Appendix

This appendix provides supplementary materials to support the main manuscript. Section A presents detailed derivations that support the theoretical foundations of our approach. Section B outlines implementation specifics, including training settings, hyperparameters, dataset construction, and other relevant details. Section C provides detailed experimental settings, extended results, and additional ablation studies. Section D presents qualitative results on camera pose estimation, as well as additional visualizations of optimization landscapes, trajectories, and score fields, offering insights into how these factors influence the pose estimation process. Section E discusses limitations and potential directions for future work.

A. Derivation Details

In this section, we provide detailed derivations and clarifications of the mathematical formulations presented in the manuscript. Section A.1 introduces the symbols and notations used throughout. Section A.2 introduces an energy-based formulation as an alternative to our score-based approach. Section A.3 derives the closed-form training objectives corresponding to the learning approaches discussed in Section 4. Section A.4 presents the detailed proofs and theoretical analyses included in the manuscript.

A.1. List of Symbols

This section summarized the symbols and notations used throughout the main manuscript and appendix. Table 6 organizes them together with brief description.

A.2. Energy-based Modeling

To further validate our design choice, we compare the score-based formulation with an energy-based alternative. Instead of explicitly learning the score function, we train an energy network $\mathcal{E}_\theta(\tilde{\mathbf{x}}, \mathbf{y})$ parameterized by θ to model the data distribution through the following objective:

$$\mathcal{L}(\theta) = \tfrac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{y}}\mathbb{E}_{\tilde{\mathbf{x}}\sim \mathrm{U}} \left\| \mathcal{E}_\theta(\tilde{\mathbf{x}}, \mathbf{y}) + \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}, \mathbf{y}) \right\|_2^2 . \quad (11)$$

After training, the gradient of the learned energy function with respect to $\tilde{\mathbf{x}}$ yields the corresponding score, which can be integrated into our two-stage optimization framework. A comparison of this energy-based approach with our proposed score-based design highlights the efficiency and stability advantages of the latter, as it directly learns the score field without requiring differentiation of the energy function. The quantitative results comparing these two modeling approaches are shown in Table 5.

A.3. Closed-Form Expression

Closed-form expressions for the score and energy targets in Eq. (3) and Eq. (11) are derived using a Gaussian perturbation kernel: $p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\tilde{\mathbf{x}}-\mathbf{x}\|^2}{2\sigma^2}\right)$, where d denotes the dimensionality of the data space and σ denotes the standard deviation. The corresponding log-density is given by

$$\log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = C - \tfrac{\|\tilde{\mathbf{x}}-\mathbf{x}\|^2}{2\sigma^2}, \quad (12)$$

where $C = -\frac{d}{2}\log(2\pi\sigma^2)$ is a contant independent of $\tilde{\mathbf{x}}$.

Score-based objective. Differentiation of Eq. (12) with respect to $\tilde{\mathbf{x}}$ yields the score function

$$\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = -\frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2}. \quad (13)$$

Substituting Eq. (13) into the training objective in Eq. (3) yields the explicit expression for the training loss:

$$\mathcal{L}(\theta) = \tfrac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{y}}\mathbb{E}_{\tilde{\mathbf{x}}\sim \mathrm{U}} \left\| s_\theta(\tilde{\mathbf{x}}, \mathbf{y}) + \tfrac{\tilde{\mathbf{x}}-\mathbf{x}}{\sigma^2} \right\|^2 . \quad (14)$$

where $s_\theta(\tilde{\mathbf{x}}, \mathbf{y})$ denotes the score network.

Energy-based objective. To construct the energy target used in energy-based modeling, we first observe that the constant term C in Eq. (12) is independent of the noised input $\tilde{\mathbf{x}}$ and can therefore be omitted during training. Consequently, the objective depends solely on the quadratic term. Substituting this expression into Eq. (11) yields the energy-based training objective

$$\mathcal{L}(\theta) = \tfrac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{y}}\mathbb{E}_{\tilde{\mathbf{x}}\sim \mathrm{U}} \left\| \mathcal{E}_\theta(\tilde{\mathbf{x}}, \mathbf{y}) - \tfrac{\|\tilde{\mathbf{x}}-\mathbf{x}\|^2}{2\sigma^2} \right\|^2 , \quad (15)$$

where $\mathcal{E}_\theta(\tilde{\mathbf{x}}, \mathbf{y})$ denotes the energy network.

A.4. Proofs

Proof of Proposition 1. To derive the optimal score function $s_\theta(\tilde{\mathbf{x}}, \mathbf{y})$, we reformulate the objective in Eq. (7) by rearranging the expectations over $\mathbf{y}$, $\tilde{\mathbf{x}}$, and $\mathbf{x}$, yielding

$$\mathcal{L}(\theta) = \tfrac{1}{2}\mathbb{E}_{\mathbf{y}}\mathbb{E}_{\tilde{\mathbf{x}}}\mathbb{E}_{\mathbf{x}} \left\| s_\theta(\tilde{\mathbf{x}}, \mathbf{y}) - \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \right\|^2 , \quad (16)$$

where the expectations are taken over $\mathbf{y} \sim p(\mathbf{y})$, $\tilde{\mathbf{x}} \sim p(\tilde{\mathbf{x}} \mid \mathbf{y})$, and $\mathbf{x} \sim p(\mathbf{x} \mid \mathbf{y}, \tilde{\mathbf{x}})$. For fixed $(\tilde{\mathbf{x}}, \mathbf{y})$, the inner expectation defines a quadratic loss:

$$\mathcal{L}(s) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y},\tilde{\mathbf{x}})} \left\| s - \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \right\|^2 , \quad (17)$$

where $s = s_\theta(\tilde{\mathbf{x}}, \mathbf{y})$ is treated as the optimization variable. By minimizing the quadratic loss in Eq. (17) and setting $\nabla_s \mathcal{L}(s) = 0$, we obtain the optimal solution:

$$s^\star(\tilde{\mathbf{x}}, \mathbf{y}) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y},\tilde{\mathbf{x}})}[\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})]. \quad (18)$$

Substituting the closed-form expression of $p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$ from Eq. (13) into Eq. (18) yields

$$\begin{aligned} s^\star(\tilde{\mathbf{x}}, \mathbf{y}) &= \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y},\tilde{\mathbf{x}})}[\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})] \\ &= \int \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x} \mid \mathbf{y}, \tilde{\mathbf{x}})\, d\mathbf{x} \\ &= \frac{\int (\mathbf{x} - \tilde{\mathbf{x}})\, p(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x} \mid \mathbf{y})\, d\mathbf{x}}{\sigma^2 \int p(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x} \mid \mathbf{y})\, d\mathbf{x}} . \end{aligned} \quad (19)$$

