STRUCTURAL PRESERVATION OF DATA IN IMPLICIT NEURAL REPRESENTATIONS VIA THE IMPLICIT FUNCTION THEOREM

Anonymous authors
Paper under double-blind review

ABSTRACT

Implicit Neural Representations (INRs) parameterize signals of interest as coordinate-based mappings, where each data sample is uniquely encoded in a specific set of weights. This data-to-weight association provides a convenient platform for studying Weight Space Learning (WSL), an emerging field with potential for tasks like meta-learning or transfer learning. So far, a precise theoretical explanation of how the underlying data structure is preserved in the learned weights is still missing. In this work, we employ the Implicit Function Theorem (IFT) to establish a rigorous correspondence between the data space and its latent weight representation space. We analyze a framework that maps instance-specific latent codes to INR weights via a shared hyper-network. Empirically, we construct a synthetic Fourier-curve testbed to verify data structure recovery. Furthermore, we demonstrate competitive performance against existing approaches on INR weight classification across 2D and 3D datasets using more compact networks. The present work offers a theoretical lens for future investigation into WSL.

1 INTRODUCTION

Implicit Neural Representations (INRs) are neural networks that map input coordinates to signal values, such as spatial locations to pixel intensities or 3D surface distances. By modeling each data sample as a coordinate-based continuous function (Park et al., 2019), INRs leverage the universal approximation property of neural networks (Hornik et al., 1989; Kidger & Lyons, 2020) to reconstruct a wide variety of data. Specifically, each data sample can be represented by a unique neural network, which establishes an explicit correspondence between the samples and their associated network weights. These INR weights can thus serve as representations of the data samples, and have been recently exploited for image classification (Dupont et al., 2022; Luigi et al., 2023; Navon et al., 2023; Zhou et al., 2023b;a; Gielisse & van Gemert, 2025; Janik & Byra, 2026). Clearly, INRs provide a natural and convenient platform for studying the Weight Space Learning (WSL) (Schürholt et al., 2024; Han et al., 2026), which is an emerging research topic, treating neural network weights as a standalone data modality. Most recent works directly investigate the weight space itself, such as meta-learning (Finn et al., 2017), parameter-efficient fine-tuning (Houlsby et al., 2019), and model merging (Yang et al., 2024). Despite empirical successes of these attempts, a solid mathematical foundation of WSL is still largely lacking.

In this work, we adopt INR weight classification as the main investigation arena, as it offers a controllable and mathematically tractable setting for examining whether network weights can serve as representations of data. One common approach in the literature adopts a two-stage pipeline: (1) training individual INRs for all data samples, and (2) learning a weight-space network to classify the resulting INR weights. Since training of neural networks is invariant with respect to arbitrary neuron permutations, most existing works address this permutation-invariant property to facilitate learning over weights. Unfortunately, difficulties of non-uniqueness in INR weights arise from not only permutation-invariance in weights, but also the optimization process itself. Even for the same data sample, independently trained networks can settle at different stationary points, which can result in disruption of neighborhood structures and severe geometric shattering in the INR weight space. As an alternative approach to individually fitting INRs, a shared hyper-network can be used

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

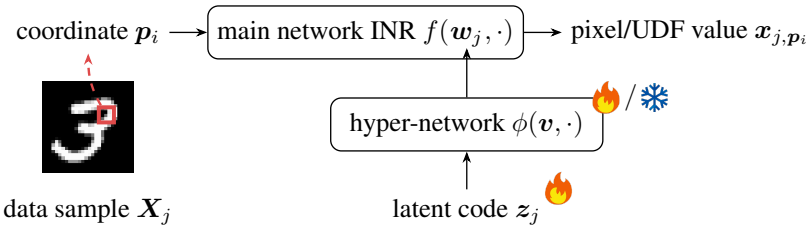


Figure 1: The main network INR maps coordinates to signals, with weights $\boldsymbol{w}_j$ generated by the hyper-network. On the training set, the hyper-network weights $\boldsymbol{v}$ and the latent codes $\boldsymbol{z}_j$ are trained jointly. On the test set, the hyper-network is frozen and only the latent codes $\boldsymbol{z}_j$ are optimized.

