CRAYM: Neural Field Optimization via Camera RAY Matching

Liqiang Lin
Shenzhen University
Shenzhen, China
linliqiang2020@gmail.com

Wenpeng Wu
Shenzhen University
Shenzhen, China
wenpengggg@gmail.com

Chi-Wing Fu
The Chinese University of Hong Kong
Hong Kong SAR, China
cwfu@cse.cuhk.edu.hk

Hao Zhang
Simon Fraser University
Burnaby, Canada
haoz@sfu.ca

Hui Huang*
Shenzhen University
Shenzhen, China
hhzhiyan@gmail.com

Abstract

We introduce *camera ray matching* (CRAYM) into the joint optimization of camera poses and neural fields from multi-view images. The optimized field, referred to as a feature volume, can be "probed" by the camera rays for novel view synthesis (NVS) and 3D geometry reconstruction. One key reason for matching camera rays, instead of pixels as in prior works, is that the camera rays can be parameterized by the feature volume to carry both geometric and photometric information. Multi-view consistencies involving the camera rays and scene rendering can be naturally integrated into the joint optimization and network training, to impose physically meaningful constraints to improve the final quality of both the geometric reconstruction and photorealistic rendering. We formulate our per-ray optimization and *matched ray coherence* by focusing on camera rays passing through *keypoints* in the input images to elevate both the efficiency and accuracy of scene correspondences. Accumulated ray features along the feature volume provide a means to discount the coherence constraint amid erroneous ray matching. We demonstrate the effectiveness of CRAYM for both NVS and geometry reconstruction, over dense- or sparse-view settings, with qualitative and quantitative comparisons to state-of-the-art alternatives.

1 Introduction

Recent advances on multi-view 3D reconstruction have been propelled by the emergence of neural fields [39], including implicit functions [42, 36, 24, 33] and radiance fields (NeRF) [26, 44, 13, 1, 10, 27, 19, 7]. A critical component to all image-to-3D reconstruction methods, including traditional approaches such as multi-view stereo (MVS) [11], is to obtain camera poses for the input images. In practice, the camera information may be available from the acquisition devices, e.g., through the GPS or inertial measurement unit (IMU), while in other cases, it is estimated, e.g., using structure-from-motion (SfM) [8, 29]. In both cases, these camera poses can be noisy, thus hindering the performance of the multi-view 3D reconstruction.

In light of the importance of having accurate camera poses, various methods have been proposed to improve their estimations. One line of approaches, which can be referred to as bundle-adjusting neural fields, jointly optimize [38, 20, 12, 2, 5] camera poses along with results from rendering and

*Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

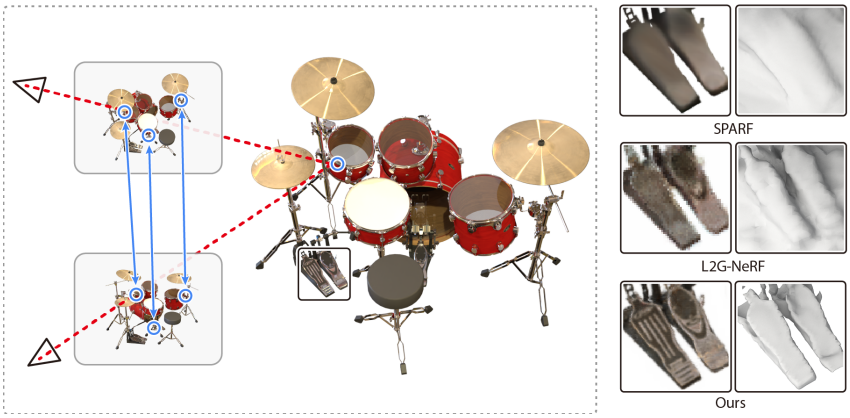

Figure 1: Our method, neural field optimization with camera ray matching (CRAYM), incorporates contextual information for per-ray processing and enforces color + geometric consistence between matched rays. Compared to SPARF [32] which utilizes dense pixel correspondences and the state-of-the-art, bundle-adjusting L2G-NeRF [5], both aimed at handling noisy camera poses, CRAYM produces superior results especially over fine details; see the zoom-ins on the right. Results are shown the Drums model from NeRF-Synthetic [26] on dense views.

geometry (e.g., depth) estimation, where the camera rays are considered *independently* for their roles in color and geometry prediction. By now, more works [3, 4, 32, 18, 16, 30, 6] realize the importance of exploiting correlations between input images, i.e., multi-view consistency, to impose additional constraints on the joint optimization. Along these lines, much effort has been invested into matching and optimization with respect to image features, whether convolutional or transformer-based.

Motivated by multi-view *spatial* analysis, several works have proposed geometric constraints involving camera rays and projections [16, 32, 18]. Most recently, SPARF [32] defines a *re-projection* loss as a spatial distance between image pixels to enforce that matched pixels between NeRF training images be back-projected onto the same 3D point. However, the effectiveness of this loss depends critically on how reliable the pixel correspondences are. In their work, these correspondences and their confidence estimates were both obtained by a pre-trained network [31], which is independent of the joint camera-scene optimization.

In this paper, we introduce *camera ray matching* into the joint optimization of camera poses and a neural field, referred to as a *feature volume*, which can be "probed" by the camera rays for both rendering, e.g., novel view synthesis as in NeRF [26], and 3D reconstruction, as in NeuS [34].

The key reasons for matching camera rays, instead of pixels [32, 18, 6], are two-fold. First, these rays carry 3D spatial information than just 2D pixel values to facilitate formulating *explicit* geometric losses when optimizing camera poses [16], as dictated by multi-view analysis. Second and more importantly, the camera rays can be *parameterized* by the feature volume — they carry both geometric and photometric information. Any constraint arising from camera ray matching can be passed onto the feature volume. Hence, both the matching itself and the associated matching confidence can be incorporated into the joint optimization and network training, to impose physically meaningful constraints to improve the final quality of both geometry reconstruction and rendering.

Our network, coined CRAYM (for Camera RAY Matching), takes as input an uncalibrated set of images capturing a 3D object, and is trained to predict the feature volume along with all the camera rays subjected to a combination of photometric rendering losses and geometric losses dedicated to ensuring multi-view consistency between camera rays. We consider two types of rays. The first are called *key rays*, which pass through keypoints detected in the input images, typically spanning regions with sharp features and rich textures over the 3D object. The other rays are called *auxiliary rays*, which pass through points around keypoints to offer contextual and local structural information as we reason about the key rays in our optimization framework.

As our main constraint for *matched ray coherence*, we enforce color consistency between renderings along two key rays whose corresponding keypoints from two different views are matched [9]. However, we must account for potential erroneous matches due to occlusion or unreliable local image features used by the matching network. To this end, we aggregate features along each key

ray through the feature volume. The matchability between two rays is defined by a cosine similarity between the accumulated ray features and applied as a weight to either accentuate or discount the color consistency constraint, allowing our optimization model to naturally degenerate itself to handle unrelated rays separately. Also, to improve the robustness of feature learning, we enhance the feature along each key ray by integrating features from surrounding auxiliary rays.

We evaluate our method on both the synthetic objects from NeRF-Synthetic [26] and the real scenes from UrbanScene3D [22], for novel view synthesis and 3D geometry reconstruction, over dense- and sparse-view settings. Compared to state-of-the-art alternatives, CRAYM produces superior results especially over fine details.

2 Related Works

Neural Fields. As a pioneer work, NeRF [26] synthesizes novel views of static objects/scenes from a set of posed images by optimizing a coordinate-based neural network, which predicts the volume density and color for a sampled point in the 3D space. Since then, numerous methods have emerged to improve the rendering quality [44, 13, 1] and rendering efficiency [10, 27, 19, 7]. To extract high-quality surfaces from the learned implicit representation, NeuS [42] and VolSDF [42] propose to learn an implicit signed distance field (SDF) representation for the scenes. These methods can achieve impressive results on both novel view synthesis and 3D reconstruction, however, the requirement of precise camera pose limits their applicability in practice.