Table 6. List of symbols and their corresponding descriptions

Symbol	Description
$\|u\| = \sqrt{u^T u} = \sqrt{\sum_i u_i^2}$	Euclidean norm of vectors u.
$\mathbf{I}_d$	$d \times d$ identity matrix.
$\mathbf{x} \in \mathbb{R}^d$	Data sample with dimension d.
$\tilde{\mathbf{x}} \in \mathbb{R}^d$	Perturbed data sample.
$\{\mathbf{x}^{(i)}\}_{i=1}^N$	A dataset consisting of N i.i.d. samples.
$\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$	The dataset consists of N pairs, where each pair includes a relative camera pose $\mathbf{x}^{(i)}$ condition on an image pair $\mathbf{y}^{(i)}$.
(Θ, Φ, ρ)	Spherical coordinates representing camera pose.
$\nabla_{\mathbf{x}} f(\mathbf{x})$	Gradient of function f with respect to $\mathbf{x}$.
$\mathcal{E}(\mathbf{x})$	Energy function.
$s(\mathbf{x})$	Score function.
$\delta(\mathbf{x})$	Dirac-delta function.
$p_{\text{data}}(\mathbf{x})$	True underlying distribution of data samples $\mathbf{x}$.
$p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 \mathbf{I}_d)$	Isotropic Gaussian kernel, where σ^2 denotes the variance.
$p(\mathbf{x} \mid \mathbf{y})$	Conditional distribution of the camera pose $\mathbf{x}$ given the image pair $\mathbf{y}$.
$SE(3)$	The Lie group representing 3D rigid body transformations.
$\mathfrak{se}(3)$	The Lie algebra associated with $SE(3)$.
$T \in SE(3)$	Transformation matrix between two coordinate systems.
$\xi \in \mathfrak{se}(3)$	Twist coordinates in the Lie algebra $\mathfrak{se}(3)$.
$\text{Exp}(\cdot)$	The exponential map from $\mathfrak{se}(3)$ to $SE(3)$.

In the last step in Eq. (19), we used the fact that the perturbation $\tilde{\mathbf{x}}$ depends only on the original variable $\mathbf{x}$ and is independent of $\mathbf{y}$. Formally, this means $p(\tilde{\mathbf{x}} \mid \mathbf{y}, \mathbf{x}) = p(\tilde{\mathbf{x}} \mid \mathbf{x})$, which allows us to factor $p(\mathbf{x} \mid \mathbf{y}, \tilde{\mathbf{x}})$ as $p(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x} \mid \mathbf{y})$ in the integral. Hence, the optimal score function of Eq. (7) takes the form above, as stated in Proposition 1. □

Proof of Lemma 1. Since the perturbation variable $\tilde{\mathbf{x}}$ is drawn from a uniform distribution U that is independent of $\mathbf{x}$, we have $p(\mathbf{x} \mid \mathbf{y}, \tilde{\mathbf{x}}) = p(\mathbf{x} \mid \mathbf{y})$. Substituting this into Eq. (18) from Proposition 1, we obtain

$$s_{\text{U}}^\star(\tilde{\mathbf{x}}, \mathbf{y}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x} \mid \mathbf{y})} \left[\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \right]. \tag{20}$$

Using the analytic form of $p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$ from Eq. (13), Eq. (20) can be expressed in the following explicit integral form:

$$s_{\text{U}}^\star(\tilde{\mathbf{x}}, \mathbf{y}) = \frac{1}{\sigma^2} \int (\mathbf{x} - \tilde{\mathbf{x}})\, p(\mathbf{x} \mid \mathbf{y})\, d\mathbf{x}. \tag{21}$$

This concludes the derivation and verifies Lemma 1. □

Proof of Lemma 2. By Bayes' rule and the conditional independence of $\tilde{\mathbf{x}}$ from $\mathbf{y}$ given $\mathbf{x}$, the posterior distribution in Eq. (18) can be written in the following form

$$p(\mathbf{x} \mid \mathbf{y}, \tilde{\mathbf{x}}) = \frac{p(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x} \mid \mathbf{y})}{\int p(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x} \mid \mathbf{y})\, d\mathbf{x}}. \tag{22}$$

The posterior depends on both the conditional prior $p(\mathbf{x} \mid \mathbf{y})$ and the likelihood $p(\tilde{\mathbf{x}} \mid \mathbf{x})$. In general, if the conditional prior $p(\mathbf{x} \mid \mathbf{y})$ has support on multiple values of $\mathbf{x}$, the posterior distribution in Eq. (22) differs from the prior. Consequently, the optimal score functions in Eqs. (18) and (20) are generally different: $s^\star(\tilde{\mathbf{x}}, \mathbf{y}) \neq s_{\text{U}}^\star(\tilde{\mathbf{x}}, \mathbf{y})$. However, in the special case where each conditional prior collapses to a Dirac delta, $p(\mathbf{x} \mid \mathbf{y}^{(i)}) = \delta(\mathbf{x} - \mathbf{x}^{(i)})$, each conditional image pair $\mathbf{y}^{(i)}$ corresponds to a single ground-truth pose $\mathbf{x}^{(i)}$. Substituting the Dirac delta function into Eq. (22) yields

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{y}^{(i)}, \tilde{\mathbf{x}}) &= \frac{p(\tilde{\mathbf{x}} \mid \mathbf{x})\, \delta(\mathbf{x} - \mathbf{x}^{(i)})}{\int p(\tilde{\mathbf{x}} \mid \mathbf{x})\, \delta(\mathbf{x} - \mathbf{x}^{(i)})\, d\mathbf{x}} \\ &= \frac{p(\tilde{\mathbf{x}} \mid \mathbf{x})\, \delta(\mathbf{x} - \mathbf{x}^{(i)})}{p(\tilde{\mathbf{x}} \mid \mathbf{x}^{(i)})} \\ &= \delta(\mathbf{x} - \mathbf{x}^{(i)}). \end{aligned} \tag{23}$$

In this case, the posterior coincides with the prior, i.e., $p(\mathbf{x} \mid \mathbf{y}^{(i)}) = p(\mathbf{x} \mid \mathbf{y}^{(i)}, \tilde{\mathbf{x}})$. As a result, the optimal score functions are identical: $s^\star(\tilde{\mathbf{x}}, \mathbf{y}) = s_{\text{U}}^\star(\tilde{\mathbf{x}}, \mathbf{y})$. □

Proof of Eq. (5). We assume the learned score function approximates the true score $s_\theta(\tilde{\mathbf{x}}, \mathbf{y}) = (\mathbf{x} - \tilde{\mathbf{x}})/\sigma^2$ for each $(\mathbf{x}, \mathbf{y})$ in the dataset. For clarity in the following analysis, we denote the ground-truth pose associated with the conditional image pair $\mathbf{y}$ as $\mathbf{x}_{\text{gt}}$. In our experiments, we set $\sigma = 1$. Substituting these into Eq. (4) yields the update equation:

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \alpha(\mathbf{x}_{\text{gt}} - \tilde{\mathbf{x}}_{t-1}) + G\mathbf{z}_t, \quad \mathbf{z}_t \sim \mathcal{N}(0, \mathbf{I}_3). \tag{24}$$

