Locality-Aware Generalizable Implicit Neural Representation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generalizable implicit neural representation (INR) enables a single continuous function, i.e., a coordinate-based neural network, to represent multiple data instances by modulating its weights or intermediate features using latent codes. However, the expressive power of the state-of-the-art modulation is limited due to its inability to localize and capture fine-grained details of data entities such as specific pixels and rays. To address this issue, we propose a novel framework for generalizable INR that combines a transformer encoder with a locality-aware INR decoder. The transformer encoder predicts a set of latent tokens from a data instance to encode local information into each latent token. The locality-aware INR decoder extracts a modulation vector by selectively aggregating the latent tokens via cross-attention for a coordinate input and then predicts the output by progressively decoding with coarse-to-fine modulation through multiple frequency bandwidths. The selective token aggregation and the multi-band feature modulation enable us to learn locality-aware representation in spatial and spectral aspects, respectively. Our framework significantly outperforms previous generalizable INRs and validates the usefulness of the locality-aware latents for downstream tasks such as image generation.

1 Introduction

Recent advances in generalizable implicit neural representation (INR) enable a single coordinate-based multi-layer perceptron (MLP) to represent multiple data instances as a continuous function. Instead of per-sample training of individual coordinate-based MLPs, generalizable INR extracts latent codes of data instances [13, 14, 40] to modulate the weights or intermediate features of the shared MLP model [8, 11, 19, 35]. However, despite the advances in previous approaches, their performance is still insufficient compared with individual training of INRs per sample.

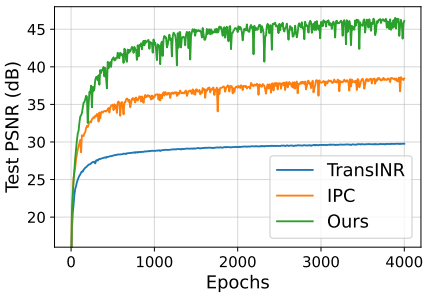

Figure 1: Learning curves of PSNRs during training on ImageNette 178×178.

We postulate that the expressive power of generalizable INR is limited by the inability to exploit the locality-aware latent representation of data. The locality of data entities has been a significant inductive bias [3] for modeling the representations of complex data, such as images, multi-views, or graphs. However, previous approaches prevent the latent codes from learning the locality of data. For example, when latent codes modulate the intermediate features [11, 12] or weight matrices [8, 19, 35] of an INR decoder, the modulation methods do not specify the location of input coordinates to exploit the latent codes. Thus,

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

the latent codes encode the global information in whole coordinates without capturing the local relationship between data entities, such as specific pixels.

To address this issue, we propose a novel framework for locality-aware generalizable INR to localize and control the fine-grained details of data. Given a data instance, our Transformer [37] encoder first extracts a set of latent tokens, while analyzing relevant local information of data into different latent tokens. Especially, our locality-aware INR decoder can guide the Transformer encoder to encapsulate the local information into each latent token and exploit the latents to predict the fine-grained details of outputs effectively. Specifically, given an input coordinate, our INR decoder selectively aggregates the spatially local information in the latent tokens and extracts a modulation vector. Then, the modulation vector is decomposed into multiple bandwidths of frequency features to amplify the high-frequency information in the modulation vector. Finally, our multi-band feature modulation progressively composes the intermediate features of the INR decoder using a coarse-to-fine approach in a frequency domain, while encouraging the INR decoder to effectively capture the high-frequency details in the outputs. We conduct extensive experiments to demonstrate the outperformance and efficacy of our locality-aware generalizable INR on benchmarks as shown in Figure 1. In addition, our locality-aware latents can also be utilized for downstream tasks such as image synthesis.

Our main contributions can be summarized as follows: 1) We propose an effective framework for generalizable INR with a Transformer encoder and locality-aware INR decoder. 2) The proposed INR decoder with selective token aggregation and multi-band feature modulation can effectively capture the local information to predict the fine-grained data details. 3) The extensive experiments validate the efficacy of our framework and show its applications to a downstream image generation task.

2 Related Work

Implicit neural representations (INRs). INRs use neural networks to represent complex data such as audio, images, and 3D scenes, as continuous functions. Especially, incorporating Fourier features [24, 36], periodic activations [31], or multi-grid features [25] significantly improves the performance of INRs. Despite its broad applications [1, 6, 10, 32, 34], INRs commonly require separate training of MLPs to represent each data instance. Thus, individual training of INRs per sample does not learn common representations in multiple data instances.

Generalizable INRs. Previous approaches focus on two major components for generalizable INRs; latent feature extraction and modulation methods. Auto-decoding [23, 26] computes a latent vector per data instance and concatenates it with the input of a coordinate-based MLP. Given input data, gradient-based meta-learning [4, 11, 12] adapts a shared latent vector using a few update steps to scale and shift the intermediate activations of the MLP. Learned Init [35] also uses gradient-based meta-learning but adapts whole weights of the shared MLP. Although auto-decoding and gradient-based meta-learning are agnostic to the types of data, their training is unstable on complex and large-scale datasets. TransINR [8] employs the Transformer [37] as a hypernetwork to predict latent vectors to modulate the weights of the shared MLP. In addition, Instance Pattern Composers [19] have demonstrated that modulating the weights of the second MLP layer is enough to achieve high performance of generalizable INRs. Our framework also employs the Transformer encoder, but focuses on extracting locality-aware latent features for the high performance of generalizable INR.

Leveraging Locality of Data for INRs Local information in data has been utilized for efficient modeling of INRs, since local relationships between data entities are widely used for effective process of complex data [3]. Given an input coordinate, the coordinate-based MLP only uses latent vectors nearby the coordinate, after a CNN encoder extracts a 2D grid feature map of an image for super-resolution [7] and reconstruction [22]. Recently, Spatial Functa [4] also demonstrates that leveraging the locality of data enables INRs to be utilized for downstream tasks such as image recognition and generation. Local information in 3D coordinates has also been effective for scene modeling using 3D feature grids [18] or the part segmentation [17] of a 3D object. However, previous approaches assume explicit grid structures of latents tailored to a specific data type. Since we do not predefine a specific relationship between latent features, our framework is flexible to learn and encode the local information of both grid coordinates in images and non-grid coordinates in light fields.

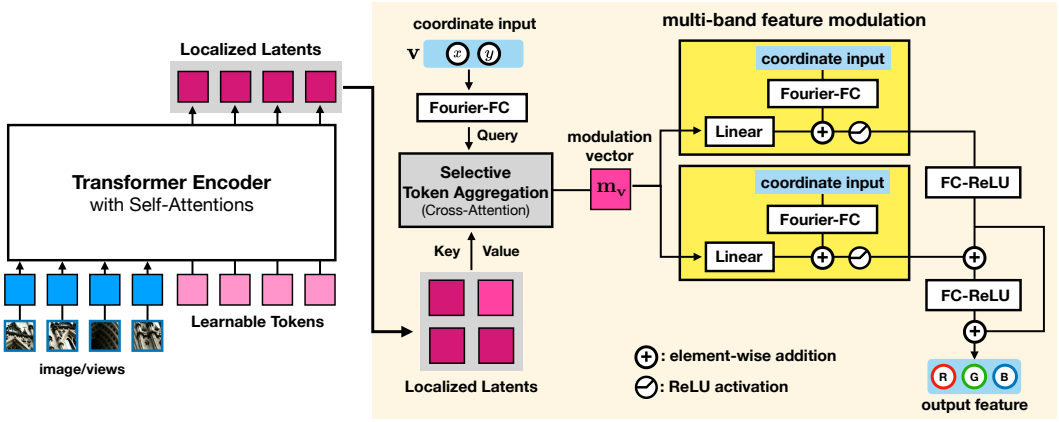

Figure 2: Overview of our framework for locality-aware generalizable INR. Given a data instance, Transformer encoder extracts its localized latents. Then, the locality-aware INR decoder uses selective token aggregation and multi-band feature modulation to predict the output for the input coordinate.