Bundle-Adjusting Neural Fields. With the realization that positional encoding is susceptible to suboptimal registration, BARF [20] applies a smooth mask on the encoding at different frequency bands for a coarse-to-fine training, while [12] presents an adaptive positional encoding. L2G-NeRF [5] first learns the pixel-wise transformations for every pixel in a frame and then aligns the frame-wise transformation with the pixel-wise transformations. Common to all the above methods is that their joint optimization of pose and scene representation processes each image and each ray *separately*, without considering their multi-view correlations. As a result, the pose optimization may not be stable, thereby leading to floaters and blurriness in both novel view renderings and 3D reconstruction. Note that our method also involves per-ray processing, by combining information from auxiliary rays with that of a key ray. This is similar to the patch-level feature processing in CR-NeRF [41], which considers multiple rays indiscriminately across the image, without the notion of key rays.

Neural Fields with Image Matching. Image matching can help establish geometric priors to improve the generalizability of NeRF, to either novel scenes or the sparse-view setting. MVSNeRF [3] constructs a cost volume by warping the image features extracted with a 3D CNN onto a plane sweep, from which a generalizable radiance field is learned. SparseNeuS [24] constructs a 3D volume with the variance of all the projected features from multi-view images. DBARF [4] optimizes camera poses and depth with a cost map constructed by the differences of image features. CorresNeRF [18] proposes to regularize the NeRF training with a pixel re-projection loss for the associated pixels and a depth loss for the predicted depth. GPNR [30] aggregates features of the image patches along epipolar lines with several stacked transformers. MatchNeRF [6] learns a generalizable NeRF with the cosine similarity of image features for each image pair as the shape prior. All these methods integrate image features and utilize the matching within or between different views. With more emphasis placed on multi-view geometry reasoning, SCNeRF [16] learns a pinhole model for each camera under the supervision of a re-projected ray distance loss, while SPARF [32] optimizes its network with a re-projection loss, measuring spatial distances between pixels in the same view. In contrast, the matched ray coherence formulation in our optimization accounts for both photometric and geometry information as obtained from the feature volume; the coherence constraint is also explicitly integrated into the network instead of only serving to define a loss.

3 Method

We are interested in neural networks that can reconstruct a 3D model, e.g., a radiance field [26] or an implicit field [34], from a set of M images $\{\mathbf{I}_i\}_{i=1}^{M}$ capturing a 3D object from multiple views. Typically, each image is associated with a known or estimated camera pose $\mathcal{T}_i = [R_i|t_i]$, where $R_i \in \mathrm{SO}(3)$ and $t_i \in \mathrm{R}^3$. The network is trained by minimizing a photometric error L_p between the

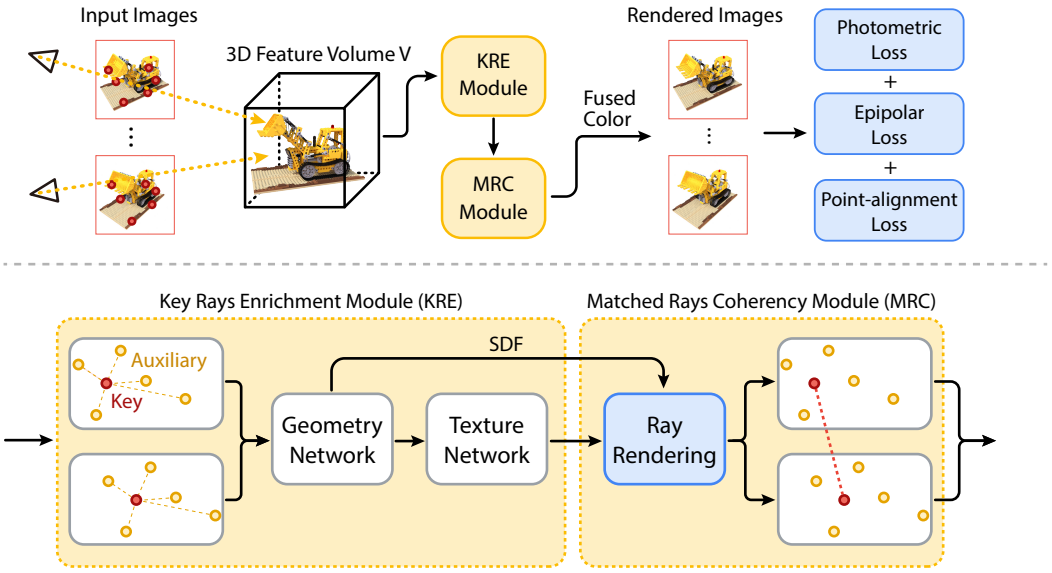

Figure 2: Overview of our CRAYM pipeline. After extracting keypoints (red dots) from input images and matching them using a pre-trained network, we train our CRAYM network to optimize a 3D feature volume $\mathcal{V}$ which encodes both geometric and photometric information about the target 3D object and can be queried by camera rays for both novel view synthesis (via the Texture Network) and 3D reconstruction (via the Geometry Network). The volume optimization is subject to photometric losses through rendering along camera rays passing through the keypoints (i.e., the key rays), which is enhanced (in the KRE) by integrating features from auxiliary rays, i.e., rays passing through nearby auxiliary points (yellow dots) in the images. Matched ray coherence (MRC) is imposed on matched key rays, in terms of color consistency, while potentially mismatched rays can be identified by comparing accumulated features along the key rays through $\mathcal{V}$. On top of the standard photometric loss, we introduce two geometric losses, the epipolar loss and point-alignment loss, to explicitly optimize ray-to-ray coherency to maximize the reconstruction quality of the feature volume.

input images and the multi-view renderings, $\{\hat{\mathbf{I}}_i\}_{i=1}^{M}$, of the target 3D object from the camera views: $\mathbf{min} \sum_i \sum_x ||\mathbf{I}_i(x) - \hat{\mathbf{I}}_i(x)||_2^2$, where $\mathbf{I}_i(x)$ is the color of image $\mathbf{I}_i$ at pixel x.

Each pixel is associated with a specific ray in 3D from the object/scene, through the pixel center, towards the camera: $\{\mathbf{r}(t) = \mathbf{r_o} + t\mathbf{r_d} | t \geq 0\}$, where $\mathbf{r_o}$ is the camera center and $\mathbf{r_d}$ is the normalized view direction of ray $\mathbf{r}$. The rendered color of ray $\mathbf{r}$, i.e., the pixel color $\mathbf{I}_i(x)$, can be produced using volume rendering by accumulating the color and opacity σ along the ray $\mathbf{r}$.

Considering that the camera poses can be noisy, the reconstructed radiance field or implicit field may not produce clean and sharp renderings with details. At a high noise level, some methods may even fail to produce results; see examples shown in Sections 4.2 and 4.4. Beyond existing approaches that map $\mathbf{r}(t)$ to opaque density (or opacity) and color implicitly with a ray-wise network, we propose CRAYM to learn the implicit field by matching rays across different images and formulating geometric priors.

3.1 The CRAYM Pipeline

Figure 2 overview our CRAYM pipeline. From the input images, our goal in the 3D neural field optimization is to construct a 3D feature volume $\mathcal{V}$ to faithfully represent the target object. In detail, we represent feature volume $\mathcal{V}$ using multi-resolution hash encoding [27] and end-to-end optimize it for the target object. The feature $f(p)$ of point p in the 3D feature volume can be extracted by

$$f(p) = \mathcal{M}(\mathcal{V}(p)), \tag{1}$$

where $\mathcal{M}$ is the progressive feature mask [19] for filtering out fine-level features during early iterations of the coarse-to-fine training. Very importantly, to account for the noise in the camera poses, we parameterize the transformation matrices of the cameras as variables in the joint optimization of the pose and implicit field with the feature volume $\mathcal{V}$.

As mentioned in the introduction, we consider two types of rays to probe the feature volume, i.e., *key rays* and *auxiliary rays*. Both rays are issued from the cameras through the pixel centers. To obtain key rays $\{\mathbf{r}_k\}$, which typically associate to surface points with rich textures and sharp features, we detect keypoints on each input image using SuperPoint [9] and perform point-to-point matching between image pairs using SuperGlue [28]. Then, we can obtain a set of *sparse ray-to-ray matchings* between image pairs. Note that these results may not be accurate for various reasons such as occlusion and unreliable matching, but they provide useful information for our pipeline to start with. As for the auxiliary rays $\{\mathbf{r}_a\}$, they are sampled around the keypoints to provide contextual or local structural information when we reason about the key rays; see Section 3.2 for details.

