JacobiNeRF: NeRF Shaping with Mutual Information Gradients

Xiaomeng Xu1,* Yanchao Yang2,3,*,† Kaichun Mo3,4 Boxiao Pan3 Li Yi1,5,6 Leonidas Guibas3,7
1Tsinghua University 2The University of Hong Kong 3Stanford University 4NVIDIA Research
5Shanghai AI Laboratory 6Shanghai Qizhi Institute 7Google Research

Abstract

We propose a method that trains a neural radiance field (NeRF) to encode not only the appearance of the scene but also semantic correlations between scene points, regions, or entities – aiming to capture their mutual co-variation patterns. In contrast to the traditional first-order photometric reconstruction objective, our method explicitly regularizes the learning dynamics to align the Jacobians of highly-correlated entities, which proves to maximize the mutual information between them under random scene perturbations. By paying attention to this second-order information, we can shape a NeRF to express semantically meaningful synergies when the network weights are changed by a delta along the gradient of a single entity, region, or even a point. To demonstrate the merit of this mutual information modeling, we leverage the coordinated behavior of scene entities that emerges from our shaping to perform label propagation for semantic and instance segmentation. Our experiments show that a JacobiNeRF is more efficient in propagating annotations among 2D pixels and 3D points compared to NeRFs without mutual information shaping, especially in extremely sparse label regimes – thus reducing annotation burden. The same machinery can further be used for entity selection or scene modifications. Our code is available at https://github.com/xxm19/jacobinerf.

1. Introduction

When a real-world scene is perturbed, the response is generally local and semantically meaningful, e.g., a slight knock on a chair will result in a small displacement of just that chair. Such coherence in the perturbation of a scene evidences high mutual information between certain scene points or entities that can be leveraged to discover instances or semantic groups [37, 38]. A NeRF scene representation, however, solely supervised with 2D photometric loss may not converge to a configuration that reflects the actual scene...
structure [41]; even if the density is correctly estimated, the network in general will not be aware of the underlying semantic structure. As shown in Fig. 1, a perturbation on a specific entity of the scene through the network weights activates almost all other entities.

This lack of semantic awareness may not be a problem for view synthesis and browsing, but it clearly is of concern when such neural scene representations are employed for interactive tasks that require understanding the underlying scene structure, e.g., entity selection, annotation propagation, scene editing, and so on. All these tasks can be greatly aided by a representation that better reflects the correlations present in the underlying reality. We take a step towards endowing neural representations with such awareness of the mutual scene inter-dependencies by asking how it is possible to train a NeRF, so that it not only reproduces the appearance and geometry of the scene, but also generates coordinated responses between correlated entities when perturbed in the network parameter space.

Current approaches that encode semantics largely treat semantic labels (e.g., instance segmentation) [17, 42] as a separate channel, in addition to density or RGB radiance. However, in the semantics case, the value of the channel (e.g., instance ID) is typically an artifact of the implementation. What really matters is the decomposition of the 2D pixels (or of the scene 3D points) the NeRF encodes into groups – this is because semantics is more about relationships than values. Thus, we introduce an information-theoretic technique whose goal is to “shape” an implicit NeRF representation of a scene to better reflect the underlying regularities (“semantics”) of the world; so as to enforce consistent variation among correlated scene pixels, points, regions, or entities, enabling efficient information propagation within and across views.

The key to the proposed “shaping” technique is an equivalence between mutual information and the normalized inner product (cosine similarity) of the Jacobians at two pixels or 3D points. More explicitly, if we apply random delta perturbations to the NeRF weights, the induced random values of two pixels share mutual information up to the absolute cosine similarity of their gradients or Jacobians with respect to the weights computed at the unperturbed NeRF. This theoretical finding ensures a large correlation between scene entities with high mutual information – and thus coherent perturbation-induced behaviors – if their tangent spaces are aligned. Based on this insight, we apply contrastive learning to align the NeRF gradients with general-purpose self-supervised features (e.g., DINO), which is why we term our NeRF “JacobiNeRF”. While several prior works [16, 30] distill 1st-order semantic information from 2D views to get a consensus 1st-order feature in 3D, we instead regularize the NeRF using 2nd-order, mutual information based contrastive shaping on the NeRF gradients to achieve semantic consensus – now encoded in the NeRF tangent space.

