MATLAT: Material Latent Space for PBR Texture Generation

Supplementary Material

In this supplementary document, we present technical discussions (Sec. A), implementation details (Sec. B), and qualitative results (Sec. C).

A. Technical Discussions

In this section, we detail our correspondence-PSNR metric (Sec. A.1), discuss the KID evaluation metric (Sec. A.2), and provide further analyses of VAE prediction types (Sec. A.3), locality regularization (Sec. A.4), and latent distribution mismatch (Sec. A.5).

A.1. Correspondence-PSNR

To quantify the multi-view consistency of the generated PBR material images, we introduced a correspondence-PSNR (c-PSNR) metric in Sec. 4 of the main paper.

Specifically, given a set of N generated view images $\{\mathbf{x}_i\}_{i=1}^N$ and a 3D mesh, for each pixel $u \in \Omega$ in the i-th view, the set of corresponding pixels in the j-th view is denoted as $\mathcal{C}_{i\to j}(u)$. We then compute the MSE (mean squared error) as follows:

$$\text{MSE} = \frac{1}{M}\sum_{i=1}^{N}\sum_{j\neq i}\sum_{\substack{u\in\Omega,\\ v\in\mathcal{C}_{i\to j}(u)}} \|\mathbf{x}_i(u) - \mathbf{x}_j(v)\|^2, \quad (1)$$

where $M = \sum_{i=1}^{N}\sum_{j\neq i}\sum_{u\in\Omega}|\mathcal{C}_{i\to j}(u)|$ is the total number of correspondence pairs. Accordingly, we define the correspondence-PSNR as

$$\text{c-PSNR} = 10\log_{10}(1/\text{MSE}),$$

which measures the discrepancy across all geometric correspondences between views. As shown in Tab. 1, MATLAT achieves higher c-PSNR than methods without CAA, indicating more consistent PBR material images across views.

A.2. Discussion on KID Evaluation

In Tab. 1 of the main paper, our full model achieves the best FID_{CLIP} under the evaluation protocol of [6], while the results on Inception-based KID are more mixed. We hypothesize that Inception features may not faithfully reflect perceptual similarity for rendered material images, as prior works have shown that they are strongly tied to ImageNet semantics and can diverge from human perception [13, 15].

To further examine this effect, we additionally report CLIP-based KID in Tab. 1. Compared with the Inception-based KID results in Tab. 1 of the main paper, CLIP-based KID is more consistent with FID_{CLIP} and with recent work that adopts CLIP-based Fréchet or kernel distances for rendered images [7, 8]. Under this metric, MATLAT consistently outperforms the baselines, including the direct adaptations of prior VAE fine-tuning schemes.

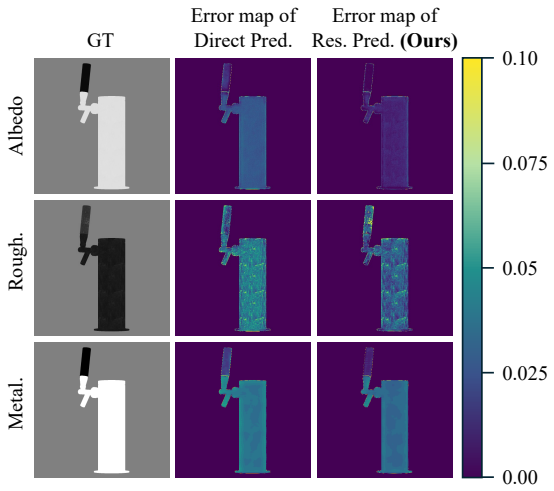

Figure 7. **Direct vs. Residual Prediction.** We show the albedo, roughness, and metallic reconstruction errors at an early training stage (10k iterations) for direct and residual prediction. Residual prediction preserves the pretrained latent representation, achieving superior albedo reconstruction quality (darker regions indicate lower errors).

Methods	Shaded	Albedo
	$\text{KID}_{\text{CLIP}}\downarrow$	$\text{KID}_{\text{CLIP}}\downarrow$
Frozen VAE [8]	2.123	3.547
Res. Pred. + $\mathcal{L}_{\text{reg}}$ (Ours)	<u>1.496</u>	**2.855**
Res. Pred. + $\mathcal{L}_{\text{id}}$ [21]	1.977	3.710
Direct Pred. + $\mathcal{L}_{\text{reg}}$ [14]	1.917	3.254
w/o $\mathcal{L}_{\text{local}}$	2.660	8.820
w/o CAA	**1.424**	<u>3.154</u>

Table 1. **Evaluation on CLIP-based KID.** We additionally measure KID in the CLIP feature space. Our proposed VAE fine-tuning scheme yields superior performance, with both our full model and the variant without CAA consistently outperforming the other baselines.

