MatPedia: A Universal Generative Foundation for High-Fidelity Material Synthesis

Supplementary Material

7. More Details

7.1. Training Details

3D VAE Training. We train the pretrained video VAE [49] on our PBR dataset (Sec. 4.4), where each sample is represented as a five-frame sequence (1024×1024 resolution) comprising one shaded RGB view and four parameter maps. During training, the encoder is frozen and only the decoder is updated. We use the AdamW optimizer [29] with a learning rate of 5×10^{-5}, a batch size of 1, pixel loss weight $\lambda_{\text{pix}} = 10$, and perceptual loss weight $\lambda_{\text{perc}} = 1$. Training is conducted in FP16 precision for 10,000 steps over 1 day on 4 nodes, each with $8\times$ NVIDIA GPUs (total 32 GPUs). We fine-tune the decoder on only the paired RGB-PBR subset of our dataset (Sec. 4.4) at 1024×1024 resolution for 10K steps.

DiT training with LoRA. For the generative backbone, we train the Wan 2.2 text-and-image-to-video DiT [49] using Low-Rank Adaptation (LoRA) [19] with rank 128, applied to the projection layers in the multi-head self-attention and the first and last linear layers in the feed-forward network. We train separate LoRA modules for the three tasks described in Sec. 4.3 (text-to-material generation, image-to-material generation, and material decomposition), each using the mixed training strategy that combines full PBR sequences with planar render-only samples. Training is performed at 1024×1024 resolution with a batch size of 16 and a learning rate of 1×10^{-4}, running for 200,000 steps per task.

7.2. Dataset Details

To support diverse material generation and robust material recovery, we construct **MatHybrid-410K**, a large-scale hybrid dataset with two complementary subsets: an RGB appearance dataset and a complete PBR material dataset.

We collect approximately 50,000 planar material images from two sources: (1) procedurally generated flat surfaces using Gemini 2.5 Flash Image [7], and (2) real-world planar material photographs from public repositories. For the procedurally generated subset, we ensure that the materials are rendered without geometric distortion and under minimal lighting variation, so that the captured appearance is purely intrinsic. The generation process is guided by descriptive prompts that specify material type, color composition, pattern structure, and texture properties. An example prompt is shown below:

This material is a porcelain tile that replicates polished limestone, with a soft beige color and faint fossil markings. The pattern is subtle and micro-veined. The tiles are arranged in a random pattern. This is a ceramic porcelain material. It is used to achieve a French country or rustic elegant look indoors and out. delight, plane material map, continuous in all directions.

Each generated image is paired with a textual description produced by Qwen2.5-VL-72B-Instruct [2], enabling text-to-material generation. The captioning process uses a structured prompt to ensure coverage of visual and semantic attributes, as shown below:

You are a material expert, and need to accurately describe the details of flat metal materials: 1. Color: Main color (including lightness/saturation), proportion of auxiliary colors, presence of gradients and transition methods. 2. Pattern: Type of pattern (striped/polka-dotted/floral, etc.), distribution density, arrangement rules (symmetrical/random). 3. Texture: Surface tactile characteristics (matte/mirror/fabric-like, etc.), texture thickness, light reflection properties. 4. Material category: Speculate on the specific material. 5. Applicable scenarios: 3 most matching application scenarios (such as desktop/wall surface/clothing fabric). Output format: Directly output the descriptive text.

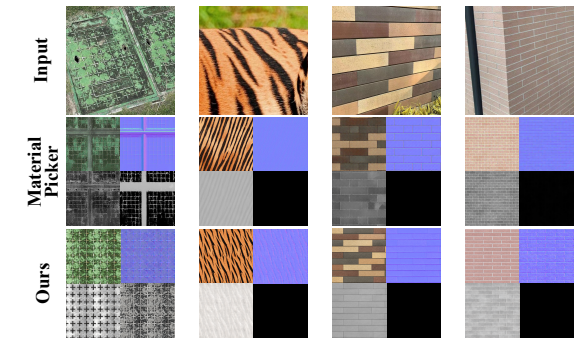
This RGB-only subset provides diverse appearance patterns for training without requiring paired PBR maps.

