A1 Appendix

This supplementary document offers further technical details, demonstrations of the datasets employed, additional insights, and comprehensive results pertaining to our LiNo-UniPS.

A1.1 Network architecture details

To provide readers with a more in-depth understanding of our LiNo-UniPS network architecture, the method primarily comprises seven key modules:(a) WaveDownSample module, (b) DINOv2-based feature extraction backbone, (c) an Enhanced Light-Normal Contextual Attention Module, (d) a DPT-Based Fusion Module, (e) a WaveUpSample module, (f) a final Decoder and (g) the training loss. An overview of our network architecture is presented in Figure 2. In the subsequent sections, we will provide a detailed account of the network's structural components and the specific transformations of tensor shapes as data progresses through the model.

A1.1.1 WaveDownSample:

To enable our encoder to extract more fine-grained contexts, we first incorporate the wavelet transform [11], chosen for its ability to separate an image's high- and low-frequency components [15, 24] while concurrently mitigating losses typically incurred during downsampling [44].

Initially, the input to our network is a batch of multi-light image sets, represented by a tensor $I \in \mathbb{R}^{B\times F\times H\times W\times 3}$. In this notation, B denotes the batch size, F is the number of images captured under different illumination conditions for each scene instance, H and W represent the spatial dimensions (height and width), and the final dimension 3 corresponds to the RGB color channels. To simplify the subsequent exposition, we will assume a batch size of $B = 1$ unless otherwise specified, effectively considering the processing of a single multi-illumination image set at a time. As part of the preprocessing, to ensure that image pixel values lie within a comparable range, each of the F images within a given scene instance is normalized by a random scalar sampled uniformly between its maximum and mean values. Following this preprocessing, for the purpose of subsequent discussion (effectively assuming $B = 1$), we obtain a set of F images $\{I_f\}_{f=1}^{F}$, where each $I_f \in \mathbb{R}^{H\times W\times 3}$.

Following SDM-UniPS [32], for each pre-processed input image $I_f \in \mathbb{R}^{H\times W\times 3}$, we perform two separate transformations: naive downsampling to obtain $I_f^d \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$ in the image domain, and a wavelet transform to yield its corresponding wavelet domain components $I_f^w \in \mathbb{R}^{4\times\frac{H}{2}\times\frac{W}{2}\times 3}$, namely $I_f^{ll} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$, $I_f^{lh} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$, $I_f^{hl} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$, and $I_f^{hh} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$.

A1.1.2 DINOv2 backbone:

Subsequently, both the downsampled image representation $I_f^d \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$ and the set of wavelet components I_f^w (comprising $I_f^{ll}, I_f^{lh}, I_f^{hl}, I_f^{hh}$, each in $\mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 3}$) are individually processed. First, each of these input components is partitioned into a sequence of patch-based tokens. These token sequences are then fed into our feature extraction backbone. To maintain a lean model architecture and leverage strong prior knowledge while minimizing the introduction of excessive additional parameters, we employ the DINOv2-small variant [43] as this backbone. This DINOv2-small backbone is initialized with its publicly available pre-trained weights and is subsequently fine-tuned during our training procedure to extract rich visual representations from these diverse inputs. In our specific implementation, we set the patch size to $P = 8$. Consequently, for each input component with spatial dimensions $H/2 \times W/2$, the resulting sequence length is $L = \frac{(H/2)\times(W/2)}{P^2}$ tokens. Given our use of the DINOv2-small variant, its feature embedding dimension is $D = 384$. Following processing by this DINOv2 backbone, we respectively obtain shallow visual feature representations $F_{s,f}^d \in \mathbb{R}^{L\times D} \in$ from the downsampled image stream (derived from I_f^d) and $F_{s,f}^w \in \mathbb{R}^{4\times L\times D}$ from the wavelet components stream (derived from I_f^w).

A1.1.3 Enhanced Light-Normal Contextual Attention Module:

To achieve a more effective decoupling of lighting and normal features that are subsequently processed by the decoder, we introduce our Enhanced Light-Normal Contextual Attention Module.

Firstly, we design the light registration tokens to improve the handling of global illumination. While lighting information predominantly exhibits global characteristics across multi-light inputs [26, 29, 32], traditional attention mechanisms in existing Universal Photometric Stereo (PS) methods often fail to fully leverage this distributed information. This deficiency can hinder effective illumination-geometry separation, motivating our specialized token-based strategy. Drawing inspiration from advancements like [10], and further considering the inherent illumination-dependency of the Universal PS task, we introduce these *light registration tokens* to explicitly capture and represent decoupled global lighting information within our framework.

To facilitate the perception of distinct lighting components, we additionally introduce three specialized light tokens: $x_{\text{hdri}} \in \mathbb{R}^{1\times C}$, designated for perceiving spatially-varying environment light; $x_{\text{point}} \in \mathbb{R}^{1\times C}$, tailored for point lights (which often contribute high-frequency illumination effects); and $x_{\text{area}} \in \mathbb{R}^{1\times C}$, for directional light (typically representing low-frequency illumination sources). Subsequently, this set of three specialized light tokens is prepended to the token sequences derived from $F^d_{s,f}$ (features from the downsampled image stream) and $F^w_{s,f}$ (features from the wavelet components stream), respectively, leading to:$F^d_{s,f,r} \in \mathbb{R}^{L'\times D}$ and $F^w_{s,f,r} \in \mathbb{R}^{4\times L'\times D}$, where $L' = L + 3$.

Then $F^d_{s,f,r}$ and $F^w_{s,f,r}$ are fed into our Interleaved Attention block [55] to enhance inter-intra feature communication. Specifically, our alternating attention block contains four attention layers, which can be represented as: frame $\rightarrow$ light-axis $\rightarrow$ global $\rightarrow$ light-axis. Previous work has found that feature communication within the encoder is very important [29, 32, 21], but their methods are often limited to patch-level local light-axis attention. Our alternating attention, however, breaks such limitations. On the one hand, it incorporates frame attention to enhance inter-image communication. On the other hand, we have added global attention, a more comprehensive global operation, allowing intra-image features to also be extended from the patch level to the image level. Upon processing by these four cascaded Interleaved Attention blocks (the number chosen to maintain a manageable parameter count, although more blocks could potentially be employed), we obtain the deep feature representations denoted as $F^d_{d,f,r} \in \mathbb{R}^{L'\times D}$ and $F^w_{d,f,r} \in \mathbb{R}^{4\times L'\times D}$. It is worth noting that the attention operations inherent in these blocks do not alter the fundamental shapes of these tensor sequences.