3 Methods

We propose a novel framework for *locality-aware generalizable INR* which consists of a Transformer encoder to localize the information in data into latent tokens and a locality-aware INR decoder to exploit the localized latents and predict outputs. First, we formulate how generalizable INR enables a single coordinate-based neural network to represent multiple data instances as a continuous function by modulating its weights or features. Then, after we introduce the Transformer encoder to extract a set of latent tokens from input data instances, we explain the details of the locality-aware INR decoder, where *selective token selection* aggregates the spatially local information for an input coordinate via cross-attention; *multi-band feature modulation* leverages a different range of frequency bandwidths to progressively decode the local information using coarse-to-fine modulation in the spectral domain.

3.1 Generalizable Implicit Neural Representation

Given a set of data instances $\mathcal{X} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$, each data instance $\mathbf{x}^{(n)} = \{(\mathbf{v}_i^{(n)}, \mathbf{y}_i^{(n)})\}_{i=1}^{M_n}$ comprises M_n pairs of an input coordinate $\mathbf{v}_i^{(n)} \in \mathbb{R}^{d_{\text{in}}}$ and the corresponding output feature $\mathbf{y}_i^{(n)} \in \mathbb{R}^{d_{\text{out}}}$. Conventional approaches [24, 31, 36] adopt individual coordinate-based MLPs to train and memorize each data instance $\mathbf{x}^{(n)}$. Thus, the coordinate-based MLP cannot be reused and generalized to represent other data instances, requiring per-sample optimization of MLPs for unseen data instances.

A generalizable INR uses a single coordinate-based MLP as a shared INR decoder $F_\theta : \mathbb{R}^{d_{\text{in}}} \rightarrow \mathbb{R}^{d_{\text{out}}}$ to represent multiple data instances as a continuous function. Generalizable INR [8, 11, 12, 19, 26] extracts the R number of latent codes $\mathbf{Z}^{(n)} = \{\mathbf{z}_k^{(n)} \in \mathbb{R}^d\}_{k=1}^{R}$ from a data instance $\mathbf{x}^{(n)}$. Then, the latents are used for the INR decoder to represent a data instance $\mathbf{x}^{(n)}$ as $\mathbf{y}_i^{(n)} = F_\theta(\mathbf{v}_i^{(n)}; \mathbf{Z}^{(n)})$, while updating the parameters θ and latents $\mathbf{Z}^{(n)}$ to minimize the errors over $\mathcal{X}$:

$$\min_{\theta, \mathbf{Z}^{(n)}} \frac{1}{NM_n} \sum_{n=1}^{N} \sum_{i=1}^{M_n} \left\| \mathbf{y}_i^{(n)} - F_\theta(\mathbf{v}_i^{(n)}; \mathbf{Z}^{(n)}) \right\|_2^2. \tag{1}$$

We remark that each previous approach employs a different number of latent codes to modulate a coordinate-based MLP. For example, a single latent vector ($R = 1$) is commonly extracted to modulate intermediate features of the MLP [11, 12, 26], while a multitude of latents ($R > 1$) are used to modulate its weights [8, 19, 35]. While we modulate the features of MLP, we extract a set of latent codes to localize the information of data to leverage the locality-awareness for latent features.

3.2 Transformer Encoder

Our framework employs a Transformer encoder [37] to extract a set of latents $\mathbf{Z}^{(n)}$ for each data instance $\mathbf{x}^{(n)}$ as shown in Figure 2. After a data instance, such as an image or multi-view images, is patchified into a sequence of data tokens, we concatenate the patchified tokens into a sequence of R learnable tokens as the encoder input. Then, the Transformer encoder extracts a set of latent tokens, where each latent token corresponds to an input learnable token. Note that our encoder does not predefine a relationship between latent tokens, since a self-attention in Transformer is a permutation-equivariant operation. Thus, whether a data instance is represented on a grid or non-grid coordinate, our framework is flexible to encode various types of data into latent tokens, while learning the local relationships of latent tokens during training.

3.3 Locality-Aware Decoder for Implicit Neural Representations

We propose the locality-aware INR decoder in Figure 2 to leverage the local information of data for effective generalizable INR. Our INR decoder comprises two primary components: i) *Selective token aggregation via cross attention* extracts a modulation vector for an input coordinate to aggregate spatially local information from latent tokens. ii) *Multi-band feature modulation* decomposes the modulation vector into multiple bandwidths of frequency features to amplify the high-frequency features and effectively predict the details of outputs.

3.3.1 Selective Token Aggregation via Cross-Attention

We remark that encoding locality-aware latent tokens is not straightforward since the self-attentions in Transformer do not guarantee a specific relationship between tokens. Thus, the properties of the latent tokens are determined by a modulation method for generalizable INR to exploit the extracted latents. For example, given an input coordinate $\mathbf{v}$ and latent tokens $\{\mathbf{z}_1, ..., \mathbf{z}_R\}$, a straightforward method can use Instance Pattern Composers [19] to construct a modulation weight $\mathbf{W}_{\text{m}} = [\mathbf{z}_1, ..., \mathbf{z}_R]^\top \in \mathbb{R}^{R \times d_{\text{in}}}$ and extract a modulation vector $\mathbf{m}_{\mathbf{v}} = \mathbf{W}_{\text{m}}\mathbf{v} = [\mathbf{z}_1^\top \mathbf{v}, ..., \mathbf{z}_R^\top \mathbf{v}]^\top \in \mathbb{R}^R$. However, the latent tokens cannot encode the local information of data, since each latent token equally influences each channel of the modulation vector regardless of the coordinate locations (see Section 4.3).

Our selective token aggregation employs cross-attention to aggregate the spatially local latents nearby the input coordinate, while guiding the latents to be locality-aware. Given a set of latent tokens $\mathbf{Z}^{(n)} = \{\mathbf{z}_k^{(n)}\}_{k=1}^R$ and a coordinate $\mathbf{v}_i^{(n)}$, a modulation feature vector $\mathbf{m}_{\mathbf{v}_i}^{(n)} \in \mathbb{R}^d$ shifts the intermediate features of an INR decoder to predict the output, where d is the dimensionality of hidden layers in the INR decoder. For the brevity of notation, we omit the superscript n and subscript i.

Frequency features We first transform an input coordinate $\mathbf{v} = (v_1, \cdots, v_{d_{\text{in}}}) \in \mathbb{R}^{d_{\text{in}}}$ into frequency features using sinusoidal positional encoding [31, 36]. We define the Fourier features $\gamma_\sigma(\mathbf{v}) \in \mathbb{R}^{d_{\text{F}}}$ with bandwidth $\sigma > 1$ and feature dimensionality d_{F} as

$$\gamma_\sigma(\mathbf{v}) = [\cos(\pi\omega_j v_i), \sin(\pi\omega_j v_i) : i = 1, \cdots, d_{\text{in}}, j = 0, \cdots, n-1] \tag{2}$$

where $n = \frac{d_{\text{F}}}{2d_{\text{in}}}$. A frequency $\omega_j = \sigma^{j/(n-1)}$ is evenly distributed between 1 and σ on a log-scale. Based on the Fourier features, we define the *frequency feature* extraction $\mathbf{h}_{\text{F}}(\cdot)$ as

$$\mathbf{h}_{\text{F}}(\mathbf{v}; \sigma, \mathbf{W}, \mathbf{b}) = \text{ReLU}\left(\mathbf{W}\gamma_\sigma(\mathbf{v}) + \mathbf{b}\right), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{d \times d_{\text{F}}}$ and $\mathbf{b} \in \mathbb{R}^d$ are trainable parameters for frequency features, d denotes the dimensionality of hidden layers in the INR decoder.