to generate network weights from latent vectors (Sitzmann et al., 2019). We refer to this framework as Hyper-Network-based INR (HN-INR), as illustrated in Figure 1. In this work, we adopt a common manifold hypothesis of the data space, i.e., high-dimensional observations $\boldsymbol{X}_j$ reside closely near an underlying low-dimensional smooth data manifold $\mathcal{X}$. Under this hypothesis, the set of corresponding INR weights is expected to also stay close to a low-dimensional manifold in the INR weight space. This construction motivates the HN-INR model to employ a low-dimensional latent code $\boldsymbol{z}_j$ to generate the weights $\boldsymbol{w}_j$. While hyper-networks can rely on an auxiliary encoder to infer latent codes from input data, we adopt an auto-decoder structure (Park et al., 2019) instead. In this setting, instance-specific latent codes are not predicted by an encoder, but rather are free parameters that can be optimized jointly with the hyper-network during reconstruction. Consequently, these learned latent codes serve as compact representations of the corresponding INR weights.

In this work, we propose to utilize the classic Implicit Function Theorem (IFT) to investigate the relationship between data samples and their latent weight representations under the HN-INR model. Technically, we analyze the non-degeneracy of the Jacobian of this implicit function to ensure a smooth local mapping from an open neighborhood around each data sample into a corresponding latent weight space. These local implicit mappings are shown to benefit downstream tasks of weight space classification. Furthermore, we empirically validate such structural preservation on a synthetic testbed with known structure (e.g., the Swiss roll), and achieve competitive classification accuracy across diverse 2D and 3D benchmarks. By using the HN-INR as a minimal, yet non-trivial setting, our approach remains compact and lightweight without relying on heavy permutation quotienting.

In summary, our contributions of this work are threefold: (1) a rigorous paradigm formulating data-weight relationships for INRs through the lens of the IFT; (2) mathematical analysis of concrete conditions to characterize, diagnose, and control structural preservation of data in weight space; (3) empirical evidence validating theoretical guarantees across controlled synthetic testbeds and competitive 2D/3D benchmarks.

2 Mathematical Modeling: HN-INR through the Lens of the IFT

As shown in Figure 1, HN-INR consists of a shared hyper-network ϕ and an INR main network f. Let $\mathcal{P} \subset \mathbb{R}^m$ be an open set of coordinates and $\mathcal{W} \subset \mathbb{R}^d$ be the set of weights of INRs. An INR main network takes sampled coordinates $\boldsymbol{p} \in \mathcal{P}$ as input and generates the corresponding pixel value or Unsigned Distance Function (UDF) value, denoted by $\boldsymbol{x} \in \mathbb{R}^c$ (Mullen et al., 2010). Specifically, for a set of sampled coordinates $\{\boldsymbol{p}_i\}_{i=1}^n \subset \mathcal{P}$, any data sample $\boldsymbol{X} := \{\boldsymbol{x}_{\boldsymbol{p}_i}\}_{i=1}^n \in \mathcal{X} \subset \mathbb{R}^{n\times c}$ can be represented as the values of an underlying function from $\mathcal{P}$ to $\mathbb{R}^c$ at these coordinates. Here, we model an INR main network as the following mapping

$$f\colon \mathcal{W} \times \mathcal{P} \to \mathbb{R}^c, \qquad (\boldsymbol{w}, \boldsymbol{p}) \mapsto f(\boldsymbol{w}, \boldsymbol{p}). \tag{1}$$

The weights $\boldsymbol{w}_j$ for each data sample are generated by a hyper-network, which maps a learnable latent code $\boldsymbol{z}_j \in \mathcal{Z} \subset \mathbb{R}^l$ to the weights $\boldsymbol{w}_j \in \mathcal{W}$ of the main network, i.e.,

$$\phi\colon \mathcal{V} \times \mathcal{Z} \to \mathcal{W}, \qquad (\boldsymbol{v}, \boldsymbol{z}_j) \mapsto \phi(\boldsymbol{v}, \boldsymbol{z}_j). \tag{2}$$

Here, $\mathcal{V} \subset \mathbb{R}^{d_v}$ denotes the space of hyper-network weights, which are shared across all data samples. The space of latent codes $\mathcal{Z} \subset \mathbb{R}^l$ is an open set. Furthermore, we assume that f and ϕ are

C^∞. For the INR main network f, this holds by the choice of INR architecture, SIREN (Sitzmann et al., 2020), whose sinusoidal activations are C^∞. For the hyper-network ϕ, we adopt SiLU activations to ensure smoothness, since they are C^∞ unlike ReLU, and the synthetic testbed uses a linear hyper-network.