The proposed NeRF shaping sets up resonances between correlated pixels or points and makes the propagation of all kinds of semantic information possible from sparse annotations – because pixels that co-vary with the annotated one are probably of the same semantics indicated by the mutual information equivalence. For example, we can use such resonances to propagate semantic or instance information as shown in Sec. 3.4, where we also show that our contrastive shaping can be applied to gradients of 2D pixels, or of 3D points. The same machinery also enables many other functions, including the ability to select an entity by clicking at one of its points or the propagation of appearance edits, as illustrated in Fig. 9. Additionally, our approach suggests the possibility that a NeRF shaped with rich 2nd-order relational information in the way described may be capable of propagating many additional kinds of semantics without further re-shaping – because the NeRF coefficients have already captured the essential “DNA” of points in the scene, of which different semantic aspects are just different expressions. In summary, our key contributions are:

- We propose the novel problem of shaping NeRFs to reflect mutual information correlations between scene entities under random scene perturbations.
- We show that the mutual information between any two scene entities is equivalent to the cosine similarity of their gradients with respect to the perturbed weights.
- We develop JacobiNeRF, a shaping technique that effectively encodes 2nd-order relational information into a NeRF tangent space via contrastive learning.
- We demonstrate the effectiveness of JacobiNeRF with state-of-the-art performance on sparse label propagation for both semantic and instance segmentation tasks.

2. Related Work

Neural Radiance Fields. Recent work has demonstrated promising results in implicitly parametrizing 3D scenes with neural networks. NeRF [23] is one notable work among many others [1, 7, 14, 19, 21, 24, 27, 35, 39, 41] that train deep networks to encode photometric attributes for novel view synthesis. Besides rendering quality, quite a few focus on the reconstructed geometry [8, 18, 25, 32, 34, 43]. There are also attempts towards making NeRF compositional, e.g., GRAF [28], CodeNeRF [13], CLIP-NeRF [31] and PNF [17] explicitly model shape and appearance so that one can modify the color or shape of an object by adjusting their separate codes. EditNeRF [20] further extends this direction by exploring different structures and tuning methods for edit propagation via delicate control of sampled rays. These methods, however, mostly work with object-level or category-level NeRFs in contrast to a whole scene
composed of many objects. Moreover, disentanglement in shape and appearance does not necessarily guarantee instance or semantic consistency. For example, [36] learns object-compositional neural radiance field for scene editing, yet instance masks are required during training. Our work studies how to further shape a holistic NeRF representation to enable awareness of the underlying semantic structure.

Self-supervised Representation Learning. Due to the absence of annotations, self-supervised feature learning has attracted much attention in recent literature [2, 4, 5, 10–12, 40]. Among these, DINO [3] proposes to perform self-distillation with a teacher and a student network, demonstrating that the features learned exhibit reasonable semantic information. There are also works utilizing self-supervised features for scene partition. For example, [22, 33] formulates unsupervised image decomposition as a traditional graph partitioning with affinity matrix computed from self-supervised features. Please refer to [15] for a more comprehensive overview of self-supervised feature learning techniques and their applications. Besides clustering with self-supervised features, CIS [37, 38] directly leverages mutual information learned in an adversarial manner to perform unsupervised object discovery. However, our focus is to encode mutual information in 3D scene representations. Moreover, our work can seamlessly incorporate arbitrary self-supervised features for learning meaningful correlations between scene entities.

2D to 3D Feature Distillation & Label Propagation 2D information annotated on NeRF 2D views (e.g., semantic labels) can be pushed into the 3D structure of a NeRF [42]. And the process can be regularized and improved, for example, N3F [30] treats 2D image features from pretrained networks as additional color channels and train a NeRF to reconstruct the augmented radiance. DFF [16] similarly distills knowledge of off-the-shelf 2D image feature extractors into NeRFs with volume rendering. In contrast to feature distillation, Semantic-NeRF [42] adds a parallel branch along the color one in NeRF [23] to predict semantic logits supervised by annotated pixels or images. With similar technique, Panoptic-NeRF [9] encodes coarse and noisy annotations into NeRFs. Both demonstrate that volume rendering improves 3D consistency, and is capable of denoising and interpolating imperfect 2D labels. However, as they are first-order, they can only replicate the provided supervision and cannot be used for other tasks.