Selective token selection via cross-attention To predict corresponding output $\mathbf{y}$ to the coordinate $\mathbf{v}$, we adopt a cross-attention to extract a modulation feature vector $\mathbf{m}_{\mathbf{v}} \in \mathbb{R}^d$ based on the latent tokens $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^R$. We first extract the frequency features of the coordinate $\mathbf{v}$ in Eq (3) as the query of the cross-attention as

$$\mathbf{q}_{\mathbf{v}} := \mathbf{h}_{\text{F}}(\mathbf{v}; \sigma_{\text{q}}, \mathbf{W}_{\text{q}}, \mathbf{b}_{\text{q}}), \tag{4}$$

where $\mathbf{W}_{\text{q}} \in \mathbb{R}^{d \times d_{\text{F}}}$ and $\mathbf{b}_{\text{q}} \in \mathbb{R}^d$ are trainable parameters, and σ_{q} is the bandwidth for query frequency features. The cross-attention in Figure 2 enables the query to select latent tokens, aggregate its local information, and extract the modulation feature vector $\mathbf{m}_{\mathbf{v}}$ for the input coordinate:

$$\mathbf{m}_{\mathbf{v}} := \text{MultiHeadAttention}(\text{Query} = \mathbf{q}_{\mathbf{v}}, \text{Key} = \mathbf{Z}, \text{Value} = \mathbf{Z}). \tag{5}$$

An intuitive implementation for selective token aggregation can employ hard attention to select only one latent token for each coordinate. However, in our primitive experiment, using hard attention leads to unstable training and a latent collapse problem that selects only few latent tokens. Meanwhile, multi-head attentions encourage each latent token to easily learn the locality in data instances.

3.3.2 Multi-Band Feature Modulation in the Spectral Domain

After the selective token aggregation extracts a modulation vector $\mathbf{m}_{\mathbf{v}}$, we use multi-band feature modulation to effectively predict the details of outputs. Although Fourier features [24, 36] reduce the spectral bias [2, 28] of neural networks, adopting a simple stack of MLPs to INRs still suffers from capturing the high-frequency data details. To address this issue, we use a different range of frequency bandwidths to decompose the modulation vector into multiple frequency features in the spectral domain. Then, our multi-band feature modulation uses the multiple frequency features to progressively decode the intermediate features, while encouraging a deeper MLP path to learn higher frequency features. Note that the coarse-to-fine approach in the spectral domain is analogous to the locally hierarchical approach in the spatial domain [21, 29, 39] to capture the data details.

Extracting multiple modulation features with different frequency bandwidths We extract L level of modulation features $\mathbf{m}_{\mathbf{v}}^{(1)}, \cdots, \mathbf{m}_{\mathbf{v}}^{(L)}$ from $\mathbf{m}_{\mathbf{v}}$ using different bandwidths of frequency features. Given L frequency bandwidths as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_L \geq \sigma_{\text{q}}$, we use Eq (3) to extract the ℓ-th level of frequency features of an input coordinate $\mathbf{v}$ as

$$(\mathbf{h}_{\text{F}})_{\mathbf{v}}^{(\ell)} := \mathbf{h}_{\text{F}}(\mathbf{v}; \sigma_\ell, \mathbf{W}_{\text{F}}^{(\ell)}, \mathbf{b}_{\text{F}}^{(\ell)}) = \text{ReLU}\left(\mathbf{W}_{\text{F}}^{(\ell)} \boldsymbol{\gamma}_{\sigma_\ell}(\mathbf{v}) + \mathbf{b}_{\text{F}}^{(\ell)}\right), \tag{6}$$

where $\mathbf{W}_{\text{F}}^{(\ell)}$ and $\mathbf{b}_{\text{F}}^{(\ell)}$ are trainable parameters and shared across data instances. Then, the ℓ-th modulation vector $\mathbf{m}_{\mathbf{v}}^{(\ell)}$ is extracted from the modulation vector $\mathbf{m}_{\mathbf{v}}$ as

$$\mathbf{m}_{\mathbf{v}}^{(\ell)} := \text{ReLU}\left((\mathbf{h}_{\text{F}})_{\mathbf{v}}^{(\ell)} + \mathbf{W}_{\text{m}}^{(\ell)} \mathbf{m}_{\mathbf{v}} + \mathbf{b}_{\text{m}}^{(\ell)}\right), \tag{7}$$

with a trainable weight $\mathbf{W}_{\text{m}}^{(\ell)}$ and bias $\mathbf{b}_{\text{m}}^{(\ell)}$. Considering that ReLU cutoffs the values below zero, we assume that $\mathbf{m}_{\mathbf{v}}^{(\ell)}$ filters out the information of $\mathbf{m}_{\mathbf{v}}$ based on the ℓ-th frequency patterns of $(\mathbf{h}_{\text{F}})_{\mathbf{v}}^{(\ell)}$.

Multi-band feature modulation After decomposing a modulation vector into multiple features with different frequency bandwidths, we progressively compose the L modulation features by applying a stack of nonlinear operations with a fully-connected layer and ReLU activation. Starting with $\mathbf{h}_{\mathbf{v}}^{(1)} = \mathbf{m}_{\mathbf{v}}^{(1)}$, we compute the ℓ-th hidden features $\mathbf{h}_{\mathbf{v}}^{(\ell)}$ for $\ell = 2, \cdots, L$ as

$$\widetilde{\mathbf{h}}_{\mathbf{v}}^{(\ell)} := \mathbf{m}_{\mathbf{v}}^{(\ell)} + \mathbf{h}_{\mathbf{v}}^{(\ell-1)} \quad \text{and} \quad \mathbf{h}_{\mathbf{v}}^{(\ell)} := \text{ReLU}(\mathbf{W}^{(\ell)} \widetilde{\mathbf{h}}_{\mathbf{v}}^{(\ell)} + \mathbf{b}^{(\ell)}), \tag{8}$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d \times d}$ and $\mathbf{b}^{(\ell)} \in \mathbb{R}^d$ are trainable weights and biases of the INR decoder. $\widetilde{\mathbf{h}}_{\mathbf{v}}^{(\ell)}$ denotes the ℓ-th pre-activation of INR decoder for coordinate $\mathbf{v}$. Note that the modulation features with high-frequency bandwidth can be processed by more nonlinear operations than the features with lower frequency bandwidths, considering that high-frequency features contain more complex signals.

Finally, the output $\hat{\mathbf{y}}$ is predicted using all intermediate hidden features of the INR decoder as

$$\hat{\mathbf{y}} := \sum_{\ell=1}^{L} f_{\text{out}}^{(\ell)}(\mathbf{h}_{\mathbf{v}}^{(\ell)}), \tag{9}$$

where $f_{\text{out}}^{(\ell)} : \mathbb{R}^d \to \mathbb{R}^{d_{\text{out}}}$ are a linear projection into the output space. Although utilizing only $\mathbf{h}_{\mathbf{v}}^{(L)}$ is also an option to predict outputs, skip connections of all intermediate features into the output layer enhances the robustness of training to the hyperparameter choices.

