Generalizable Implicit Neural Representations via Instance Pattern Composers

Chiheon Kim*
Kakao Brain
chiheon.kim@kakaobrain.com

Doyup Lee*
Kakao Brain
doyup.lee@kakaobrain.com

Saehoon Kim
Kakao Brain
shkim@kakaobrain.com

Minsu Cho
POSTECH
mscho@postech.ac.kr

Wook-Shin Han†
POSTECH
wshan@postech.ac.kr

Abstract

Despite recent advances in implicit neural representations (INRs), it remains challenging for a coordinate-based multi-layer perceptron (MLP) of INRs to learn a common representation across data instances and generalize it for unseen instances. In this work, we introduce a simple yet effective framework for generalizable INRs that enables a coordinate-based MLP to represent complex data instances by modulating only a small set of weights in an early MLP layer as an instance pattern composer; the remaining MLP weights learn pattern composition rules for common representations across instances. Our generalizable INR framework is fully compatible with existing meta-learning and hypernetworks in learning to predict the modulated weight for unseen instances. Extensive experiments demonstrate that our method achieves high performance on a wide range of domains such as an audio, image, and 3D object, while the ablation study validates our weight modulation.

1. Introduction

Implicit neural representations (INR) have shown the potential to represent complex data as continuous functions. Assuming that a data instance comprises the pairs of a coordinate and its output features, INRs adopt a parameterized neural network as a mapping function from an input coordinate into its output features. For example, a coordinate-based MLP [23] predicts RGB values at each 2D coordinate as an INR of an image. Despite the popularity of INRs, a trained MLP cannot be generalized to represent other instances, since each MLP learns to memorize each data instance. Thus, INRs necessitate separate training of MLPs to represent a lot of data instances as continuous functions.

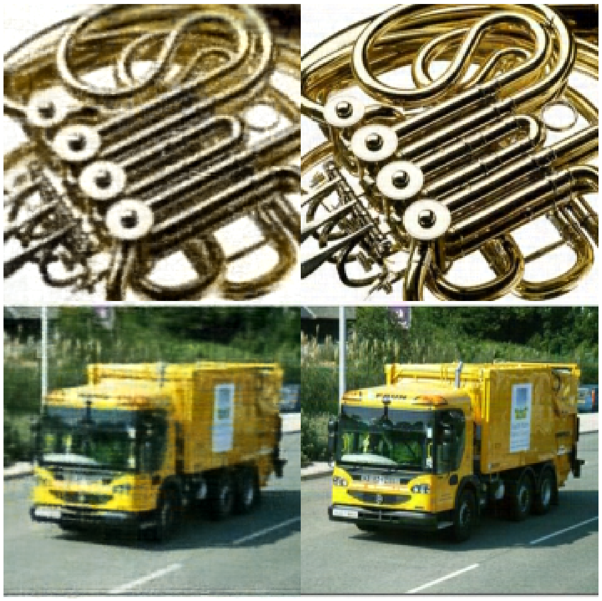

Figure 1. The reconstructed images of 178×178 ImageNette by TransINR [4] (left) and our generalizable INRs (right).

Generalizable INRs aim to learn common representations of a MLP across instances, while modulating features or weights of the coordinate-based MLP to adapt unseen data instances [4, 7, 24]. The feature-modulation method exploits the latent vector of an instance to condition the activations in MLP layers through concatenation [17] or affine-transform [8, 18]. Despite the computational efficiency of feature-modulation, the modulated INRs have unsatisfactory results to represent complex data due to their limited modulation capacity. On the other hand, the weight-modulation method learns to update the whole MLP weights to increase the modulation capacity for high performance. However, modulating whole MLP weights leads to unstable and expensive training [4, 7, 9, 24].

In this study, we propose a simple yet effective frame-

[*] Equal contribution
[†] Corresponding author

work for generalizable INRs via *Instance Pattern Composers* to modulate only a small set of MLP weights. We postulate that a complex data instance can be represented by composing low-level patterns in the instance [23]. Thus, we rethink and categorize the weights of MLP into i) *instance pattern composers* and ii) *pattern composition rule*. The instance pattern composer is a weight matrix in the early layer of our coordinate-based MLP to extract the instance content patterns of each data instance as a low-level feature. The remaining weights of MLP is defined as a pattern composition rule, which composes the instance content patterns in an instance-agnostic manner. In addition, our framework can adopt both optimization-based meta-learning and hypernetworks to predict the instance pattern composer for an INR of unseen instance. In experiments, we demonstrate the effectiveness of our generalizable INRs via instance pattern composers on various domains and tasks.

Our main contributions are summarized as follows. 1) *Instance pattern composers* enable a coordinate-based MLP to represent complex data by modulating only one weight, while *pattern composition rule* learns the common representation across data instances. 2) Our instance pattern composers are compatible with optimization-based meta-learning and hypernetwork to predict modulated wights of unseen data during training. 3) We conduct extensive experiments to demonstrate the effectiveness of our framework through quantitative and qualitative analysis.

2. Related Work

Implicit neural representations (INRs). INRs train a parameterized neural network to represent complex and continuous data such as audios, images, and 3D objects and scenes. Seminal works incorporate Fourier features [15, 25, 27] and sinusoidal activations [21] in the coordinate-based MLP to avoid the spectral bias [2, 19]. Consequently, recent advances of INRs have shown broad impacts on various applications such as data reconstruction and compression [3, 6, 21, 25], and 3D representations [1, 12, 15, 21, 22, 25]. A coordinate-based MLP can learn to represent each data instance with high-resolution and complex patterns, but the learned MLP cannot be generalized to represent other data instances and requires re-training from the scratch.