3. Method

We seek scene representations that not only mirror the appearance and geometry of a scene, but also encode mutual correlations between scene regions and entities – in the sense that the representation facilitates the generation of semantically meaningful scene perturbations which will change such regions and entities in coordinated ways. Our study here mainly focuses on NeRFs, due to their capability for high-fidelity view synthesis. We show, however, that in the absence of correlation regularization on the learning dynamics, solely reconstructing the scene with standard photometric losses, does not guarantee semantic synergies between entities under various scene perturbations.

In order to analyze how NeRFs render mutual correlation under perturbations, we derive that the mutual information (MI) between any two scene entities is equivalent to the cosine similarity of their gradients with respect to the perturbed network parameters. Based on this equivalence, we propose a training regimen that biases the parameters to properly encode mutual correlation between 3D points or 2D pixels. Our method shapes NeRFs in a way that minimizes the synthesis discrepancy, while maximizing the synergy between correlated scene entities when perturbing along the gradients, without changing the architecture. We further show how this mutual-information-induced synergy can be leveraged to propagate labels for both semantic and instance segmentation.

3.1. NeRF preliminaries

Given a set of posed images \( \{I_k\} \), NeRFs aim at learning an implicit field representation of the scene from which new views can be generated through volume rendering density and radiance values. Denote \( x \in \mathbb{R}^3 \) as a point in 3D whose radiance is determined by a color function \( c : \mathbb{R}^3 \times S^2 \to \mathbb{R}^3 \), mapping the point coordinate \( x \) and a viewing direction \( v \in S^2 \) to an RGB value. Also, denote \( p \) as a pixel on the image plane specified by a camera center \( o \) and a direction \( v \), the volume rendering procedure that generates the pixel value of \( p \) can be described by:

\[
I(p) = \Phi(o, v; \theta) = \int_0^{+\infty} w(t; \theta)c(p(t), v; \theta)dt,
\]

where \( \{p(t) = o + tv \mid t \geq 0\} \) is the camera ray passing through the camera center \( o \) and the pixel \( p \), and \( w \) is a weight function computed from the density values.
MLP formulation. Please refer to [23] for more details on volume rendering and \( w \), together with how to perform the integration in Eq. (1) under discretization.

The natural way to perturb a scene represented by such a NeRF is to perturb the MLP parameters. Unless otherwise mentioned, all trainable parameters are compacted into the vector \( \theta \) in Eq. (1). Traditional NeRFs losses are 1st-order, optimizing values such as density and color. In this work we focus additionally and crucially on 2nd-order supervision, optimizing correlations between scene entities when the scene is perturbed through changes in the MLP parameters. In this way we aim to “shape” the NeRF MLP to better reflect any supervision we may have on the co-variation structure of the scene contents. Next, we detail the perturbations and the derivation of mutual information in an analytical form.

### 3.2. Mutual information approximation by Jacobian inner products

We study the correlation under MLP parameter perturbations of the values produced by the NeRF, either at 3D points, or (after volume rendering) at 2D view pixels. For concreteness, we focus on two pixels \( p_i \) and \( p_j \) and their gray-scale values \( I(p_i) \) and \( I(p_j) \), respectively (the pixels may or may not come from the same view):

\[
I(p_i) = \Phi(o_i, v_i; \theta), \\
I(p_j) = \Phi(o_j, v_j; \theta).
\]

Further, we denote by \( \theta^D \) the set of parameters that will be perturbed by a random noise vector \( n \in \mathbb{R}^D \) sampled from a uniform distribution on the sphere \( S^{D-1} \). Please see Fig. 2 for different selection patterns of \( \theta^D \). The random variables representing the perturbed pixel values are then:

\[
\hat{I}(p_i) = \Phi(o_i, v_i; \theta^D + n), \\
\hat{I}(p_j) = \Phi(o_j, v_j; \theta^D + n),
\]

and we omit parameters that remain unchanged for clarity.