Once optimized, the feature volume can be used for novel view synthesis or for multi-view 3D reconstruction. The color prediction for novel view synthesis is accomplished by a texture network Φ_t, as in a typical NeRF [26] setting, and the latter is accomplished by a geometry network Φ_g, as in a typical NeuS [34] setting. Specifically, the geometry network takes a 3D point p sampled along $\mathbf{r}$ and the feature at p as input to produce an SDF value and then an opaque density σ to render the 3D object and extract the 3D reconstructions. Here, we propose the Key Rays Enrichment (KRE) module (Section 3.2) to improve the robustness in the process by enhancing the features along the key ray using the features sampled by the auxiliary rays.

Subsequently, the texture network takes the output features from geometry network, ray directions, and normal at p as inputs to predict color $c(p)$ at point p. Further, we design the Matched Rays Coherency (MRC) module (Section 3.3) to enhance the volume rendering quality by considering matchability between rays and learning to maintain coherency between ray matchings. Particularly, the MRC module can effectively reduce the influence of mismatched rays by disambiguating the camera ray matchings.

A pair of the matched key rays, $\mathbf{r_k}$ and $\mathbf{r_k}'$, are sampled with the corresponding auxiliary rays during each iteration. The geometry network, texture network, and feature volume optimization are jointly trained end-to-end. Besides the photometric loss, we formulate the epipolar loss and point-alignment loss (Section 3.4) to explicitly promote coherency among the ray matchings and boost performance.

3.2 Key Rays Enrichment Module

As the input images are captured through a perspective projection, all rays in 3D through the same image should converge at a common camera point. In previous works, for each iteration, rays are optimized separately, so the pose optimization may not be stable. As different rays may back propagate gradients in different directions, the optimized poses may oscillate during the training. Hence, we introduce the KRE module to stabilize the optimization by learning structural information around each key ray. This is done by sampling auxiliary rays around the key ray to enrich the feature of the key ray with more contextual information:

$$f'(p_k) = \sum_j g(f(p_k), f(q_j)), \tag{2}$$

where p_k is a point along key ray $\mathbf{r}_k$; $\{q_j\}$ are points around p_k sampled along the j-th auxiliary ray around $\mathbf{r}_k$; and function g fuses features $f(p_k)$ and $f(q_i)$. Then, we employ the geometry network Φ_g to predict the SDF value at p_k and feature vector $f''(p_k)$, from which we can further obtain the color of point p_k with the texture network. Please refer to the supplemental materials for the details.

3.3 Matched Rays Coherency Module

Next, we propose to learn the coherency of features accumulated in the 3D feature volume between the matched key rays. The purpose is to enhance the camera ray matching and account for imprecise ray matchings, since keypoints matching is performed only on local image features.

Similar to color accumulation in volume rendering, we calculate the aggregated feature along a key ray $\mathbf{r_k}$ as the feature of $\mathbf{r_k}$:

$$f(\mathbf{r_k}) = \int_0^\infty \mathcal{T}(p_k)\sigma(p_k)f''(p_k)dt. \tag{3}$$

The function $\mathcal{T}(\mathbf{r_k}(t)) = \exp(-\int_0^t \sigma(s)ds)$ denotes the accumulated transmittance along key ray $\mathbf{r_k}$.

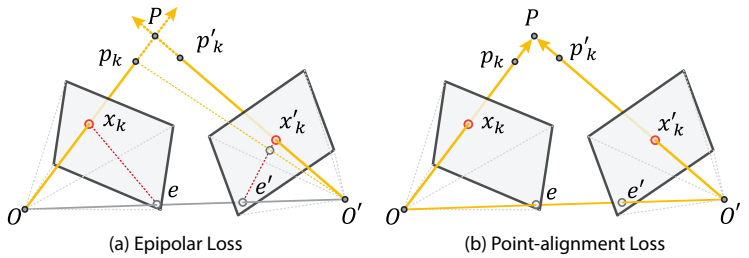

Figure 3: Illustrating of our geometric losses. The red lines in the left subfigure are epipolar lines. The epipolar loss constrains the relative transformations between cameras, so that the projection of a keypoint p_k onto the image plane of the other camera should lie on the epipolar line $e'x'_k$. With the camera poses constrained by the epipolar loss, the point-alignment loss further constrains the depth of x_k and x'_k, aiming to align p_k and p'_k with P.

Essentially, the accumulated ray feature $f(\mathbf{r_k})$ is an integration of density-weighted features along a ray. When the network converges, the actual surface point at which ray $\mathbf{r_k}$ intersects with the paired matched key ray should have the highest density. Hence, we consider coherency between the matched rays to optimize the learning of the opacity density and point color, such that we can enhance the coherency of features accumulated along the matched key rays. In return, this will help to optimize the parameters and the 3D feature volume, when training the pipeline. Therefore, we fuse the rendered color $\mathbf{c}(\mathbf{r}_k)$ of the matched rays based on the cosine similarity between their accumulated features:

$$\mathbf{c}(\mathbf{r}_k) = w\mathbf{c}(\mathbf{r}'_k) + (1-w)\mathbf{c}(\mathbf{r}_k), \tag{4}$$

where w is the matchability calculated as the cosine distance between the accumulated features of the matched rays. When two key rays are mismatched, e.g., due to occlusion or ambiguities of weak texture areas and similar structures, our formulation can learn to degenerate itself to a form that separately optimizes individual rays.

3.4 Loss Function

Further, we introduce the following two geometric losses to more explicitly promote the coherency of the ray matchings:

Epipolar loss. Given a pair of matched keypoints x_k and x'_k on two different input images, which associate with camera centers O and O', respectively, (see the illustration in Figure 3), we can estimate the depths at x_k and x'_k by using a depth accumulation formulation similar to Equation 3, and then project points x_k and x'_k into the 3D object space to obtain 3D locations p_k and p'_k, respectively.

If the camera poses, the matchings, and the depths are precise, the two rays through x_k and x'_k should precisely intersect at a common point, say P, on the target object surface, such that p_k and p'_k align with P. Also, we denote e and e' as the epipolar points on the two images; these points are the image-space locations at which the line OO' intersects the two image planes; see Figure 3(a).

During the training, the depth estimation of x_k can vary, so p_k may vary along ray r_k. If the camera poses are precise, the projection of p_k onto the image plane of the other camera should lie on the epipolar line $e'O'$. In case of noisy camera poses, the projection of p_k may not lie exactly on $e'O'$, so we explicitly enforce the epipolarity during the training by minimizing the distance between p_k's projection and the epipolar line $e'x'_k$ using

$$L_e = \frac{1}{N_k}\sum_{i=1}^{N_k} \mathrm{Dist}(Proj(p_k), e'x'_k). \tag{5}$$

Since the epipolar loss is not affected by depth, we decouple the unreliable depth estimation from the epipolar loss with ray marching to constrain the camera poses.

Point-alignment loss. The epipolar loss focuses on enhancing the projection consistency for producing more precise camera poses. To complement it, we introduce the point-alignment loss to facilitate depth convergence for improving the reconstruction of fine details. In detail, we consider the triangle

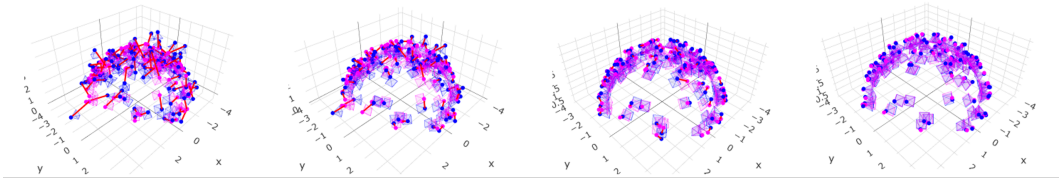

Figure 4: Visualization of the initial and optimized camera poses for the LEGO scene in the NeRF-Synthetic dataset [26]. (Purple: ground-truth poses; blue: initial or optimized poses; red lines: translation errors.)

formed by intersection point P, line segment p_kO, and line segment p'_kO' (Figure 3(b)), and aim to minimize the distance between points p_k and p'_k and align them:

$$L_a = \frac{1}{N_k} \sum_{i=1}^{N_k} \text{Dist}(p_k, p'_k). \tag{6}$$

Note that depth estimation may be unreliable and likely unstable early in the training, so we use the point-alignment loss only after a certain number of training iterations.

Overall loss. After constructing the epipolar loss L_e and the point-alignment loss L_a, we put them together with the photometric loss L_p and SSIM loss L_s to form the overall loss function

$$L = \lambda_1 * L_p + \lambda_2 * L_s + \lambda_3 * L_e + \lambda_4 * L_a, \tag{7}$$

where L_p is calculated between the input images and the rendered images and is modeled as an MSE loss.