A.3. Additional Analysis of Direct and Residual Prediction

As discussed in Sec. 3.1 of the main paper, residual prediction can be more effective than direct prediction for VAE fine-tuning. In this section, we present an empirical com-

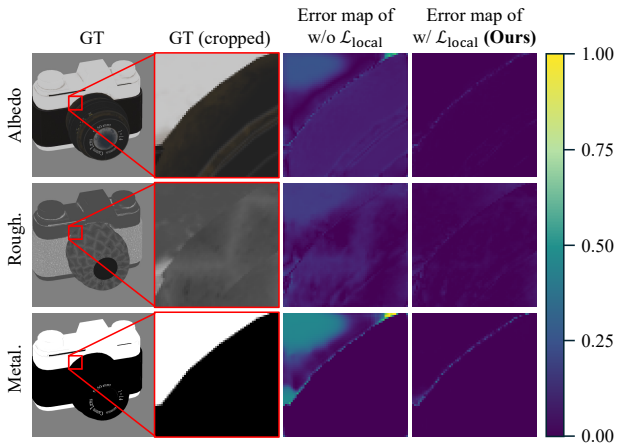

Figure 8. **Visualization of Reconstruction Error Maps on Cropped Patches.** We show albedo, roughness, and metallic reconstruction error maps for MATVAE trained with and without $\mathcal{L}_{\text{local}}$. Applying $\mathcal{L}_{\text{local}}$ significantly reduces patch reconstruction errors, indicating improved latent–image spatial alignment.

parison of residual and direct prediction for VAE during the early stage of fine-tuning, highlighting their optimization behavior and stability.

We present a qualitative comparison in Fig. 7. The left column shows the ground-truth albedo, roughness, and metallic maps. The middle and right columns show reconstruction error maps, computed between the ground-truth images and the outputs of VAEs fine-tuned with direct prediction and residual prediction (Ours), respectively, where brighter regions indicate higher errors. All results are evaluated at the same early fine-tuning step (after 10k iterations) for a fair comparison.

Note that the reconstruction error of the albedo map under residual prediction is smaller than that under direct prediction. This is because residual prediction initializes the final layer weights of the offset encoder to zero, producing zero residuals at the start of training and thereby preserving the original latent representation. In contrast, direct prediction is not constrained to remain aligned with the pretrained latent space and, even after several update steps, already exhibits noticeable degradation in albedo reconstruction quality. While the errors for roughness and metallic maps remain comparable between the two schemes, we empirically observe that the improved stability of residual prediction on albedo images translates into better downstream generation performance, as reported in Tab. 1 of the main paper.

A.4. Additional Analysis of Locality Regularization

As discussed in Sec. 3.2 of the main paper, applying locality regularization $\mathcal{L}_{\text{local}}$ improves latent-image spatial alignment and leads to superior performance when combined with CAA. In Fig. 8, we visualize the reconstruction error of cropped patches using a VAE fine-tuned with and without

Crop range	$\text{FID}^{\text{CLIP}}_{\text{shaded}}$	$\text{FID}^{\text{CLIP}}_{\text{albedo}}$	c-PSNR
$[3, 32]$	**3.083**	**4.599**	**21.934**
$[24, 48]$	3.208	5.027	20.571
$[32, 64]$	3.339	4.719	20.980

Table 2. **Crop-range ablation for locality regularization.** We vary the crop-size range used in $\mathcal{L}_{\text{local}}$ on a 64×64 latent grid. Smaller crop ranges yield the best overall performance, while all cropped settings outperform training without cropping.

$\text{SWD}(\mathbf{z}, \mathbf{z}_{\text{a}})$	$\text{SWD}(\mathbf{z}, \mathbf{z}_{\text{rm}})$	$\text{SWD}(\mathbf{z}, \mathbf{z}_{\text{ours}})$
0.539	0.897	0.560

Table 3. **Sliced Wasserstein distances to the pretrained diffusion prior.** Lower values indicate that the encoded latents are better aligned with the pretrained latent space.

