POSTECH

AMI Algorithmic Machine Intelligence Lab.

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

https://kim-youwang.github.io/paint-it

Paint-it: Text-to-Texture Synthesis via Deep Convolutional Texture Map Optimization and Physically-Based Rendering

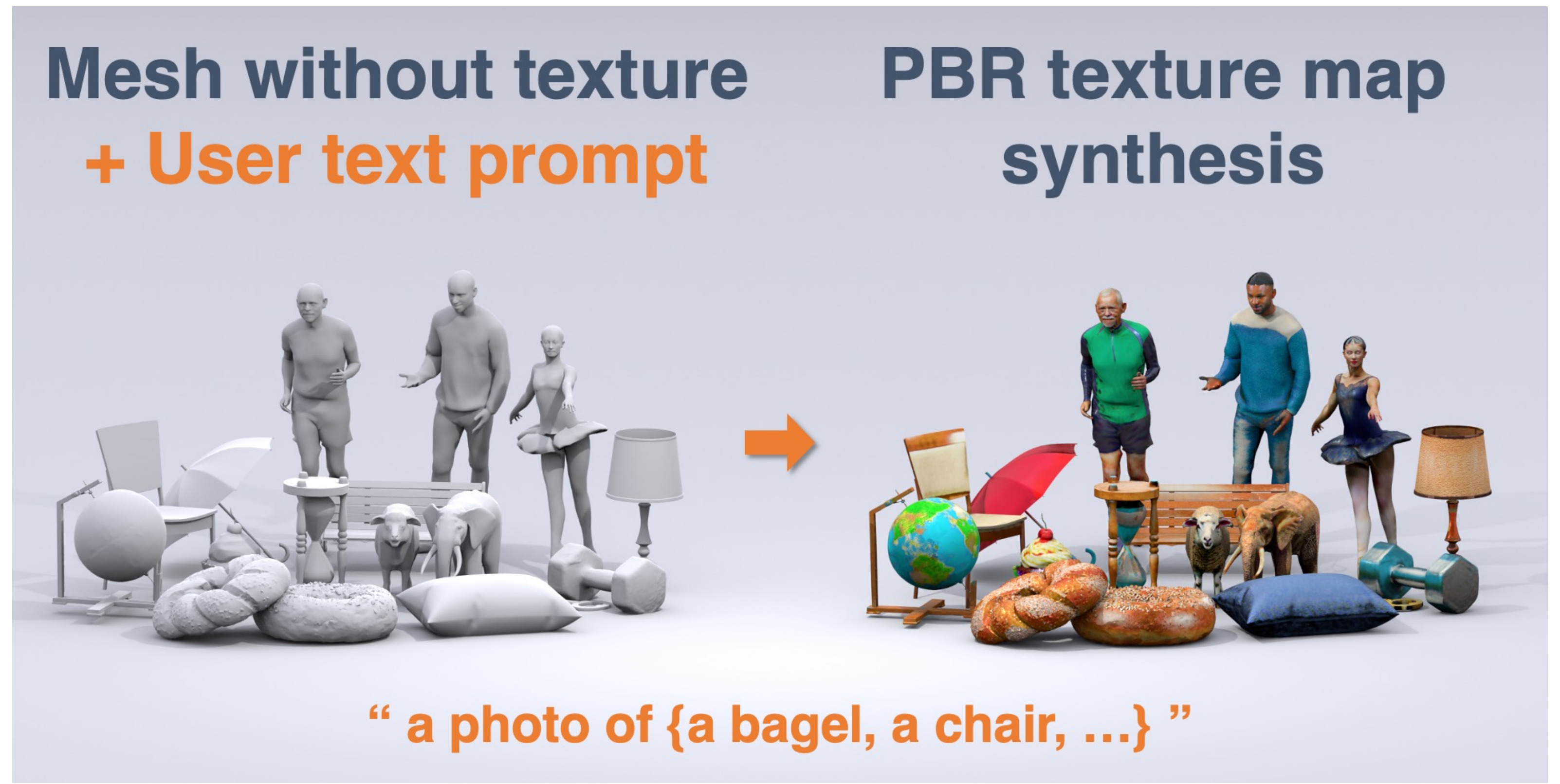
Kim Youwang[1] Tae-Hyun Oh[1] Gerard Pons-Moll[2]