8. More Results

8.1. Text-to-Material Generation

We provide additional qualitative comparisons with MatFuse[46] in Fig. 11 and Fig. 12. For all examples, both methods are conditioned on the same text prompts and we show the generated PBR maps (Basecolor, Normal, Roughness, Metallic/Specular) and the corresponding render. Fig. 11 includes stone, blue–white porcelain pattern, wooden floor, and red brick wall. Our method produces clearer patterns, more coherent structure, and more plausible roughness/metallic distributions, leading to renders that better match the prompts (e.g., regular brick and mortar layout, sharper porcelain motifs, and realistic wood grain). Fig. 12 covers fur, damask fabric, a space-themed pattern, and rusted metal. In these cases, our results exhibit more faithful semantic details (such as hair-like streaks, repetitive textile ornaments, well-defined planets and stars, and spatially varying rust) and more consistent material attributes than MatFuse. These visual observations are aligned with

Figure 9. Qualitative comparison of image-conditioned PBR generation.



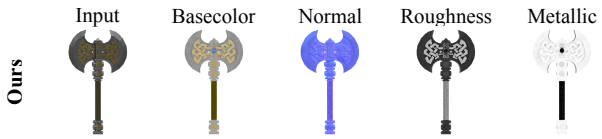


Figure 10. Qualitative example of object-level material decomposition from a single RGB image. From left to right: input rendering, and our predicted basecolor, normal, roughness, and metallic maps for a weapon model. The decomposed PBR maps are clean and structurally consistent, capturing fine geometric details and the spatial variation between metallic and non-metallic regions, demonstrating that our method generalizes well from planar materials to full 3D objects.

the quantitative improvements reported in Table 1.

8.2. Image-to-Material Generation

We present additional qualitative comparisons with MatFuse[46] and Material Palette[28] in Fig. 13, Fig. 14, and Fig. 15. Examples cover bricks, leather, wood, fabrics, cracked soil, and outdoor pavements. Across all cases, our method recovers undistorted basecolor maps and clean normal/roughness maps that remove perspective and shading while preserving fine structural details, leading to consistent, physically plausible renders. MatFuse often produces blurrier detail and less accurate roughness, and Material Palette frequently carries over illumination and geometric distortions from the input, in line with the quantitative trends in Table2.

we provide new comparisons with MaterialPicker in Fig. 9 on the image-to-material task. Given distorted input images, our method successfully extracts the dominant planar region and synthesizes visually coherent tileable patterns consistent with the input (as illustrated in column 3) with more accurate PBR properties (e.g., metallic maps).

8.3. Material Decomposition

Planer Material Decomposition. Fig. 16 presents additional qualitative comparisons for material decomposition from single RGB images against Material Palette[28] and RGB↔X[56]. We show diverse categories including brick, marble, and rusty metal. Consistent with Table 3 and Fig. 6, our method produces basecolor maps with clean albedo and reduced noise, normal maps with sharp yet stable geometric details, and roughness maps with coherent structural patterns, resulting in renders that closely match the input appearance. Both baselines exhibit noticeable artifacts such as over-sharpened normals, residual shading in basecolor, and inconsistent roughness, especially for high-frequency and specular textures.

Object-level Material Decomposition. To assess the generalization ability of our method beyond planar material samples, we further apply it to object-level inputs with complex geometry and spatially varying appearance. Fig. 10 shows a representative example on a weapon asset rendered from a single RGB image. Our approach successfully decomposes the input into physically meaningful PBR components, including basecolor, normal, roughness, and metallic maps. The recovered basecolor is largely free of shading and specular highlights, while the normal map preserves fine-scale engravings and edge details without introducing excessive noise. In addition, the roughness and metallic maps are spatially coherent and accurately reflect the distribution of glossy metallic parts versus more diffuse regions along the handle and decorations. These results indicate that our model, trained primarily on planar materials, can robustly generalize to full objects and produce high-quality material decompositions that are directly usable in standard rendering pipelines.

On the Necessity of Asymmetric RGB-PBR Design. Integrating abundant RGB data overcomes the scalability bottleneck and limited diversity of existing PBR-only datasets. To incorporate RGB, we design an asymmetric 3D VAE mirroring video compression: RGB, containing primary appearance, serves as the "first frame"; PBR maps encode only complementary material properties conditioned on RGB, compressing into one additional latent instead of four. In contrast, **symmetric encoding** (3D VAE with all maps as image frames, or 2D VAE per map) requires 2.5× tokens (limiting resolution) and additional attention for multi-channel consistency (IntrinsicX). Table 5 validates this on intrinsic decomposition: asymmetric outperforms symmetric across all reconstruction metrics.