$\mathcal{L}_{\text{local}}$ to further analyze the effect of the regularizer. Specifically, we show cropped patches from the albedo, roughness, and metallic images, together with their reconstruction error maps, $\|\mathbf{x} - \mathcal{D}(\mathcal{E}(\mathbf{x}))\|^2$, computed at an 8×8 latent resolution for VAEs fine-tuned without and with $\mathcal{L}_{\text{local}}$, where brighter regions indicate higher errors.

Note that applying $\mathcal{L}_{\text{local}}$ significantly reduces the reconstruction error, indicating that the learned latent representation achieves improved latent–image spatial alignment. In contrast, removing $\mathcal{L}_{\text{local}}$ disrupts the spatial alignment, indicating that the fine-tuned encoder entangles information across spatially distant latent tokens. Consequently, the latent–pixel mapping is no longer enforced to be spatially local, and applying CAA propagates information across unrelated regions, leading to degraded multi-view consistency and reduced overall performance, as observed in Tab. 1 of the main paper.

We further study the sensitivity of locality regularization to the crop size used during training. In our default setting on a 64×64 latent grid, cropping is applied with 50% probability, using a square crop whose size is uniformly sampled from $[3, 32]$ at a random location. In Tab. 2, we compare this setting with larger crop ranges, $[24, 48]$ and $[32, 64]$. All crop ranges outperform training without cropping, while the default range $[3, 32]$ achieves the best overall performance. This suggests that smaller patches provide a stronger and more robust locality prior, which is well aligned with CAA operating on local correspondence windows.

A.5. Analysis of Latent Distribution Mismatch

As discussed in Sec. 3.1 of the main paper, a key challenge in adapting an RGB-pretrained diffusion model to PBR texture generation is the latent distribution mismatch caused by additional material channels. To mitigate this, MATLAT

regularizes the learned latents toward the pretrained latent space.

Tab. 3 reports sliced Wasserstein distances between the pretrained diffusion prior and encoded PBR latents. We estimate the SD3.5-medium latent prior from 5K samples generated using the text prompts paired with our PBR dataset, and compare it with the latents obtained by encoding the corresponding material maps using MATVAE and the pretrained VAE.

The pretrained VAE latent of albedo, $\mathbf{z}_{\mathrm{a}}$, remains relatively close to the prior, $\mathbf{z}$, while the latent of roughness+metallic, $\mathbf{z}_{\mathrm{rm}}$, exhibits a larger discrepancy. This supports our claim that additional PBR channels induce a mismatch with the pretrained diffusion latent space. In contrast, MATVAE reduces this mismatch substantially, suggesting that $\mathcal{L}_{\mathrm{reg}}$ helps preserve alignment with the pretrained latent space while incorporating material information.

B. Implementation Details

In this section, we present implementation details for MATVAE (Sec. B.1) and MATLAT (Sec. B.2), along with the data preprocessing pipeline (Sec. B.3).

B.1. MATVAE

Architecture. Our MATVAE is initialized from the pretrained VAE of `Stable Diffusion 3.5-Medium`. We freeze the original encoder $\mathcal{E}_{\mathrm{pre}}$ and decoder $\mathcal{D}_{\mathrm{pre}}$, and introduce an offset encoder $\mathcal{E}_{\mathrm{res}}$ to adapt the latent space to 5-channel PBR inputs.

The offset encoder $\mathcal{E}_{\mathrm{res}}$ shares the same architecture as $\mathcal{E}_{\mathrm{pre}}$ except for the first convolution layer, which is modified to accept 5-channel inputs $[\mathbf{a}, \mathbf{r}, \mathbf{m}]$. All intermediate layers of $\mathcal{E}_{\mathrm{res}}$ are initialized from the corresponding weights of $\mathcal{E}_{\mathrm{pre}}$, while the output layer is zero-initialized such that $\boldsymbol{\mu}_{\mathrm{res}} = \mathbf{0}$ and $\boldsymbol{\sigma}_{\mathrm{res}} = \mathbf{1}$ at initialization. The decoder $\mathcal{D}$ extends the final convolution of $\mathcal{D}_{\mathrm{pre}}$ to output 5 channels; all other layers are copied from the pretrained decoder.

Training Configurations. We train MATVAE with the loss function in Eq. 8 of the main paper. We use the Adam optimizer with learning rate 3×10^{-5} and batch size 8, training for 200k iterations on 8 NVIDIA RTX Pro6000 GPUs for 60 hours. The loss weights are set to $\lambda_{\mathrm{local}} = 3$, $\lambda_{\mathrm{KL}} = 10^{-6}$, $\lambda_{\mathrm{disc}} = 0.02$, and $\lambda_{\mathrm{reg}} = 3 \times 10^{-9}$. For locality regularization, we randomly crop a square region covering 0.2%–25% of the image area with probability 0.5.