Now we characterize the mutual information between \( \hat{I}(p_i) \) and \( \hat{I}(p_j) \) under the perturbation-induced joint probability distribution \( \mathbb{P}(\hat{I}(p_i), \hat{I}(p_j)) \). However, calculating the joint distribution under a push-forward of the MLP is complicated due to non-linearities. Thus, we proceed by constraining the magnitude of the perturbations, namely, multiplying the random noise \( n \) by \( \sigma \ll 1.0 \). This constraint improves compliance with the fact that a small perturbation in the physical scene is enough to reveal mutual information between scene entities. For example, slightly pushing a chair can generate a motion that shows correlations between different parts of the environment. Moreover, it makes it likely that the perturbed representation still represents a legitimate scene. With this constraint, we can explicitly write the random variables under consideration as:

\[
\hat{I}(p_i) = I(p_i) + \sigma n \cdot \frac{\partial \Phi(o_i, v_i; \theta)}{\partial \theta^D}, \\
\hat{I}(p_j) = I(p_j) + \sigma n \cdot \frac{\partial \Phi(o_j, v_j; \theta)}{\partial \theta^D},
\]

following a Taylor expansion. We denote the respective Jacobians as \( \partial \Phi_i / \partial \theta^D \) or \( \partial \Phi_j \) for notational ease. We can then show that the mutual information is:

\[
\mathbb{I}(\hat{I}(p_i), \hat{I}(p_j)) = \mathbb{H}(\hat{I}(p_j)) - \mathbb{H}(\hat{I}(p_j) | \hat{I}(p_i)) \\
= \mathbb{H}(\sigma n \cdot \partial \Phi_j) - \mathbb{H}(\sigma n \cdot \partial \Phi_j | \sigma n \cdot \partial \Phi_i),
\]

leveraging the fact that entropy is translation-invariant. Furthermore, by writing the random noise and the Jacobians in spherical coordinates, we can derive that:

\[
\mathbb{I}(\hat{I}(p_i), \hat{I}(p_j)) = \log\left( \frac{1}{\sqrt{1 - \cos^2 \alpha}} \right) + \text{const}. \quad (5)
\]

Here, \( \alpha \) is the angle between \( \partial \Phi_i \) and \( \partial \Phi_j \). For more derivation details, please refer to the Appendix. The key insight is that the mutual information between the perturbed pixels is positively correlated with the absolute value of the cosine similarity of their gradients with respect to the perturbed parameters. More explicitly, if \( \partial \Phi_i \) and \( \partial \Phi_j \) are pointing at the same or opposite direction, i.e., \( \| \cos \alpha \| \) is close to 1, then the mutual information between \( \hat{I}(p_i) \) and \( \hat{I}(p_j) \) becomes infinity (maximized). Otherwise, if \( \partial \Phi_i \) and \( \partial \Phi_j \) are perpendicular, i.e., \( \| \cos \alpha \| \) is 0, then the mutual information between the perturbed pixels is minimized.

With this analytical expression of the mutual information, we can efficiently check how different entities from the same scene are correlated. As observed in Fig. 3, NeRF
weights obtained solely by the reconstruction loss do not reveal meaningful correspondences. Next, we describe our method that biases the training dynamics of NeRFs so that the shaped weights not only reconstruct the scene well but also reflect the mutual correlation between scene entities.

3.3. Shaping neural radiance fields with mutual information gradients

According to Eq. (5), if we want to render two pixels or points correlated (with high mutual information), we can align their gradients regarding the perturbed parameters. On the other, for entities that share little mutual information, we like their gradients to be orthogonal. Suppose $(p_i, p_{i+})$ is a pair of highly correlated pixels, whereas $(p_i, p_{i-})$ are independent, then we should observe:

$$\|\partial \Phi_i^T \partial \Phi_{i+}\| > \|\partial \Phi_i^T \partial \Phi_{i-}\|.$$  

(6)

Since our goal is to encode the relative correlations between different pairs of scene entities instead of the exact mutual information values, it suffices to minimize the InfoNCE [26] loss with positive and negative gradient pairs:

$$L_{\text{MIG}} = -\log \frac{\exp([\|\cos(\partial \Phi_i, \partial \Phi_{i+})\|/\tau])}{\sum_i \sum_{i'} \exp([\|\cos(\partial \Phi_i, \partial \Phi_{i'})\|/\tau])},$$  

(7)

where $\tau$ is the temperature and we (ab)use $\cos$ for cosine similarity. Note, $L_{\text{MIG}}$ encourages highly correlated (positive) points to have large cosine similarity by the pull through the numerator.