Table 5. Ablation study on asymmetric RGB-PBR design and LoRA rank.

Design	MSE ↓	SSIM ↑	LPIPS ↓	Rank	CLIP ↑	FID ↓
Symmetric	0.018	0.842	0.513	32	0.269	1.68
Asymmetric	**0.008**	**0.935**	**0.423**	128	**0.283**	**1.31**
–	–	–	–	256	0.277	1.38

LoRA Rank Choice and Runtime Analysis. **1) Rank:** We selected rank 128 empirically—lower ranks (32) lose high-frequency details, higher ranks (256) yield diminishing returns (see Table 5). **2) Runtime:** 1024^2 generation takes 20s (+8s for 4K upsampling), significantly faster than MaterialPalette ($\sim$5min) and ControlMat ($\sim$350s).

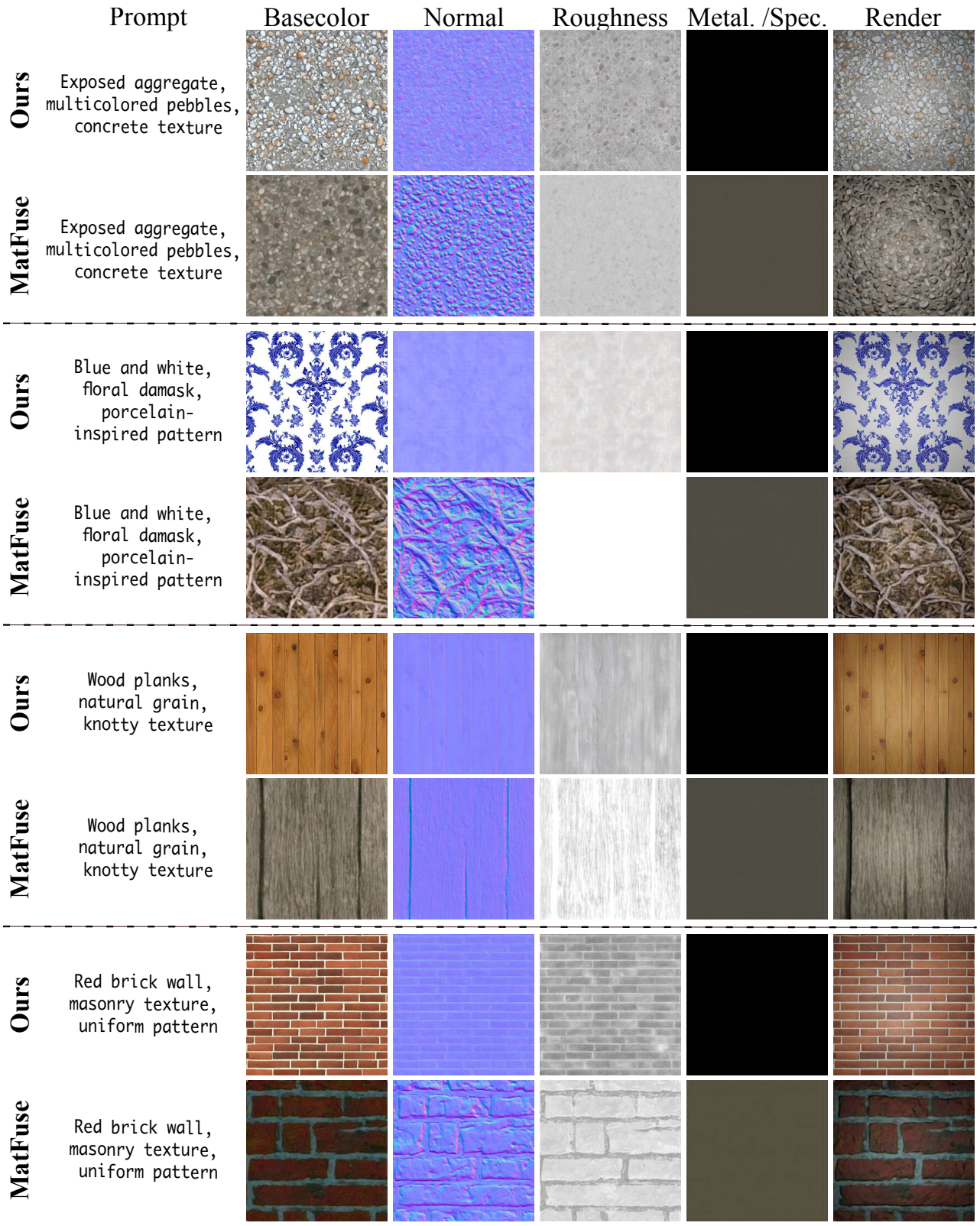