The classic residual loss for assessing the reconstruction of the INR against a single data sample can be defined as

$$\begin{aligned}\ell\colon \mathcal{V}\times\mathcal{Z}\times\mathcal{X}\to\mathbb{R},\quad \ell(\boldsymbol{v},\boldsymbol{z},\boldsymbol{X}) &:= \frac{1}{nc}\left\|\left\{f(\phi(\boldsymbol{v},\boldsymbol{z}),\boldsymbol{p}_i)\right\}_{i=1}^n-\boldsymbol{X}\right\|_F^2\\ &= \frac{1}{nc}\sum_{i=1}^n\sum_{k=1}^c\Big(f(\phi(\boldsymbol{v},\boldsymbol{z}),\boldsymbol{p}_i)_k - x_{\boldsymbol{p}_i,k}\Big)^2,\end{aligned} \tag{3}$$

where $x_{\boldsymbol{p}_i,k}$ denotes the k-th channel of the value of sample $\boldsymbol{X}$ at coordinate $\boldsymbol{p}_i$, and $\|\cdot\|_F$ denotes the Frobenius norm of matrices. Minimizing the summation of the loss ℓ of all data samples yields a converged parameter $\boldsymbol{v}^*$ for the hyper-network, which allows us to define the following function

$$\ell_{\boldsymbol{v}^*}\colon \mathcal{Z}\times\mathcal{X}\to\mathbb{R},\qquad (\boldsymbol{z},\boldsymbol{X})\mapsto\ell(\boldsymbol{v}^*,\boldsymbol{z},\boldsymbol{X}). \tag{4}$$

Let us denote $\boldsymbol{w}:=\phi(\boldsymbol{v}^*,\boldsymbol{z})$, and the per-coordinate residual $\boldsymbol{\epsilon}_i := f(\phi(\boldsymbol{v}^*,\boldsymbol{z}),\boldsymbol{p}_i)-\boldsymbol{x}_{\boldsymbol{p}_i}$. We compute the differential map of $\ell_{\boldsymbol{v}^*}$ with respect to the weight latent code $\boldsymbol{z}$ as

$$\mathrm{D}_1\,\ell_{\boldsymbol{v}^*}(\boldsymbol{z},\boldsymbol{X}) = \frac{2}{nc}\sum_{i=1}^n\boldsymbol{\epsilon}_i^\top\,\mathrm{D}_1\,f(\boldsymbol{w},\boldsymbol{p}_i)\,\mathrm{D}_2\,\phi(\boldsymbol{v}^*,\boldsymbol{z}), \tag{5}$$

where $\mathrm{D}_1\,f(\boldsymbol{w},\boldsymbol{p}_i)$ and $\mathrm{D}_2\,\phi(\boldsymbol{v}^*,\boldsymbol{z})$ are the differential maps of f with respect to its two variables, respectively. Assuming joint minimization of both hyper-network weights and latent codes converges to a critical point of the residual loss for each sample $\boldsymbol{X}\in\mathcal{X}$, we have the vanishing gradients as

$$\mathrm{D}_1\,\ell_{\boldsymbol{v}^*}(\boldsymbol{z},\boldsymbol{X}) = \boldsymbol{0}. \tag{6}$$

Note, that this result is attainable under the exact reconstruction setting of all data samples. More discussions will be found in Section 3.2. Through the lens of the IFT, equation 6 establishes a potential mapping from $\boldsymbol{X}$ to $\boldsymbol{z}$. We thus construct the following *implicit function* that anchors our theoretical development

$$\xi_{\boldsymbol{v}^*}\colon \mathcal{Z}\times\mathcal{X}\to\mathbb{R}^l,\quad (\boldsymbol{z},\boldsymbol{X})\mapsto(\mathrm{D}_1\,\ell_{\boldsymbol{v}^*}(\boldsymbol{z},\boldsymbol{X}))^\top. \tag{7}$$

This function is essentially the gradient of $\ell_{\boldsymbol{v}^*}$ with respect to $\boldsymbol{z}$ under the canonical Euclidean metric. By viewing HN-INR training as an implicit correspondence, we leverage equation 7 to investigate how the learned weight representations can achieve structural preservation of the data.