The question now is how to select positive and negative gradient pairs – which seems require knowledge about the mutual information between scene entities. Unfortunately, since the posed images for training NeRFs are from a (static) snapshot of the physical scene, the joint distribution needed to compute the mutual information is difficult to recover. This necessitates that we resort to external sources for surrogates of the mutual information, for which, fortunately, we have several candidates. For example, self-supervised features can come from contrastive learning methods. We choose DINO features [3] as our primary surrogate due to their capability to capture semantic similarity while maintaining reasonable discriminability. We can of course also accept direct supervision through off-the-shelf external semantic and instance segmentation tools. An interesting issue that we investigate is exactly how much such supervision is needed for meaningful NeRF shaping. We detail the selection process and our results in Sec. 4.2.

With positive and negative samples, we can write the training loss that endows NeRFs with mutual-information-awareness in the tangent space (perturbations) as:

$$L_{\text{TM}} = L_{\text{NeRF}} + \lambda L_{\text{MIG}} + \gamma(1.0 - \|\partial \Phi_i\|)^2,$$  

(8)

where $L_{\text{NeRF}}$ is for photometric reconstruction, and $L_{\text{MIG}}$ shapes NeRF with mutual information (MI-shaping) through cosine similarity of gradients. The last term improves the training efficiency by compacting the gradients onto a unit sphere, which also facilitates label propagation with perturbation response in the following. Please see Fig. (4) (left) for a visual illustration of MI-shaping.

We term our 2nd-order NeRF “JacobiNeRF” exactly be-
cause of the use of the inner products of Jacobians to define these 2nd-order losses that capture mutual information. Our MI-shaping regiment can be thought of a NeRF operator, acting to better align the tangent space of a given NeRF with information we have on mutual correlations between scene pixels, points, regions, or entities.

3.4. Label propagation with JacobiNeRF

Besides revealing correlations under small perturbations, we can also leverage the synergy between different scene entities in a JacobiNeRF to perform label propagation by either transporting annotations from one view to another or to densify labels from a few annotated points to the remaining ones. Next, we illustrate the propagation procedure from a source view \( I(p_i^s) = \Phi^s(o^s, v_i^s; \theta) \) to a target \( I(p_i^t) = \Phi^t(o^t, v_i^t; \theta) \) with semantic segmentation – note that the same method can be directly applied to instance segmentation as well as to different combinations of views.

Suppose we have a list of labeled pixels from the source view \( \{(p_k^s, l_k^s)\}_{k=1,...,K} \) with one label for each of the \( K \) classes, i.e., \( l_k^s = k \). The goal is to determine the semantic labels for every pixel \( p_i^t \) of the target view. In principle, we can perform a maximum a posteriori (MAP) estimation for each target pixel by:

\[
      l_i^t = \arg \max_{l_i^t} \mathbb{P}(l_i^t | \{(p_k^s, l_k^s\}), \Phi^s).  \quad \text{(9)}
\]

If the mutual-information shaping described in Sec. 3.3 converges properly, we can assume conditional independence between uncorrelated entities. Then the target label \( l_i^t \) depends only on the source pixel \( p_k^s \), which conveys the maximum mutual information towards \( p_i^t \). Thus, it is legitimate to assign \( l_k^s \) or \( k^* \) to \( p_i^t \) in order to maximize the posterior in Eq. (9). In other words, \( l_i^t = l_k^{s*} \), so that:

\[
       k^* = \arg \max_k \mathbb{I}(\Phi^s(o^s, v_k^s; \theta), \Phi^t(o^t, v_i^t; \theta)), \quad \text{(10)}
\]

\[
       = \arg \max_k \frac{||\partial \Phi_k^{1,s}||}{\partial \Phi_i^{1,t}} ||\partial \Phi_k^{1,s}||. \quad \text{(11)}
\]