Figure 11. Qualitative comparison of text-conditioned PBR material generation among our method, MatFuse [46]. For each prompt, we show the generated PBR maps (Basecolor, Normal, Roughness, Metallic) followed by a render view under point-light illumination. We note that MatFuse generates a specular map rather than a metallic map.

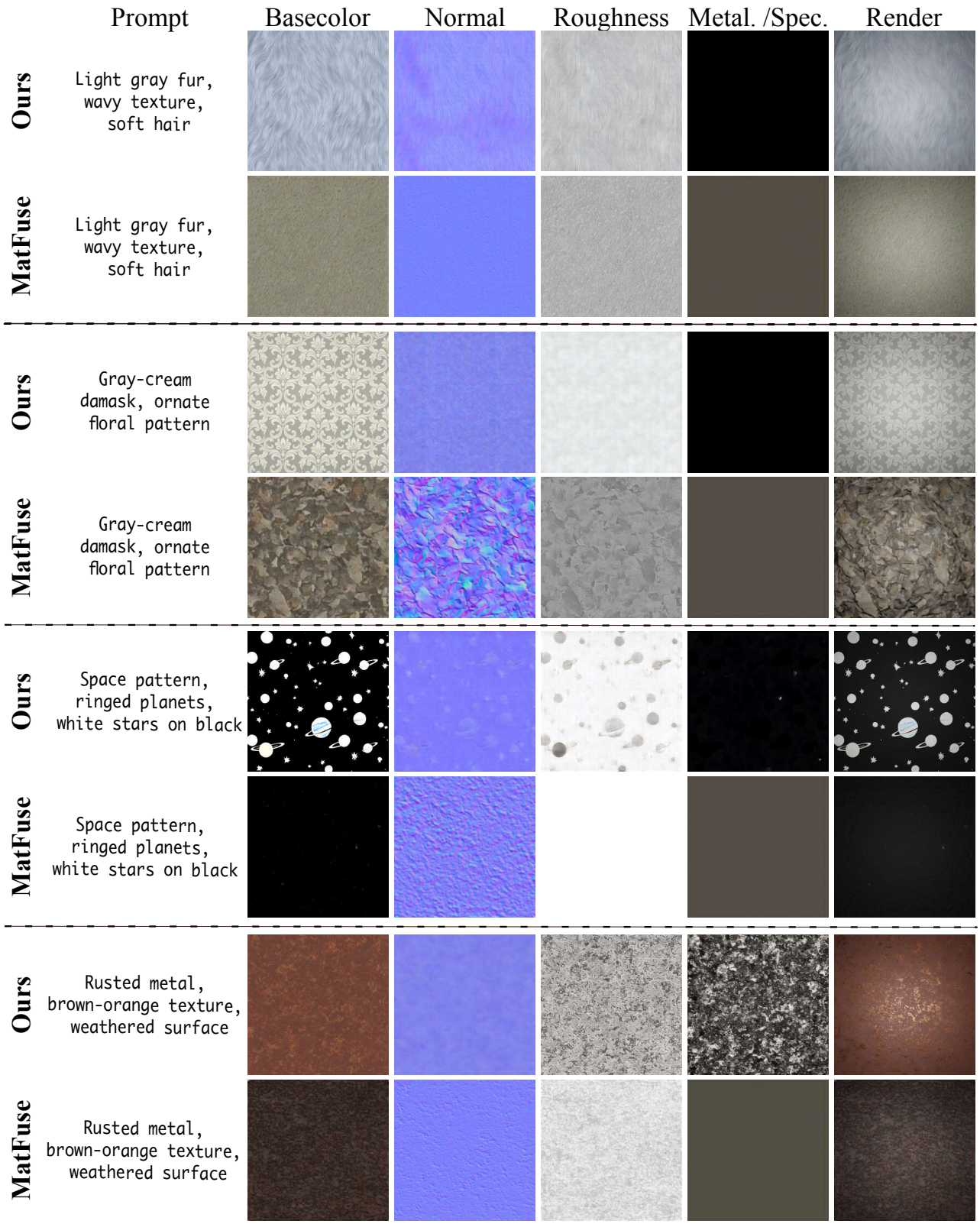


Figure 12. Qualitative comparison of text-conditioned PBR material generation among our method, MatFuse [46]. For each prompt, we show the generated PBR maps (Basecolor, Normal, Roughness, Metallic) followed by a render view under point-light illumination. We note that MatFuse generates a specular map rather than a metallic map.