4 Results

We evaluate our method on the NeRF-Synthetic dataset [26] with eight synthetic objects (Section 4.2), the LLFF dataset [25], and the real scenes from the UrbanScene3D dataset [22] (Section 4.3). We compare our method on both novel view synthesis and 3D reconstruction with NeRF [26], NeuS [34], BARF [21], L2G-NeRF [5], PET-NeuS [37], SPARF [32], and BAA-NGP [23]. Since NeRF [26], NeuS [34], and PET-NeuS [37] are designed for neural implicit field with fixed and precise poses, we set the camera transformations as variables to be optimized jointly with the neural field, as in our method.

While other methods optimize the radiance field, in which the target values, radiances, of points are more independent of each other, the optimization of SDFs in NeuS and PET-NeuS poses a challenge to the requirements of non-local geometric constraints to correctly form the shape, making them more vulnerable to unstable pose optimization. As the camera rays are parameterized by our feature volume to carry both geometric and photometric information, our geometric constraints on the camera ray matching can effectively lead to better optimization of the geometry. The joint optimization of camera pose and implicit SDF may also fail to produce results for NeuS and PET-NeuS, when the camera poses are at a high noise level. With the assistance of camera ray matching, CRAYM outperforms other methods on both novel view synthesis and 3D reconstruction at varying noise levels. We report the PSNR, SSIM, and LPIPS [43] for quantitative comparisons on novel view synthesis and Chamfer distance (CD) for the 3D reconstruction. A test-time photometric pose optimization is performed to evaluate these metrics, following prior works [21, 32, 5]. The quantitative evaluations on the other metrics are provided in the supplementary materials.

4.1 Pose Alignment

To evaluate the registration quality of the optimized training poses, we use Procrustes analysis [14] to find the 3D similarity transformation that aligns the optimized training poses with the calibrated camera poses, following BARF [21]. As Figure 4 shows, the optimized poses produced by CRAYM align well with the ground-truth poses with lower translation errors.

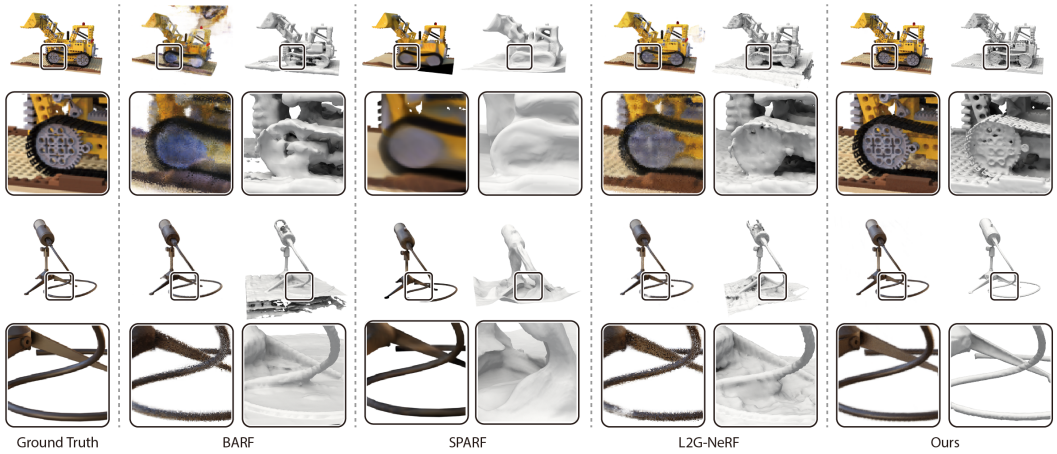

Figure 5: Qualitative comparison results of novel view synthesis and surfaces reconstruction on the synthetic objects.

The average translation errors and rotation errors are reported in Table 1. With the contextual information and feature coherency of camera ray matching, the camera poses produced by CRAYM are better optimized for the construction of the implicit filed, so that CRAYM is able to produce high-quality rendered views, as well as more precise 3D reconstructions.

Table 1: Poses registration errors evaluated on the LEGO scene in the NeRF-Synthetic dataset [26].

Method		Rotation (°) ↓	Translation ↓
BARF [21]	ICCV'21	9.02	0.81
SPARF [32]	CVPR23'21	10.64	0.53
L2G-NeRF [5]	CVPR23'21	2.90	**0.10**
CRAYM (ours)		**1.21**	0.19

4.2 Evaluation on Synthetic Objects

For the evaluation on the NeRF-Synthetic dataset, we follow the same setting of noisy poses as L2G-NeRF [5], which perturbs the ground-truth camera poses with additive noise as the initial poses. As NeuS [34] and PET-NeuS [37] fail to produce results at such a setting, we present only the results of NeRF [26], BARF [21], SPARF [32], and L2G-NeRF [5]. The mean results of the eight objects and four of them are given in Table 2. NeRF fails to extract meshes from the reconstructed radiance fields on Hotdog and Ship. Figure 5 shows the results of view synthesis and 3D reconstruction visually for BARF, SPARF, L2G-NeRF, and our method. SPARF produces over smoothing results with dense input, as shown in Figure 5. As can be seen in Figure 5, CRAYM is able to produce clean and complete renderings and reconstructions with fewer floaters and less blurriness.

Table 2: Results on the NeRF-Synthetic dataset.

Metrics	Method	PSNR↑	SSIM↑	LPIPS↓	CD↓
Chair	NeRF [26]	16.69	0.77	0.39	2.23
	BARF [21]	28.55	0.93	0.06	0.09
	SPARF [32]	23.80	0.87	0.19	0.14
	L2G-NeRF [5]	30.99	0.95	0.05	0.10
	CRAYM (ours)	**34.18**	**0.98**	**0.02**	**0.06**
Hotdog	NeRF [26]	15.07	0.74	0.42	N/A
	BARF [21]	30.12	0.95	0.04	0.38
	SPARF [32]	29.10	0.93	0.13	0.07
	L2G-NeRF [5]	34.56	0.97	0.03	0.38
	CRAYM (ours)	**36.42**	**0.98**	**0.02**	**0.05**
LEGO	NeRF [26]	11.11	0.60	0.58	0.58
	BARF [21]	22.54	0.79	0.12	**0.04**
	SPARF [32]	22.47	0.80	0.25	0.09
	L2G-NeRF [5]	27.71	0.91	0.06	0.12
	CRAYM (ours)	**31.60**	**0.96**	**0.03**	**0.04**
Mic	NeRF [26]	13.08	0.73	0.53	0.48
	BARF [21]	30.37	0.96	**0.05**	0.28
	SPARF [32]	28.36	0.91	0.17	0.24
	L2G-NeRF [5]	30.91	0.97	**0.05**	0.17
	CRAYM (ours)	**31.02**	**0.97**	**0.05**	**0.04**
Mean	NeRF [26]	13.29	0.68	0.49	0.64
	BARF [21]	23.09	0.84	0.18	0.24
	SPARF [32]	23.90	0.84	0.23	0.18
	L2G-NeRF [5]	28.62	0.93	0.07	0.17
	CRAYM (ours)	**30.34**	**0.95**	**0.05**	**0.06**

4.3 Evaluation on Real Scenes

We first evaluate our method on the LLFF dataset [25] for high-fidelity view synthesis of the eight real scenes. Compared with BARF [21], L2G-NeRF [5], and BAA-NGP [23], our method is able to produce high-quality results with fewer artifacts and better scores in terms of PSNR, SSIM, and LPIPS, as shown in the Table 3.

Table 3: Qualitative comparison on novel view synthesis on the LLFF dataset [25].