As encouraged by the third term of the shaping loss in Eq. (8), the norm of the gradients should be close to 1.0, which allows us to approximate the cosine similarity between \( \partial \Phi_k^{1,s} \) and all \( \partial \Phi_k^{1,t} \)s by a single delta perturbation along \( \partial \Phi_k^{1,s} \) as evidenced by the Taylor expansion. Namely, for each of the labeled pixels, we first generate a perturbed JacobiNeRF along its gradient, i.e.,

\[
       \Phi^t(:, \theta + \sigma \partial \Phi_k^{1,s}), \quad k = 1...K. \quad \text{(12)}
\]

Then each of the perturbed JacobiNeRFs will be used to synthesize a perturbed image in the target view:

\[
       I_k(p_i^t) = \Phi^t(o^t, v_i^t; \theta + \sigma \partial \Phi_k^{1,s}), \quad k = 1...K. \quad \text{(13)}
\]

Next, we calculate the perturbation response as the absolute difference between the perturbed and original target images:

\[
       R_k(p_i^t) = | I_k(p_i^t) - I(p_i^t) |, \quad k = 1...K. \quad \text{(14)}
\]

Finally, since the perturbation response \( R_k(p_i^t) \) resembles the mutual information between \( I(p_k^s) \) and \( I(p_i^t) \) (Eq. (11)), we can treat the concatenation \([R_k(p_i^t)]\) as the logits for a K-way semantic segmentation on the target image. Thus, we obtain the semantic label for \( p_i^t \) as:

\[
       l_i^t = \arg \max_k R_k(p_i^t). \quad \text{(15)}
\]

Please note that the propagation principle discussed above is applicable to any task that is view-invariant. For example, \( K \) can be the number of entity instances in the scene, and we perform propagation for instance segmentation. Here, the absolute difference logits are computed in the 2D domain (after volume rendering) in Eq. (14). However, we can also measure the perturbation differences in 3D first, e.g., estimate difference logits for a 3D point along the ray emanating from a certain pixel (following the hierarchical sampling strategy of [23]), and then leverage volume rendering to accumulate the sampled perturbation differences and arrive at a 2D logit. In this 3D case, noise in the perturbation may be averaged out, improving the result.

We name the results obtained directly in 2D as J-NeRF 2D while the latter as J-NeRF 3D.

4. Experiments

We first describe the datasets in Sec. 4.1, which we use for evaluating JacobiNeRF on semantic and instance label propagation in Sec. 4.2 and Sec. 4.3, respectively. We then perform an extensive ablation in Sec. 4.4 on the hyper-parameters. Results show that JacobiNeRF is effective in propagating annotations with the encoded correlations, especially in the very sparse label regimes.

4.1. Datasets

Replica [29] is a synthetic indoor dataset with high-quality geometry, texture, and semantic annotations. We pick the 7 scenes selected by [42] for fair comparison. Each of the scenes comes with 900 posed frames, which are partitioned into a training set and a test set of 180 frames respectively.

ScanNet [6] is a real-world RGB-D indoor dataset. Again, we uniformly sample training and test frames, and ensure that the training and test sets do not overlap and contain approximately the same number of frames (from 180 to 200).

4.2. Label propagation for semantic segmentation

Experimental setting. The training of JacobiNeRF follows Sec. 3.3, and for test-time label propagation, we implement two schemes, i.e., J-NeRF 2D and J-NeRF 3D following Sec. 3.4. Please also refer to the appendix for more details.

We test the effectiveness of segmentation label propagation under two settings. In the sparse setting, we only provide a single randomly selected pixel label for each class in
the source view, simulating the real-world annotation scenario of user clicking. While for the dense setting, all pixels in the source view are annotated. This is useful when we want to obtain fine-grained labels with a higher annotation cost. The propagation strategy for the sparse setting is elaborated in Sec. 3.4. For the dense setting, we employ an adaptive sampling strategy to select the most representative gradients for each class to save possibly redundant perturbations. Furthermore, to best leverage the given dense labels, we apply a lightweight decoding MLP to enhance the argmax operator described in Sec. 3.4. More details can be found in the appendix.