3 Theoretical analysis via the IFT

3.1 Derivative formulation and decomposition

To evaluate conditions required by the IFT, we need to examine the derivative of $\xi_{\boldsymbol{v}^*}(\boldsymbol{z},\boldsymbol{X})$ with respect to $\boldsymbol{z}$. This derivative $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z},\boldsymbol{X})$ also coincides with the second-order derivative of $\ell_{\boldsymbol{v}^*}$, namely, the Hessian of the reconstruction loss. A straightforward computation yields

$$\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z},\boldsymbol{X}) = G^*_{\boldsymbol{X}}(\boldsymbol{z})+E^*_{\boldsymbol{X}}(\boldsymbol{z}), \tag{8}$$

where $G^*_{\boldsymbol{X}}(\boldsymbol{z})$ is the Gauss-Newton matrix and $E^*_{\boldsymbol{X}}(\boldsymbol{z})$ denotes the residual curvature perturbation, i.e.,

$$G^*_{\boldsymbol{X}}(\boldsymbol{z}) := \frac{2}{nc}\sum_{i=1}^n\boldsymbol{J}_i^\top\boldsymbol{J}_i,\qquad E^*_{\boldsymbol{X}}(\boldsymbol{z}) := \frac{2}{nc}\sum_{i=1}^n\boldsymbol{\epsilon}_i^\top(\mathcal{B}_i+\mathcal{C}_i). \tag{9}$$

In this decomposition, the per-coordinate Jacobian $\boldsymbol{J}_i\in\mathbb{R}^{c\times l}$ is defined as

$$\boldsymbol{J}_i := \mathrm{D}_1\,f(\boldsymbol{w},\boldsymbol{p}_i)\,\mathrm{D}_2\,\phi(\boldsymbol{v}^*,\boldsymbol{z}), \tag{10}$$

and the terms $\mathcal{B}_i$ and $\mathcal{C}_i$ capture the second-order variations of the INR and hyper-network, given by

$$\mathcal{B}_i := (\mathrm{D}_2\,\phi(\boldsymbol{v}^*,\boldsymbol{z}))^\top\,\mathrm{D}_1^2\,f(\boldsymbol{w},\boldsymbol{p}_i)\,\mathrm{D}_2\,\phi(\boldsymbol{v}^*,\boldsymbol{z}),\qquad \mathcal{C}_i := \mathrm{D}_1\,f(\boldsymbol{w},\boldsymbol{p}_i)\,\mathrm{D}_2^2\,\phi(\boldsymbol{v}^*,\boldsymbol{z}). \tag{11}$$

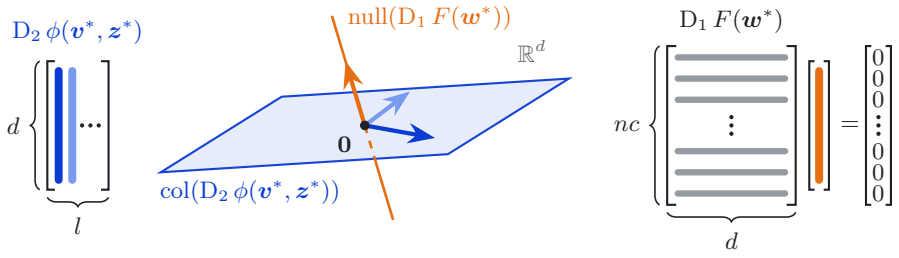


Figure 2: Linear subspace intersection (Condition 2). The columns of the hyper-network Jacobian span the blue subspace, and the vectors orthogonal to all rows of the main network Jacobian form the orange subspace, which intersect only at $\mathbf{0}$.

