Breathing New Life into 3D Assets with Generative Repainting: Supplementary Materials

Tianfu Wang[1]
Menelaos Kanakis[1]
Konrad Schindler[2]
Luc Van Gool[1,3,4]
Anton Obukhov[1→2]

[1] ETH Zürich
Computer Vision Laboratory

[2] ETH Zürich
Photogrammetry and Remote Sensing

[3] KU Leuven

[4] INSAIT, Sofia

Table S1: Comparison of geometry painting with various methods on ShapeNetSem [3] dataset measured through Kernel Inception Distance (KID ↓) [1] metric with various feature extractors. Standard deviations are given in small font for all values. Lower values are better.

KID ↓ [1] Features Multiplier	Methods	All (11992)	Misc. (2912)	Chair (682)	Lamp (655)	ChstDrw. (503)	Table (416)	Couch (405)	Computer (241)	TV (229)	WallArt (220)	Bed (218)	Cabt. (216)
Inception [1, 2, 3] ×0.01	Orig. texture [8]	1.19 ±0.04	1.18 ±0.09	1.40 ±0.20	2.03 ±0.38	7.89 ±0.79	1.61 ±0.26	4.47 ±0.56	2.76 ±0.47	3.04 ±0.40	1.84 ±0.52	2.72 ±0.36	5.98 ±0.69
	Latent-Paint [8]	1.31 ±0.05	1.37 ±0.11	2.02 ±0.25	1.95 ±0.36	4.52 ±0.54	1.05 ±0.21	4.26 ±0.39	3.84 ±0.35	4.17 ±0.34	3.81 ±0.58	2.14 ±0.32	4.19 ±0.47
	TEXTure [3]	0.75 ±0.04	0.71 ±0.08	1.19 ±0.20	1.61 ±0.35	**1.79** ±0.38	1.97 ±0.32	2.37 ±0.48	2.14 ±0.33	2.36 ±0.31	2.10 ±0.41	1.90 ±0.32	1.54 ±0.29
	Ours	**0.44** ±0.03	**0.38** ±0.06	**0.65** ±0.18	**0.80** ±0.24	1.88 ±0.48	**0.94** ±0.22	**1.14** ±0.30	**1.74** ±0.36	**1.06** ±0.24	**0.53** ±0.16	**1.09** ±0.22	**1.32** ±0.41
CLIP [1, 3] ×0.01	Orig. texture [8]	9.33 ±0.26	8.92 ±0.50	13.1 ±1.38	14.6 ±1.38	21.1 ±1.71	13.0 ±1.19	18.7 ±1.73	8.36 ±1.12	14.3 ±1.41	5.89 ±1.22	14.2 ±1.32	17.7 ±1.76
	Latent-Paint [8]	7.87 ±0.18	7.87 ±0.37	9.36 ±0.57	6.44 ±0.67	17.1 ±1.23	5.47 ±0.48	13.1 ±0.85	12.1 ±0.81	11.2 ±0.67	10.7 ±0.99	10.5 ±0.91	15.7 ±1.20
	TEXTure [3]	3.18 ±0.09	3.04 ±0.21	5.12 ±0.46	4.84 ±0.62	5.81 ±0.67	5.67 ±0.52	4.82 ±0.60	4.68 ±0.43	4.63 ±0.47	3.99 ±0.65	5.61 ±0.50	4.40 ±0.48
	Ours	**1.36** ±0.06	**1.30** ±0.13	**1.69** ±0.32	**1.51** ±0.27	**3.73** ±0.70	**1.90** ±0.34	**2.07** ±0.39	**2.98** ±0.61	**2.14** ±0.48	**0.96** ±0.27	**2.09** ±0.36	**3.12** ±0.62
DINOv2 [1] ×1.0	Orig. texture [8]	2.40 ±0.06	2.36 ±0.15	3.65 ±0.37	4.95 ±0.47	11.9 ±0.98	5.94 ±0.61	14.1 ±1.46	5.42 ±0.85	9.80 ±0.96	3.75 ±0.75	7.56 ±0.69	9.74 ±0.79
	Latent-Paint [8]	1.01 ±0.02	1.04 ±0.05	1.85 ±0.25	2.09 ±0.25	5.51 ±0.51	1.62 ±0.25	6.61 ±0.83	5.49 ±0.54	8.87 ±0.71	3.37 ±0.49	4.26 ±0.50	5.21 ±0.52
	TEXTure [3]	0.53 ±0.02	0.50 ±0.03	1.18 ±0.24	1.76 ±0.27	2.56 ±0.43	1.67 ±0.27	3.32 ±0.62	3.72 ±0.50	3.97 ±0.61	2.17 ±0.44	2.21 ±0.31	2.37 ±0.39
	Ours	**0.38** ±0.01	**0.35** ±0.03	**0.63** ±0.21	**1.07** ±0.21	**2.01** ±0.48	**1.03** ±0.21	**1.90** ±0.64	**3.14** ±0.66	**2.24** ±0.43	**0.86** ±0.19	**1.08** ±0.23	**1.58** ±0.37

S1 Large-Scale Study of ShapeNetSem

Out of 12,288 models in the dataset, we processed 11,992 with all methods. The remaining 296 models either had flat geometry or could not be processed by the Latent-Paint [8] pipeline, TEXTure [12], or both. The failure cases happened most commonly due to the complex geometry not fitting 16GB GPU RAM within the respective method pipeline or failures in `xatlas` texture UV unwrapping module [7]. Our method produced results consistently even on these models, but for a fair comparison, we excluded these models completely.

In addition to the FID [5] evaluation from Tab. 1 of the main paper, we provide a quantitative evaluation of all pipelines on ShapeNetSem with the KID metric [1] in Tab. S1.