Generalizable INRs. Generalizable INRs learn to modulate or adapt the coordinate-based MLP to unseen data instances. Given a latent vector of each data instance, autodecoding [14, 17] concatenates the latent vector into the features of MLP as the input condition, while sharing whole MLP weights across data instances. Inspired by the success of feature modulations [11, 18], a hypernetwork [10] is trained to predict the modulation vectors for each data instance to scale and shift the activations in all layers of the shared MLP [7, 8, 13]. Both approaches are simple and computationally efficient, since they do not need to change the whole weights of MLP. However, the scope and capacity of feature modulations are limited and insufficient to adapt the shared MLP for a multitude of data instances.

Existing studies adopt optimization-based meta-learning to training generalizable INRs. The bilevel optimizations such as MAML [9] and CAVIA [28] train the weight initialization of coordinate-based MLP, where the inner optimization achieves rapid adaptation of MLP to unseen data instances in a few gradient steps [20, 24]. Despite high performance using direct weight updates, the training is unstable and memory intensive due to the computation of high-order gradients [7] and requires an exhaustive search of hyperparameters. Interpreting the inner optimization of MAML as the inference of transformers [5, 26], TransINR [4] uses a hypernetwork comprised of a transformer to predict the column vectors in the weight matrix at every MLP layer.

3. Methods

In this section, we propose an effective framework for generalizable INRs via instance pattern composer. We first present the formulation of INR for a data instance and generalize it for multiple instances in a dataset. Then, we propose the *instance pattern composer* for our generalizable INRs to modulate a small set of weights in the second MLP layer, and *pattern composition rule* to generate complex data based on the extracted patterns. Finally, we explain how our framework can be combined with optimization-based meta-learning and transformer-based hypernetwork to preidct the instance pattern composer of data instances.

3.1. Preliminary

An INR represents a data instance as a continuous function by learning a parameterized neural network, *e.g.*, a coordinate-based MLP, which maps a coordinate into its corresponding features. Given a dataset $\mathcal{X} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$ with N instances, we assume that each instance $\mathbf{x}^{(n)}$ is expressed by M_n pairs of an input coordinate $\mathbf{v}_i^{(n)}$ and its corresponding output feature $\mathbf{y}_i^{(n)}$:

$$\mathbf{x}^{(n)} = \{(\mathbf{v}_i^{(n)}, \mathbf{y}_i^{(n)})\}_{i=1}^{M_n}, \tag{1}$$

where $\mathbf{v}_i^{(n)} \in \mathbb{R}^{d_{\text{in}}}$ and $\mathbf{y}_i^{(n)} \in \mathbb{R}^{d_{\text{out}}}$. For example, an $H \times W$ image $\mathbf{x}^{(n)}$ takes as input a 2D coordinate ($d_{\text{in}} = 2$) and produces as output its RGB values ($d_{\text{out}} = 3$), resulting in $M_n = HW$ for all pixels. Given trainable parameters $\phi^{(n)}$, a coordinate-based MLP $f_{\phi^{(n)}} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$ learns to implicitly represent an instance $\mathbf{x}^{(n)}$, while minimizing the mean-squared error over coordinates:

$$\mathcal{L}_n(\phi^{(n)}; \mathbf{x}^{(n)}) := \frac{1}{M_n} \sum_{i=1}^{M_n} \|\mathbf{y}_i^{(n)} - f_{\phi^{(n)}}(\mathbf{v}_i^{(n)})\|_2^2. \tag{2}$$

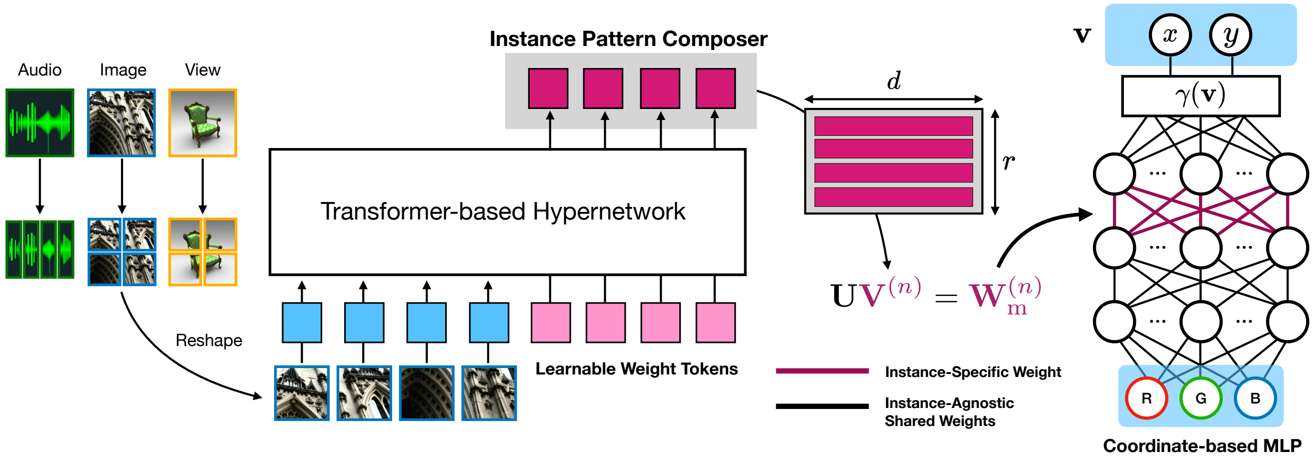
Audio
Image
View
Reshape
Instance Pattern Composer
Transformer-based Hypernetwork
Learnable Weight Tokens
d
r
$\mathbf{U}\mathbf{V}^{(n)} = \mathbf{W}_{\mathrm{m}}^{(n)}$
Instance-Specific Weight
Instance-Agnostic Shared Weights
$\mathbf{v}$
x
y
$\gamma(\mathbf{v})$
R
G
B
Coordinate-based MLP

Figure 2. Overview of our framework with Instance Pattern Composer for generalizable INRs. Instance pattern composer modulates the weight matrix in the second lowest MLP layer, while the remaining weights learns an instance-agnostic pattern composition rule.