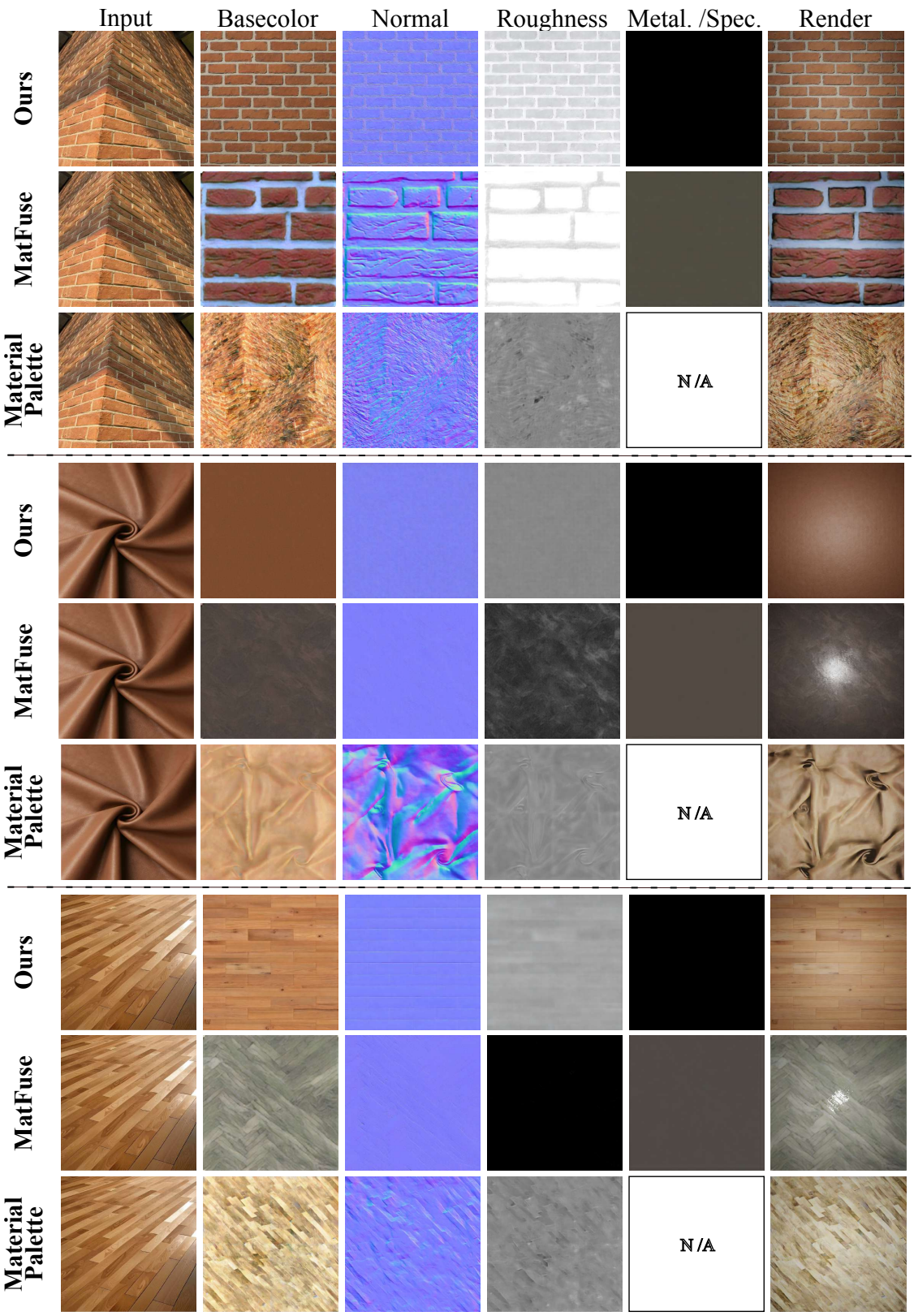


Figure 13. Qualitative comparison of image-conditioned PBR generation. For each sample, the first column shows the distorted input image (cropped from the scene), and the second to last columns present the generated material maps together with a rendering under point-light illumination. Our method produces geometrically flattened and artifact-free maps, while MatFuse shows reduced roughness fidelity and Material Palette retains geometric distortions from the input.