4 Experiments

We conduct extensive experiments to demonstrate the effectiveness of our locality-aware generalizable INR on image reconstruction and novel view synthesis. In addition, we conduct in-depth analysis to validate the efficacy of our selective token aggregation and multi-band feature modulation to

Figure 3: Reconstructed images of FFHQ with 512×512 resolution by TransINR [8] (left), IPC [19] (middle), and our locality-aware generalizable INR (right).

localize the information of data to capture fined-grained details. We also show that our locality-aware latents can be utilized for image generation by training a generative model on the extracted latents. Our implementation and experimental settings are based on the official codes of Instance Pattern Composers [19] for a fair comparison. We attach the implementation details to Appendix.

4.1 Image Reconstruction

We follow the protocols in previous studies [8, 19, 35] to evaluate our framework on image reconstruction of CelebA, FFHQ, and ImageNette with 178×178 resolution. Our framework also outperforms previous approaches on high-resolution images with 256×256, 512×512, and 1024×1024 resolutions of FFHQ. We compare our framework with Learned Init [35], TransINR [8], and IPC [19]. The Transformer encoder predicts $R = 256$ latent tokens, while the INR decoder uses $d_{\text{in}} = 2$, $d_{\text{out}} = 3$, $d = 256$ dimensionality of hidden features, $\sigma_{\text{q}} = 16$ and $(\sigma_1, \sigma_2) = (128, 32)$ bandwidths.

178×178 Image Reconstruction Table 1 shows that our generalizable INR significantly outperforms previous methods by a large margin. We remark that TransINR, IPC, and our framework use the same capacity of the Transformer encoder, latent tokens, and INR decoder except for the modulation methods. Thus, the results imply that our locality-aware INR decoder with selective token aggregation and multi-band feature modulation is effective to capture local information of data and fine-grained details for high-quality image reconstruction.

Table 1: PSNRs of reconstructed images of 178×178 CelebA, FFHQ, and ImageNette.

	CelebA	FFHQ	ImageNette
Learned Init [35]	30.37	-	27.07
TransINR	33.33	33.66	29.77
IPC	35.93	37.18	38.46
Ours	**50.74**	**43.32**	**46.10**

High-Resolution Image Reconstruction We further evaluate our framework on the reconstruction of FFHQ images with 256×256, 512×512, 1024×1024 resolutions to demonstrate our effectiveness to capture fine-grained data details in Table 2. Although the performance increases as the MLP dimensionality d and the number of latents R increases, we use the same experimental setting with 178×178 image reconstruction to validate the efficacy of our framework. Our framework consistently achieves higher PSNRs than TransINR and IPC for all resolutions. Figure 3 also shows that TransINR and IPC cannot reconstruct the fine-grained details of a 512×512 image, but our framework provides a high-quality result of reconstructed images. The results demonstrate that leveraging the locality of data is crucial for generalizable INR to model complex and high-resolution data.

Table 2: PSNRs on the reconstructed FFHQ with 256×256, 512×512, and 1024×1024 resolutions.

	256×256	512×512	1024×1024
TransINR	30.96	29.35	-
IPC [19]	34.68	31.58	28.68
Ours	**39.88**	**35.43**	**31.94**

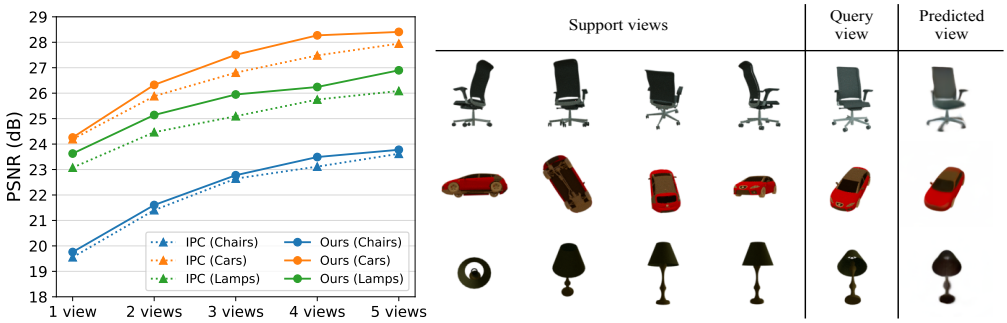

Figure 4: (a) PSNRs on novel view synthesis of ShapeNet Chairs, Cars, and Lamps according to the number of support views (1-5 views). (b) Examples of novel view synthesis with 4 support views.

4.2 Few-Shot Novel View Synthesis

We evaluate our framework on novel view synthesis with the ShapeNet Chairs, Cars, and Lamps datasets to predict a rendered image of a 3D object under an unseen view. Given few views of an object with known camera poses, we employ a light field [32] for novel view synthesis. A light field does not use computationally intensive volume rendering [24] but directly predicts RGB colors for the input coordinate for rays with $d_{\text{in}} = 6$ using the Plücker coordinate system. Our INR decoder uses $d = 256$ and two levels of feature modulations with $\sigma_{\text{q}} = 2$ and $(\sigma_1, \sigma_2) = (8, 4)$.

Figure 4(a) shows that our framework outperforms IPC for novel view synthesis. Our framework shows competitive performance with IPC when only one support view is provided. However, the performance of our framework is consistently improved as the number of support views increases, while outperforming the results of IPC. Note that defining a local relationship between rays is not straightforward due to its non-grid property of the Plücker coordinate. Our Transformer encoder can learn the local relationship between rays to extract locality-aware latent tokens during training and achieve high performance. We analyze the learned locality of rays encoded in the extracted latents in Section 4.3. Figure 4(b) shows that our framework correctly predicts the colors and shapes of a novel view corresponding to the support views, although the predicted views are blurry due to the lack of training objectives with generative modeling. We expect that combining our framework with generative models [5, 38] to synthesize a photorealistic novel view is an interesting future work.

4.3 In-Depth Analysis

Learning Curves on ImageNette 178×178 Figure 1 juxtaposes the learning curves of our framework and previous approaches on ImageNette 178×178. Note that TransINR, IPC, and our framework use the same Transformer encoder to extract data latents, while adopting different modulation methods. While the training speed of our framework is about 80% of the speed of IPC, we remark our framework achieves the test PSNR of 38.72 after 400 epochs of training, outperforming the PSNR of 38.46 achieved by IPC trained for 4000 epochs, hence resulting in $8\times$ speed-up of training time. That is, our locality-aware latents enables generalizable INR to be both efficient and effective.

Selective token aggregation and multi-band feature modulations We conduct an ablation study on ImageNette 178×178 and FFHQ 256×256 to validate the effectiveness of the selective token aggregation and the multi-band feature modulation. We replace the multi-band feature modulations with a simple stack of MLPs (ours w/o multiFM), and the selective token aggregation with the weight modulation of IPC (ours w/o STA). If both two modules are replaced together, the INR decoder becomes the same architecture as IPC. We use single-head cross-attention for the selective token aggregation to focus on the effect of two modules. Table 3 demonstrates that both the selective token aggregation and the multi-band feature modulation are required for the performance improvement, as there is no significant improvement when only one of the modules is used.

Table 3: Ablation study on ImageNette 178×178 and FFHQ 256×256.