To ensure the light registers tokens x_{hdri}, x_{point}, x_{area} effectively capture global illumination, as direct supervision of the decoded light map is challenging, we introduce a light-aware feature alignment strategy during training [62, 61, 51]. Since we train LiNo-UniPS on our own rendered synthetic dataset PS-Verse, for every scene within this training dataset, we have access to its corresponding lighting information from the rendering process. This includes: the environment light HDRI map, the directions and intensities of point lights, and the direction and intensity of the directional light. Mathematically, these are denoted as $L_{\text{hdri}} \in \mathbb{R}^{H\times W\times 3}$, $L_{\text{point}} \in \mathbb{R}^{M_1\times 4}$, and $L_{\text{area}} \in \mathbb{R}^{M_2\times 5}$, respectively. For $L_{\text{point}} \in \mathbb{R}^{M_1\times 4}$, where M_1 is the number of point lights, its first three dimensions denote position and the fourth denotes intensity. For the directional light component $L_{\text{area}} \in \mathbb{R}^{M_2\times 5}$, where M_2 denotes the number of directional lights, its first three dimensions specify position, the fourth denotes the directional light size, and the fifth indicates intensity. Subsequently, we encode the lighting components L_{hdri}, L_{point}, and L_{area} by projecting them into a D-dimensional feature space, thereby obtaining their respective representations $\mathbf{l}^h_{\text{hdri}} \in \mathbb{R}^D$, $\mathbf{l}^h_{\text{point}} \in \mathbb{R}^D$, and $\mathbf{l}^h_{\text{area}} \in \mathbb{R}^D$. Similarly, the light tokens x_{env}, x_{point}, and x_{dir}, after being processed by the cascaded alternating attention blocks, are projected into the same D-dimensional feature space. This yields their respective high-dimensional representations: $\mathbf{x}^h_{\text{hdri}}$, $\mathbf{x}^h_{\text{point}}$, and $\mathbf{x}^h_{\text{area}}$, all in $\mathbb{R}^D$. Specifically, this projection is realized using three structurally similar two-layer Multi-Layer Perceptrons (MLPs). Within this common embedding space, we employ cosine similarity to supervise and align their respective feature distributions. Cosine similarity is chosen as it effectively measures the directional concordance between feature vectors, making it suitable for aligning representations of different lighting characteristics. This supervision translates into three distinct loss functions, denoted as $\mathcal{L}_{\text{hdri}}$, $\mathcal{L}_{\text{point}}$, and $\mathcal{L}_{\text{area}}$. Their respective mathematical formulations are:

$$\mathcal{L}_{\text{hdri}} = 1 - \sum(\mathbf{l}^h_{\text{hdri}} \cdot \mathbf{x}^h_{\text{hdri}}) \tag{A1a}$$

$$\mathcal{L}_{\text{point}} = 1 - \sum(\mathbf{l}^h_{\text{point}} \cdot \mathbf{x}^h_{\text{point}}) \tag{A1b}$$

$$\mathcal{L}_{\text{area}} = 1 - \sum(\mathbf{l}^h_{\text{area}} \cdot \mathbf{x}^h_{\text{area}}) \tag{A1c}$$

A1.1.4 DPT-Based Fusion Module:

The subsequent discussion details the fusion module applied to features extracted during the initial stages of our Encoder. The overall feature fusion process is DPT-based [46]. For clarity, we will first illustrate this process using the features derived from the downsampled image features $F^{d}_{d,f,r}$ as the primary example. Within each Interleaved Attention module of the encoder, features obtained from its four internal attention—frame, light-axis, global, and ight-axis—are first concatenated along the feature dimension. This operation yields an aggregated feature set for each module, denoted as $F^{d,(i)}_{d,f} \in \mathbb{R}^{L\times 4D}$, where $i \in \{1, 2, 3, 4\}$ represents the index of the i-th attention block. It is crucial to note that the three additional *light registration tokens* (introduced previously) do not participate in this specific feature aggregation (concatenation) process. Then, the aggregated features $F^{d,(i)}_{d,f}$ from four selected Interleaved Attention blocks then undergo a series of projection and resizing operations. These operations transform them into a four-level feature pyramid, with hierarchical features H_1, H_2, H_3, and H_4. Their respective shapes are $F \times C \times H_0/2 \times W_0/2$, $F \times 2C \times H_0/4 \times W_0/4$, $F \times 4C \times H_0/8 \times W_0/8$, and $F \times 4C \times H_0/16 \times W_0/16$. In this context, F represents the number of multi-light input images, C is a base feature channel dimension (256), and H_0, W_0 refer to the spatial resolution of the original, full-sized input images. These four levels of hierarchical features (H_1 through H_4) are subsequently progressively fused using residual convolutional blocks [23], ultimately producing a fused feature representation $F^{d}_{\text{fused}} \in \mathbb{R}^{F\times \frac{H}{2}\times \frac{W}{2}\times C}$. A similar fusion process is applied to the features derived from the wavelet components path ($F^{w}_{d,f,r}$), yielding a corresponding fused representation, $F^{w}_{\text{fused}} \in \mathbb{R}^{4\times F\times \frac{H}{2}\times \frac{W}{2}\times C}$. It is noteworthy that this strategy of selecting four feature levels for hierarchical fusion can be adapted if more than four Interleaved Attention blocks are employed in the encoder's initial stages. For instance, if six such blocks are utilized, features from blocks indexed 1, 2, 4, and 6 might be selected to form the pyramid. Similarly, for an eight-block configuration, features from blocks 1, 3, 5, and 7 could be chosen as inputs to the hierarchical fusion pathway.

A1.1.5 WaveUpSample:

To obtain the final encoder output $F_{\text{enc}} \in \mathbb{R}^{F\times H\times W\times C}$, the features derived from the downsampled image path F^{d}_{fused} and those from the wavelet components path F^{w}_{fused} should be integrated. The process is as follows: first, the feature map F^{d}_{fused} is upsampled, yielding an representation $F^{\text{up}}_{\text{fused}} \in \mathbb{R}^{F\times H\times W\times C}$. Concurrently, for F^{w}_{fused}, which originates from the wavelet-transformed inputs, an inverse wavelet transform is applied to convert it back to the spatial domain, resulting in $F^{\text{dwt}}_{\text{fused}} \in \mathbb{R}^{F\times H\times W\times C}$. Finally, these two processed feature sets, F^{up}_{fused} and F^{dwt}_{fused}, are element-wise summed. A Gaussian blur is subsequently applied to this sum to promote a smoother and more effective fusion of these potentially cross-domain features, ultimately producing the final encoder representation $F_{\text{enc}} \in \mathbb{R}^{F\times H\times W\times C}$.