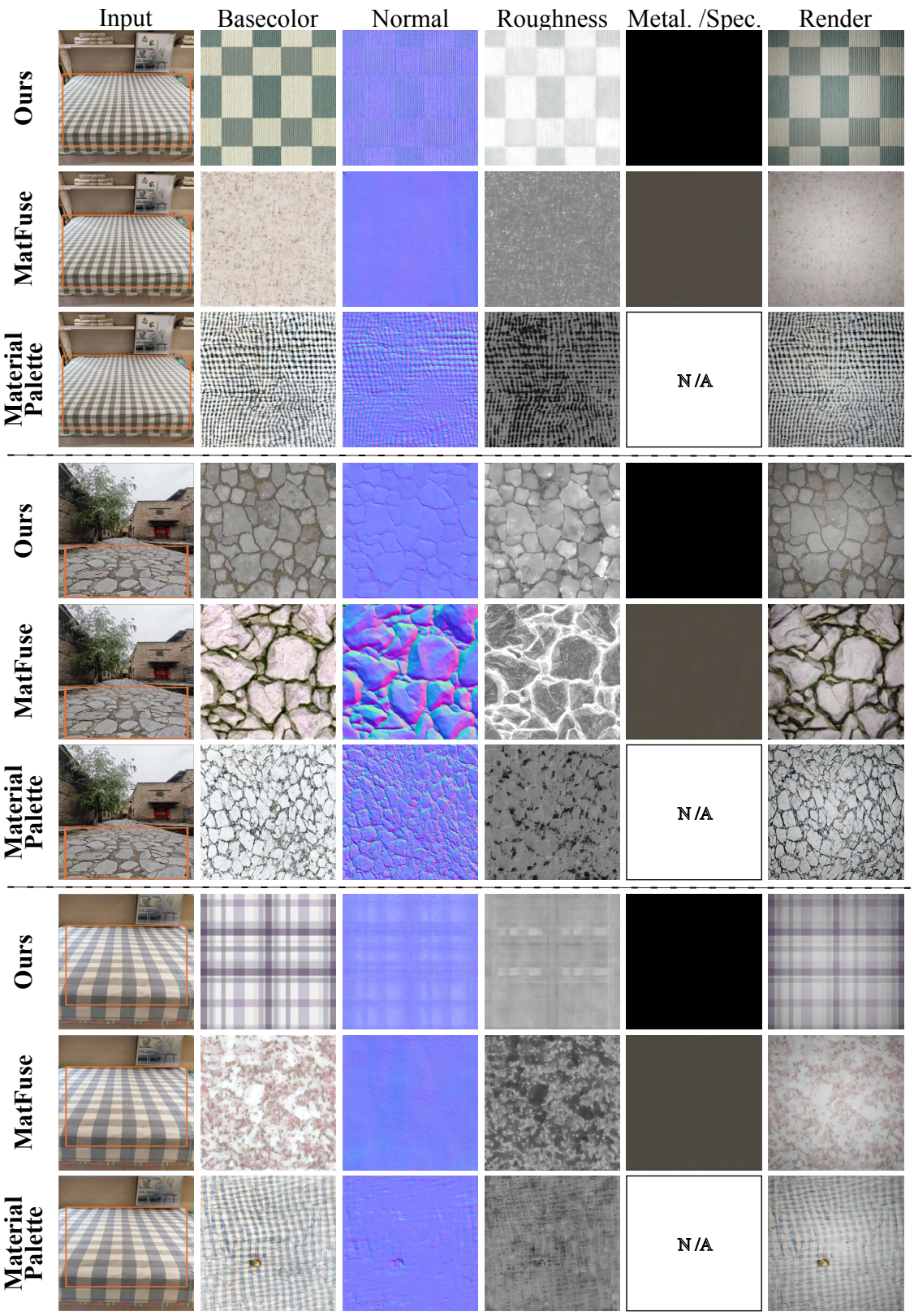


Figure 14. Qualitative comparison of image-conditioned PBR generation. For each sample, the first column shows the distorted input image (cropped from the scene), and the second to last columns present the generated material maps together with a rendering under point-light illumination. Our method produces geometrically flattened and artifact-free maps, while MatFuse shows reduced roughness fidelity and Material Palette retains geometric distortions from the input.