	ImageNette	FFHQ
Ours	**37.46**	**38.01**
w/o STA	34.54	34.52
w/o multiFM	33.90	33.65
IPC [19]	34.11	34.68

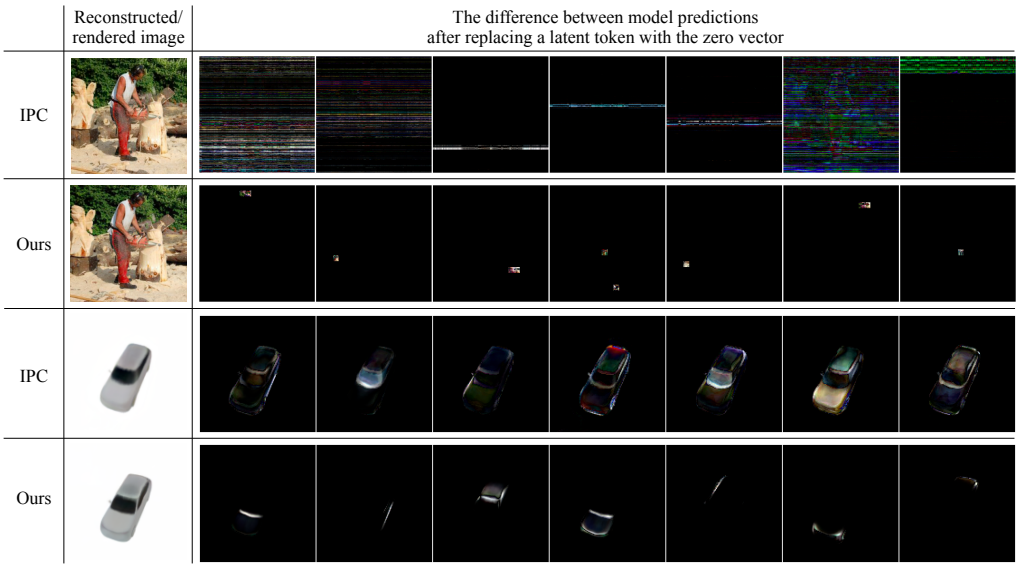

Figure 5: Visualization of differences between model predictions after replacing a latent token with the zero vector, for IPC [19] and our framework.

Choices of frequency bandwidths Table 5 shows that the ordering of frequency bandwidths in Eq. (4) and Eq. (6) can affect the performance. We train our framework with two-level feature modulations on ImageNette 178×178 during 400 epochs with different settings of the bandwidths $\sigma_1, \sigma_2, \sigma_{\mathrm{q}}$. Although our framework outperforms IPC regardless of the bandwidth settings, the best PSNR is achieved with $\sigma_1 \geq \sigma_2 \geq \sigma_{\mathrm{q}}$. The results imply that selective token aggregation does not require high-frequency features, but the high-frequency features need to be processed by more nonlinear operations than lower-frequency features as discussed in Section 3.3.2.

Table 4: PSNRs of reconstructed ImageNette 178×178 with various frequency bandwidths.

(σ_1, σ_2)	σ_{q}	ImageNette
(128, 32)	16	**37.46**
(32, 128)	16	35.00
(128, 128)	16	35.30
(128, 32)	128	35.58
IPC ($\sigma = 128$)		34.11

The role of extracted latent tokens Figure 5 shows that our framework encodes the local information of data into each latent token, while IPC cannot learn the locality in data coordinates. To visualize the information in each latent token, we randomly select a latent token to be replaced with the zero vector. Then, we visualize the difference between the model predictions with or without the replacement. Each latent token of our framework encapsulates the local information in different regions of images and light fields. However, the latent tokens of IPC cannot exploit the local information of data, while encoding the global information over whole coordinates. Note that our framework *learns* the structure of locality in light fields during training, although the structure of the Plücker coordinate system is not regular as the grid coordinates of images. Thus, our framework can learn the locality-aware latents of data for generalizable INR regardless of the types of coordinate systems.

4.4 Generating INRs for Conditional Image Synthesis

We examine the potentials of the extracted latent tokens to be utilized for a downstream task such as class-conditional image generation of ImageNet [9]. Note that we cannot use the architecture of U-Net in conventional image diffusion models [4, 30], since our framework is not tailored to the 2D grid coordinate. Thus, we adopt a Transformer-based diffusion model [27, 15] to predict a set of latent tokens after corruping the latents by Gaussian noises. We train 458M parameters of Transformers during 400 epochs to generate our locality-aware latent tokens. When we train a diffusion model to generate latent tokens

Table 5: Reconstructed PSNRs and FID of generated images on ImageNet 256×256.

	Latent Shape	rPSNR	FID
Ours	256×256	37.7	9.3
Spatial	16×16×256	37.2	11.7
Functa [4]	32×32×64	37.7	8.8
LDM [30]	64×64×3	27.4	3.6

Figure 6: The examples of generated 256×256 images by generating latents of IPC (left) and ours (right), trained on ImageNet.

of IPC in Figure 6, the generated images suffer from severe artifacts, because the prediction error of each latent token for IPC leads to the artifacts over all coordinates. Contrastively, the diffusion model for our locality-aware latents generates realistic images. In addition, although we do not conduct exhaustive hyperparamter search, the FID score of generated images achieves 9.3 with classifier-free guidance scale [16]. Thus, the results validate the potential applications of the local latents for INRs. Meanwhile, a few generated images may exhibit checkerboard artifacts, particularly in simple backgrounds. We leave the elaboration of diffusion models for INR latents as future work.

4.5 Comparison with Overfitted INRs

Figure 7 shows that our generalizable INR efficiently provides meaningful INRs compared with individual training of INRs per sample. To evaluate the efficiency of our framework, we select ten images of FFHQ 256×256 and train randomly initialized FFNet [36] per sample using one NVIDIA V100 GPU. The individual training of FFNets requires over 10 seconds of optimization to achieve the same PSNRs of our framework, where our inference time is negligible. Moreover, when we apply the test-time optimization (TTO) only for the extracted latents, it consistently outperforms per-sample FFNets for 30 seconds while maintaining the structure of latents. When we consider the predicted INR as initialization and finetune all parameters of the INR decoder per each sample, our framework consistently outperforms the per-sampling training of INRs from random initialization. Thus, the results imply that leveraging generalizable INR is computationally efficient to model unseen data as INRs regardless of a TTO.

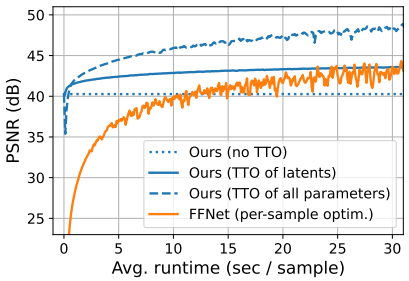

Figure 7: Comparison with individually trained FFNets [36] per sample.

5 Conclusion

We have proposed an effective framework for generalizable INR with the Transformer encoder and locality-aware INR decoder. The Transformer encoder capture the locality of data entities and learn to encode the local information into different latent tokens. Our INR decoder selectively aggregates the locality-aware latent tokens to extract a modulation vector for a coordinate input and exploits the multiple bandwidths of frequency features to effectively predict the fine-grained data details. Experimental results demonstrate that our framework significantly outperforms previous generalizable INRs on image reconstruction and few-shot novel view synthesis. In addition, we have conducted the in-depth analysis to validate the effectiveness of our framework and shown that our locality-aware latent tokens for INRs can be utilized for downstream tasks such as image generation to provide realistic images. Considering that our framework can learn the locality in non-grid coordinates, such as the Plücker coordinate for rays, leveraging our generalizable INR to generate 3D objects or scenes is a worth exploration. Furthermore, we expect that elaborating on the architecture and techniques for diffusion models to effectively generate INRs is an interesting future work.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[2] Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *International Conference on Machine Learning*, pages 685–694. PMLR, 2020.