A1.1.6 Decoder:

The decoder architecture in our LiNo-UniPS is largely identical to that of SDM-UniPS [32]; for clarity, we briefly outline its key components and rationale here.

A common initial step in PS for surface normal estimation is the pixel-wise aggregation of spatial-light features along the illumination axis, effectively reducing F light channels to a single representation per pixel using input images I_f and their corresponding encoded features F^{f}_{enc}. We introduce an approach, termed the pixel-sampling Transformer [60, 9, 59, 58], which uniquely operates on a fixed count (m, e.g., $m = 2048$) of randomly chosen pixel locations. This strategy offers distinct advantages: it maintains a constant memory footprint per sample set regardless of image dimensions, thus ensuring excellent scalability; furthermore, by processing a sparse, randomly distributed set of points, it substantially curtails over-smoothing artifacts often prevalent in dense convolutional operations. The practical implementation of the *pixel-sampling Transformer* involves selecting m random pixels, denoted $\{x_i\}_{i=1}^{m}$, from the valid (masked) region of the input image. For each sampled pixel x_i, its associated features $F^{f}_{\text{enc}}(x_i) \in \mathbb{R}^{C}$ are obtained. These features $F^{f}_{\text{enc}}(x_i)$ are then combined through element-wise addition with $I'_f(x_i) \in \mathbb{R}^{C}$. The term $I'_f(x_i)$ represents a high-dimensional projection of the raw pixel observations $I_f(x_i) \in \mathbb{R}^{3}$, and this projection is performed by a two-layer MLP with the objective of enhancing the representational power of these

raw observations by mapping them to this higher-dimensional space. Notably, our strategy of first projecting the raw observations to $\mathbb{R}^C$ and then performing addition differs from SDM-UniPS [32], which typically employ direct concatenation of features and the raw observations. These added per-pixel features $F^f_{\text{add}}(x_i)$ are subsequently condensed into compact descriptors $A(x_i)$ by employing Pooling by Multi-head Attention (PMA) [36]. The resulting collection of m descriptors, $A(x_i)_{i=1}^m$, is then fed into a Transformer network. The processing for each of the m sampled points involves applying frame attention and light-axis attention to aggregate non-local context and cross-image information. Following this, a two-layer MLP is utilized to predict the surface normal vector for each location. These sparsely predicted normals are then systematically merged—for instance, through spatial interpolation or a dedicated upsampling module—to reconstruct the full-resolution surface normal map corresponding to the original input image dimensions. In essence, the pixel-sampling Transformer facilitates the modeling of robust non-local dependencies with notable computational efficiency, while concurrently preserving fine details in the output normal map. This makes the approach particularly well-suited for physics-based vision tasks that involve high-resolution imagery.

A1.1.7 Training Loss:

In this part, we elaborate on the composition of our total training loss and the design of the respective weights for its constituent components. Based on the definitions provided in Eq. 4, Eq. 3.1 and Eq. A1, the overall training loss $\mathcal{L}$ for our LiNO-UniPS method can be expressed as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{hdri}} + \lambda_2 \mathcal{L}_{\text{point}} + \lambda_3 \mathcal{L}_{\text{area}} + \lambda_4 \sum (N - \tilde{N})^2 \odot C + \lambda_5 \sum (\tilde{G} - G)^2,$$

Let us define the confidence-weighted normal reconstruction loss as $\mathcal{L}_{\text{conf}}$ and the normal gradient supervision loss as $\mathcal{L}_{\text{gradient}}$. The overall training loss $\mathcal{L}$ can then be expressed as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{hdri}} + \lambda_2 \mathcal{L}_{\text{point}} + \lambda_3 \mathcal{L}_{\text{area}} + \lambda_4 \mathcal{L}_{\text{conf}} + \lambda_5 \mathcal{L}_{\text{gradient}},$$

Since our primary objective is surface normal reconstruction, $\mathcal{L}_{\text{conf}}$ (our confidence-weighted reconstruction loss) is established as the principal component of our total loss function. First, we provide a detailed explanation for our selection of $\mathcal{L}_{\text{conf}}$ as the primary loss function. We elaborate on why this specific formulation was chosen over other potential candidates, such as a direct MSE, $\sum (N - \tilde{N})^2$, or an alternative loss weighted by $e^G, G = \nabla N$, namely $\sum (N - \tilde{N})^2 \odot e^G$.

A primary motivation for our LiNO-UniPS framework is to advance beyond prior Universal PS methods by specifically improving the handling of challenging high-frequency regions. We identify these regions based on large magnitudes of the surface normal gradients, as these directly reflect geometric complexity. We deliberately avoid using gradients derived from the input RGB multi-light images as the primary criterion for this identification. The rationale is that while RGB gradients are indeed large in areas of intricate geometric detail, they can also exhibit high magnitudes in regions with significant basecolor variations, which do not necessarily correspond to the geometric high-frequency features we aim to emphasize and reconstruct accurately. Consequently, our methodology incorporates a loss function that is directly informed by surface normal gradients, rather than relying on a naive MSE.

A crucial aspect of this gradient-informed loss strategy concerns the source of the gradients utilized for weighting or guidance. We opt to utilize network-estimated normal gradients $\tilde{G}$ for this purpose, rather than directly employing ground truth normal gradients G. This design choice is primarily motivated by two factors: Firstly, it compels the network to intrinsically estimate high-frequency components from the input, thereby fostering its inherent capability to process and represent fine-grained details. Secondly, refraining from direct weighting by ground truth normal gradients typically leads to a more stable and manageable training process, especially during the initial stages when network predictions may significantly deviate from the ground truth.