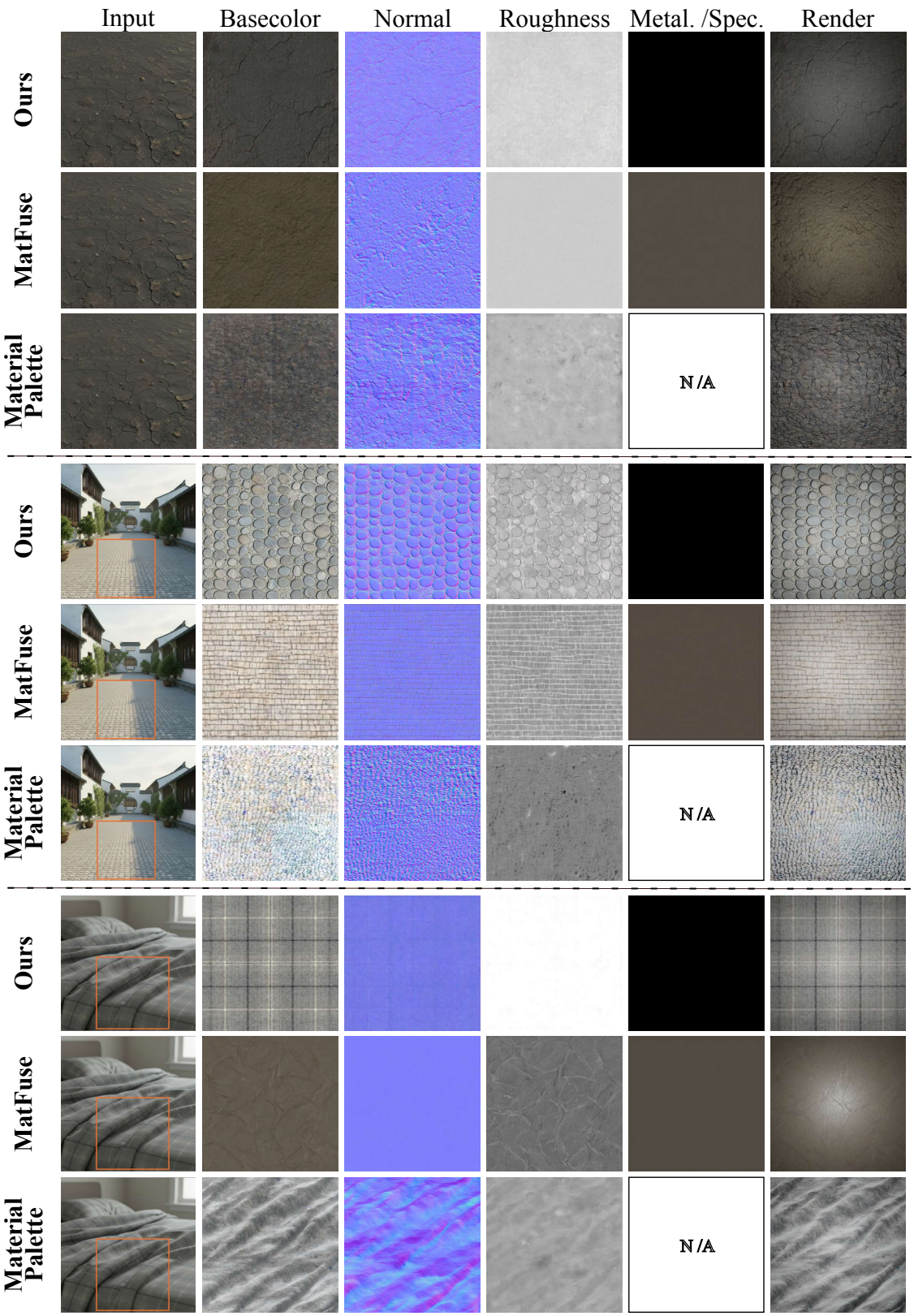


Figure 15. Qualitative comparison of image-conditioned PBR generation. For each sample, the first column shows the distorted input image (cropped from the scene), and the second to last columns present the generated material maps together with a rendering under point-light illumination. Our method produces geometrically flattened and artifact-free maps, while MatFuse shows reduced roughness fidelity and Material Palette retains geometric distortions from the input.