[3] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[4] Matthias Bauer, Emilien Dupont, Andy Brock, Dan Rosenbaum, Jonathan Schwarz, and Hyunjik Kim. Spatial functa: Scaling functa to imagenet classification and generation. *arXiv preprint arXiv:2302.03130*, 2023.

[5] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In *arXiv*, 2023.

[6] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021.

[7] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021.

[8] Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pages 170–187. Springer, 2022.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Emilien Dupont, Adam Golinski, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: COmpression with implicit neural representations. In *Neural Compression: From Information Theory to Applications – Workshop @ ICLR 2021*, 2021.

[11] Emilien Dupont, Hyunjik Kim, SM Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *International Conference on Machine Learning*, pages 5694–5725. PMLR, 2022.

[12] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. Coin++: Data agnostic neural compression. *arXiv preprint arXiv:2201.12904*, 2022.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[14] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[17] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.

[18] Animesh Karnewar, Tobias Ritschel, Oliver Wang, and Niloy Mitra. Relu fields: The little non-linearity that could. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.

[19] Chiheon Kim, Doyup Lee, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Generalizable implicit neural representations via instance pattern composers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[21] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[22] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223, 2021.

[23] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[24] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[25] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[26] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

[28] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[29] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, pages 14866–14876, 2019.

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[31] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[32] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34:19313–19325, 2021.

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[34] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.

[35] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021.

[36] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[38] Daniel Watson, William Chan, Ricardo Martin Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

[39] Tackgeun You, Saehoon Kim, Chiheon Kim, Doyup Lee, and Bohyung Han. Locally hierarchical auto-regressive modeling for image generation. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

[40] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR, 2019.

A Implementation Details

We describe the implementation details of our locality-aware generalizable INR with the Transformer encoder and locality-aware INR decoder. We implement our framework based on the official open-sourced implementation of IPC[1] for a fair comparison. Our Transformer encoder comprises six blocks of self-attentions with 12 attention heads, where each head uses 64 dimensions of hidden features, and $R = 256$ latent tokens for all experiments. We use the Adam [20] optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$ and constant learning rate of 0.0001. The batch size is 16 and 32 for image reconstruction and novel view synthesis, respectively.

A.1 Image Reconstruction

178×178 image reconstruction For the image reconstruction of CelebA, FFHQ, and ImageNette with 178×178 resolution, we use $L = 2$ level of modulation features for multi-band feature modulation of locality-aware INR decoder. The dimensionality of frequency features and hidden layers in the INR decoder is 256, where $(\sigma_1, \sigma_2, \sigma_{\text{q}}) = (128, 32, 16)$. We represent a 178×178 resolution of the image as 400 tokens, where each token corresponds to a 9×9 size of the image patch with zero padding. We use a multi-head attention block with two attention heads for our selective token selection via cross-attention. Following the experimental setting of previous studies [8, 19], we train our framework on CelebA, FFHQ, and ImageNette during 300, 1000, and 4000 epochs, respectively. When we use four NVIDIA V100 GPUs, the training takes 5.5, 6.7, and 4.3 days, respectively.

ImageNet 256×256 We use $L = 2$ level of feature modulation for the image reconstruction of ImageNet with 256×256 resolution. We use eight heads of selective token aggregation, 256 dimensionality of frequency features and hidden layers of the INR decoder, and $(\sigma_1, \sigma_2, \sigma_{\text{q}}) = (128, 32, 16)$. An image is represented as 256 tokens, where each token corresponds to a 16×16 patch in the image. We use eight NVIDIA A100 GPUs to train our framework on ImageNet during 20 epochs, where the training takes about 2.5 days.

FFHQ 256×256, 512×512, and 1024×1024 Our framework for FFHQ 256×256 and 512×512 uses $L = 2$ level of feature modulation with $(\sigma_1, \sigma_2, \sigma_{\text{q}}) = (128, 32, 16)$. The size of each patch is 16 and 32 for 256×256 and 512×512 resolutions, respectively, the number of latent tokens is $R = 256$, and the dimensionality of the INR decoder is $d_{\text{F}} = d = 256$. Our selective token aggregation uses two and four heads of cross-attention for FFHQ 256×256 and 512×512, respectively. We randomly sample the 10% of coordinates to be decoded at each training step to increase the efficiency of training. We train our framework during 400 epochs, while the training takes about 1.5 days using four NVIDIA V100 GPUs for FFHQ with 256×256 and about 1.4 days using eight V100 GPUs for FFHQ with 512×512. For FFHQ 1024×1024, we use 48 patch size to represent an image as 484 data tokens and $L = 2$ level of feature modulation with $(\sigma_1, \sigma_2, \sigma_{\text{q}}) = (256, 64, 32)$. The training of 400 epochs takes about 3.4 days using eight NVIDIA V100 GPUs.

A.2 Novel View Synthesis

We train our framework for the task of novel view synthesis on ShapeNet Cars, Chairs, and Lamps. Given a few known camera views as support views of a 3D object, our framework predicts a light field of the 3D object to predict unseen camera views. For a fair comparison, we use the same splits of train-valid samples with previous studies of generalizable INR [8, 19, 35]. While each rendered view has the 128×128 resolution of an image, we patchify each rendered image into 256 tokens with 8×8 size of patches. We use the Plücker coordinate to represent a ray for a pixel as an embedding with six dimensions and concatenate the ray embedding into each pixel along the channel dimension. Since our INR decoder estimates a light field of a 3D object, the INR decoder has six input channels $d_{\text{in}} = 6$ for a ray coordinate and three output channels $d_{\text{out}} = 3$ for a RGB pixel. Our INR decoder uses $L = 2$ level of feature modulation with $(\sigma_1, \sigma_2, \sigma_{\text{q}}) = (8, 4, 2)$. We use $d_{\text{F}} = d = 256$ dimensionality of the frequency features and hidden features of the INR decoder. We use 1000 training epochs for ShapeNet Cars and Chairs, while using 500 epochs for ShapeNet Lamps.

[1] https://github.com/kakaobrain/ginr-ipc

A.3 Diffusion Model for INR generation

We implement a diffusion model to generate the latent tokens for INRs of ImageNet 256×256. Different from the conventional approaches, which use a U-Net architecture to generate an image, we use a vanilla Transformer with a simple stack of self-attentions, since the latent tokens do not predefine 2D grid structure but are permutation-equivariant. The Transformer for the diffusion model has 458M parameters having 24 self-attention blocks with 1024 dimensions of embeddings and 16 heads. We remark that the locality-aware generalizable INR is not updated during the training of diffusion models. For the training of the diffusion model, we follow the formulation of DDPM [15]. The linear noise schedule with $T = 1000$ is used to randomly corrupt the latent tokens for INRs using isotropic Gaussian noises, and then we train our Transformer to denoise the latent tokens. Instead of the ϵ-parameterization that predicts the noises used for the corruption, our Transformer $\mathbf{x}_0$-parameterization to predict the original latent tokens. We drop 10% of class conditions for our model to support classifier-free guidance [16]. For the stability of training, we standardize the features of latent tokens, after computing the mean and standard deviation of feature channels of each latent token based on the training data. We use eight NVIDIA A100 GPUs to train the model with 256 batch size during 400 epochs, where the training takes about 7 days. The Adam [20] optimizer with constant learning rate 0.0001 and $(\beta_1, \beta_2) = (0.9, 0.999)$ is used without learning rate warm-up and any weight decaying. During training, we further compute the exponential moving average (EMA) of model parameters with a decaying rate of 0.9999. During the evaluation, we use the EMA model with 250 DDIM steps [33] and 2.5 scales of classifier-free guidance [16].