	PSNR ↑				SSIM ↑				LPIPS ↓			
	BARF [21]	L2G-NeRF [5]	BAA-NGP [23]	CRAYM (ours)	BARF [21]	L2G-NeRF [5]	BAA-NGP [23]	CRAYM (ours)	BARF [21]	L2G-NeRF [5]	BAA-NGP [23]	CRAYM (ours)
Fern	23.88	24.57	19.37	**24.83**	0.71	0.75	0.50	**0.79**	0.31	0.26	0.38	**0.25**
Flower	24.29	24.90	**25.16**	25.04	0.71	0.74	**0.81**	0.76	0.20	0.17	**0.10**	0.13
Fortress	29.06	29.27	29.24	**29.39**	0.82	0.84	0.83	**0.85**	0.13	**0.11**	0.14	0.12
Horns	23.29	23.12	19.71	**23.30**	0.74	0.74	0.72	**0.75**	0.29	0.26	**0.24**	**0.24**
Leaves	18.91	19.02	**19.96**	19.57	0.55	0.56	**0.68**	0.60	0.35	0.33	**0.23**	0.29
Orchids	19.46	19.71	12.45	**19.81**	0.57	**0.61**	0.14	0.59	0.29	0.25	0.42	**0.23**
Room	32.05	32.25	29.72	**32.44**	0.94	**0.95**	0.90	0.91	0.10	**0.08**	0.12	0.09
T-rex	22.92	23.49	**24.56**	23.68	0.78	0.80	**0.86**	0.83	0.20	0.16	0.11	**0.10**
Mean	24.23	24.54	22.52	**24.76**	0.73	0.75	0.68	**0.76**	0.23	0.20	0.22	**0.18**

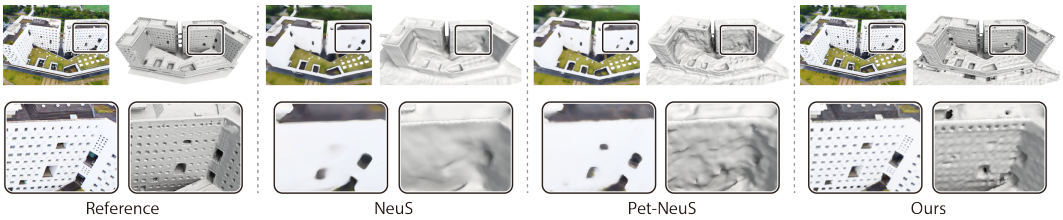

Figure 6: Qualitative results of novel view synthesis and surfaces reconstruction on real scenes captured by high-resolution cameras.

As the images in the LLFF dataset are captured with a restricted range of angles, we further assess our method on the real scenes PolyTech and ArtSci from the UrbanScene3D [22] dataset with two sets of drones captured images [45] associated with GPS information. The large-scale scenes are captured with hundreds of images, which share smaller overlaps than the images from the NeRF-Synthetic dataset. Considering the limitations of memory, we reduce the size of the original high-resolution images by using bicubic interpolation. As the GPS information may not be reliable and the positions in GPS may shift in meters, we preprocess the poses from GPS with COLMAP [29] and add a small noise to the calibrated poses, following L2G-NeRF [5].

Table 4: Results on the real scenes PolyTech and ArtSci in the UrbanScene3D [22] dataset.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	CD↓
PolyTech	NeuS [34]	14.27	0.42	0.71	0.10
	PET-NeuS [37]	13.64	0.40	0.75	0.07
	CRAYM (ours)	**22.51**	**0.63**	**0.43**	**0.01**
ArtSci	NeuS [34]	13.49	0.33	0.89	0.10
	PET-NeuS [37]	15.18	0.36	0.88	0.08
	CRAYM (ours)	**19.37**	**0.42**	**0.63**	**0.02**

The UrbanScene3D dataset contains high-precision LiDAR scans for the target buildings, PolyTech and ArtSCi. Therefore, we evaluate the reconstructions quality with the point cloud scans as the ground truths. We crop the reconstructed meshes according to the LiDAR scans and align them using ICP. NeRF, BARF, and L2G-NeRF produce blurry renderings and degenerated meshes for the real scenes, while SPARF may fail to process such data with dense correspondences. Therefore, we only provide the visual results of NeuS, PET-NeuS, and our method.

The quantitative results of NeuS, PET-NeuS, and our method are provided in Table 4. As Figure 6 shows, NeuS and PET-NeuS tend to produce over smoothing results, while our method is able to extract meshes with fine details. A mesh reconstructed directly from the original high-resolution images using ContextCapture[2], a commercial MVS solution, is provided as a reference.

4.4 Ablation Study

Comparison on varying noise levels. To evaluate model robustness, we evaluate on the LEGO data sample with poses at varying noise levels. The "high noise level" means we use the same noise setting as L2G-NeRF [5]; the "low noise level" means we perturb the ground-truth poses with half of the noise as L2G-NeRF; and "w/o noise" means the poses are initialized as ground-truth poses without noise. The transformation matrices of the camera poses are all set as variables to be optimized. Table 5 summarizes the results. We can see that NeuS [34] and PET-NeuS [37] are more sensitive to noise and cannot effectively handle the high-noise setting, while the other methods can not produce

[2]https://www.bentley.com/en/products/brands/contextcapture

accuracy reconstructions. With the contextual information learned in the camera ray matching and the explicit utilization of matched rays, CRAYM is able to obtain better results for all the noise settings.

Ablation of major modules and losses. To demonstrate the efficiency of the proposed modules and geometric losses, we conduct an ablation study on the LEGO data sample. The results are reported in Table 6. Similar with BARF, which applies a smooth mask on the encoding at different frequency bands for neural radiance field, we apply a progressive feature mask on the hash encoding with a coarse-to-fine training of the neural implicit field as our baseline, which combines BARF [21] and NeuS2[35]. The KRE module improves the robustness to noisy poses in the training, whereas the MRC module effectively enhances the quality of the volume renderings with ray matching. In addition to that, the proposed geometric losses further help our framework to obtain better camera pose optimization.

Table 5: Comparing different methods at varying noise levels.

Method	PSNR↑	SSIM↑	LPIPS↓	CD↓
w/o Noise				
NeRF [26]	29.08	0.94	0.04	0.34
NeuS [34]	21.18	0.82	0.09	0.04
BARF [21]	28.33	0.93	0.05	0.36
SPARF [32]	22.73	0.80	0.25	0.09
PET-NeuS [37]	21.37	0.82	0.14	0.10
L2G-NeRF [5]	27.94	0.92	0.06	0.14
CRAYM (ours)	**32.72**	**0.97**	**0.02**	**0.03**
Low Noise Level				
NeRF [26]	24.86	0.88	0.09	0.34
NeuS [34]	21.76	0.83	0.14	0.05
BARF [21]	28.32	0.93	0.05	0.36
SPARF [32]	22.55	0.80	0.25	0.09
PET-NeuS [37]	21.34	0.82	0.11	0.55
L2G-NeRF [5]	27.75	0.92	0.06	0.21
CRAYM (ours)	**32.68**	**0.97**	**0.02**	**0.04**
High Noise Level				
NeRF [26]	11.36	0.81	0.56	0.57
NeuS [34]	N/A	N/A	N/A	N/A
BARF [21]	14.48	0.69	0.29	0.04
SPARF [32]	22.47	0.80	0.25	0.09
PET-NeuS [37]	N/A	N/A	N/A	N/A
L2G-NeRF [5]	27.71	0.91	0.06	0.12
CRAYM (ours)	**31.60**	**0.96**	**0.03**	**0.03**

Table 6: Ablation of major modules and losses.

Method	PSNR↑	SSIM↑	LPIPS↓	CD↓
L2G-NeRF [5]	27.71	0.91	0.06	0.12
NeuS2 [35]	26.83	0.86	0.17	0.08
Baseline	27.30	0.91	0.10	0.06
+ KRE	28.64	0.93	0.07	0.05
+ KRE + MRC	30.41	0.95	0.04	**0.04**
+ L_e	29.43	0.92	0.07	0.05
+ L_e + L_a	29.95	0.94	0.06	**0.04**
Our full pipeline	**31.60**	**0.96**	**0.03**	**0.04**

5 Conclusion and discussion

Our method, CRAYM, addresses the issue of noise camera poses for multi-view 3D reconstruction and view synthesis. The key idea is to jointly optimize a neural field and camera poses by incorporating contextual information (via KRE) and enforcing geometric and photometric consistency (via MRC and geometric losses) through camera ray matching.

Experiments demonstrate that our method outperforms state-of-the-art alternatives under various settings: dense- vs. sparse-views, and different noise levels. However, the implicit field and optimizable pose transformations may not converge when the poses are randomly initialized or extremely noisy. A stronger pose regularization prior to the field optimization may resolve this problem. Furthermore, the meshes extracted from the constructed SDFs may still contain messy inner structures over invisible areas.

A promising future work is to apply the ray matching to the 3D Gaussian splatting, which will greatly improve the rendering efficiency of CRAYM. However, extracting reconstructions with fine geometric structures from 3D Gaussians is still an open problem.