Our design for the loss weights is as follows:

$$\lambda_1 = \frac{0.1}{(\mathcal{L}_{\text{hdri}}/\mathcal{L}_{\text{conf}})_{\text{sg}}}, \lambda_2 = \frac{0.1}{(\mathcal{L}_{\text{point}}/\mathcal{L}_{\text{conf}})_{\text{sg}}}, \lambda_3 = \frac{0.1}{(\mathcal{L}_{\text{area}}/\mathcal{L}_{\text{conf}})_{\text{sg}}}, \lambda_4 = 1, \lambda_5 = \frac{0.1}{(\mathcal{L}_{\text{gradient}}/\mathcal{L}_{\text{conf}})_{\text{sg}}}$$

where the subscript ‘sg‘ denotes that the term within the parenthesis is treated as a constant (i.e., its gradient is not computed during backpropagation for the purpose of this scaling factor, akin to `.detach()` in PyTorch).

Our decision to set λ_4 to 1 is because $\mathcal{L}_{\text{conf}}$ serves as the principal component in our overall loss function. The remaining auxiliary losses are then scaled using the adaptive weighting mechanism detailed in Eq. A1.1.7. This mechanism constrains their magnitudes to 0.1 times that of the primary loss's detached value, $(\mathcal{L}_{\text{conf}})_{\text{sg}}$, while still allowing their gradients to backpropagate fully. Such a strategy effectively positions these auxiliary losses to act as regularizers to the main learning objective, rather than allowing disparate loss magnitudes to vie for dominance and potentially destabilize training. This controlled weighting is crucial for ensuring stable and efficient training of LiNO-UniPS, mitigating issues such as excessively slow convergence or even training failure that can arise from an unbalanced multi-term loss function.

A1.2 Dataset Analysis and Presentation

A1.2.1 Categorization Methodology

To rigorously evaluate and enhance the capability of our LiNO-UniPS method for reconstructing surface normals of objects that feature high-frequency geometric details on complex surfaces, we curated a dedicated set of objects exhibiting diverse geometric complexities. These objects were subsequently graded by difficulty into five distinct levels, designated Level 1 to Level 5. Specifically, Levels 1–4 are classified following Dora [8] criterion, based on the number of salient edges N_Γ. Level 5, in contrast, is distinguished by the use of normal maps in its rendering. The specific criteria for this classification are as follows:

- Level 1 (Less Detail): $0 < N_\Gamma \leq 5000$;
- Level 2 (Moderate Detail): $5000 < N_\Gamma \leq 20000$;
- Level 3 (Rich Detail): $20000 < N_\Gamma \leq 50000$;
- Level 4 (Very Rich Detail): $N_\Gamma > 50000$.
- Level 5 : With Normal Map.

Fig. A1 shows representative cases from the different defined levels. PS-Verse comprises 100,000 scenes. For each of these scenes, two distinct renderings are typically generated: one that utilizes normal map to incorporate fine geometric details, and another rendered without this normal map. Levels 1 4 consist exclusively of scenes rendered without normal map, with each of these four levels containing 25,000 scenes. Level 5 is composed entirely of the 100,000 scenes rendered with normal map enhancement.

A1.2.2 The Use of Normal Map

To enhance PS normal reconstruction for objects characterized by intricate, high-frequency details, training data rich in such geometric features is essential. However, 3D models genuinely possessing fine-grained geometric intricacies are often scarce and prohibitively expensive, which impedes the creation of diverse, large-scale, high-fidelity datasets. To overcome this limitation within the Universal PS framework, our work pioneers the integration of normal mapping directly into the dataset generation process. Normal mapping, a 3D computer graphics technique, imbues low-polygon models with the visual appearance of high-frequency geometric details by applying a specialized texture, known as a normal map, which encodes fine-scale perturbations of the surface normals. During rendering, these stored normal variations are then utilized to simulate intricate surface details without actually increasing the underlying geometric complexity or polygon count of the model.

A visual comparison of renderings with and without the use of normal maps is presented in Fig. A2. It is clearly evident from the figure that employing normal maps during the rendering process yields a significantly higher level of detail in the resulting surface normals.

A1.2.3 Lighting Setup

When rendering PS-Verse, we use four types of light sources; (a) environment lighting, (b) directional lighting, (c) point lighting, (d) uniform background lightings.

During rendering, we generate ten distinct lighting configurations by combining several base lighting components (conceptually denoted here as (a), (b), (c), and (d)). These specific configurations are as

Figure A1: The objects are displayed sequentially from left to right, representing Level 1 to Level 5. Across Levels 1 through 5, there is a progressive rise in geometric complexity. Specifically, Level 5 features exceptionally complex surface geometry due to the utilization of normal maps in its rendering process.

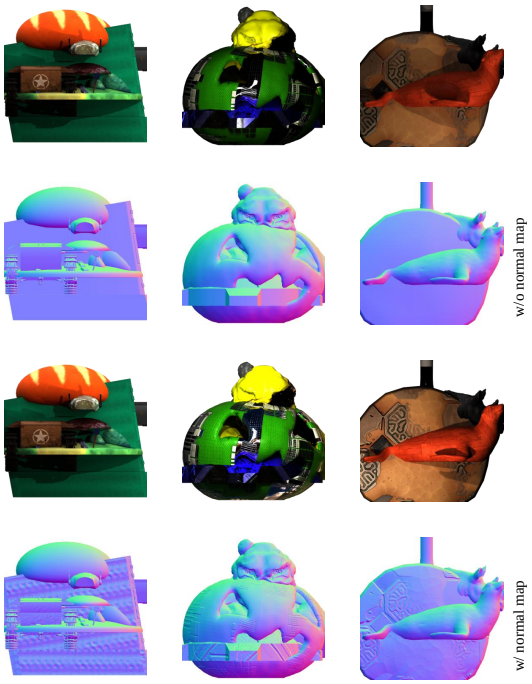

Figure A2: Effect of normal mapping on rendered surface detail. The top two rows display renderings without normal maps, while the bottom two rows showcase the same scenes rendered with normal map. It is evident that employing normal maps (bottom rows) results in significantly more high-frequency surface normal detail compared to renderings without (top rows).

follows: (1) Component (a), (2) Component (b), (3) Component (b), (4) Components (a) + (b), (5) Components (a) + (c), (6) Components (b) + (c), (7) Components (a) + (b) + (c), (8) Components (a) + (d), (9) Components (b) + (d), (10) Components (a) + (b) + (d). The lighting setup includes: one directional light; a variable number of point lights, randomly ranging from one to three; and uniform background lighting, introduced to better simulate realistic global illumination. Every scene in PS-Verse is rendered as 20 images, each employing a lighting setup randomly chosen from our ten predefined lighting configurations. An example of such an image set for a single scene is illustrated in Figure A3.