[1]POSTECH [2]University of Tübingen

CVPR SEATTLE, WA JUNE 17-21, 2024

Overview

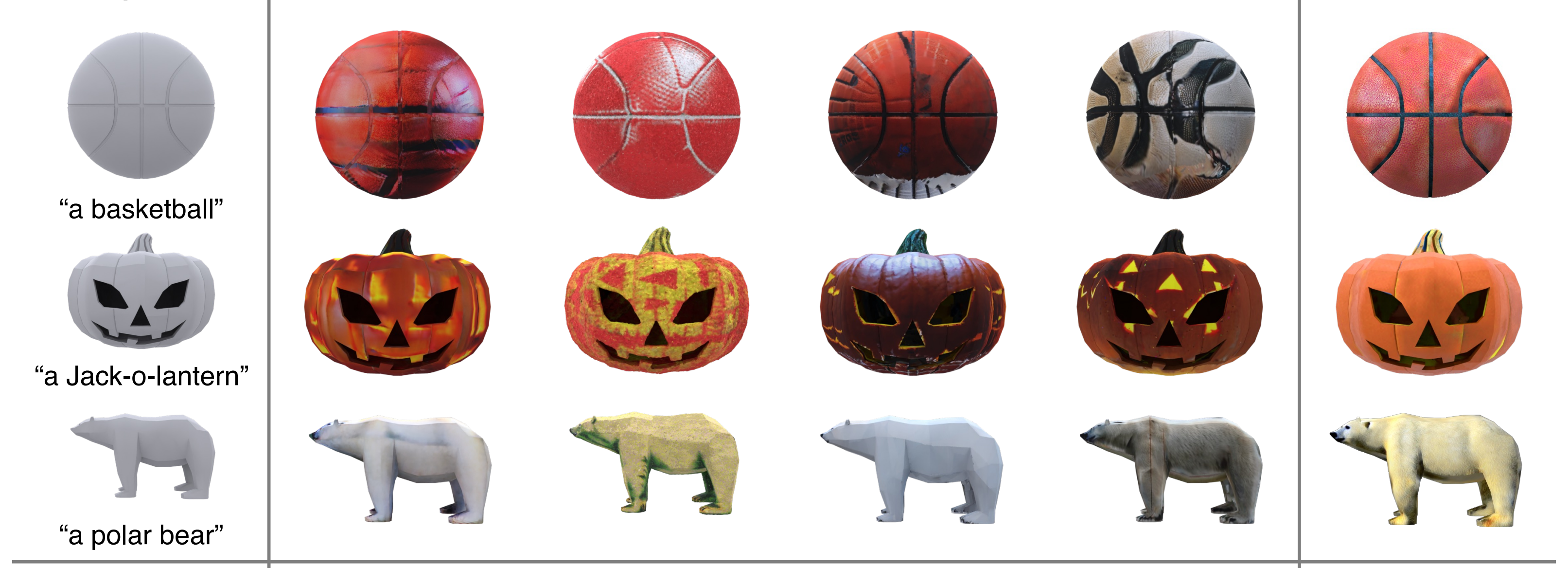
TL;DR Convolutional parameterization of PBR texture maps helps high-quality text-to-texture synthesis!