B.2. Diffusion Model Fine-Tuning

Our diffusion model is built on STABLE DIFFUSION 3.5-MEDIUM with MMDiT [5] backbone. For each joint attention layer, we introduce a parallel correspondence-aware attention (CAA) branch that attends over geometrically corresponding pixels across views, using the precomputed 3D correspondences described in Sec. 3.2 of the main paper. The CAA branch uses the same base projection weights as the original joint attention layers, while introducing additional LoRA layers [10] with rank 32. The CAA output is added residually to the original attention output.

Additionally, to align the generated images with the input geometry, we follow previous works [1, 8, 12] and condition the diffusion model on rendered position and normal maps from the corresponding camera view. These geometric features are concatenated with the noisy latent to form the diffusion input.

Training Configurations. We optimize MATLAT using the Conditional Flow Matching [16] objective defined in Eq. 9 of the main paper, where timesteps are sampled from the same logit-normal distribution as in SD3 [5]. We use the Adam optimizer with learning rate 5×10^{-5} and batch size 4, training for 20k iterations on 8 NVIDIA RTX Pro6000 GPUs, which takes about 24 hours in total.

Inference. At inference time, we generate multi-view images from $N = 6$ canonical views (front, back, left, right, top, and bottom) using the Euler sampler with 30 steps and a Classifier-Free Guidance [9] scale of 4.0. Given the generated PBR latent samples, we first decode them into 5-channel material images and then convert the multi-view PBR outputs into a final UV texture map following the pipeline of MVAdapter [12]: we upscale the images, unproject them into UV space using the given camera poses and mesh, and finally perform inpainting in UV space to fill occluded regions.

Baseline Implementations. For MeshGen [2] and MaterialMVP [8], which operate in an image-conditional setting, we use STABLE DIFFUSION 2-DEPTH to generate the reference images used as conditional inputs. All other baselines are evaluated using their official implementations with default configurations.

B.3. Data Processing

We curate $40,851$ meshes with PBR textures from Objaverse-XL [3], holding out 128 meshes for evaluation. For each mesh, we render material images from 26 fixed camera views surrounding the object. We then render albedo, roughness, and metallic images, along with normal and position maps for conditioning, from each view. During diffusion training, we randomly sample 6 of these 26 views per iteration to construct the multi-view training batch. Additionally, for text conditioning, we use captions from Bootstrap3D [18] when available; otherwise, we use captions from Cap3D [17].

During evaluation, we render the PBR-textured assets under environment lighting using 785 HDR environment

Figure 9. **Relighting results.** Example objects generated by MATLAT and rendered under four different environment maps.

maps from Poly Haven. For each image, we then randomly sample an environment map and apply the same rendering setup to both the generated and ground-truth images to ensure a fair comparison.

C. Additional Qualitative Results

In this section, we present representative qualitative example for the ablation studies (Sec. C.1), baseline comparisons (Sec. C.2), and generated textures of MATLAT (Sec. C.3). **Please refer to the project page for more results and video demonstrations:** https://matlat-proj.github.io/

C.1. Additional Qualitative Results for Ablation Studies

Fig. 10 extends the ablation study in Sec. 4.2 of the main paper with additional qualitative examples. We observe that Frozen VAE produces unrealistic material appearance, as evidenced by the overly shiny metallic shoe surface. Additionally, Res. Pred. + $\mathcal{L}_{\text{id}}$ and Direct Pred. + $\mathcal{L}_{\text{reg}}$ miss fine details such as *metal* buckles, whereas MATLAT generates PBR textures with correct and realistic material appearances.

C.2. Additional Qualitative Comparisons with Previous Methods

Fig. 11 presents qualitative comparisons against representative baselines, including models trained from scratch (MeshGen [2], TexGaussian [19]), SDS-based optimization methods (Paint-it [20], DreamMat [22], FlashTex [4]), and prior multi-view diffusion models (MaterialAnything [11], MaterialMVP [8]). Models trained from scratch often produce textures with suboptimal quality and exhibit weak alignment with the input text prompts. Additionally, SDS-based methods lack fine details and tend to generate textures with oversaturated colors. In contrast, our method MATLAT generates high-quality textures with realistic material properties.