Finally, CRAYM has been designed to rely on sparse key rays for dense-view reconstruction, while a dense counterpart may bring up extra overhead. In our Matched Ray Coherency formulation, we explicitly account for potentially erroneous (i.e., low-quality) 2D matches by using the matchability between two rays as a weight to either accentuate or discount the color consistency constraint. In terms of sensitivity with respect to the density of the 2D matches, in our experiments, we have observed that even with sparse input views and sparsely distributed matched rays, CRAYM can still notably improve the optimization convergence. An effective approach to utilize ray matching for both sparse and dense inputs may further boost the performance of CRAYM.

Acknowledgement

We thank the reviewers for their valuable comments. This work was supported in parts by NSFC (U21B2023, U2001206), ICFCRT(W2441020), Guangdong Basic and Applied Basic Research Foundation (2023B1515120026), DEGP Innovation Team (2022KCXTD025), Shenzhen Science and Technology Program (KQTD20210811090044003, RCJC20200714114435012), and Scientific Development Funds from Shenzhen University.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRf: A multiscale representation for anti-aliasing neural radiance fields. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5855–5864, 2021.

[2] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-NeRF: Optimising neural radiance field with no pose prior. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4160–4169, 2023.

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proc. Int. Conf. on Computer Vision*, pages 14124–14133, 2021.

[4] Yu Chen and Gim Hee Lee. DBARF: Deep bundle-adjusting generalizable neural radiance fields. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 24–34, 2023.

[5] Yue Chen, Xingyu Chen, Xuan Wang, Qi Zhang, Yu Guo, Ying Shan, and Fei Wang. Local-to-global registration for bundle-adjusting neural radiance fields. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 8264–8273, 2023.

[6] Yuedong Chen, Haofei Xu, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Explicit correspondence matching for generalizable neural radiance fields, 2023.

[7] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 16569–16578, 2023.

[8] David Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 3001–3008, 2011.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 224–236, 2018.

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5501–5510, 2022.

[11] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 32(8):1362–1376, 2010.

[12] Zelin Gao, Weichen Dai, and Yu Zhang. Adaptive positional encoding for bundle-adjusting neural radiance fields. In *Proc. Int. Conf. on Computer Vision*, pages 3284–3294, 2023.

[13] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-MipRF: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 19774–19783, 2023.

[14] John R Hurley and Raymond B Cattell. The procrustes program: Producing direct rotation to test a hypothesized factor structure. *Behavioral science*, 7(2):258, 1962.

[15] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 406–413, 2014.

[16] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5846–5854, 2021.

[17] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. on Graphics*, pages 1–13, 2017.

[18] Yixing Lao, Xiaogang Xu, Zhipeng Cai, Xihui Liu, and Hengshuang Zhao. CorresNeRF: Image correspondence priors for neural radiance fields. In *Proc. Conf. on Neural Information Processing Systems*, 2023.

[19] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H. Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 8456–8465, 2023.

[20] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5741–5751, 2021.

[21] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *Proc. Int. Conf. on Computer Vision*, 2021.

[22] Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, reconstructing, and simulating: the UrbanScene3D dataset. In *Proc. Euro. Conf. on Computer Vision*, pages 93–109, 2022.

[23] Sainan Liu, Shan Lin, Jingpei Lu, Alexey Supikov, and Michael Yip. BAA-NGP: Bundle-adjusting accelerated neural graphics primitives. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 850–857, 2024.

[24] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. SparseNeuS: Fast generalizable neural surface reconstruction from sparse views. In *Proc. Euro. Conf. on Computer Vision*, pages 210–227, 2022.

[25] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. on Graphics*, 2019.

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. Euro. Conf. on Computer Vision*, 2020.

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, pages 1–15, 2022.

[28] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4938–4947, 2020.

[29] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4104–4113, 2016.

[30] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *Proc. Euro. Conf. on Computer Vision*, pages 156–174, 2022.

[31] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2021.

[32] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. SPARF: Neural radiance fields from sparse and noisy poses. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4190–4200, 2023.

[33] Aditya Vora, Akshayand Gadi Patil, and Hao Zhang. DiViNeT: 3D reconstruction from disparate views via neural template regularization. In *Proc. Euro. Conf. on Computer Vision*, pages 210–227, 2022.

[34] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Proc. Conf. on Neural Information Processing Systems*, pages 27171–27183, 2021.

[35] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. NeuS2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proc. Int. Conf. on Computer Vision*, 2023.

[36] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. HF-NeuS: Improved surface reconstruction using high-frequency details. *Proc. Conf. on Neural Information Processing Systems*, pages 1966–1978, 2022.

[37] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. PET-NeuS: Positional encoding tri-planes for neural surfaces. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 12598–12607, 2023.

[38] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[39] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022.

[40] Weidan Xiong, Hongqian Zhang, Botao Peng, Ziyu Hu, Yongli Wu, Jianwei Guo, and Hui Huang. Twintex: Geometry-aware texture generation for abstracted 3D architectural models. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, pages 227:1–227:14, 2023.

[41] Yifan Yang, Shuhai Zhang, Zixiong Huang, Yubing Zhang, and Mingkui Tan. Cross-ray neural radiance fields for novel-view synthesis from unconstrained image collections. In *Proc. Int. Conf. on Computer Vision*, pages 15901–15911, 2023.

[42] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Proc. Conf. on Neural Information Processing Systems*, pages 4805–4815, 2021.

[43] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 586–595, 2018.

[44] Kun Zhou, Wenbo Li, Yi Wang, Tao Hu, Nianjuan Jiang, Xiaoguang Han, and Jiangbo Lu. NeRFLiX: High-quality neural view synthesis by learning a degradation-driven inter-viewpoint mixer. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 12363–12374, 2023.

[45] Xiaohui Zhou, Ke Xie, Kai Huang, Yilin Liu, Yang Zhou, Minglun Gong, and Hui Huang. Offsite aerial path planning for efficient urban scene reconstruction. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, pages 192:1–192:16, 2020.

A Appendix / supplemental material

A.1 Implementation

A.1.1 Camera Pose Parameterization.

The camera poses $\{\mathbf{T}_i\}$, where $\mathbf{T}_i = [\mathbf{R}_i|\mathbf{t}_i] \in \mathrm{SE}(3)$, need to be parameterized and optimized in the training process. As a high frequency operation, pose parameterization needs to be simple and efficient, since the conversion between pose parameterization and transformation matrix is performed in every training iteration.

The position component $\mathbf{t}$ of $\mathbf{T}$ is simply represented as a three-dimensional vector. Instead of parameterizing the rotation $\mathbf{R}$ as an exponential map $\exp(\mathbf{r})$ from the Lie algebra $\mathfrak{so}(3)$ to the Lie group $\mathrm{SO}(3)$, we represent $\mathbf{R}$ as a six-dimensional vector composed by $\mathbb{R} = [v_a|v_b]$, $v_a \in \mathrm{R}^3$, $v_b \in \mathrm{R}^3$. v_a, v_b, and $v_c = v_a \times v_b$ span the three bases of the camera space. To obtain rotation matrix $\mathbf{R}$ from $\mathbb{R}$, we perform a Gram-Schmidt orthonormalization on v_a, v_b, and v_c, resulting in three orthonormal bases of the camera space, $\bar{v}_a$, $\bar{v}_b$, and $\bar{v}_c$. Thus, the rotation matrix is

$$R = [\bar{v}_a^T|\bar{v}_b^T|\bar{v}_c^T]. \tag{8}$$

A.1.2 Key Rays Enrichment Module

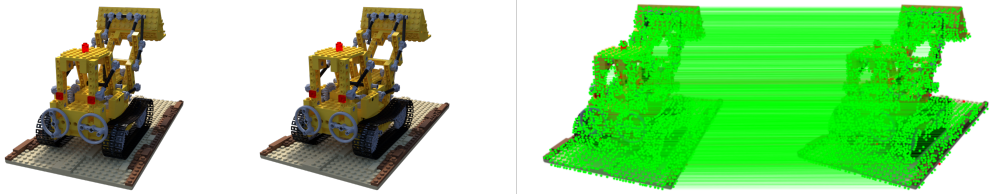

Figure 7: Visualization of the matches between two images from the LEGO scene in the NeRF-Synthetic dataset [26].

Figure 7 shows an example of matched pixels. Due to the noise in the camera poses, the surface point captured by the associated keypoint on the input image may not intersect with the key ray $\mathbf{r}_k$ exactly, the integration of features from the neighboring points $\{q_i\}$ sampled by auxiliary rays $\{\mathbf{r}_o\}$ can greatly facilitate a more stable optimization associated with key ray $\mathbf{r_k}$; see Figure 8.