Mesh without texture
+ User text prompt
PBR texture map synthesis
" a photo of {a bagel, a chair, ...} "

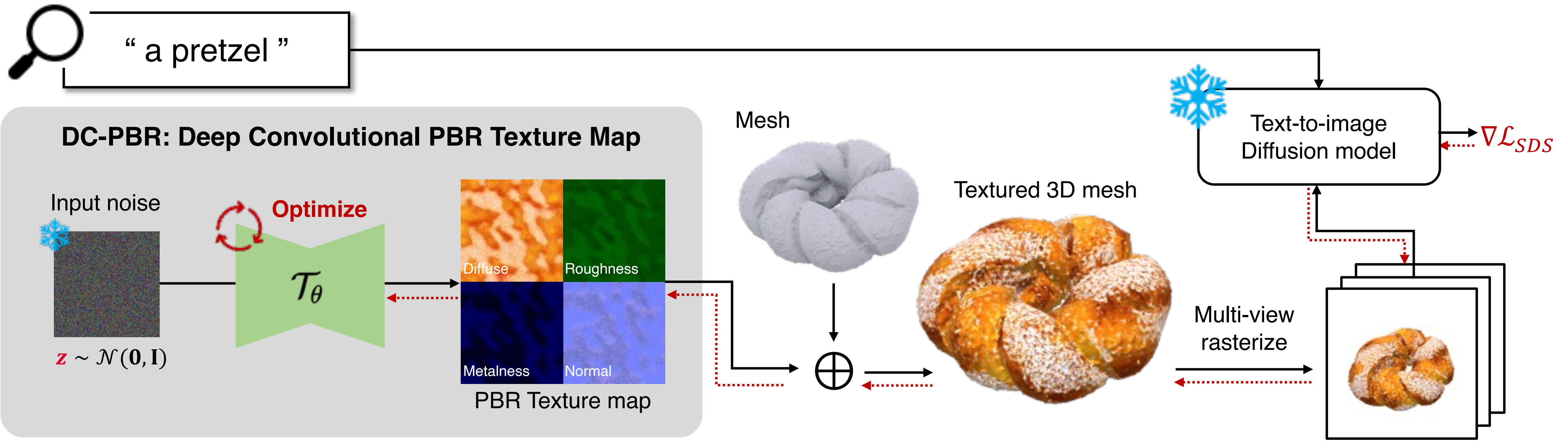
Contributions

- **Paint-it**: Text → 3D mesh texture synthesis method, generating PBR maps compatible with graphics engines
- **DC-PBR**: CNN re-parameterization of PBR texture maps guides frequency-scheduled texture synthesis



Objaverse mesh w/ input text	Latent-Paint (CVPR 2023)	Fantasia3D (ICCV 2023)	Text2Tex (ICCV 2023)	TEXTure (SIGGRAPH 2023)	Paint-it (Ours)
Texture type / Relightable?	RGB / ✗	Per-point PBR / ✓	RGB / ✗	RGB / ✗	DC-PBR / ✓