Let $\boldsymbol{z}^*$ denote the fitted latent code of a sample $\boldsymbol{X}$, so that $\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X}) = \mathbf{0}$. When the derivative $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})$ in equation 8 has full rank, the classic IFT implies that there exist an open neighborhood $U(\boldsymbol{X}) \subset \mathcal{X}$ of $\boldsymbol{X}$ and $V(\boldsymbol{z}^*) \subset \mathcal{Z}$ of $\boldsymbol{z}^*$ and a unique C^∞ function $g\colon U(\boldsymbol{X}) \to V(\boldsymbol{z}^*)$ satisfying

$$g(\boldsymbol{X}) = \boldsymbol{z}^*, \quad \text{and} \quad \xi_{\boldsymbol{v}^*}(g(\boldsymbol{X}'), \boldsymbol{X}') = \mathbf{0}, \quad \forall \boldsymbol{X}' \in U(\boldsymbol{X}). \tag{12}$$

In the rest of this section, we analyze the non-degeneracy of $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})$ under two scenarios. We first consider the idealized setting of exact reconstruction in Section 3.2, where we assume that the capacity of the INRs is sufficiently large to ensure exact reconstruction across the dataset $\mathcal{X}$. We then address a more realistic setting with non-vanishing reconstruction residuals in Section 3.3.

3.2 Local embedding under exact reconstruction

Under exact reconstruction of HN-INR on all data samples, the network achieves zero residual at the fitted latent code $\boldsymbol{z}^*$, i.e., $\boldsymbol{\epsilon}_i = \mathbf{0}$ for all coordinates $i = 1, \ldots, n$. In this setting, the residual curvature perturbation vanishes, namely, $E^*_{\boldsymbol{X}}(\boldsymbol{z}^*) = \mathbf{0}$. We define the stacked INR output over all sampled coordinates as

$$F(\boldsymbol{w}) \coloneqq [f(\boldsymbol{w}, \boldsymbol{p}_1)^\top, \ldots, f(\boldsymbol{w}, \boldsymbol{p}_n)^\top]^\top \in \mathbb{R}^{nc}. \tag{13}$$

Differentiating the composite forward map $F(\phi(\boldsymbol{v}^*, \boldsymbol{z}))$ yields the composite system Jacobian as

$$\tilde{\boldsymbol{J}} \coloneqq \mathrm{D}_1\,F(\boldsymbol{w}^*)\,\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*) \in \mathbb{R}^{nc \times l}, \tag{14}$$

which aggregates the per-coordinate Jacobians $\boldsymbol{J}_i$ from equation 10 and thus simplifies equation 8 to

$$\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X}) = G^*_{\boldsymbol{X}}(\boldsymbol{z}^*) = \frac{2}{nc}\tilde{\boldsymbol{J}}^\top\tilde{\boldsymbol{J}} \in \mathbb{R}^{l \times l}. \tag{15}$$

To ensure that $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})$ is strictly positive definite, the composite Jacobian $\tilde{\boldsymbol{J}}$ must have full column rank. We establish this property through two subspace conditions.

Condition 1 (Hyper-network immersion)**.** For any $\boldsymbol{X} \in \mathcal{X}$, the fitted latent code $\boldsymbol{z}^*$ satisfies $\mathrm{rank}(\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*)) = l$.

It is not a strong condition for the hyper-network because its output dimension is much bigger than its input dimension ($d \gg l$). In practice, this property implies that every latent dimension actively generates distinct weight variations without dimensional collapse.

Condition 2 (Trivial subspace intersection)**.** For any $\boldsymbol{X} \in \mathcal{X}$, the fitted latent code $\boldsymbol{z}^*$ and the weights $\boldsymbol{w}^* = \phi(\boldsymbol{v}^*, \boldsymbol{z}^*)$ satisfy $\mathrm{null}(\mathrm{D}_1\,F(\boldsymbol{w}^*)) \cap \mathrm{col}(\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*)) = \{\mathbf{0}\}$.