Starting from the initial point $\tilde{\mathbf{x}}_0$, the iterative formula in Eq. (24) updates the noisy prediction $\tilde{\mathbf{x}}_{t-1}$ to $\tilde{\mathbf{x}}_t$. Taking the expectation on both sides of Eq. (24) gives

$$\mathbb{E}[\tilde{\mathbf{x}}_t] = (1-\alpha)\,\mathbb{E}[\tilde{\mathbf{x}}_{t-1}] + \alpha\,\mathbf{x}_{\text{gt}}, \tag{25}$$

where the noise term vanishes since $\mathbb{E}[\mathbf{z}_t] = \mathbf{0}$. Eq. (25) can be rewritten as the following recurrence:

$$\delta_t = (1-\alpha)\,\delta_{t-1}, \tag{26}$$

where $\delta_t = \mathbb{E}[\tilde{\mathbf{x}}_t - \mathbf{x}_{\text{gt}}]$. Since both $\tilde{\mathbf{x}}_0$ and $\mathbf{x}_{\text{gt}}$ are fixed, the expectation is redundant at $t = 0$, giving $\delta_0 = \tilde{\mathbf{x}}_0 - \mathbf{x}_{\text{gt}}$. Solving Eq. (26) gives $\delta_t = (1-\alpha)^t \delta_0$. Substituting the definition of δ_t and taking norms on both sides then yields

$$\|\mathbb{E}[\tilde{\mathbf{x}}_t - \mathbf{x}_{\text{gt}}]\| = M\,(1-\alpha)^t, \tag{27}$$

where $M = \|\tilde{\mathbf{x}}_0 - \mathbf{x}_{\text{gt}}\|$ denotes the initial prediction error and provides an upper bound on the expected distance from the ground-truth pose. Next, we compute the variance of the noised prediction $\tilde{\mathbf{x}}_t$. By taking the variance on both sides of the iterative update in Eq. (24), we obtain

$$\begin{aligned}\text{Var}[\tilde{\mathbf{x}}_t] &= \text{Var}[(1-\alpha)\,\tilde{\mathbf{x}}_{t-1} + G\,\mathbf{z}_t] \\ &= (1-\alpha)^2\,\text{Var}[\tilde{\mathbf{x}}_{t-1}] + G\,\text{Var}[\mathbf{z}_t]\,G^\top. \end{aligned} \tag{28}$$

Since $\mathbf{z}_t \sim \mathcal{N}(0, \mathbf{I}_3)$, we have $\text{Var}[\mathbf{z}_t] = \mathbf{I}_3$. As the multiplicative factor $(1-\alpha)^2 < 1$, the variance recursion is guaranteed to converge. By setting $\text{Var}[\tilde{\mathbf{x}}_t] = \text{Var}[\tilde{\mathbf{x}}_{t-1}] = \Sigma$ and using the fact that G is a diagonal matrix so that $G\,G^\top = G^2$, we obtain the equation $\Sigma = (1-\alpha)^2\Sigma + G^2$. By solving the equation for Σ, the solution is

$$\Sigma = \frac{G^2}{1-(1-\alpha)^2} = \frac{G^2}{2\alpha-\alpha^2} \approx \frac{G^2}{2\alpha}. \tag{29}$$

With the expectation result in Eq. (27) and the variance result in Eq. (29), the dynamics of the iterative update are fully characterized. This completes the derivation and thus concludes the proof of Eq. (5). □

Eq. (5) indicates that, under an accurate score approximation, the distance between the predicted pose and the ground-truth pose decays exponentially, while the variance is governed by the coordinate-wise noise scales γ_i. These results provide a theoretical justification for the design of our score-based first-stage optimization.

B. Implementation Details

In this section, we provide implementation details to support reproducibility. Section B.1 introduces the spherical coordinate system used to represent camera poses in our work. Section B.2 describes the dataset construction and rendering procedures. Section B.3 details the training setup, model architecture, and our proposed two-stage optimization strategy. Section B.4 summarizes the computation resources employed in our experiments..

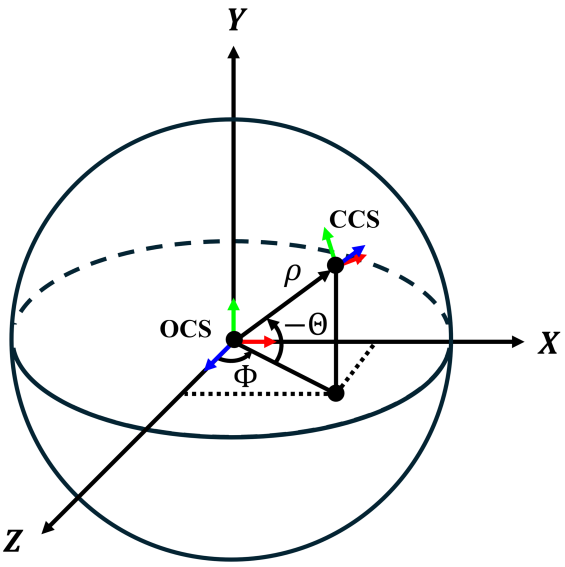

Figure 7. **Spherical Coordinate System. CCS** denotes camera coordinate system and **OCS** denotes object coordinate system.

B.1. Coordinate System

Following iFusion [59], we represent camera positions and their relative transformations using a spherical coordinate system centered at the object's origin, as illustrated in Fig. 7. The coordinates Θ, Φ, and ρ denote the vertical angle from the equatorial plane, the horizontal angle within that plane, and the radial distance from the origin, respectively. Since we assume an object-centric camera pose, the extrinsic matrix between the camera coordinate system (CCS) and the object coordinate system (OCS) is uniquely determined by the camera's location in this spherical coordinate system. During training, given two images captured from different viewpoints with camera positions $(\Theta_1, \Phi_1, \rho_1)$ and $(\Theta_2, \Phi_2, \rho_2)$, we define their relative transformation as the difference vector $(\Theta_2 - \Theta_1, \Phi_2 - \Phi_1, \rho_2 - \rho_1)$. For clarity and ease of interpretation, we refer to Θ as latitude and Φ as longitude throughout this paper to more intuitively represent the camera pose.

B.2. Dataset Construction

GSO and OO3D. We construct our training and testing datasets using 3D models from the GSO and OO3D datasets, following the data preparation pipeline of iFusion. Both datasets use the same 70 objects as in the iFusion dataset. To assess the generalization capability of our framework, we also selected 10 additional objects from the GSO dataset that were not used during training. Each object was normalized to fit within a unit cube by scaling its maximum side length to one. Rendering was performed using Pyrender [31] with a perspective camera whose field of view (FoV) was set to $49.1°$. All images were rendered

at a resolution of 512×512 pixels with transparent backgrounds. Camera positions were defined using the spherical coordinate system introduced in Section B.1.