Paint-it: Text-Driven PBR Texture Map Synthesis



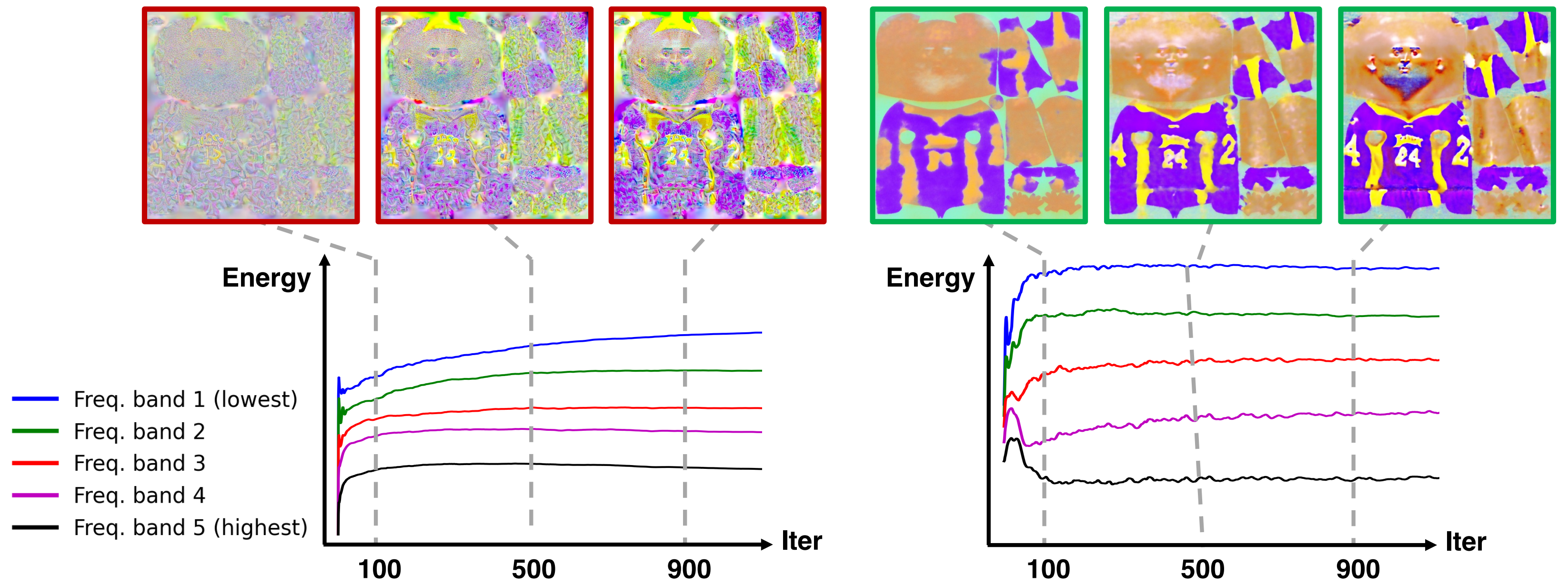
" a pretzel "
DC-PBR: Deep Convolutional PBR Texture Map
Input noise
$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Optimize
$\mathcal{T}_\theta$
Diffuse
Roughness
Metalness
Normal
PBR Texture map
Mesh
Textured 3D mesh
Multi-view rasterize
Text-to-image Diffusion model
$\nabla \mathcal{L}_{SDS}$

- Step 1: Randomly initialize CNN U-Net ($\mathcal{T}_\theta$) & Fixed input noise ($\mathbf{z}$)
- Step 2: Feed fixed $\mathbf{z}$ to U-Net $\mathcal{T}_\theta$ → Synthesize PBR texture maps
- Step 3: Render 3D mesh with the synthesized PBR texture maps
- Step 3: Update **CNN kernels θ**, using Score-Distillation Sampling (SDS) gradient

$$\underbrace{[\mathbf{K}^{\mathrm{d}}_\theta, \mathbf{K}^{\mathrm{rm}}_\theta, \mathbf{K}^{\mathrm{n}}_\theta]}_{\text{Explicit PBR texture maps}} = \underbrace{\mathcal{T}_\theta(\mathbf{z})}_{\text{DC-PBR : Our PBR representation}}$$

Deep Convolutional PBR Texture Map Re-param. (DC-PBR)

Our observation: Re-parameterizing PBR texture maps with CNN kernels naturally induces **"coarse-to-fine frequency scheduling"** for texture synthesis & noise robustness



Per-pixel optimization (Baseline)
Increases all freq. despite the TV loss

DC-PBR optimization (Ours)
Learns low freq. to high freq.