The ability of our method to handle complex geometry, low memory footprint, weak dependence on the geometry format or the rendering pipeline, and potentially unknown

Figure S1: **Subjective Study.** Left: User instruction with quality and realism judgment examples; **Right**: Two validation questions "Which one is better?" shared among all subjects to ensure engagement (the right column answers were expected for a pass).

texture coordinates – all these properties make our method a reliable go-to solution for 3D assets revamping.

S2 Subjective User Study

We conducted a limited crowd-sourced perceptual comparison between Latent-Paint [8], TEXTure [12], and our method. The study was based on 50 randomly sampled models from 10 categories, c.f. Tab. S1. Subjects were instructed (Fig. S1, left) to analyze and vote for higher quality and realism after observing a full 360° spin of models painted with a pair of methods, side by side. Each subject submitted 20 votes, plus 2 validation questions with predefined correct answers (Fig. S1, right). 35 subjects participated in our study, of which 29 (83%) passed the validation. 638 votes were collected, ensuring at least 3 votes for every pair, and aggregated into preference scores with the Crowd Bradley-Terry [2] model. The resulting scores were (log-scale, up to additive constant, 95% confidence intervals, higher is better): $S_{\text{Latent}-\text{Paint}} = 0.15_{\pm 0.15}, S_{\text{TEXTure}} = 0.30_{\pm 0.11}, S_{\text{ours}} = \mathbf{1.86}_{\pm 0.13}$. The scores agree with the quantitative results.

S3 ShapeNet Rendering Settings

To facilitate a fair comparison of different methods on the ShapeNetSem dataset [3], we choose the mesh rendering settings in all pipelines such that the output result is adequate for all methods. Notably, TEXTure [12] relies on mesh normals to determine inpainting regions. However, a subset of ShapeNetSem [3] meshes have faces with inappropriately oriented surface normals. For these meshes, directly passing them as input to TEXTure [12] produces corrupt texturing.

To address this issue, we utilize back-face culling of mesh to disable the rendering of mesh

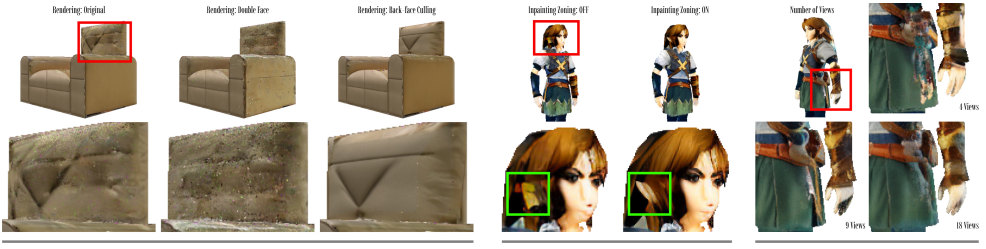

Figure S2: **Left:** Rendering Settings Comparison with TEXTure [12] method. Back-face culling achieves the best result compared with the original[1] and double-face rendering settings. **Middle:** Visibility Score Refinement produces more realistic details on surfaces seen at sharp angles, such as the ear. **Right:** Low number of generated views (4) leads to poor coverage of the input geometry, 18 results in over-smoothing, and 9 is a trade-off.

faces that are oriented away from the camera. We build our method on top of PyTorch3D [11], which provides a built-in implementation of back-face culling. However, since both Latent-Paint [8] and TEXTure [12] pipelines rely on the Kaolin renderer [4], which did not implement back-face culling as of the time of writing, we implemented back-face culling in software. This allowed us to address the rendering discrepancy and level the settings for all pipelines.

We experimented with double-face rendering as an alternative approach to resolving face orientation issues. However, the result of using double-face rendering is worse than that of using back-face culling, as seen in Fig. S2 (left). We suspect this is due to areas of the mesh having overlapping front-facing faces in the double-face rendering setting, thereby negatively affecting texture back-projection in the TEXTure [12] method. Overall, our rendering protocol is chosen to maximize the output quality of the pipelines relying on differential rendering under complex geometry.

S4 Ablation: Inpainting Zoning

Inpainting zoning works in areas of the mesh that face away from the camera in one generated view so that they can be further refined in the subsequent views. Fig. S2 (middle) shows that our refinement scheme brings more details to the areas of the model with challenging visibility constraints.

S5 Ablation: Number of Input Views

We show a qualitative comparison between models painted using various numbers of input views in Fig. S2 (right). With just 4 input views, we find holes and artifacts on the object's surface. With 18 views, the shape is smooth, but the generated color lacks detail. The choice of 9 views achieves the best quality.

S6 Prompt Augmentation

Our method transparently exposes the style guidance functionality of the underlying generative models. It permits prompt augmentation, enabling greater variety in the generated painting

Figure S3: **Style Specifier Guidance**. In the given examples, prompts take the following form: "*A photo of a* {{*material*}} *dresser*" (top) and "*A photo of a* {{*color*}} *dragon*" (bottom).

while preserving 3D consistency. Specifically, our pipeline extends the input object description prompt as follows: "*A photo of a {{modifier}} {{object}}, {{dir}} view*". The "*{{modifier}}*" style specifier term could be the color or the material of the object. In the same vein as text guides image generation models, the texture of our 3D models changes according to the modifier, as shown in Fig. S3.

References

[1] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

[2] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*, 2015.

[4] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A pytorch library for accelerating 3d deep learning research, 2022.

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[6] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. In *ICLR*, 2023.

[7] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics (ToG)*, 21(3):362–371, 2002.

[8] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv:2211.07600*, 2022.

[9] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023.

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[11] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[12] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv:2302.01721*, 2023.

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.