The training data was generated by uniformly sampling camera viewpoints on the unit sphere. The latitude Θ were sampled from the range $[-30°, 30°]$ in $7.5°$ increments, and the longitude Φ from $[0°, 360°]$ in $15°$ increments. The camera distance ρ was uniformly sampled from the interval $[1.2, 2.0]$. This resulted in a total of 15120 training images for each dataset. The testing viewpoints were randomly sampled from the same parameter ranges, with latitude $\Theta \in [-30°, 30°]$, longitude $\Phi \in [0°, 360°]$, and camera distance $\rho \in [1.2, 2.0]$. For each object, 8 viewpoints were randomly selected, resulting in a total of 560 testing images under this configuration.

HOPEv2. For the HOPEv2 dataset, which contains 28 objects, we uniformly sampled 100 images per object from the original dataset to cover diverse viewpoints, resulting in a total of 2800 training images. For testing, 4 images per object were selected, yielding a total of 112 testing images. To satisfy the object-centric assumption, each image was cropped around the object, and the corresponding ground-truth annotations were adjusted accordingly. The images were preprocessed using the object masks to isolate the objects from the background.

B.3. Training and Evaluation

Model Architecture and Hyperparameters. The score network is conditioned on an image pair and a relative noised pose. The reference and query images are first passed through a ResNet-50 [15] backbone to extract 2048-dimensional feature vectors for each image. The conditioned pose is embedded using a sinusoidal positional encoding, and these features are concatenated to form the model input. The concatenated input is then passed through a lightweight MLP with residual connections [15], ReLU activations, and layer normalization [1]. The number of hidden channels is set to 256. The network outputs a three-dimensional vector corresponding to the conditional score of the data distribution. Through empirical tuning, we found that this architecture is sufficient to accurately learn the score function and effectively guide the perturbed pose toward the ground-truth pose.

Our implementation is based on PyTorch [38]. For all experiments, we used the Adam [23] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 0.0001, a batch size of 256, and 60 epochs. For each object, we consider all possible pairs of image. For each image pair, the noised pose is randomly sample from the uniform distribution $U = [-\frac{\pi}{3}, \frac{\pi}{3}] \times [0, 2\pi] \times [1.2, 2.0] \subset \mathbb{R}^3$. In the second stage of our two-stage optimization pipeline, the pose-conditioned diffusion model uses Zero123-XL as its 3D geometric prior. Comparison with other alternatives are provided in Section C.4.

Two Stage Optimization. After training, the learned score function can be used to guide arbitrary initial poses toward higher-density regions of the pose distribution. The first-stage optimization is performed by iteratively applying the update rule defined in Eq. (4). In practice, we set the hyperparameters to $\alpha = 0.1, \gamma_1 = \gamma_3 = 0, \gamma_2 = 0.3$, and run the optimization for 50 iterations.

In the second stage, following iFusion, the prediction is refined using the pretrained Zero123 model with an MSE loss. This involves computing the gradient with respect to the pose condition (i.e., the noised pose estimate), followed by applying a gradient-based iterative solver for optimization. To ensure that the estimated pose remains on the $\mathrm{SE}(3)$ manifold during this process, we parameterize it as $T_{r\to q} = \mathrm{Exp}(\xi)$, where $\xi \in \mathbb{R}^6$ denotes the twist vector in the Lie algebra [48] $\mathfrak{se}(3)$ associated with the Lie group $\mathrm{SE}(3)$. The exponential map $\mathrm{Exp}(\cdot)$ converts the twist into a corresponding rigid body transformation in $SE(3)$. The loss function is defined as:

$$\hat{\xi} = \mathrm{argmin}_{\xi \in \mathfrak{se}(3)} \mathcal{L}(I_q, (I_r, \mathrm{Exp}(\xi))) + \mathcal{L}(I_r, (I_q, \mathrm{Exp}(-\xi))), \quad (30)$$

where $\mathcal{L}(I_q, (I_r, \mathrm{Exp}(\xi))) = \mathbb{E}_{\mathbf{z},\epsilon,t}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, (I_r, \mathrm{Exp}(\xi)))\|_2^2\right]$, ϵ is drawn from an isotropic Gaussian distribution, $t \in [0, 1, \ldots, 99]$, $\mathbf{z}$ denoting the latent representation of the query image I_q. The Adam optimizer is applied with a learning rate of 0.1 for 50 iterations. A `ReduceLROnPlateau` scheduler with a decay factor of 0.7 and a patience of 10 adaptively reduces the learning rate when the optimization plateaus. Each iteration corresponds to a diffusion timestep, obtained by linearly mapping the iteration index to the predefined diffusion time range.

B.4. Computation Resources

The training of our model was performed using a single NVIDIA RTX 6000 Ada GPUs, with 48 GB of memory, together with an AMD EPYC 7313 16-core CPU. Training the score model on the 70-object GSO dataset (Table 1) takes approximately 1.5 days.

C. Additional Experimental Results

In this section, we provide further experimental information to supplement the results reported in the main manuscript. Section C.1 describes the processing used for comparison with the baseline methods. Section C.2 presents a comparison of inference time between iFusion and our method, both of which are gradient-based solver. Section C.4 presents further ablation studies of our proposed approach.

C.1. Details of Baseline Settings

Our proposed method is compared against three baseline approaches: iFusion [59], DUSt3R [57] and VGGT [56].

DUSt3R and VGGT estimates relative camera transformations from monocular image pairs, and the absolute scale of the translation cannot be determined without additional depth information. In our work, we follow the settings of Zero123 and iFusion, and learn a normalized representation of the object to predict a normalized camera scale, which is then refines in a second stage using Zero123. To ensure a fair comparison, we determine the optimal scale for these baseline methods. This ensures consistency with the assumed camera pose representation in our framework.

C.2. Analysis of Inference Time

Table 8 compares the inference time of iFusion and our proposed method across different numbers of initial poses. For comparable recall values, our method requires significantly less computation time due to the sample efficiency of our two-stage optimization. For instance, to achieve a recall near 0.90, our approach takes only 12.86 seconds with 2 initial poses, whereas iFusion requires 91.92 seconds with 8 initial poses to achieve similar performance. Even for the same number of initial poses, our approach is faster because of its lightweight score model, compared to the heavy Zero123 diffusion UNet. These result demonstrates that our method is more robust to variations in the initial poses and achieves comparable performance with substantially fewer samples, resulting in a substantial reduction in runtime.

C.3. Additional Quantitative Results

Table 7 provides further comparisons with ID-Pose and iFusion on the large-scale CO3Dv2 3D category dataset. Training and testing data are sampled from the CO3Dv2 single-sequence subset, comprising 29 scenes, with viewpoints uniformly sampled for each scene. Our framework continues to produce strong results across all metrics, notably achieving the highest success rate among all methods.