(Baseline) Per-pixel PBR optimization

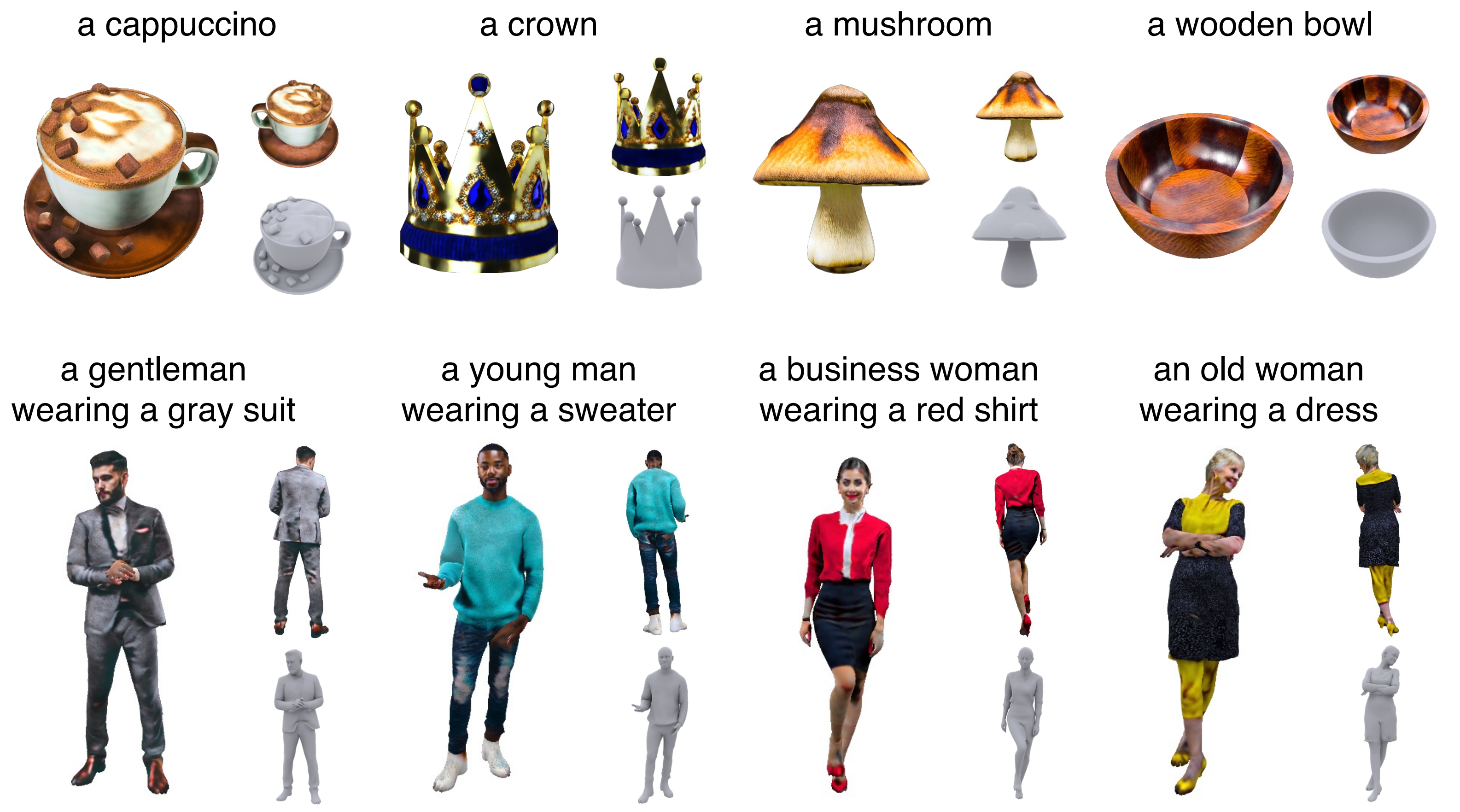
$$[\mathbf{K}^{\mathrm{d}*}, \mathbf{K}^{\mathrm{rm}*}, \mathbf{K}^{\mathrm{n}*}] = \arg\min_{\mathbf{K}^{\mathrm{d}}, \mathbf{K}^{\mathrm{rm}}, \mathbf{K}^{\mathrm{n}}} \mathbb{E}_{t,\epsilon}\left[\|\hat{\epsilon}_\phi(\mathcal{R}^{\mathrm{M}}_t(\mathbf{K}^{\mathrm{d}}, \mathbf{K}^{\mathrm{rm}}, \mathbf{K}^{\mathrm{n}}); y, t) - \epsilon\|^2_2\right] + \mathcal{L}_{\mathrm{TV}}$$