Note that since a coordinate-based MLP model $f_{\phi^{(n)}}$ is trained to represent only *one* instance $\mathbf{x}^{(n)}$, the model cannot represent other instances in the dataset as well as unseen instances after training. The straightforward approach to obtain INRs for a dataset is to separately train an MLP per instance from scratch, but it is computationally infeasible for a large-scale dataset. Furthermore, this separate training cannot leverage common information across instances, limiting efficiency and generalizability.

3.2. Generalizable Implicit Neural Representations with Instance Pattern Composers

To overcome the limitation of conventional INRs, we propose the framework for generalizable INRs with *Instance Pattern Composers*. Our framework can efficiently modulate a small set of MLP weights, while learning common representations across instances. After we present the formulation of generalizable INRs, we explain the details of our framework with instance pattern composers.

3.2.1 Generalizable INRs.

We define the two types of parameters of a coordinate-based MLP for generalizable INRs: i) instance-specific parameter $\phi^{(n)}$ for an instance $\mathbf{x}^{(n)}$ and ii) instance-agnostic parameter θ. The instance-specific parameter $\phi^{(n)}$ characterizes a data instance $\mathbf{x}^{(n)}$, while the instance-agnostic parameter θ is shared across all instances to learn the underlying structural information in a dataset. Then, the coordinate-based MLP for generalizable INRs is trained to minimize the average of mean-square errors over training dataset $\mathcal{X}$:

$$\mathcal{L}(\theta, \{\phi^{(n)}\}_{n=1}^{N}; \mathcal{X}) := \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_n(\theta, \phi^{(n)}; \mathbf{x}^{(n)}). \quad (3)$$

Previous studies first train whole MLP weights as the instance-agnostic parameter θ, and then a modulation function g is used to convert the whole MLP weights to be instance-specific as $\phi^{(n)} = g(\theta, \mathbf{x}^{(n)})$, where g is executing the inner optimization steps in meta-learning [9, 24] or directly predicting weights in hypernetwork [4, 10]. However, we remark that modulating whole MLP weights is limited in terms of efficiency and effectiveness due to unstable training and expensive computational costs.

3.2.2 Generalizable INRs by Modulating One Weight Matrix as Instance Pattern Composer

We propose a simple yet powerful weight modulation of coordinate-based MLPs for generalizable INRs. Inspired by the first usage of coordinate-based MLP, which composes low-level patterns to synthesize complex visual patterns [23], we postulate that our coordinate-based MLP can represent complex data by composing simple patterns of instance-specific contents. Thus, we first categorize the weights of MLP into the following two types: i) *Instance Pattern Composer* as instance-specific parameter $\phi^{(n)}$, and ii) *Pattern Composition Rule* as instance-agnostic parameter θ. Especially, we assign one weight matrix in the early MLP layer for the instance pattern composer to be modulated, while the remaining weights are instance-agnostic pattern composition rule. For the brevity of notation, we omit the subscript i and denote $\mathbf{v}_i^{(n)}$ as $\mathbf{v}^{(n)}$ in this section.

Low-level frequency patterns. We convert a coordinate $\mathbf{v}^{(n)}$ of an instance $\mathbf{x}^{(n)}$ into its Fourier features $\gamma(\mathbf{v}^{(n)}) \in \mathbb{R}^{d_{\mathrm{f}}}$ with a dimensionality d_{f} [25]. Then, a fully-connected (FC) layer generates low-level frequency patterns $\mathbf{h}_{\mathrm{f}}$ as

$$\mathbf{h}_{\mathrm{f}} = \sigma(\mathbf{W}_{\mathrm{f}}\gamma(\mathbf{v}^{(n)}) + \mathbf{b}_{\mathrm{f}}), \quad (4)$$

where $\mathbf{W}_\mathrm{f} \in \mathbb{R}^{d \times d_\mathrm{f}}$ and $\mathbf{b}_\mathrm{f} \in \mathbb{R}^d$ are a learnable weight matrix and a bias vector, respectively, d is the dimensionality of hidden layers, and $\sigma(\cdot)$ is an element-wise nonlinearity function *e.g.* ReLU. Since $\mathbf{W}_\mathrm{f}$ and $\mathbf{b}_\mathrm{f}$ are instance-agnostic, data instances have the same frequency patterns $\mathbf{h}_\mathrm{f}$.

Instance Pattern Composer. An *instance pattern composer* characterizes the INR of a data instance $\mathbf{x}^{(n)}$ and extracts *instance content patterns* $\mathbf{h}^{(n)}$ based on frequency patterns $\mathbf{h}_\mathrm{f}$. Given a modulated weight matrix $\mathbf{W}_\mathrm{m}^{(n)} \in \mathbb{R}^{d \times d}$, we define an instance pattern composer $\mathbf{V}^{(n)} \in \mathbb{R}^{r \times d}$ of an instance $\mathbf{x}^{(n)}$ as a factorized matrix with rank r

$$\mathbf{W}_\mathrm{m}^{(n)} = \mathbf{U}\mathbf{V}^{(n)}, \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{d \times r}$ is an instance-agnostic weight. Then, an instance content pattern $\mathbf{h}^{(n)}$ of $\mathbf{x}^{(n)}$ is predicted as

$$\mathbf{h}^{(n)} = \sigma(\mathbf{W}_\mathrm{m}^{(n)}\mathbf{h}_\mathrm{f} + \mathbf{b}_\mathrm{m}), \tag{6}$$

where $\mathbf{b}_\mathrm{m} \in \mathbb{R}^d$ is an instance-agnostic bias vector. Note that the instance pattern composer $\mathbf{V}^{(n)}$ extracts the instance-specific representations of $\mathbf{x}^{(n)}$ to characterize an instance of modulated MLPs as a continuous representation of $\mathbf{x}^{(n)}$. Thus, our generalizable INRs only modulate the one weight matrix $\mathbf{V}^{(n)}$ to represent complex data, while sharing other MLP weights across data instances.