C.4. More Ablation Studies

Ablation on Noise Scale γ. This ablation demonstrates that the noise scale γ in Eq. (4) provides a mechanism to trade off between recall and success rate. As shown in Table 9, increasing the noise scale leads to higher recall but decreases success rate. This is because a larger noise scale increases the variance of the first stage predictions, as indicated in Eq. (5), allowing the second-stage energy-based optimization to start from more diverse initial points. Some of these diverse points may lie closer to the global optimum, thereby improving recall. However, others may start farther from feasible or optimal regions, which can reduce the success rate. Overall, the noise scale γ effectively controls the balance between exploration (recall) and convergence stability (success rate). In our experiments, we also found that adding noise to the latitude and radius coordinates significantly decreases recall. We conjecture that this is because the energy landscape is steeper along these dimensions, so even small perturbations in latitude or radius can cause large changes in energy, resulting in implausible starting points for second stage refinement.

Ablation on Pose-Conditioned Priors. We study different pose-conditioned diffusion models for second-stage refinement, including Zero123 and Zero123-XL. As shown in Table 10, using Zero123-XL consistently improves performance across all evaluation metrics. This improvement occurs because the refinement stage relies on energy-based optimization, whose landscape is implicitly defined by the pretrained pose-conditioned diffusion model. In other words, the optimization aims to determine the relative pose that best matches the query view. Consequently, a stronger generative prior naturally leads to a more effective and robust refinement process.

Failure Case Analysis. While our method significantly improves the overall success rate and recall across most objects, there remain certain failure cases where performance degrades. To better understand these issues, we analyze two representative examples.

As shown in the Figs. 8 (a) and (b), the Bowl object exhibits a continuous symmetrical shape along its axis and features repetitive surface patterns. Even for humans, estimating its horizontal rotation angle is challenging due to the lack of distinctive visual cues. To investigate whether the performance drop stems from limitations of Zero123, we visualize novel views of the object generated by Zero123, as shown in Fig. 8 (c). The results indicate that Zero123 also struggles to generate meaningful and consistent views for this object, particularly when varying the longitude. Therefore, the refinement stage no longer provides any improvement in this case.

The second example involves the Screwdriver object, as shown in Fig. 9. Unlike the previous case, this failure arises from the pose representation used by Zero123. Since Zero123 conditions on the difference in spherical coordinates between two views, it must implicitly estimate the pose of the object in the reference image to uniquely define the novel view's pose. However, in this example, the camera pose in reference image is positioned directly in front of the screwdriver, making it impossible to determine the object's latitude. To better understand this issue, we vary ϕ from 0° to 315° in 45° increments and generate images using Zero123, as shown in Fig. 9 (d). From these results, we hypothesize that Zero123 interprets the camera pose of the reference image as being from above the object. In reality, the ground-truth θ value is nearly 0°, indicating that the camera is positioned at the side. The ground-truth views for this object are shown in Fig. 9 (c).

Table 7. **Evaluation results on the CO3Dv2 dataset.** For large-scale, scene-level evaluation, we compare the Zero123 gradient-based method on 29 sampled CO3Dv2 scenes. **Red** marks the best result, and **blue** the second-best.

Dataset	Method	@5		@15		@30		@5		@15		@30		Rot. ↓	Trans. ↓
		R ↑	R(R) ↑	R ↑	R(R) ↑	R ↑	R(R) ↑	SR ↑	SR(R) ↑	SR ↑	SR(R) ↑	SR ↑	SR(R) ↑		
CO3Dv2	ID-Pose [6]	0.022	0.035	0.134	0.297	0.181	0.491	0.004	0.006	0.030	0.070	0.050	0.140	30.95	0.267
	iFusion [59]	**0.103**	**0.116**	**0.392**	**0.522**	**0.509**	**0.711**	**0.041**	**0.052**	**0.176**	**0.233**	**0.237**	**0.334**	**13.98**	**0.160**
	Ours	**0.069**	**0.078**	**0.401**	**0.517**	**0.599**	**0.884**	**0.054**	**0.071**	**0.344**	**0.469**	**0.567**	**0.838**	**14.59**	**0.150**

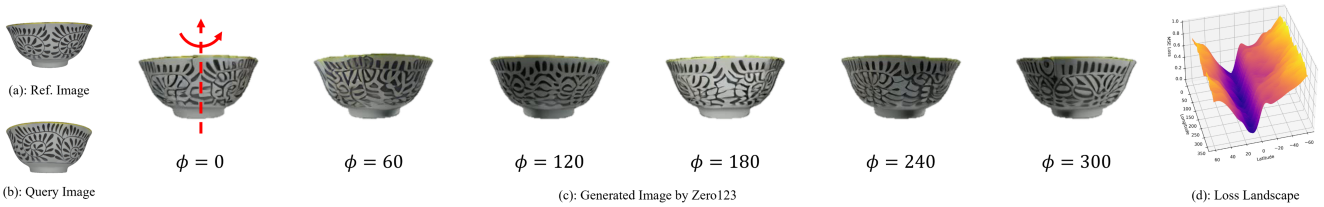

Figure 8. (a) Reference image and (b) query image of the object, captured from different camera poses. (c) Image generated by Zero123 by varying ϕ from 0° to 300° in 60° increments. (d) Corresponding MSE loss landscape.

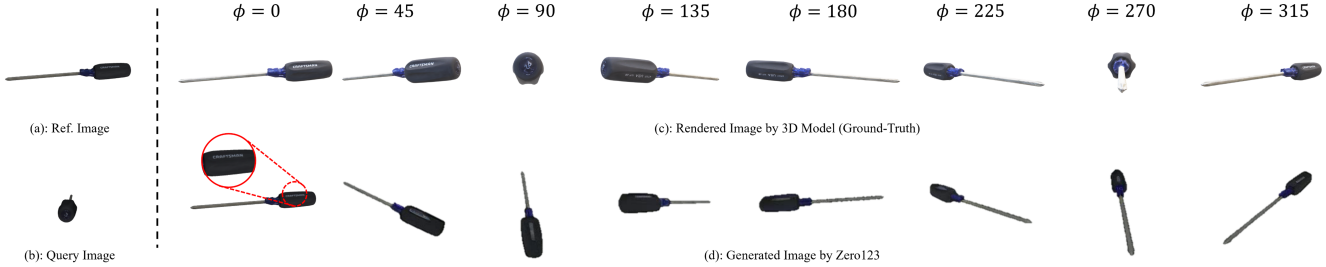

Figure 9. (a) Reference image and (b) query image of the object, captured from different camera poses. (c) Rendered image from the 3D CAD model. (d) Image generated by the Zero123 model. Both vary ϕ from 0° to 315° in 45° increments.

Table 8. **Comparison of inference time and performance.** We report recall (**R@30**) and inference time for different numbers of initial poses for each method. Our method achieves comparable performance while requiring substantially less inference time.

Method	iFusion				Ours			
# of Initial Poses	2	4	6	8	2	4	6	8
R@30	0.661	0.817	0.893	0.907	0.901	0.913	0.943	0.938
Time (s)	23.30	45.89	69.72	91.92	12.86	25.61	36.97	51.29

Table 9. **Ablation on the noise scale γ.** The noise scale γ controls the magnitude of the noise added during each iteration, which in turn affects the variance of the first-stage predictions. Here, we ablate the noise scale for the longitude dimension. Higher noise scales lead to higher recall, whereas lower noise scales yield higher success rates. **Bold** indicates the best result for each metric.