A1.2.4 Comparison with Other Datasets

Here, we primarily compare several training datasets: CyclesPS-Train [26], PS-Wild [29], PS-Mix [32], PS-Uni MS-PS [20], and our PS-Verse. For a qualitative comparison of these datasets, please refer to Tab. A4. We now present illustrative qualitative comparisons in Fig. A4. Our comprehensive evaluation, encompassing both qualitative and quantitative aspects, leads us to conclude that PS-Verse is the premier training dataset in terms of quality for the Universal PS task.

A1.3 Additional Discussion

A1.3.1 Why Feature Consistency Matters

The fundamental objective of the Universal PS task is to reconstruct illumination-invariant surface normals from a series of multi-light input images. Consequently, the decoder must effectively aggregate contextual information provided by the encoder to derive a lighting-agnostic representation of these surface normals. Existing Universal PS methods predominantly utilize encoder-decoder architectures, wherein encoders typically possess a significantly larger parameter count than their decoder counterparts. Therefore, assuming a relatively consistent decoder design, empowering a more capable encoder to effectively disentangle lighting from normal information and refine these into robust normal-specific features is expected to yield superior normal reconstruction.

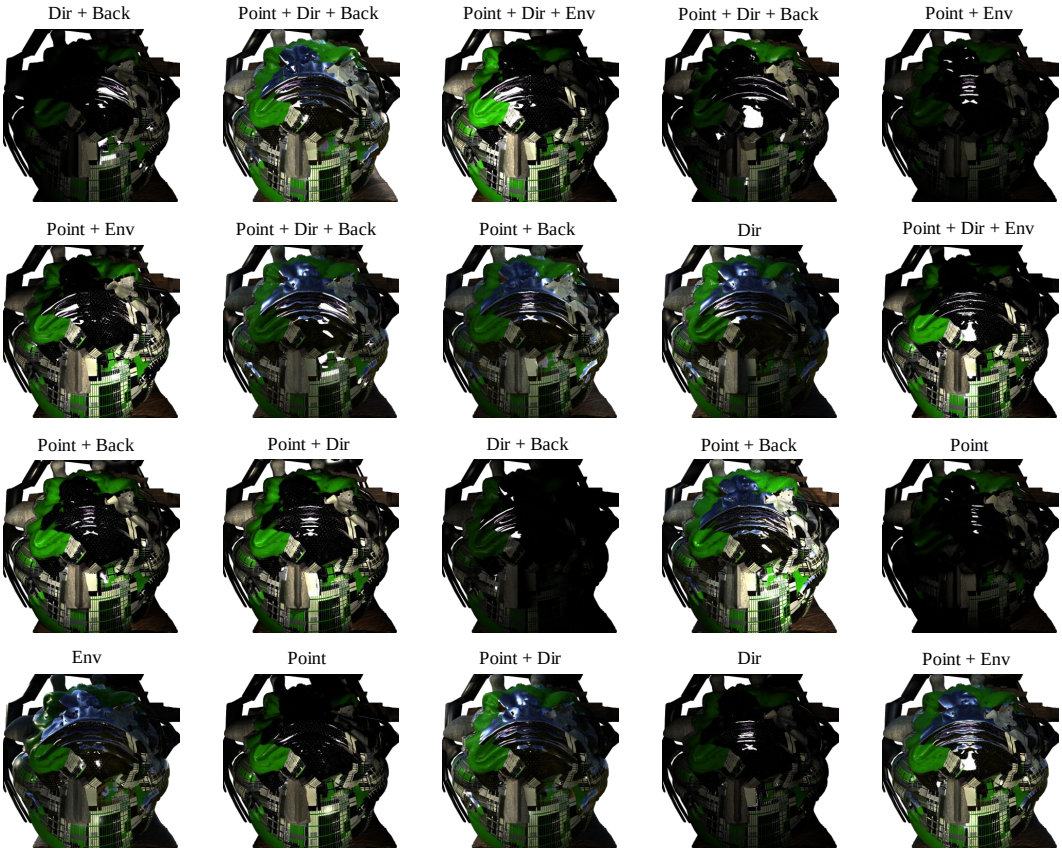

Figure A3: 20 multi-light images of one scene and their respective lighting configurations.

Building upon approaches such as UniPS [29] and SDM-UniPS [32] that utilize comparable decoder designs, a primary driver for our LiNO-UniPS is the introduction of an improved encoder. The objective of this encoder is to yield more consistent features, thereby achieving superior results. However, it is not an oversimplification that greater feature consistency directly translates to superior normal reconstruction; the decoder's architecture and capabilities also represent a non-negligible factor in overall performance. Nonetheless, the decoders utilized in UniPS, SDM-UniPS and our LiNO-UniPS are architecturally similar, all adhering to the pixel-sampling paradigm. Consequently, when analyzing the relationship between feature consistency (e.g., as measured by CSIM/SSIM) and normal reconstruction performance in Fig. 1, we group these three methods together to facilitate a more direct comparison of the impact of their respective encoder-derived features.

Fig. A8 presents Principal Component Analysis (PCA) visualizations of features extracted by the encoders of various methods, alongside their corresponding feature similarity metrics (CSIM/SSIM). In the following discussion, we primarily focus our analysis on our Ours w/mlp variant and Uni MS-PS.

Ours w/MLP refers to a configuration of our LiNO-UniPS where the standard decoder is replaced by a simple two-layer MLP. As illustrated in Fig. A8, this variant gives the highest feature similarity. Visual inspection of the PCA plots further reveals that its extracted features have effectively disentangled lighting information. We hypothesize that the superior feature similarity of Ours w/mlp compared to the full LiNO-UniPS (with its original decoder) stems from the constraints imposed by the weaker MLP decoder; this limited decoder capacity compels the encoder to learn more consistent features to facilitate accurate normal reconstruction. While this enhanced feature consistency from the encoder may not entirely compensate for the reduced representational power of the simpler decoder in terms of absolute normal reconstruction quality (when compared to the full LiNO-UniPS), Ours w/MLP variant nevertheless significantly outperforms SDM-UniPS. This finding strongly corroborates our