Pattern composition rule. Based on the instance content patterns $\mathbf{h}^{(n)}$ at coordinate $\mathbf{v}^{(n)}$, the subsequent FC layers are trained to predict the output features $\mathbf{y}^{(n)}$. We assume that the subsequent MLP layers learn to compose the instance content patterns $\mathbf{h}^{(n)}$ to represent complex output features $\mathbf{y}^{(n)}$, while learning the underlying structural information across data instances. Specifically, when the total number of MLP layers is L, the parameters of the remaining $L-2$ layers are shared across data instances to determine the *pattern composition rule* of MLP. Given $\mathbf{z}_2^{(n)} := \mathbf{h}^{(n)}$, the remaining hidden activations of MLP are computed as

$$\mathbf{z}_\ell^{(n)} = \sigma(\mathbf{W}_\ell \mathbf{z}_{\ell-1}^{(n)} + \mathbf{b}_\ell), \tag{7}$$

where $\mathbf{W}_\ell \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_\ell \in \mathbb{R}^d$ are weight and bias of l-th layer, and $l \in \{3, \cdots, L-1\}$. Finally, given the weight $\mathbf{W}_L \in \mathbb{R}^{d_\mathrm{out} \times d}$ and bias $\mathbf{b}_L \in \mathbb{R}^{d_\mathrm{out}}$, the output layer predicts the output features $\mathbf{y}^{(n)}$ as

$$f_{\theta,\phi^{(n)}}(\mathbf{v}^{(n)}) := \mathbf{W}_L \mathbf{z}_{L-1}^{(n)} + \mathbf{b}_L, \tag{8}$$

where $\phi^{(n)} = \mathbf{V}^{(n)}$ is the instance-specific parameter, and $\theta = \{\mathbf{W}_\mathrm{f}, \mathbf{b}_\mathrm{f}, \mathbf{U}, \mathbf{b}_\mathrm{CP}, \mathbf{W}_3, \mathbf{b}_3, \cdots, \mathbf{W}_\mathrm{L-1}, \mathbf{b}_\mathrm{L-1}\}$ is the instance-agnostic parameter. Since the pattern composition rule is the set of instance-agnostic parameters, our coordinate-based MLP shares all trainable parameters except for $\mathbf{V}^{(n)}$. That is, our generalizable INRs use the same rule to compose the content patterns $\mathbf{h}^{(n)}$ of different instances to represent complex data instances. In summary, our generalizable INRs learn the common pattern composition rule of extracted instance content patterns to generalize the learned representations for unseen data instances.

Algorithm 1 Optimization-based meta-learning for generalizable INRs via instance pattern composer.

```
Require: Randomly initialized $\theta$, $\phi$, a dataset $\mathcal{X}$, the number of inner steps $N_\text{inner}$, and learning rates $\epsilon, \epsilon'$.
while not done do
    for $n = 1, \cdots, N$ do
        Initialize instance-specific parameter $\phi^{(n)} \leftarrow \phi$
    end for
    /* inner-loop updates for $\theta^{(n)}$ */
    for all step $\in \{1, \cdots, N_\text{inner}\}$ and $\mathbf{x}^{(n)} \in \mathcal{X}$ do
        $\phi^{(n)} \leftarrow \phi^{(n)} - \epsilon \|\phi^{(n)}\|^2 \nabla_{\phi^{(n)}} \mathcal{L}_n(\theta, \phi^{(n)}; \mathbf{x}^{(n)})$
    end for
    /* outer-loop updates for $\theta$, $\phi$ */
    Update $\phi \leftarrow \phi - \epsilon' \nabla_\phi \mathcal{L}(\theta, \{\phi^{(n)}\}_{n=1}^N; \mathcal{X})$
    Update $\theta \leftarrow \theta - \epsilon' \nabla_\theta \mathcal{L}(\theta, \{\phi^{(n)}\}_{n=1}^N; \mathcal{X})$
end while
```

3.3. Predicting Modulation Weights

Thanks to the simple method of weight modulation, our framework for generalizable INRs is compatible with existing methods to predict the modulated weight $\phi^{(n)} = \mathbf{V}^{(n)}$ to characterize the INR of $\mathbf{x}^{(n)}$. This section shows that how optimization-based meta-learning [7–9, 24, 28] and hypernetworks [4, 10] can be combined with our framework.

Optimization-based meta-learning. An optimization-based meta-learning can learn the initialization of instance-specific parameter $\phi^{(n)} = \phi$ to be adapted to $\mathbf{x}^{(n)}$ in few optimization steps of Eq. (2). Since we do not require the adaptation of the whole weights in the test time, we modify CAVIA [28] for our generalizable INRs in Algorithm 1. Different from the original CAVIA, we train the initialization of $\phi^{(n)}$ as ϕ in the outer update to encourage the training of $\mathbf{U}$ in Eq. (5). We also scale the learning rate ϵ in the inner loop by the square of the norm of adapted parameter $\|\phi^{(n)}\|^2$ to improve the stability of the inner-loop updates.