		R@5	R@15	R@30	SR@5	SR@15	SR@30	Rot.	Trans.
γ_2	**0**	0.593	0.864	0.884	0.562	**0.818**	0.835	4.11	0.049
	0.1	0.629	0.866	0.886	**0.582**	0.808	**0.836**	3.61	0.048
	0.3	0.645	**0.907**	**0.927**	0.501	0.728	0.755	3.57	**0.045**
	0.5	**0.677**	**0.907**	0.923	0.395	0.582	0.612	**3.41**	0.048

D. Visualization

In this section, additional visualizations are provided. Section D presents the Zero123 MSE loss landscape to supplement the analysis in the main manuscript. Section D.2 illustrates the optimization trajectories. Section D.3 shows qualitative comparisons between iFusion and our method on camera pose estimation. Section D.4 visualizes the learned score and energy fields. Collectively, these results offer deeper insights into the optimization dynamics and demonstrate the effectiveness of our proposed approach.

Table 10. **Ablation on pose-conditioned priors**. Comparison of different pose-conditioned diffusion models used for second-stage refinement.

	R@5	R@15	R@30	SR@5	SR@15	SR@30	Rot.	Trans.
Zero123-XL	**0.645**	**0.907**	**0.927**	**0.501**	**0.728**	**0.755**	**3.57**	**0.045**
Zero123	0.534	0.880	0.904	0.389	0.659	0.699	4.52	0.054

D.1. Additional Zero123 MSE Landscape Visualization

We provide further details on visualizing the MSE loss landscape with Zero123, as shown in Fig. 1. Given a pair of images, we follow the Zero123 noise prediction pipeline illustrated in Fig. 3 (b) of the main manuscript to compute MSE loss. To explore the loss landscape, we sweep the conditioning pose over latitude from $-60°$ to $60°$ in steps of $8°$ and longitude from $0°$ to $360°$ in steps of $12°$. At each pose, the MSE loss is computed and averaged over five random noise samples. Additional examples are shown in Fig. 10,

offering a more comprehensive visualization of the results. These extended visualizations further demonstrate that the Zero123 MSE loss landscape contains multiple local minima and flat regions (plateaus).

D.2. Visualization of Optimization Trajectories

Additional examples corresponding to Fig. 2 (d) are presented in Fig. 11. These visualization clearly demonstrate the local minimum issue that occurs when directly optimizing the conditioning pose using Zero123 with the MSE loss. For each case, the optimization process is initialized from four different starting poses at longitudes $0°$, $90°$, $180°$, and $270°$. In most cases, only one or two of the initial points converge to the ground-truth pose, while the others become trapped in local minima.

D.3. Additional Qualitative Results

Additional qualitative results comparing iFusion and our method are shown in Fig. 12. These results clearly demonstrate that our gradient-based two-stage optimization effectively avoids local minima and consistently guides the pose updates toward the ground-truth pose. Importantly, our method is robust to different initial points and achieves strong performance across a wide range of starting poses, on par with state-of-the-art approaches.

D.4. Visualization of Score Field

In Fig. 13, we visualize the score and energy fields from the energy-based method, alongside the score field from the score-based method. The energy-based model obtains the score by differentiating the learned energy, while the score-based model directly learns it via score matching, resulting in a consistently more accurate score field.

E. Limitations

From the failure case analysis in Section C.4, we identify two limitations of our method. The first limitation arises from the limited generative capability of Zero123 model on specific objects. Due to multi-view inconsistencies in Zero123, refinement can be challenging for such objects. Some recent works [5, 46] address this issue and show promising improvements. Combing these pose-conditioned diffusion models with our score-based guidance framework suggests a promising direction for future work. The second limitation concerns the pose representation used in Zero123. The model represent relative poses in polar coordinates without explicitly defining an object coordinate system. This leads to ambiguity, especially for symmetric objects. Consequently, the same relative pose may correspond to multiple plausible target views, making pose-conditioned generation inherently ambiguous and adversely affecting our method's performance. We leave these two limitations as opportunities for future improvement.