(Ours) DC-PBR re-parameterized optimization

$$\theta^* = \arg\min_\theta \mathbb{E}_{t,\epsilon}\left[\|\hat{\epsilon}_\phi(\mathcal{R}^{\mathrm{M}}_t(\mathbf{K}^{\mathrm{d}}_\theta, \mathbf{K}^{\mathrm{rm}}_\theta, \mathbf{K}^{\mathrm{n}}_\theta); y, t) - \epsilon\|^2_2\right],$$
$$[\mathbf{K}^{\mathrm{d}}_\theta, \mathbf{K}^{\mathrm{rm}}_\theta, \mathbf{K}^{\mathrm{n}}_\theta] = \mathcal{T}_\theta(\mathbf{z})$$

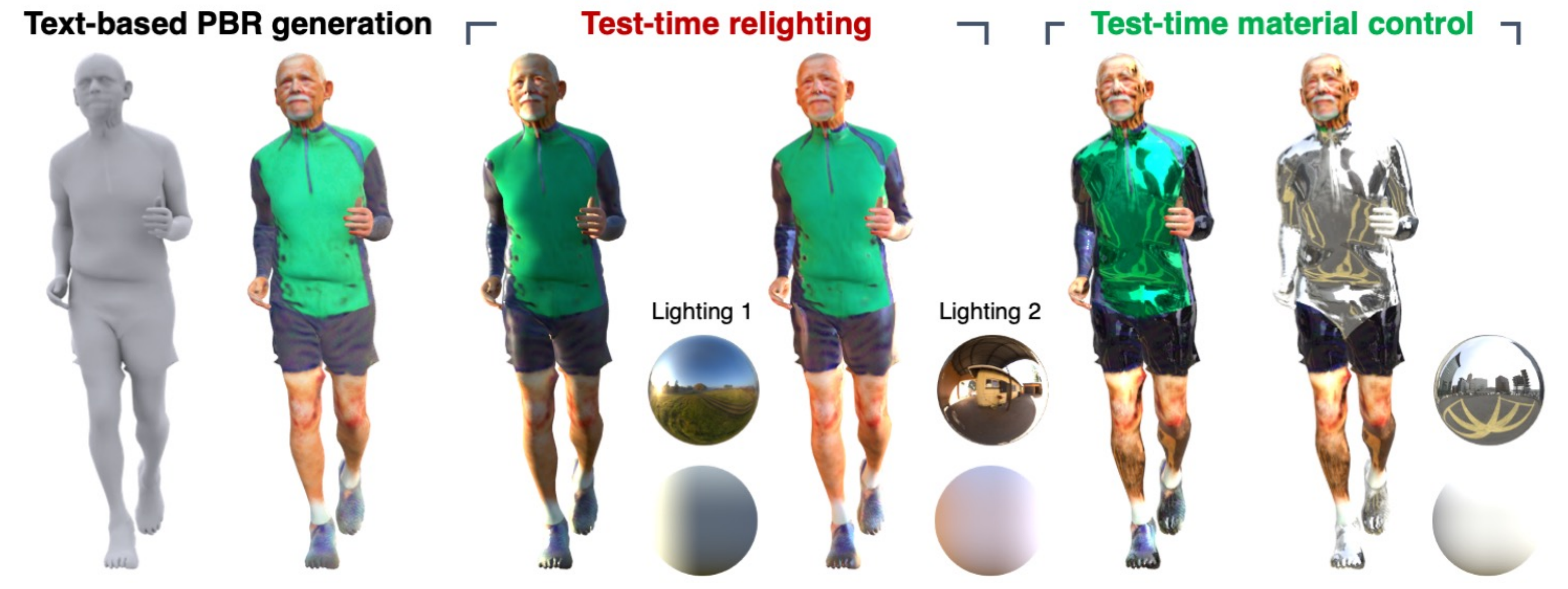
PBR Texture Synthesis Results

Qualitative results - Objaverse, RenderPeople



a cappuccino
a crown
a mushroom
a wooden bowl
a gentleman wearing a gray suit
a young man wearing a sweater
a business woman wearing a red shirt
an old woman wearing a dress

Test-time applications with CG engines



Take-home message

The SDS loss is known to be noisy. We show a better PBR texture representation (parameterization) sufficiently stabilizes the SDS loss and its empirical mechanism.