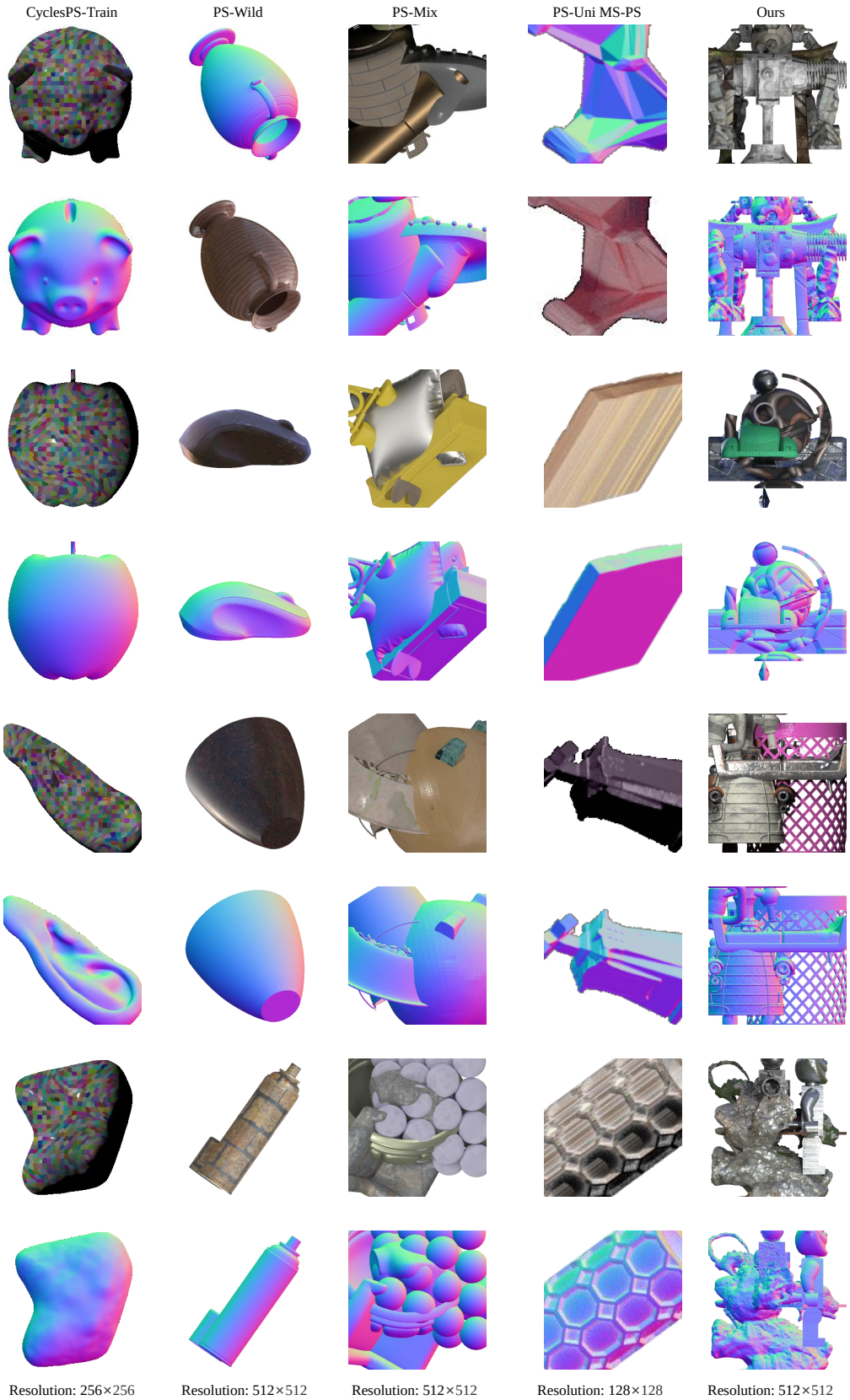

Figure A4: Visual comparison of different datasets. The spatial resolution of the images corresponding to each column is indicated beneath it.

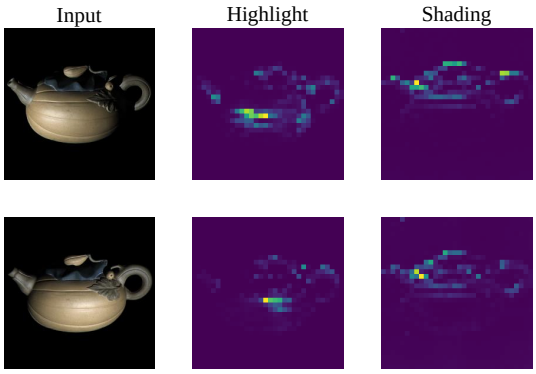

Figure A5: Attention maps for light registers tokens, showcasing their ability to focus on illumination-related regions within the input.

central hypothesis regarding the critical role of a powerful and well-regularized encoder in achieving effective feature disentanglement and consistency.

Uni MS-PS also demonstrates high feature similarity. However, visual analysis of its features (Fig. A8) suggests that they remain considerably entangled with lighting information. Consequently, we infer that its high reported feature similarity may be more attributable to geometric self-consistency within its representations rather than successful illumination decoupling. Despite this apparent lack of complete feature decoupling, Uni MS-PS often achieves commendable reconstruction results. We attribute this primarily to its multi-scale architecture: beyond the initial stage, each subsequent network stage in Uni MS-PS incorporates predicted normals from the preceding stage as an additional input, effectively leveraging them as a strong geometric prior. This iterative refinement, guided by intermediate normal predictions, places Uni MS-PS in a distinct operational paradigm compared to methods like UniPS, SDM-UniPS, and our LiNO-UniPS.

Furthermore, We need to figure out that Uni MS-PS exhibits certain practical limitations. (a) Its multi-scale nature leads to considerable inference latency, particularly when processing multiple high-resolution input images (e.g., handling 16 images at 4K resolution can extend to several hours). In contrast, our LiNO-UniPS method typically completes inference within tens of seconds for similar inputs. (b) While Uni MS-PS can reconstruct detailed surface normals, its reliance on potentially lower-resolution training datasets and its patch-based inference mechanism can lead to a loss of global contextual information, sometimes resulting in reconstructions that are locally detailed but globally inconsistent or erroneous 1.

A1.3.2 Light Registers Tokens

To achieve a more effective decoupling of lighting from normal features within our encoder, we introduce light register tokens. These tokens are specifically designed with the expectation that they will primarily capture and encode illumination information. Fig. A5 presents attention maps derived from the final layer of our encoder. These visualizations demonstrate that the light register tokens indeed attend to diverse illumination-related regions within the input images, including areas characterized by highlights and shading.