We compare JacobiNeRF (J-NeRF) to Semantic-NeRF [42] (S-NeRF), which adds a semantic branch to the original NeRF, and hence predicts semantic labels from the color feature integrated into the radiance field. We also compare to DINO-2D, which extracts DINO features from the images and propagates with DINO feature similarity; and DINO-NeRF [16] (D-NeRF), which distills DINO features to a DINO branch appended to NeRF, and propagates labels with feature similarity of the volume rendered DINO features. All methods are tested given the same source view labels. We evaluate with three standard metrics. Namely, mean intersection-over-union (mIoU), averaged class accuracy, and total accuracy. The scores are obtained by averaging all test views for each scene. Since we only provide sparse or dense labels from one view, some classes in the test views may not be seen from the source view. Therefore, we exclude them and only evaluate with the seen classes.

Results. Tab. 1 summarizes the quantitative results for semantic segmentation propagation on the 7 scenes from the Replica [29] dataset. Our method consistently achieves the best performance in both sparse and dense settings by utilizing correlations encoded in the tangent space.

Figure 5 compares JacobiNeRF with the baselines qualitatively. As observed, with sparse annotations, JacobiNeRF propagates to novel views with much better quality and smoothness than the baselines. In the dense setting, JacobiNeRF demonstrates the ability to propagate labels at a finer granularity on both synthetic and real-world data.

4.3. Label propagation for instance segmentation

Experimental setting. The training and test settings are the same as the semantic segmentation procedure in Sec. 4.2.

Results. In Tab. 2, we show the results of instance segmen-
4.4. Ablation study

Table 2. Label propagation for instance segmentation on ScanNet with sparse (S.) and dense (D.) annotations.

<table>
<thead>
<tr>
<th>View Distance</th>
<th>S-NeRF</th>
<th>J-NeRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>mIoU</td>
<td>mIoU</td>
</tr>
<tr>
<td></td>
<td>0.77</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>0.72</td>
<td>0.77</td>
</tr>
<tr>
<td>Far</td>
<td>0.28</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td>0.54</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 3. Left: propagation performance on near and distant views (dense). Right: comparison with multiview dense supervision.

4.4. Ablation study

Label density. Fig. 7 shows the semantic label propagation performance of our method on one scene under various label density settings. We provide dense labels from one view, and randomly select a sub-region of labels for each class, following the setting in [42] with varying density. The horizontal axis denotes the percentage (in area) of used labels for each class. As the density of the label increases, the propagation performance also gets better.

Views far from the source view. We find that S-NeRF overfits to the source view with low-quality propagation on distant views. In contrast, J-NeRF generalizes much better as the shaping can be applied on all views (Tab. 3, left).

Multiview dense supervision. We report the results with multiview supervision in Tab. 3 (right). The gap between S-NeRF and J-NeRF decreases as the view number increases, due to the sub-optimal downsampling of dense labels with limited computation. A more sufficient and efficient downsampling scheme is our next goal.

Perturbation magnitude. We study how different perturbation scales, i.e., the $\sigma$ in Eq. 12, affect the label propagation performance. Fig. 8 shows the results of J-NeRF 2D in the sparse setting. As observed, if the magnitude is too small, the propagation magnitude $\sigma$ for label propagation is not good due to weak (noisy) responses. However, if the magnitude is too large, the approximation with Taylor expansion in Eq. 2 becomes invalid, thus producing worse results. So we treat it as a hyper-parameter and empirically set $\sigma = 0.1$ for all runs.

5. Conclusion

We have demonstrated a way to regularize the learning dynamics of a NeRF, so it reflects the correlations between high mutual information entities. This is achieved through the MI-shaping technique that aligns gradients of these correlated entities with respect to the network parameters via a contrastive learning. Once shaped in the tangent space with second-order training, the induced resonances in a NeRF can propagate value information from selected quantities to other correlated quantities leveraging the mutual-information-induced synergies. We demonstrate this capability by the propagation of semantic and instance information, as well as semantic entity selection and editing. Currently, the shaping scheme relies on self-supervised visual features, but it can be easily oriented to consume features from foundation models to encode cross-modal mutual correlations. We expect more correlation-informed applications to be possible using the proposed mutual information gradient alignment techniques for NeRFs.

Acknowledgement: We thank the support of a grant from the Stanford Human-Centered AI Institute, an HKU-100 research award, and a Vannevar Bush Faculty Fellowship.
References


[24] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multires-


[40] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 2, 3, 6, 7, 8