Procedure-wise, we propose to first fuse the feature of p_k with the features of the surrounding $\{q_i\}$ to produce the enriched feature $f'(p_k)$, which can better describe the local radiance feature and geometry feature of the captured object with structure information.

$$f'(p_k) = \sum_{i=1}^{N_o} \mathrm{Softmax}(f(p_k) * f(q_i)) * f(q_i). \tag{9}$$

The features $\{f(q_i)\}$ of the auxiliary rays $\{\mathbf{r}_a\}$ remain untouched: $f'(q_i) = f(q_i)$. Further, we adopt the geometry network Φ_g to process the features $f'(p)$ of both key rays and auxiliary rays to extract the SDF value at point p in the radiance field. From the extracted SDF values, we can then produce the 3D reconstruction of the target object:

$$[\mathrm{SDF}(p)|f''(p)] = \Phi_g(f'(p)), \tag{10}$$

where $f''(p)$ is the output feature of point p. $f''(p)$ is one of the inputs to the texture network Φ_t. The contextual information of ray matchings facilitates a more stable pose optimization and promotes the details of the geometry, i.e., the accuracy of the SDF values.

The color of point p sampled on key rays or auxiliary rays can then be obtained with the texture network Φ_t as

$$c(p) = \Phi_t(f''(p), \mathbf{r_d}, \mathrm{Normal}(p)), \tag{11}$$

where $\mathbf{r_d}$ is the normalized direction of ray $\mathbf{r}$ and $\mathrm{Normal}(p)$ is the normal of the implicit surface at p computed as the gradient of $\mathrm{SDF}(p)$. We take the normal at p into account to boost the training of Φ_t.

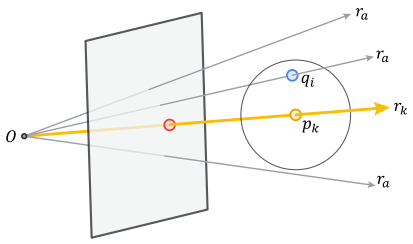

Figure 8: Illustrating the Key Ray Enrichment Module. The yellow ray $\mathbf{r}_k$ is the key ray and the gray rays $\{\mathbf{r_a}\}$ are the neighboring auxiliary rays. The surface point (the blue point) captured by the associated keypoint (the red point) on the input image may not intersect with $\mathbf{r}_k$ exactly due to the unreliable pose, while we can learn the contextual information of the surface point with the neighboring rays of point p_k sampled on $\mathbf{r}_k$.

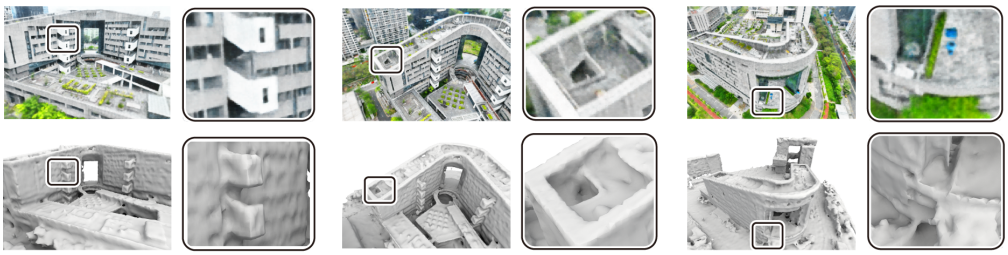

Figure 9: View synthesis and 3D reconstruction results on the real scenes CSSE produced by our method.

The color $c(\mathbf{r})$ of ray $\mathbf{r}$ is then rendered with the opaque densities and colors of all the points sampled along $\mathbf{r}$, where the opaque densities are obtained with the SDF values, normal vectors, and ray directions [34].

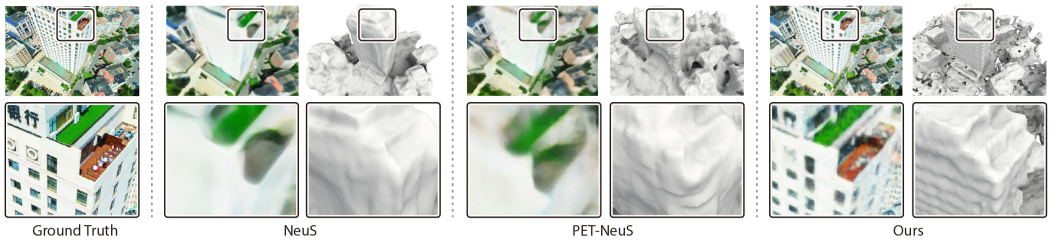

Figure 10: Comparison of view synthesis results on the real scenes Bank.

A.2 Training Details

We use a 16-level hash grid [27] to encode the feature volume $\mathcal{V}$. The feature length of each level is two. Therefore, the base resolution of $\mathcal{V}$ is 32. We apply a progressive feature mask [19] on the hash encoding, which starts at level 4 and is updated to the next level every 1,000 iterations. The geometry network ϕ_g is implemented as a three-layer MLP with the ReLU activation for the input and hidden layers. The texture network ϕ_t is implemented as a four-layer MLP with the ReLU activation for the input and hidden layers. The ray directions are encoded using the spherical harmonic representation [10] and fed into the texture network ϕ_t together with the output features of ϕ_g to predict the color of the sampled points on the ray. The whole network is optimized with the AdamW optimizer with a learning rate of 0.01, $\beta = [0.9, 0.99]$, and $\epsilon = 1.0^{-15}$. The variance [34] of the geometry network ϕ_g is initialized as 0.3 and is optimized with a learning rate of 0.001. We adopt a warm-up training for the first 500 iterations with the LinearLR scheduler. All the experiments are conducted with an Nvidia GV100.

Table 7: Comparison of reconstruction quality on the NeRF-Synthetic [26] dataset.

Metrics	Method		Mean	Chair	Drums	Ficus	Hotdog	LEGO	Materials	Mic	Ship
HD↓	NeRF [26]	ECCV'20	2.19	3.22	2.05	1.63	N/A	2.56	1.78	1.89	N/A
	BARF [21]	ICCV'21	1.52	**0.38**	1.42	0.28	1.13	0.44	1.90	1.26	5.34
	SPARF [32]	CVPR'23	1.11	0.80	1.42	1.09	**0.54**	0.92	1.35	1.64	1.10
	L2G-NeRF [5]	CVPR'23	1.57	0.43	1.57	0.48	1.14	0.99	2.15	1.09	4.69
	CRAYM (ours)		**0.98**	**0.38**	**0.73**	**0.23**	0.59	**0.37**	**0.35**	**0.57**	**4.62**
Precision↑	NeRF [26]	ECCV'20	1.17	N/A	1.99	0.71	N/A	3.72	2.12	0.81	N/A
	BARF [21]	ICCV'21	15.97	24.60	15.96	9.23	7.44	42.89	18.74	3.279	5.63
	SPARF [32]	CVPR'23	15.12	28.40	10.14	0.22	30.18	28.06	10.85	5.30	7.83
	L2G-NeRF [5]	CVPR'23	17.36	21.81	23.01	10.59	3.38	43.71	18.31	10.56	7.54
	CRAYM (ours)		**30.59**	**33.53**	**45.44**	**17.55**	**30.50**	**49.58**	**19.16**	**38.75**	**15.60**
Recall↑	NeRF [26]	ECCV'20	3.18	N/A	1.69	11.82	N/A	3.36	5.75	2.80	N/A
	BARF [21]	ICCV'21	20.43	24.25	3.33	17.47	24.33	61.29	18.75	12.24	1.81
	SPARF [32]	CVPR'23	19.38	16.17	6.09	17.54	**37.34**	13.68	12.60	5.40	19.38
	L2G-NeRF [5]	CVPR'23	26.98	19.97	48.06	21.10	10.99	68.00	19.51	19.34	8.84
	CRAYM (ours)		**42.29**	**43.65**	**50.11**	**45.95**	31.17	**81.63**	**21.67**	**40.29**	**29.77**
F-score↑	NeRF [26]	ECCV'20	1.38	N/A	1.83	1.34	N/A	3.53	3.10	1.25	N/A
	BARF [21]	ICCV'21	16.32	24.42	5.51	12.08	11.39	50.47	18.75	5.17	2.74
	SPARF [32]	CVPR'23	16.03	35.19	12.46	0.42	22.18	32.04	12.10	7.46	6.39
	L2G-NeRF [5]	CVPR'23	20.64	20.85	31.12	14.10	5.17	53.22	18.89	13.66	8.14
	CRAYM (ours)		**34.76**	**37.93**	**47.66**	**25.40**	**30.83**	**61.69**	**20.34**	**39.50**	**20.48**