We introduce three additional *light registration tokens*: x_{hdri} intended for environment light, x_{point} for point light sources, and x_{area} for area light sources. Fig. A6 shows the attention map visualizations for these different tokens. It is observable that the HDRI register typically attends to broader regions.

The point light register token and the area light register token exhibit similar attentional patterns, both demonstrating a more localized and sharper focus. This shared characteristic aligns well with the generally low-frequency nature of global HDRI illumination and the more localized, often high-frequency, impact typically associated with point and area light sources.

A1.3.3 Limitations

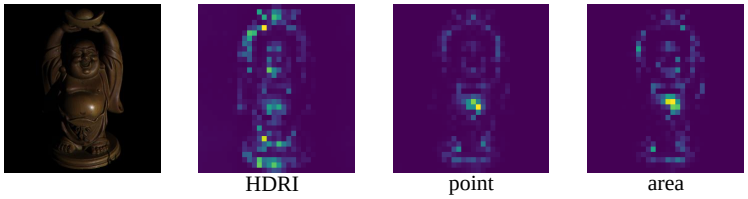

Figure A6: Attention maps for our different light registers tokens reveals distinct focusing patterns. The HDRI register token typically attends to broader, more diffuse regions, while the point light register token and the area light register token exhibit similar behaviors, both demonstrating a more localized and sharper attentional focus.

Figure A7: For near-planar objects possessing intricate concave and convex surface details, our LiNO-UniPS tends to invert the predicted surface normals. The objects are from DiLiGenT-II [54]

Despite the commendable performance demonstrated by LiNO-UniPS, certain limitations nonetheless remain, offering avenues for future research.

Firstly, our incorporation of global attention in the encoder, while aimed at enhancing intra-image feature communication for more effective decoupling of lighting from normal features and successfully improving disentanglement, also introduces computational burden. A key direction for future work is therefore to explore more computationally efficient mechanisms that can achieve comparable decoupling efficacy at a reduced operational cost.

Secondly, a limitation arises in the estimation of surface normals for near-planar objects that exhibit intricate concave and convex surface details, where the performance of LiNO-UniPS can be suboptimal. We attribute this primarily to a fundamental challenge inherent in the Universal Photometric Stereo (PS) paradigm: the absence of explicit light source parameters makes it difficult for the network to unambiguously determine the precise illumination direction (e.g., to distinguish between light originating from above versus below the surface). Consequently, this ambiguity can frequently, and in some instances, even lead to the estimated surface normals being inverted A7. In future investigations, we plan to explore whether more sophisticated lighting alignment strategies could effectively mitigate this specific issue.

A1.3.4 Broader Impacts

Our research offers several positive societal implications. It promises to drive technological advancements in digital content creation, including fields such as virtual reality (VR) and cinematic visual effects (VFX). Moreover, enhanced 3D reconstruction capabilities can bolster machine perception for applications like robotics and autonomous driving, and provide valuable tools for auxiliary scientific research.

Nevertheless, the advanced capabilities described also entail potential negative societal effects. High-fidelity 3D reconstruction, if misused, could be exploited to generate disinformation or fabricate convincing false realities. Furthermore, such technology raises concerns regarding potential infringements on personal privacy, and could pose risks to intellectual property rights and security.

A1.4 Extended Results

In this section, we present additional results to further demonstrate the capabilities of our LiNO-UniPS.

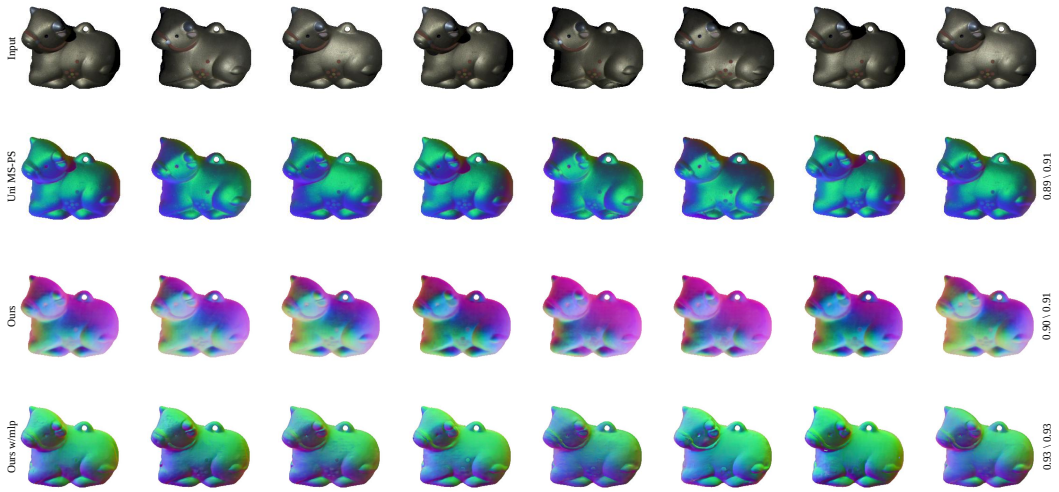

Figure A8: Post-PCA visualization of features extracted by different method encoders for the CowPNG from the DiLiGenT [49]. Metrics displayed to the right of each row is (CSIM/SSIM), higher values indicate higher feature similarity.