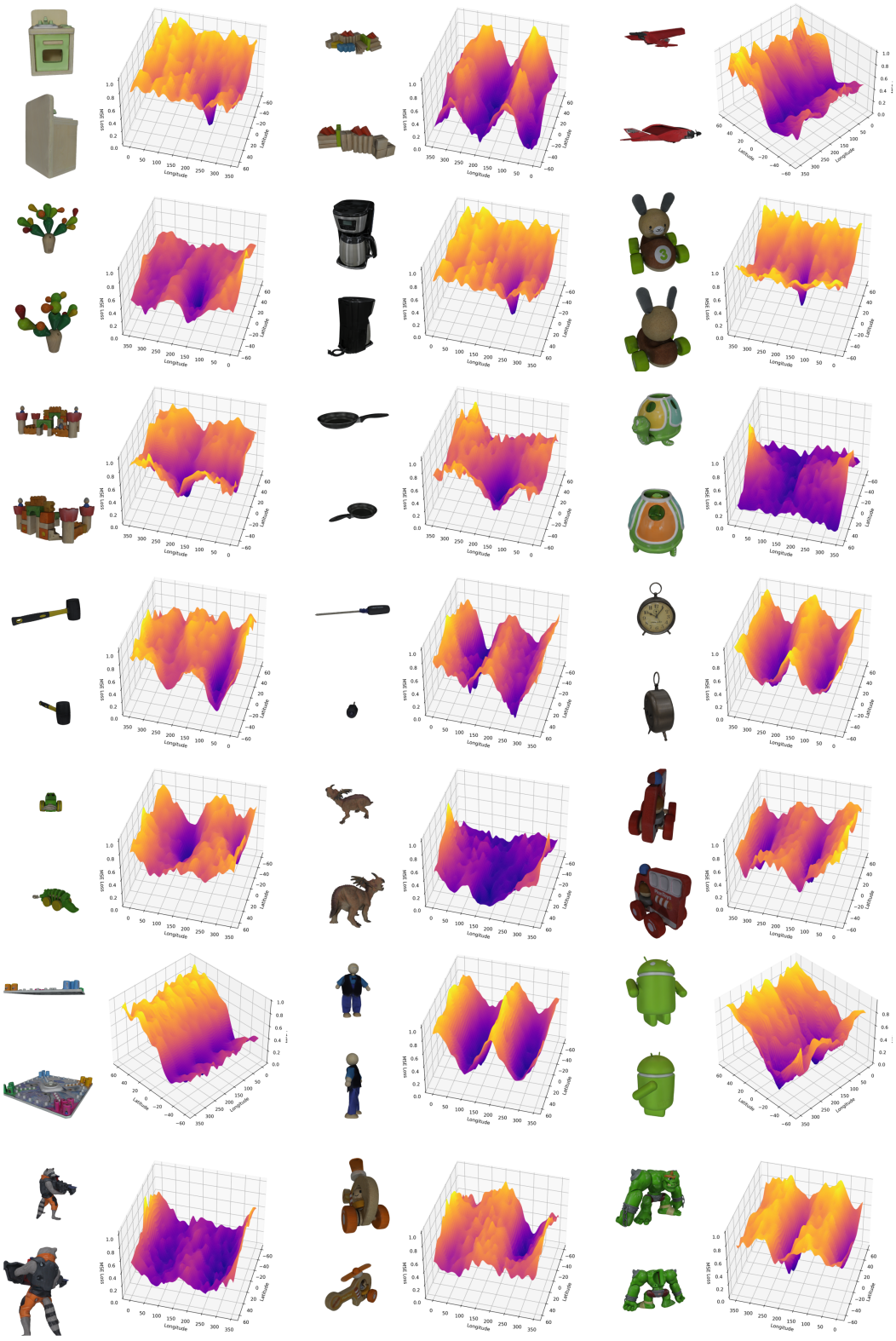

Figure 10. **3D MSE Landscape.** Using image pairs from the GSO dataset [10], we visualize the Zero123 MSE loss landscape for each object. The plots clearly reveal the presence of local minima and plateau regions, highlighting inherent optimization challenges.

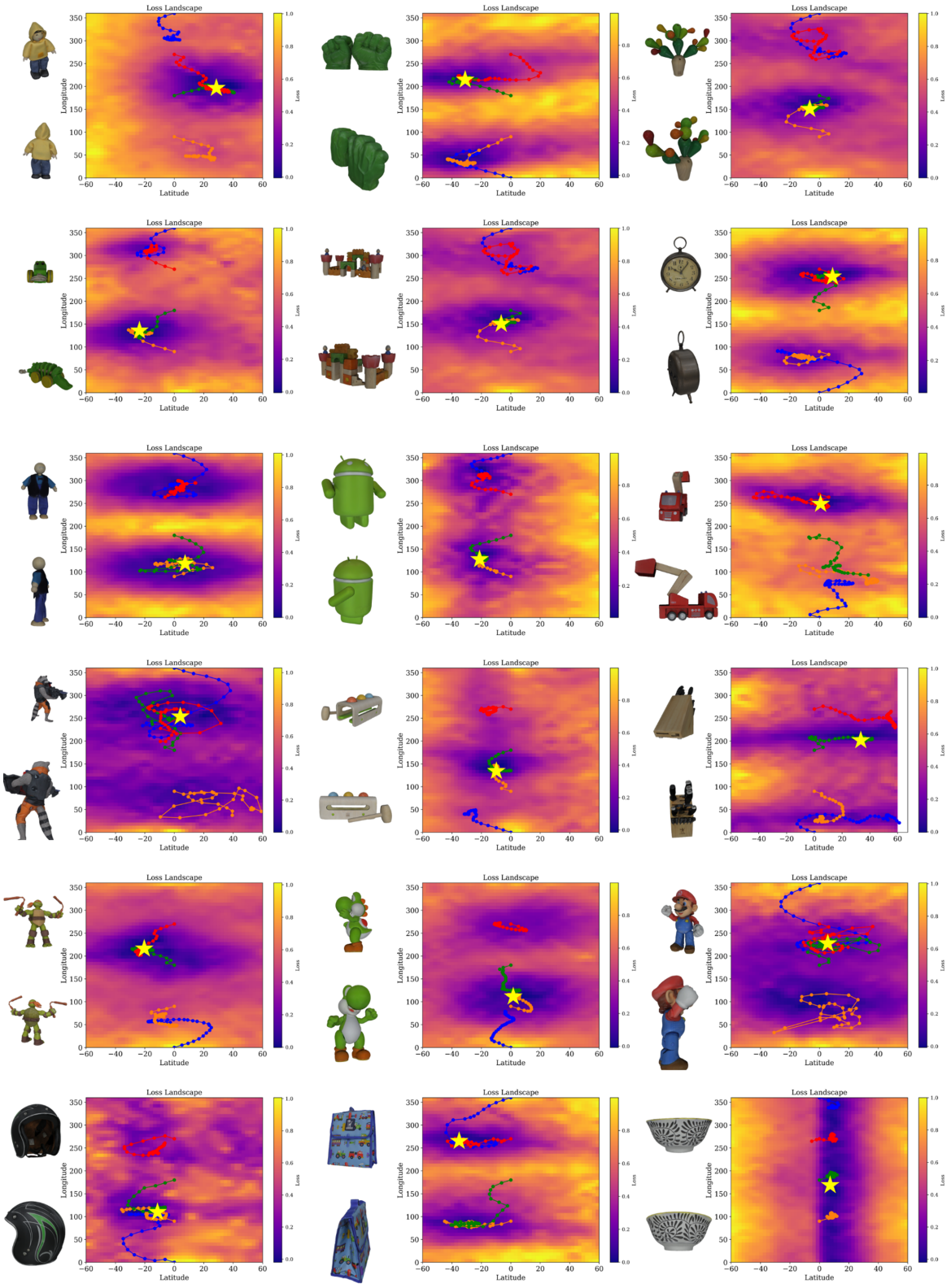

Figure 11. **Optimization Trajectories.** Trajectories generated by optimizing the conditioning pose with Zero123 using MSE loss. Starting from four initial longitudes (0°, 90°, 180°, 270°), only some converge to the ground-truth pose, while others fall into local minima.