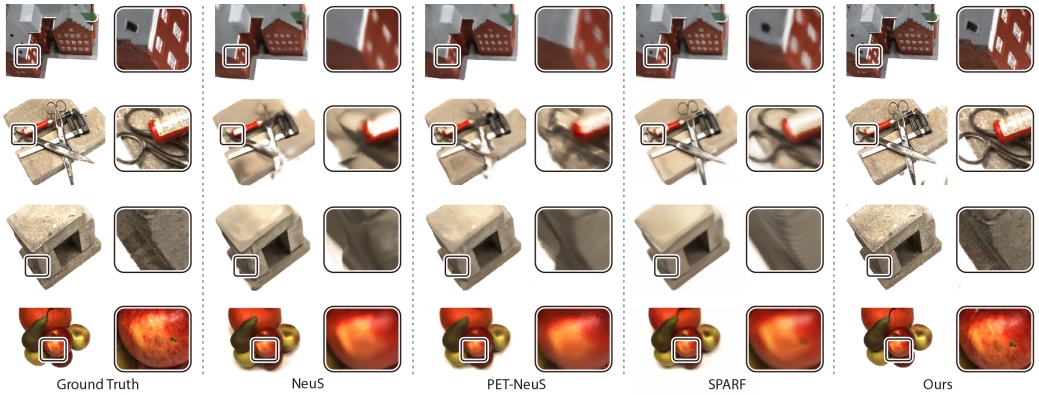

Figure 11: View Synthesis on the DTU [15] dataset.

Table 8: Comparison of reconstruction quality on the real scenes PolyTech and ArtSci of the UrbanScene3D [22] dataset.

Scene	Method	HD↓	Precision↑	Recall↑	F-score↑
PolyTech	NeuS [34]	0.63	41.32	55.68	47.44
	PET-NeuS [37]	0.46	43.29	46.83	44.99
	CRAYM (ours)	**0.04**	**81.59**	**91.21**	**88.13**
ArtSci	NeuS [34]	0.67	33.73	31.89	30.24
	PET-NeuS [37]	0.32	21.30	15.69	18.07
	CRAYM (ours)	**0.19**	**59.01**	**52.54**	**55.59**

Table 9: Comparison of novel view synthesis on the real scene Bank from TwinTex [40].

Method		PSNR↑	SSIM↑	LPIPS↓
NeuS [34]	NIPS'21	11.69	0.25	0.91
PET-NeuS [37]	CVPR'23	13.22	0.27	0.93
CRAYM (ours)		**18.68**	**0.46**	**0.59**

Table 10: Comparison of novel view synthesis on the DTU [15] dataset.

Scene	Method		PSNR↑	SSIM↑	LPIPS↓
Scan24	NeuS [34]	NIPS′21	20.55	0.64	0.41
	PET-NeuS [37]	CVPR′23	20.77	0.65	0.42
	SPARF [32]	CVPR′23	22.80	0.78	0.29
	CRAYM (ours)		**30.53**	**0.96**	**0.02**
Scan37	NeuS [34]	NIPS′21	19.27	0.72	0.30
	PET-NeuS [37]	CVPR′23	18.68	0.71	0.30
	SPARF [32]	CVPR′23	21.99	0.72	0.31
	CRAYM (ours)		**28.64**	**0.95**	**0.03**
Scan40	NeuS [34]	NIPS′21	23.88	0.64	0.52
	PET-NeuS [37]	CVPR′23	22.78	0.63	0.51
	SPARF [32]	CVPR′23	23.85	0.72	0.38
	CRAYM (ours)		**26.86**	**0.91**	**0.06**
Scan55	NeuS [34]	NIPS′21	16.83	0.66	0.39
	PET-NeuS [37]	CVPR′23	23.10	0.73	0.38
	SPARF [32]	CVPR′23	19.80	0.63	0.49
	CRAYM (ours)		**30.96**	**0.98**	**0.02**
Scan63	NeuS [34]	NIPS′21	28.45	0.92	0.10
	PET-NeuS [37]	CVPR′23	26.89	0.90	0.09
	SPARF [32]	CVPR′23	27.45	0.92	0.12
	CRAYM (ours)		**31.30**	**0.98**	**0.01**

A.3 Reconstruction Quality

Next, we report the Hausdorff distance (HD), precision, recall, and F-score [17] for the reconstruction quality evaluated on the NeRF-Synthetic [26] dataset and the UrbanScene3D [22] dataset. The NeRF-Synthetic dataset contains eight synthetic objects, which are captured with 100 images. We evaluate our method on the two real scenes, PolyTech and ArtSci, from the UrbanScene3D [22] dataset, on which we measure the quality of the reconstructions with the high-resolution LiDAR scans as the ground truths.

Tables 7 and 8 report the reconstruction quality, compared with NeRF [26], BARF [21], SPARF [32], and L2G-NeRF [5] on the two datasets. A threshold of 0.01 is used to extract the inliers and outliers for the calculation of the precision, recall, and F1 score. The precision measures the reconstruction accuracy by calculating the distances from the reconstructed models to the ground truths. The recall measures the reconstruction completeness by determining the extent of the ground-truth points covered by the reconstructed models. A high F-score means both high accuracy and high completeness of the reconstructed models. As we can see from Tables 7 and 8, our method is able to produce high-quality reconstructions with both high accuracy and high completeness. It is worth to note that compared with other methods, our method achieves the top performance on all metrics consistently for both datasets.

A.4 Results on Real Scenes

Further, we evaluate our method on the real scene Bank from TwinTex [40], which is a set of high-resolution drone-captured images. We preprocess the images with COLMAP [29] to obtain the calibrated poses and perturb the poses with additive noise ξ, where $\xi \in \mathfrak{se}(3)$ and $\xi \in \mathcal{N}(0, n\mathbf{I})$, as the initial poses, following the procedure on real scenes PolyTech and ArtSci of UrbanScene3D [22]. For these real scenes, we set n as a small value 0.01.

Since TwinTex does not provide a LiDAR scan for the scene Bank, we only report the evaluation results of novel view synthesis in Table 9. Figure 10 shows the comparison on the real scene Bank visually with NeuS [34] and PET-NeuS [37]. While the other methods tend to generate renderings with blurriness, our method is able to produce sharper results with more fine details. Figure 9 further shows results of our method on another real scenes CSSE.

Table 11: Results of sparse input (3 views) on the LEGO scene from the NeRF-Synthetic dataset [26].

Method		PSNR↑	SSIM↑	LPIPS↓	CD↓
SPARF [32]	CVPR′23	15.91	0.69	**0.40**	1.27
CRAYM (ours)		**16.08**	**0.70**	0.41	**0.09**

A.5 Results on the DTU Dataset

The DTU [15] dataset is aimed at multi-view stereo (MVS) evaluation, containing image sets captured with structured light scanners mounted on an industrial robot arm. We evaluate our method on the first five image sets used in the NeuS [34], comparing with NeuS [34], PET-NeuS [37], and SPARF [32]. Each image set contains 48 images. We use 90% of them as training set and the remaining 10% images as testing data. The quantitative results of novel view synthesis on these data are shown in the Table 10. Figure 11 shows some of the novel view synthesis results visually. As we can see in Figure 11, our method produces renderings with much more details.

A.6 Sparse Views

Since SPARF is originally designed for sparse input with re-projection loss of dense pixel correspondences, we further evaluate our method on the LEGO data with sparse input, which contains only three views as the training images. The comparison of SPARF [32] and our method is shown in Table 11. Though CRAYM aims for bundle-adjusting neural implicit field with dense views as input, it still obtains a result comparable with SPARF, for sparse input views.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. All the claims are demonstrated in the experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the proposed method is discussed in the Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The details of the CRAYM are given in the Section 3, Appendix A.1.1, and Appendix A.1.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: All the experiments use open access dataset. The code is not released yet.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The data splits are set the same as previous paper. The training details are given in the Appendix A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 4 and Appendix A.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms with the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The proposed method is not tied to particular applications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the related papers are cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.