As visualized in Figure 2, the column space $\mathrm{col}(\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*))$ (blue plane) contains all first-order weight variations driven by latent perturbations. In contrast, the null space $\mathrm{null}(\mathrm{D}_1\,F(\boldsymbol{w}^*))$ (orange line) collects weight directions that induce zero first-order change in coordinate outputs. Their trivial intersection at $\mathbf{0}$ ensures that every non-zero latent perturbation produces an observable response across coordinates, rather than vanishing into unobservable first-order blind spots.

Lemma 1 (Local non-degeneracy and embedding via trivial subspace intersection)**.** *Let the data lie on a smooth manifold $\mathcal{X}$ of intrinsic dimension m^*, and let the fitted latent code $\boldsymbol{z}^*$ reconstruct $\boldsymbol{X} \in \mathcal{X}$ exactly. If Condition 1 and Condition 2 are satisfied, the following properties hold:*

(a) ***Positive definiteness****: the derivative* $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X}) = G^*_{\boldsymbol{X}}(\boldsymbol{z}^*)$ *is strictly positive definite, with* $\sigma_{\min}(\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})) = \lambda_{\min}(\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})) > 0$*, so the implicit function* g *in equation 12 exists.*

(b) ***Local embedding****: if exact reconstruction also holds along* g*, namely* $F(\phi(\boldsymbol{v}^*, g(\boldsymbol{X}'))) = \boldsymbol{X}'$ *for all* $\boldsymbol{X}' \in U(\boldsymbol{X})$*, then* g *is a smooth embedding of* $U(\boldsymbol{X})$ *into* $\mathcal{Z}$*.*

Lemma 1 establishes that the learned latent codes faithfully preserve local data structure. Because g is a smooth embedding, it continuously maps nearby data samples to adjacent latent codes, preventing optimization-induced shattering. Importantly, this guarantee does not require the latent dimension to match the intrinsic dimension, as any latent space satisfying $l \geq m^*$ preserves the m^*-dimensional data structure. Part (b) makes $m^* \leq l$ necessary but not sufficient, since it still requires exact reconstruction around $\boldsymbol{X}$. The detailed proof is provided in Appendix B.1.

3.3 Non-degeneracy under non-vanishing residuals

In real applications, data samples are not guaranteed to sit perfectly on a smooth manifold. Meanwhile, due to potential optimization constraints, exact reconstruction is rarely achieved even on synthetic data. As a sequel, the residuals $\boldsymbol{\epsilon}_i$ and the curvature perturbation $E^*_{\boldsymbol{X}}(\boldsymbol{z})$ in equation 8 do not vanish. Although a sufficiently small reconstruction error can make $E^*_{\boldsymbol{X}}(\boldsymbol{z})$ negligible, we still need to examine how these higher-order curvature terms influence $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})$. We adopt an L_2 penalty $\frac{\lambda}{2}\|\boldsymbol{z}\|_2^2$ to regularize the original loss with regularization weight $\lambda > 0$, i.e., an L_2 regularized residual loss defined as

$$\ell_\lambda(\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{X}) := \frac{1}{nc}\left\|\left\{f(\phi(\boldsymbol{v}, \boldsymbol{z}), \boldsymbol{p}_i)\right\}_{i=1}^n - \boldsymbol{X}\right\|_F^2 + \frac{\lambda}{2}\|\boldsymbol{z}\|_2^2. \tag{16}$$

At critical points of this L_2-regularized loss, the implicit function in equation 7 becomes

$$\xi^\lambda_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X}) \coloneqq \xi_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X}) + \lambda \boldsymbol{z} = \boldsymbol{0}. \tag{17}$$

Differentiating $\xi^\lambda_{\boldsymbol{v}^*}$ with respect to $\boldsymbol{z}$ yields

$$\mathrm{D}_1\,\xi^\lambda_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X}) = \mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X}) + \lambda \boldsymbol{I}_l = (G^*_{\boldsymbol{X}}(\boldsymbol{z}) + \lambda \boldsymbol{I}_l) + E^*_{\boldsymbol{X}}(\boldsymbol{z}). \tag{18}$$