Transformer-based hypernetwork. Our framework can adopt the transformer-based hypernetwork in Figure 2 to predict the r number of row vectors in Instance Pattern Composer $\mathbf{V}^{(n)}$ for each instance $\mathbf{x}^{(n)}$. Specifically, we first patchify a data instance $\mathbf{x}^{(n)}$, such as an audio, image, or multiple views, into non-overlapping patches and convert them into a sequence of data tokens in the raster-scan

Table 1. PSNRs of the reconstruction of the LibriSpeech test-clean dataset whose sample is trimmed into one and three seconds.

	LibriSpeech (1s)	LibriSpeech (3s)
TransINR	39.22	33.17
Ours	**40.11**	**35.38**

ordering. Then, we concatenate r learnable tokens into the sequence of data tokens, and use the concatenated token sequence as the input of the bidirectional transformer. Finally, r output tokens corresponding to learnable query tokens are linearly mapped into $\mathbb{R}^d$ to form an $r \times d$ matrix to predict the instance-specific factorized matrix $\mathbf{V}^{(n)}$ in Eq. (5). Since the transformer predicts the instance-specific weights $\theta^{(n)} = \mathbf{V}^{(n)}$, the parameters of the transformer are trained in an end-to-end manner by the optimization process of Eq. (3). Although the transformer-based hypernetwork has been already proposed, our framework does not require a heuristic method of weight grouping [4], but significantly improve the performance of the hypernetwork.

4. Experiments

We evaluate our framework on a wide range of domains such as audios, images, and 3D objects. We mainly use the transformer-based hypernetwork in Figure 2, since its training does not requires exhaustive hyperparameter search. Nonetheless, we also validate that our framework is also compatible with optimization-based meta-learning in Section 4.4. The implementation details are in the Appendix.

4.1. Audio Reconstruction

Our framework is trained on LibriSpeech-clean [16] for audio reconstruction. The MLP has five layers with $d = 256$, $d_{\text{in}} = 1$, and $d_{\text{out}} = 1$ for an audio. Our transformer-based hypernetwork predicts $r = 256$ weight tokens for $\mathbf{V}^{(n)}$, while TransINR predicts 257 weight tokens to modulate whole MLP weights via weight grouping [4]. We train our framework on randomly cropped audio during 1000 epochs, while test audio is trimmed for evaluation.

Table 1 shows the PSNRs of reconstructed audios. Although the reconstructed audios with three seconds have lower PSNRs than one second of audios, our framework consistently outperforms TransINR. Since the main difference from TransINR is the weight modulation method, the results validate the effectiveness of our instance pattern composers to modulate a small set of MLP weights.

4.2. Image Reconstruction

We evaluate our generalizable INRs on image reconstruction of 178×178, 256×256, 512×512 resolution of images usingfive layers of MLPs with $d_{\text{out}} = 3$ and $d_{\text{in}} = 2$.

Table 2. PSNRs of reconstructed images for178×178 resolution of images in the CelebA, FFHQ, and ImageNette test dataset.

	CelebA	FFHQ	ImageNette
Learned Init [24]	30.37	-	27.07
TransINR	33.33	33.66	29.77
Ours	**35.93**	**37.18**	**38.46**

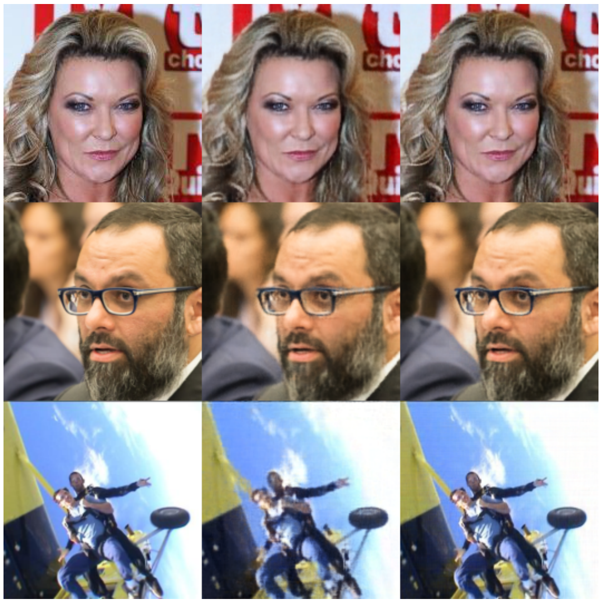

Figure 3. The reconstruction examples of TransINR [4] (middle) and our framework (right), given 178×178 original images (left) in CelebA, FFHQ, and ImageNette in each row, respectively.

Table 3. PSNRs on high-resolution FFHQ reconstruction according to MLP dimensions d and the number of weight tokens r.

	d	r	256×256	512×512
TransINR	256	64×4+3	30.96	29.35
TransINR	256	256×4+3	32.92	31.00
Ours	256	256	**34.68**	**31.58**
TransINR	1024	64×4+3	33.83	31.57
TransINR	1024	256×4+3	36.50	32.68
Ours	1024	256	38.43	35.22
Ours	1024	1024	**40.37**	**36.27**

178×178 Image Reconstruction We evaluate our generalizable INRs of MLP with $d = 256$ on 178×178 image reconstruction. Our transformer uses $r = 256$ weight tokens, since TransINR uses 259 (64×4+3) weight tokens to modulate all MLP layers [4]. Table 2 shows that our framework significantly outperforms Learned Init [24] and TransINR on the three datasets by a large margin. TransINR cannot precisely reconstruct the images of ImageNette, which contains complex patterns in images, but our framework produces high quality of reconstructed images. Figure 3 shows