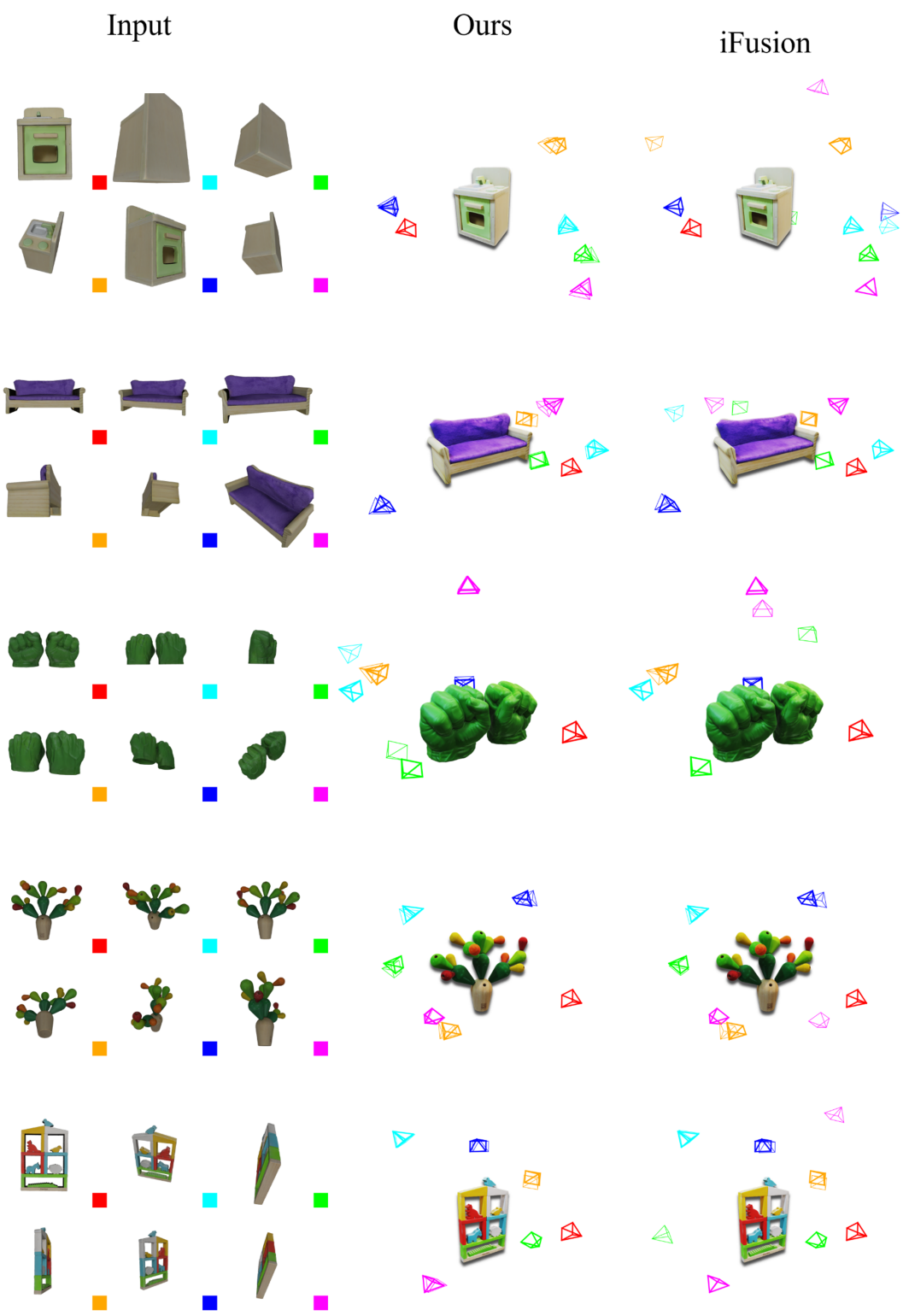

Figure 12. **Qualitative Results for Camera Pose Estimation.** The figure visualizes predicted camera poses (thin) alongside ground-truth poses (bold). For each object, we estimate the relative camera poses of five target views from a single reference image, shown in **red**. Both methods start from two randomly initialized poses. Our method consistently converges to the correct poses, while iFusion often gets stuck in local minima, resulting in incorrect predictions.

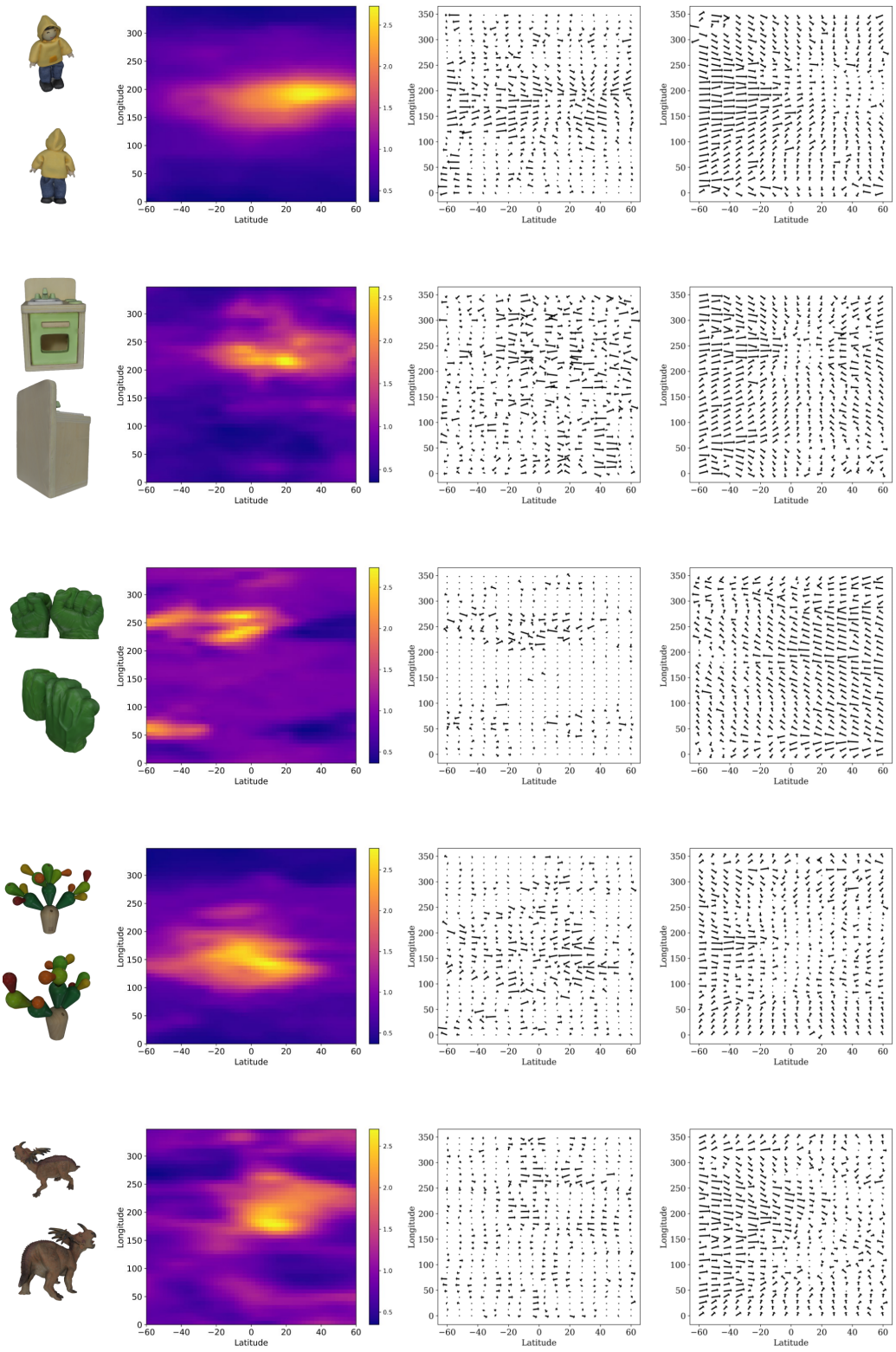

Figure 13. **Comparison Between Score and Energy.** The leftmost column shows the image pair, with the top image as the reference and the bottom image as the query. The second column shows the learned energy field, visualized as $\exp(-\mathcal{E}(x))$, where $\mathcal{E}(x)$ denotes the energy function. The higher value represents the high probability region. The third column shows the score field corresponding to the energy field, calculated by automatic differentiation. The last column shows the score field learned by the score-based method.