C.3. Additional Qualitative Results of MATLAT

Extending the results in Fig. 6 of the main paper, we present additional PBR textures generated by our method MATLAT with diverse prompts, meshes, and environment maps in Fig. 12. Note that our method generates physically plausible material properties that are well aligned with both the object characteristics and the provided text prompt. These results demonstrate that our pipeline exhibits strong generalization across diverse object categories and material types.

Fig. 9 further presents relighting examples, where the same object is rendered under four different environment maps. The consistent appearance changes across lighting conditions demonstrate that MATLAT produces coherent PBR textures with plausible illumination-dependent effects.

Figure 10. **Additional Qualitative Results of Ablation Studies.** Extended qualitative results of the ablation studies presented in Fig. 5 of the main paper: Frozen VAE, Res. Pred. + $\mathcal{L}_{\text{id}}$, Direct Pred. + $\mathcal{L}_{\text{reg}}$, and Res. Pred. + $\mathcal{L}_{\text{reg}}$ (Ours). **Best viewed in video.**

Figure 11. **Additional Qualitative Comparisons with Previous Methods.** Each column shows the shaded output rendered under identical lighting conditions. Our method produces high-quality, physically plausible materials with superior text-visual alignment and detail preservation. **Best viewed in video.**

Figure 12. **Additional Qualitative Results of MatLat.** Gallery of assets textured by MatLat for various text prompts and environment maps. **Best viewed in video.**

References

[1] Raphael Bensadoun, Yanir Kleiman, Idan Azuri, Omri Harosh, Andrea Vedaldi, Natalia Neverova, and Oran Gafni. Meta 3d texturegen: Fast and consistent texture generation for 3d objects. *arXiv preprint arXiv:2407.02430*, 2024. 3

[2] Zilong Chen, Yikai Wang, Wenqiang Sun, Feng Wang, Yiwen Chen, and Huaping Liu. Meshgen: Generating pbr textured mesh with render-enhanced auto-encoder and generative data augmentation. In *CVPR*, 2025. 3, 4

[3] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *NeurIPS*, 2023. 3

[4] Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. Flashtex: Fast relightable mesh texturing with lightcontrolnet. In *ECCV*, 2024. 4

[5] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 3

[6] Fan Fei, Jiajun Tang, Fei-Peng Tian, Boxin Shi, and Ping Tan. Pacture: Efficient pbr texture generation on packed views with visual autoregressive models. *arXiv preprint arXiv:2505.22394*, 2025. 1

[7] Yifei Feng, Mingxin Yang, Shuhui Yang, Sheng Zhang, Jiaao Yu, Zibo Zhao, Yuhong Liu, Jie Jiang, and Chunchao Guo. Romantex: Decoupling 3d-aware rotary positional embedded multi-attention network for texture synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17203–17213, 2025. 1

[8] Zebin He, Mingxin Yang, Shuhui Yang, Yixuan Tang, Tao Wang, Kaihao Zhang, Guanying Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, and Wenhan Luo. Materialmvp: Illumination-invariant material generation via multi-view pbr diffusion. In *ICCV*, 2025. 1, 3, 4

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[11] Xin Huang, Tengfei Wang, Ziwei Liu, and Qing Wang. Material anything: Generating materials for any 3d object via diffusion. In *CVPR*, 2024. 4

[12] Zehuan Huang, Yuan-Chen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. In *ICCV*, 2025. 3

[13] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9307–9315. IEEE, 2024. 1

[14] Akshay Krishnan, Xinchen Yan, Vincent Casser, and Abhijit Kundu. Orchid: Image latent diffusion for joint appearance and geometry generation. In *ICCV*, 2025. 1

[15] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[16] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 3

[17] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. In *NeurIPS*, 2023. 3

[18] Zeyi Sun, Tong Wu, Pan Zhang, Yuhang Zang, Xiaoyi Dong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Bootstrap3d: Improving multi-view diffusion model with synthetic data. In *ICCV*, 2025. 3

[19] Bojun Xiong, Jialun Liu, Jiakui Hu, Chenming Wu, Jinbo Wu, Xing Liu, Chen Zhao, Errui Ding, and Zhouhui Lian. Texgaussian: Generating high-quality pbr material via octree-based 3d gaussian splatting. In *CVPR*, 2025. 4

[20] Kim Youwang, Tae-Hyun Oh, and Gerard Pons-Moll. Paint-it: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. In *CVPR*, 2024. 4

[21] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. In *SIGGRAPH*, 2024. 1

[22] Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, and Xiaogang Jin. Dreammat: High-quality pbr material generation with geometry- and light-aware diffusion models. In *SIGGRAPH*, 2024. 4