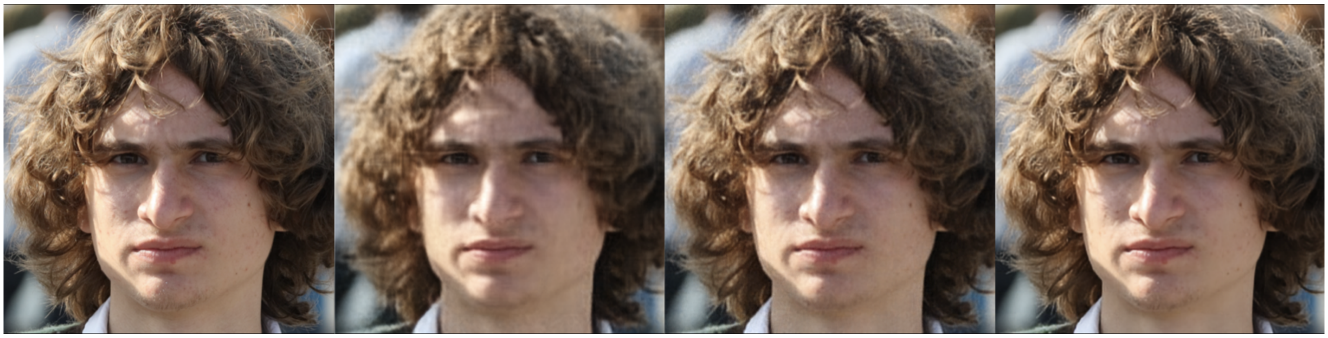

Figure 4. Examples of original 512×512 images (left), and reconstructed images by TransINR [4] with $d = 256$ and $r = 259$ (middle left), our framework with $d = 256$ and $r = 256$ (middle right), and $d = 1024$ and $r = 1024$ (right).

Table 4. Performace comparison of generalizable INRs on novel view synthesis from a single support view.

	Chairs	Cars	Lamps
Matched Init [24]	16.30	22.39	20.79
Shuffled Init [24]	10.76	11.30	13.88
Learned Init [24]	18.85	22.80	22.35
TransINR	19.05	**24.18**	22.89
Ours	**19.30**	**24.18**	**23.41**

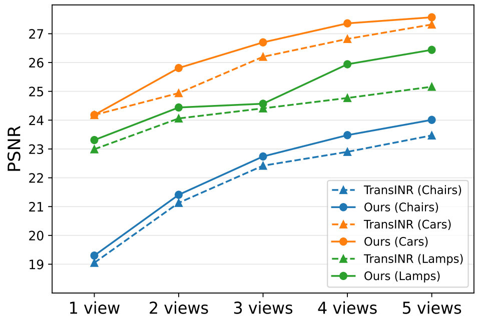

Figure 5. PSNRs on novel view synthesis of Chairs, Cars, Lamps according to the number of support views (1-5 views).

that our framework reconstructs images with high precision.

High-Resolution Image Reconstruction We evaluate our framework on high-resolution FFHQ images with 256×256 and 512×512 resolutions. As high-resolution images would require a larger capacity of INRs, MLP models with $d = 256$ and $d = 1024$ are modulated by instance pattern composers with $r = 256$ and $r = 1024$. In Table 3, our framework significantly outperforms previous TransINR on high-resolution image reconstruction in various settings. When we increase the d to 1024, our framework significantly improves PSNRs for 256×256 and 512×512 images. The results show that our coordinate-based MLP can adapt to unseen data despite the minimal changes in MLP weights. In Figure 4, TransINR cannot reconstruct high-frequency details of original images, but our framework precisely reconstructs those details. Considering that previous studies have not achieved high performance on high-resolution images, our results demonstrate that the weight modulation is the key to generalizable INRs.

4.3. Novel View Synthesis

We evaluate our framework on novel view synthesis of a 3D object based on the ShapeNet Chairs, Cars, and Lamps datasets. Given a 3D object and a few view images with known camera poses, we train the coordinate-based MLP, which has six layers with $d = 256$, $d_{\text{in}} = 3$ for (x, y, z) coordinates, and $d_{\text{out}} = 4$ for outputs of RGB values and its density, to estimate the view of a 3D object under unseen camera poses. For evaluation, we randomly sample a camera pose. To synthesize a novel view image, we use the simple volumetric rendering [15] to focus on the effectiveness of our weight modulation method instead of achieving state-of-the-art performance. We follow the experimental settings of previous studies [4, 24] except for the manual decay of learning rate in TransINR [4], but use a constant learning rate until the training converges.

In Table 4, our generalizable INRs outperform previous approaches on novel view synthesis under a single support view. Note that the results of our framework and TransINR are not benefited from the test-time optimization (TTO), but the other approaches use TTO by the nature of optimization-based meta-learning. Figure 5 also shows our framework consistently outperforms TransINR as the number of support views increases, while our performance is continuously improved. Although Figure 6 shows that our framework provides blurry views due to the simple volumetric rendering, our framework can capture and synthesize the shapes and colors of 3D objects based on given support views.

Table 5 shows the performance after 100 TTO steps on ShapeNet-Lamps with 1-5 support views. We use the following two types of TTO. The first approach optimizes the whole MLP weights, but the other only optimizes one

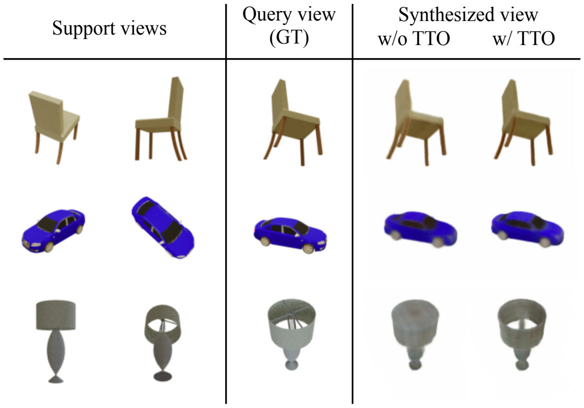

Figure 6. Novel view synthesis examples by our framework with two support views of ShapeNet Chairs, Cars, and Lamps.