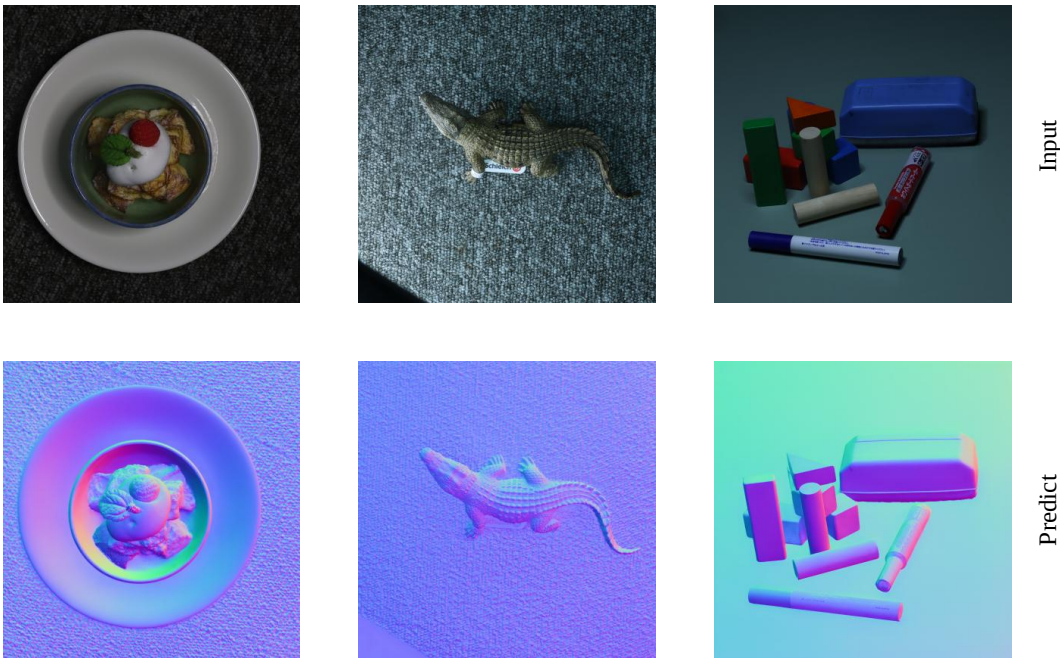

Figure A9: Top row: Example from the input multi-light images. Bottom row: Surface normal map reconstructed by our LiNO-UniPS. The data utilized is from SDM-UniPS [32], and the resolution is 4K .

A1.4.1 Real Data

Fig. A9 shows reconstruction results on challenging real-world data, which was captured from diverse scenes, is mask-free, and features a 4K spatial resolution. These results demonstrate that our LiNO-UniPS is also scale, mask-free and detailed.

Fig. A10 presents examples of real-world captured objects, the overall quality of these reconstructions underscores our approach's strong generalization capabilities.

Fig. A11 presents a comparison of various Universal PS methods on our PS-Verse Testdata. Given that PS-Verse Testdata is a synthetic dataset, ground truth is readily available, facilitating precise quan-

Figure A10: Real-world data with masks and corresponding LiNO-UniPS reconstruction results; data sourced from UniPS [29] and SDM-UniPS [32].

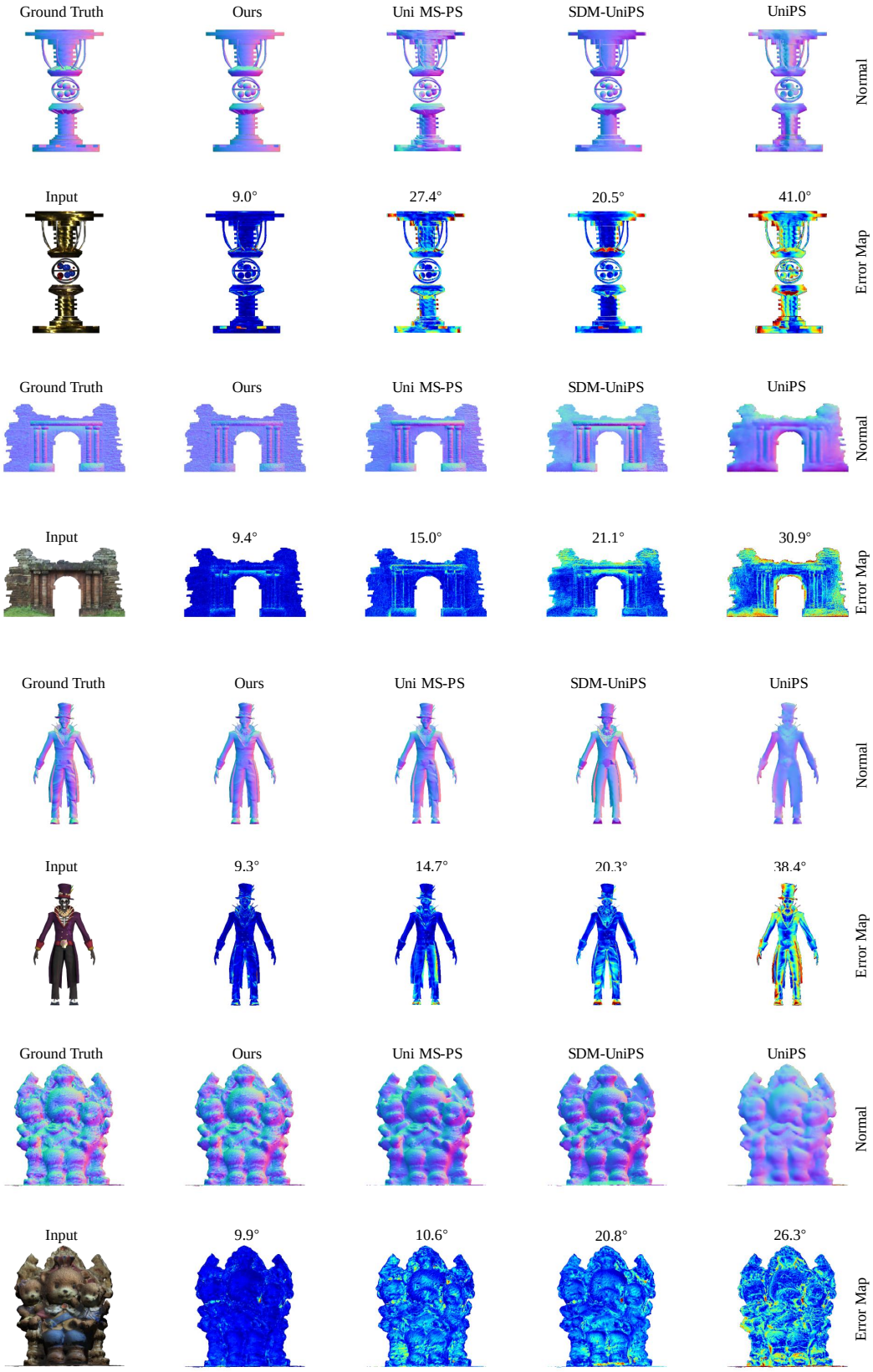

Figure A11: Comparison of different Universal PS methods on our PS-Verse Testdata, showcasing ground truth normals, reconstruction normals, and corresponding error maps. The error maps depict the Mean Angular Error (MAE), measured in degrees; lower MAE values signify a more accurate reconstruction.

titative evaluation. The results clearly demonstrate that our LiNo-UniPS significantly outperforms contemporary approaches, including Uni MS-PS [20], SDM-UniPS [32], and UniPS [29].