B Additional Experiments

B.1 Ablation Study on the Number of Levels

Table 6 demonstrates the effect of the number of levels L on image reconstruction benchmarks of FFHQ images with 256×256, 512×512, and 1024×1024 resolutions. Our INR decoder uses bandwidths $\sigma_{\text{q}} = 16$ and $(\sigma_\ell)_{\ell=1}^{L}$ equal to (128), $(128, 32)$, $(128, 64, 32)$ and $(128, 90, 64, 32)$ for $L = 1, 2, 3, 4$ respectively in case of 256×256 and 512×512 resolution, and all bandwidths are doubled for 1024×1024 to leverage high-frequency details.

Table 6: PSNRs on the reconstructed FFHQ with 256×256, 512×512, and 1024×1024 resolutions for different number of levels.

	256×256	512×512	1024×1024
TransINR	30.96	29.35	-
IPC [19]	34.68	31.58	28.68
Ours ($L = 1$)	37.09	34.84	31.56
Ours ($L = 2$)	39.88	35.43	31.94
Ours ($L = 3$)	**40.13**	**35.58**	**32.40**
Ours ($L = 4$)	39.79	35.40	32.32

Note that our framework outperforms previous studies [8, 19] even with $L = 1$. Moreover, the results demonstrate that increasing L improves the performance, while the performance saturates beyond $L \geq 3$. We postulate that higher resolution requires a larger number of levels, as the performance gap between $L = 3$ and $L = 4$ decreases as the resolution increases.

B.2 Additional Examples of Novel View Synthesis

In Figure 8, we show additional examples of novel view synthesis of ShapeNet Chairs, Cars, and Lamps with one to five support views.

B.3 Additional Examples of High-resolution Image Reconstruction

Figure 9 and 10 shows image reconstruction examples of FFHQ with 256×256, 512×512, and 1024×1024 resolution by previous studies [8, 19] and our locality-aware generalizable INR. Unlike previous studies, our framework can successfully reconstruct fine-grained details in high resolutions.

B.4 Additional Examples of Conditional Image Synthesis

Figure 11 shows additional examples of generated images with 256×256 resolution by generating locality-aware latents of our framework.

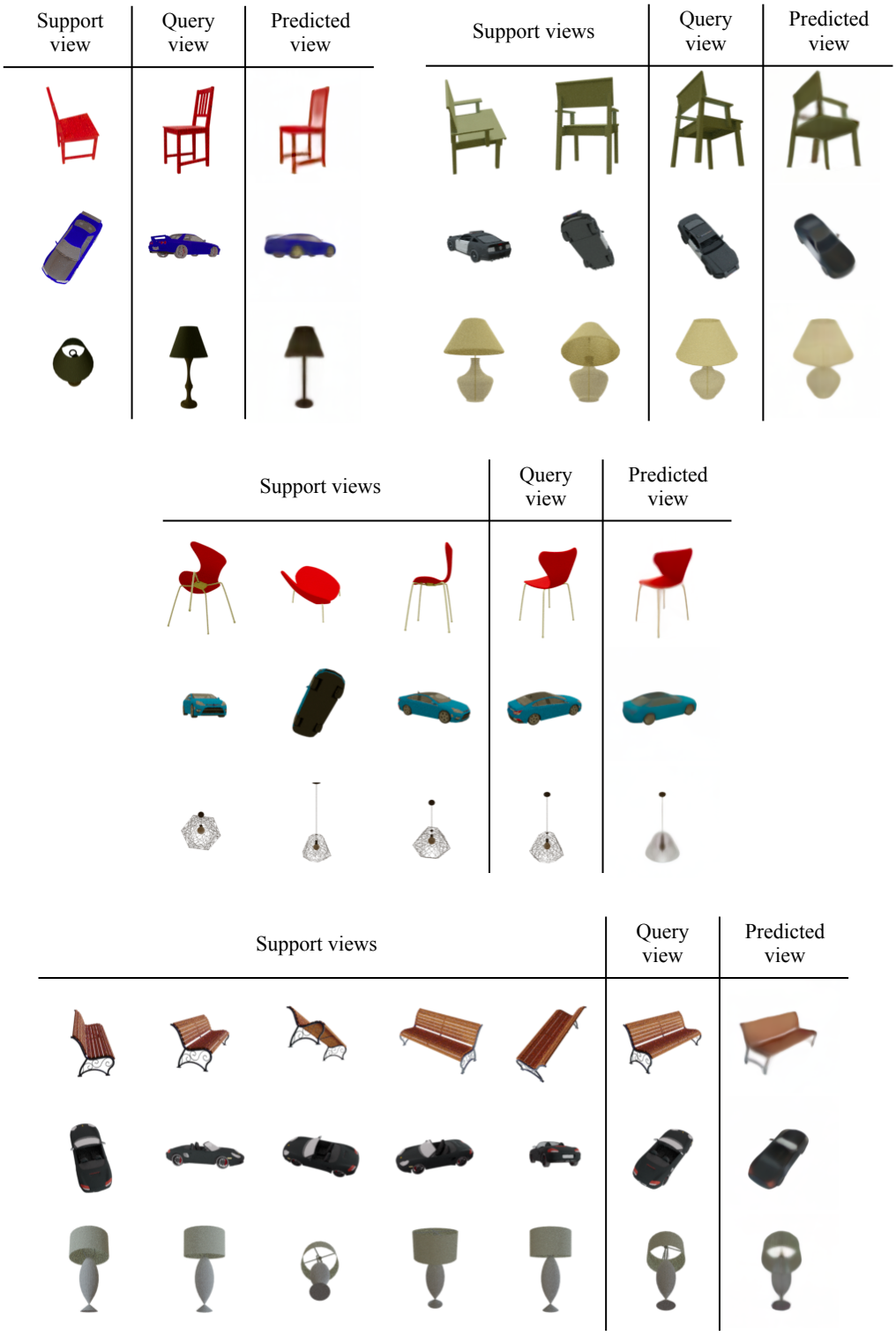

Figure 8: Examples of novel view synthesis of ShapeNet Chairs, Cars and Lamps with one, two, three, and five support views.

Figure 9: Examples of reconstructed images of FFHQ with 256×256 resolution (top row) and 512×512 resolution (bottom row) by TransINR [8] (left), IPC [19] (middle), and our locality-aware generalizable INR (right).

Figure 10: Examples of reconstructed images of FFHQ with 1024×1024 resolution by IPC (left) and our locality-aware generalizable INR (right).