Table 5. The improvements after test-time optimization (TTO) on novel view synthesis of ShapeNet-Lamps with support views.

	the number of views				
	1	2	3	4	5
TransINR	22.99	24.06	24.41	24.77	25.16
w/ TTO (all weights)	25.13	27.28	28.09	28.56	28.93
Ours	23.40	24.33	24.62	26.05	26.79
w/ TTO ($\mathbf{V}^{(n)}$)	**25.47**	27.53	28.34	**29.52**	**30.19**
w/ TTO (all weights)	25.40	**27.57**	**28.40**	29.43	30.07

Table 6. PSNRs of our generalizable INRs on ImageNette and Lamps (2 views) according to the types of modulation methods.

$\mathbf{W}_{\text{CP}}^{(n)}$	ImageNette	Lamps (2 views)
$\mathbf{V}^{(n)}$	30.01	**24.69**
$\mathbf{U}^{(n)}\mathbf{V}^{(n)}$	32.35	23.04
$\mathbf{U} \odot \mathbf{V}^{(n)}$	30.64	24.18
$\mathbf{U}\mathbf{V}^{(n)}$ (ours)	**35.93**	24.44

weight matrix of instance pattern composer $\mathbf{V}^{(n)}$. Table 5 shows that our framework consistently outperforms TransINR after TTO. Moreover, despite updating only one weight matrix, the improvement after TTO of $\mathbf{V}^{(n)}$ is competitive with or even better than TTO of all weights. In other words, our model learns a generalizable and instance-agnostic pattern composition rule to achieve high performance if the instance pattern composers $\mathbf{V}^{(n)}$ can be accurately predicted. Figure 6 demonstrates that the synthesized images also become sharp and precise after TTO.

4.4. Ablation Study

Methods for Weight Modulation We first validate the design of our modulation method in Eq. (5), where the modulated weights $\mathbf{W}^{(n)}$ consists of the matrix multiplication of instance-agnostic weight $\mathbf{U}$ and instance-specific parameter $\mathbf{V}^{(n)}$. We compare our method with three variants in Table 6 to predict the modulated weights $\mathbf{W}^{(n)}$: the direct prediction $\mathbf{W}^{(n)} = \mathbf{V}^{(n)}$, Hadamard product $\mathbf{U} \odot \mathbf{V}^{(n)}$, and an instance-specific $\mathbf{U}^{(n)}$. In the case of $\mathbf{U}^{(n)}\mathbf{V}^{(n)}$, our transformer predicts each column vector of $\mathbf{U}^{(n)}$ and row vector of $\mathbf{V}^{(n)}$. Although the variants provide reasonable results in Table 6, our method shows high performance on both image reconstruction and novel view synthesis.

Table 7. PSNRs of our generalizable INRs on image reconstruction according to the location of modulated weights in MLP.

	the modulated layer of MLP				
	1	2	3	4	5
ImageNette	31.00	**35.93**	32.99	31.10	20.26
FFHQ	36.04	**36.20**	34.2	31.09	22.92

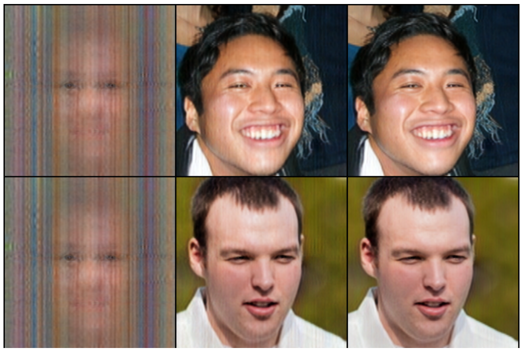

Figure 7. Reconstructions of FFHQ after inner-loop updates for $\mathbf{V}^{(n)}$ (left: initial, middle: first update, right: second update).

The Location of Modulated Weights We change the location of weight modulation from the first layer to the fifth layer and evaluate its effects. Table 7 shows that modulating the second MLP weight achieves the best performance on image reconstruction of both ImageNette and FFHQ. Interestingly, the performance on ImageNette significantly deteriorates when we modulate the first layer, which takes Fourier features as its input. Considering the high complexity of ImageNette, a coordinate-based MLP necessitates complex frequency patterns rather than simple and periodic Fourier features to generate instance content patterns of an image. The PSNRs also deteriorate when we modulate the third or above layers, since the MLP cannot learn the pattern composition rule enough. Thus, we modulate the second layer for generalizable INRs in experiments.

Optimization-based Meta-Learning Our instance pattern composer can improve the performance of generalizable INRs with optimization-based meta-learning in Algorithm 1. We train a coordinate-based MLP on FFHQ 256×256 for 100 epochs with $\epsilon = 0.01$, $\epsilon' = 0.001$, and