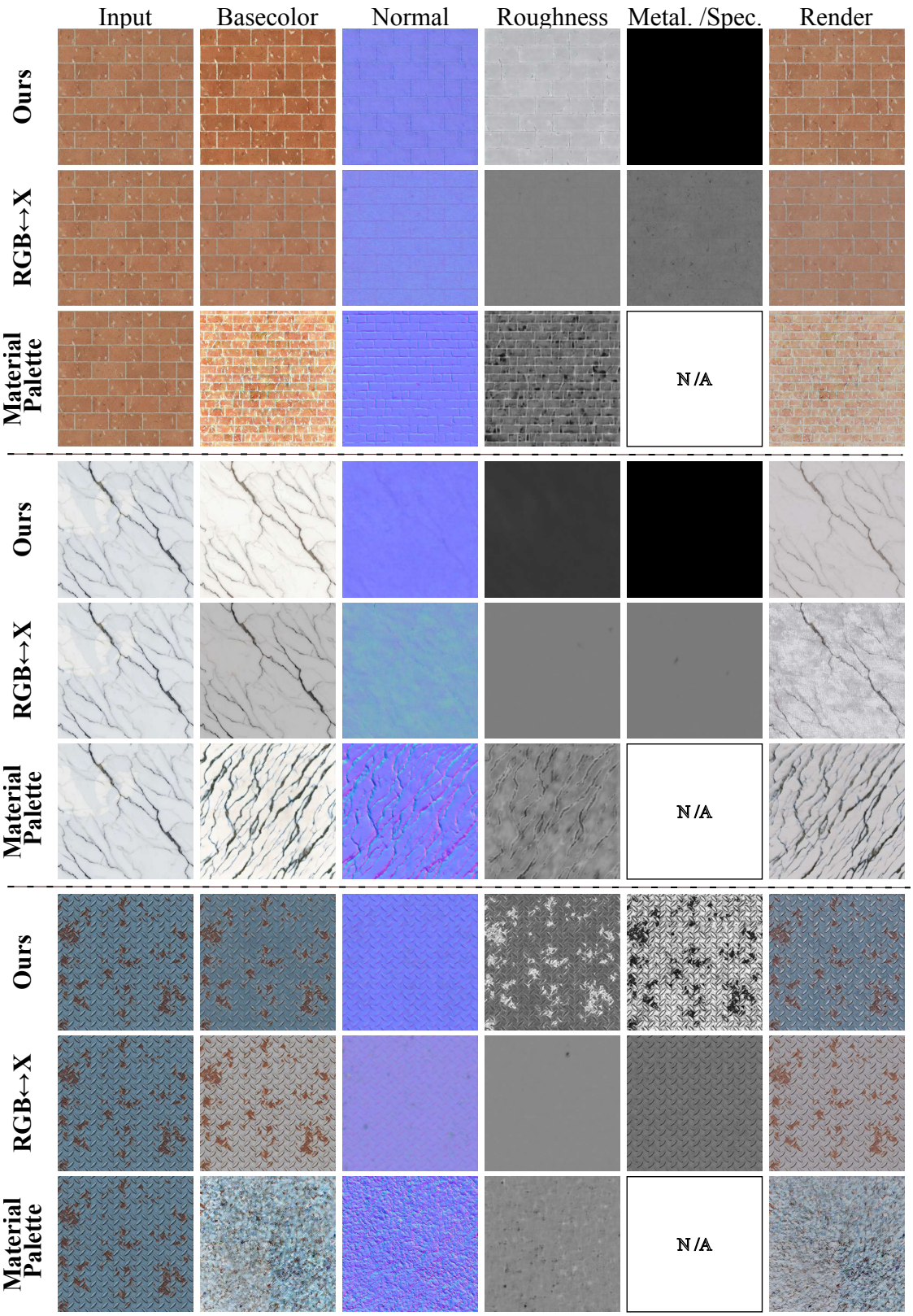


Figure 16. Qualitative comparison of material decomposition. For each sample, the first column shows the planar input image, and the second to last columns present the generated material maps together with a rendering under environment lighting. Our method produces consistent structural patterns, yielding rendered views that closely match the input appearance.