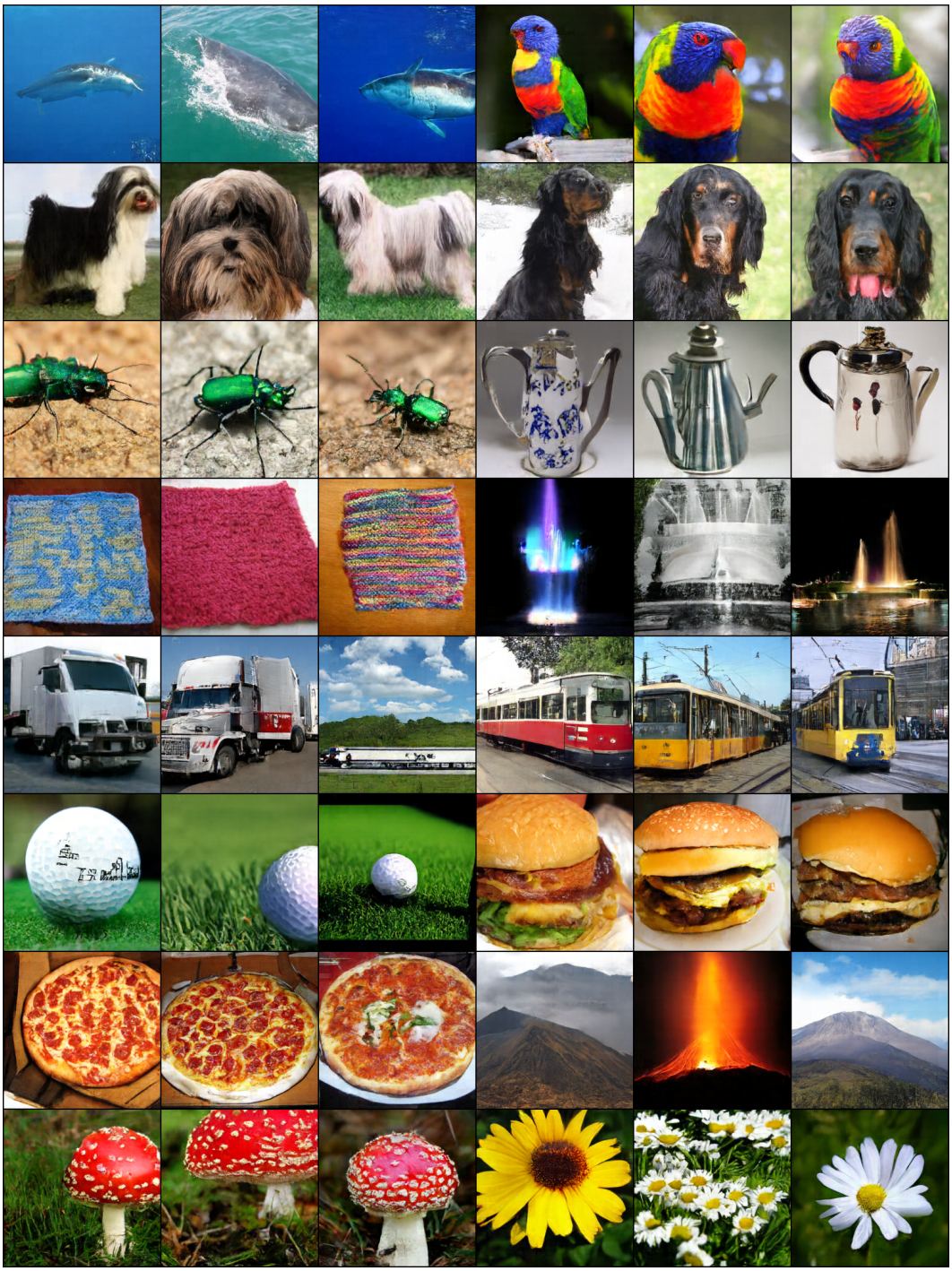

Figure 11: Additional examples of class-conditional image synthesis by generating the locality-aware latents of our framework via a transformer-based diffusion model with 458M parameters. All images are generated with classifier-free guidance at scale 2.5.

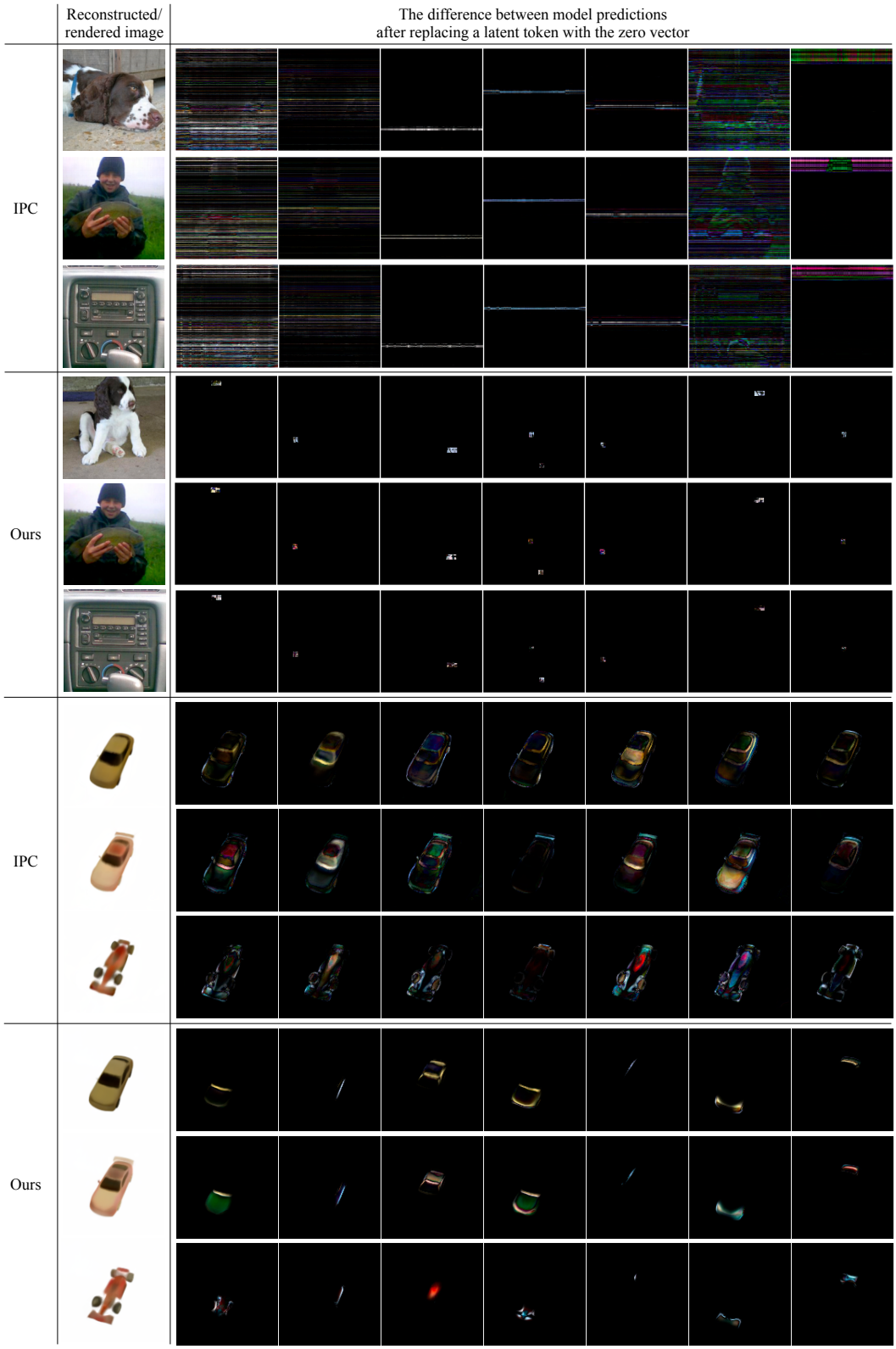

Figure 12: Additional visualization of differences between model predictions after replacing a latent token with the zero vector for IPC [19] and our framework.

B.5 Additional Visualization for Locality Analysis

Figure 12 visualizes which local information of data is encoded in each latent token of IPC [19] and our locality-aware generalizable INR in addition to Figure 5. We randomly select a latent token and replace it with the zero vector, then visualize the difference between the model predictions with or without the replacement as described in Section 4.3. The differences are rescaled to have the maximum value of 1 for clear visualization. Furthermore, we fix the set of replaced latent tokens for different samples in Figure 12 to emphasize the role of each latent token. Note that each latent token of our framework encodes the local information in a particular region of images or light fields, while latent tokens of IPC encode global information over whole coordinates.