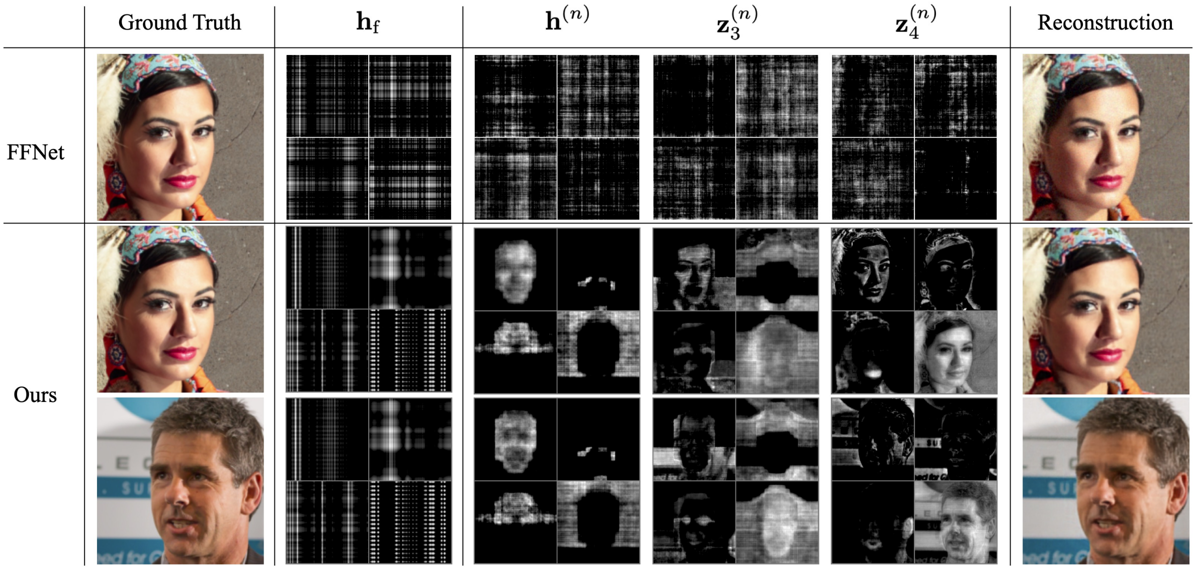

Figure 8. Activation maps of FFNet [25] to be separately trained to memorize a data instance (top row) and our generalizable INRs (bottom two rows). We select four neurons from each hidden layer to visualize and interpret the activation maps over input coordinates.

$N_\text{inner} = 2$. We also train another model using MAML [9] to adapt the whole MLP weights for a data instance during 100 epochs with $N_\text{inner} = 2$ and $\epsilon = 0.001$ and $\epsilon' = 0.0003$. While the model trained with MAML achieves a PSNR of 32.84 after adaptations, our model achieves a higher PSNR of 33.74. We remark that our model only adapts one weight matrix $\mathbf{V}^{(n)}$ in the adaptation steps.The results imply that a coordinate-based MLP can effectively compose instance content patterns to represent unseen data, while exploiting the shared representations across instances. Figure 7 also shows that our generalizable INRs can adapt to unseen instances after two update steps $\mathbf{V}^{(n)}$.

4.5. Visualization Analysis of MLP Activations

Figure 8 visualize the activations of trained coordinate-based MLP on FFHQ to understand how our generalizable INRs work. First, we train a coordinate-based MLP on a data instance seperately [25], and visualize the activations of selected neurons in each MLP layer. However, a semantic structure does not exist in the activation maps, since the MLP is trained to memorize a data instance without learning the underlying structures across different instances.

Contrastively, our generalizable INRs have common structures in activation maps of different images. After non-periodic and instance-agnostic frequency patterns $\mathbf{h}_\text{f}$ are composed, instance-specific content patterns $\mathbf{h}^{(n)}$ are generated. Regardless of data instances, each neuron in the second layer is activated at similar coordinates, but shows different signals of patterns. Our instance pattern composers learn to assign each neuron in the second layer to different coordinate regions for generating instance-specific patterns. Then, while subsequent layers use the instance-agnostic rule to compose $\mathbf{h}^{(n)}$, each neuron has a similar role across instances, enlarges the activated regions, and synthesizes complex and global patterns as the layer goes up. That is, our generalizable INRs learn underlying structures across instances to represent complex data as the composition of instance-specific low-level patterns.

5. Conclusion

This study has proposed the framework for generalizable INRs via instance pattern composers, which modulate one weight matrix of the early MLP layer to generalize the learned INRs for unseen data instances. Thanks to the simplicity, our framework is compatible with both optimization-based meta-learning and hypernetworks to significantly improve the performance of generalizable INRs. Experimental results demonstrate the broad impacts of the proposed method on various domains and tasks, since our generalizable INRs effectively learn underlying representations across instances. Our study remains a theoretical analysis of our generalizable INRs as worth exploration.

6. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2018-0-01398: Development of a Conversational, Self-tuning DBMS, 35%; No.2022-0-00113: Sustainable Collaborative Multi-modal Lifelong Learning, 30%) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2021R1A2B5B03001551, 35%)

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

[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. 2

[3] 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. 2

[4] Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pages 170–187. Springer, 2022. 1, 2, 3, 4, 5, 6

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[6] 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. 2

[7] 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. 1, 2, 4

[8] 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. 1, 2, 4

[9] 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. 1, 2, 3, 4, 8

[10] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017. 2, 3, 4

[11] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 2

[12] Yuzhe Lu, Kairong Jiang, Joshua A Levine, and Matthew Berger. Compressive neural representations of volumetric scalar fields. In *Computer Graphics Forum*, volume 40, pages 135–146. Wiley Online Library, 2021. 2

[13] 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. 2

[14] 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. 2

[15] 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. 2, 6

[16] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015. 5

[17] 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. 1, 2

[18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 2

[19] 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. 2

[20] Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems*, 33:10136–10147, 2020. 2

[21] 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. 2

[22] 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. 2

[23] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007. 1, 2, 3

[24] 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. 1, 2, 3, 4, 5, 6

[25] 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. 2, 3, 8

[26] 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. 2

[27] Ellen D. Zhong, Tristan Bepler, Joseph H. Davis, and Bonnie Berger. Reconstructing continuous distributions of 3d protein structure from cryo-em images. In *International Conference on Learning Representations*, 2020. 2

[28] 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. 2, 4