji, X., ... Chen, J. (2025). Learning pseudo 3d guidance for view-consistent texturing with 2d diffusion. *European conference on computer vision* (pp. 18–34).

Nguyen-Phuoc, T., Liu, F., Xiao, L. (2022). Snerf: stylized neural implicit representations for 3d scenes. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2022*, , https://doi.org/10.1145/3528223.3530107

Phillips, F., & Mackintosh, B. (2011). Wiki art gallery, inc.: A case for critical thinking. *Issues in Accounting Education*, *26*(3), 593–608,

Teed, Z., & Deng, J. (2020). Raft: Recurrent all-pairs field transforms for optical flow. *Computer vision–eccv 2020: 16th european conference, glasgow, uk, august 23–28, 2020, proceedings, part ii 16* (pp. 402–419).