Since the underlying scalar objective is twice continuously differentiable, $\mathrm{D}_1\,\xi^\lambda_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})$ is a real symmetric matrix, whose invertibility is governed entirely by its real eigenvalues. While the regularized Gauss-Newton term satisfies $G^*_{\boldsymbol{X}}(\boldsymbol{z}) + \lambda \boldsymbol{I}_l \succ 0$, the perturbation $E^*_{\boldsymbol{X}}(\boldsymbol{z})$ can still introduce negative eigenvalues. Applying Weyl's inequality (Weyl, 1912) provides the spectral lower bound

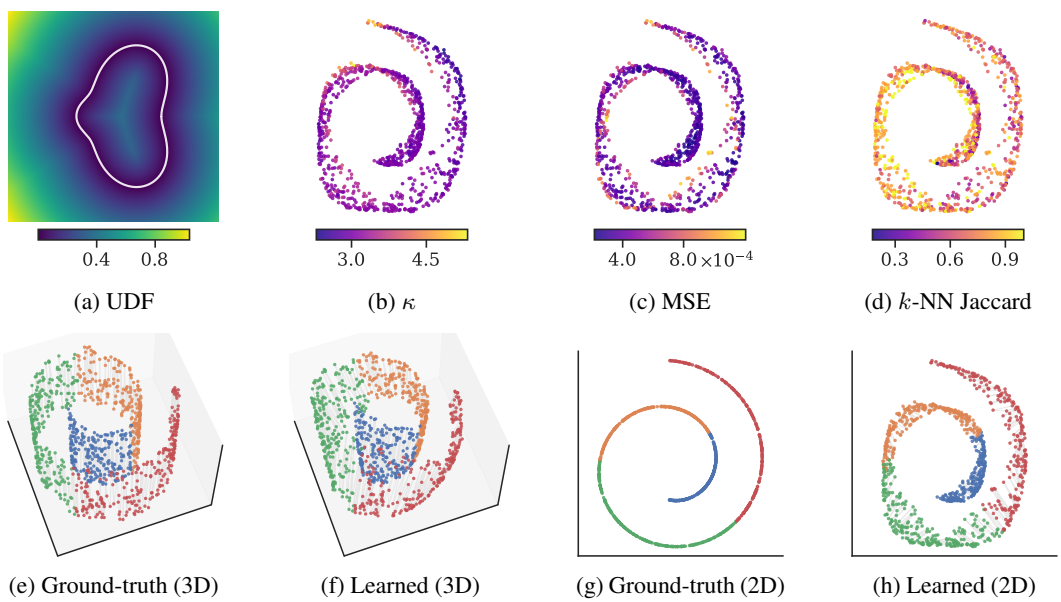
$$\lambda_{\min}(\mathrm{D}_1\,\xi^\lambda_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})) \geq \lambda_{\min}(G^*_{\boldsymbol{X}}(\boldsymbol{z}) + \lambda \boldsymbol{I}_l) + \lambda_{\min}(E^*_{\boldsymbol{X}}(\boldsymbol{z})). \tag{19}$$

A strictly positive lower bound guarantees that $\mathrm{D}_1\,\xi^\lambda_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})$ remains positive definite and therefore invertible. This estimate represents a worst-case scenario that assumes the minimal eigenvectors of $G^*_{\boldsymbol{X}}(\boldsymbol{z}) + \lambda \boldsymbol{I}_l$ and $E^*_{\boldsymbol{X}}(\boldsymbol{z})$ lie along the same direction. In practice, these eigenvectors do not necessarily coincide, meaning that the true spectrum of $\mathrm{D}_1\,\xi^\lambda_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})$ can remain above this theoretical minimum.

The spectral lower bound in equation 19 establishes two operational levers to preserve non-degeneracy. First, adjusting the weight λ directly shifts the spectrum upward through $\lambda \boldsymbol{I}_l$, lowering the condition number $\kappa \coloneqq \sigma_{\max}/\sigma_{\min}$ of $\mathrm{D}_1\,\xi^\lambda_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})$ as evaluated in Table 4. Second, lowering the reconstruction error reduces the residual magnitudes, suppressing the negative spectrum of $E^*_{\boldsymbol{X}}(\boldsymbol{z})$ toward zero (Figure 4). Optimizing reconstruction fidelity thus simultaneously enhances representation quality and mathematical non-degeneracy.

4 Examining downstream weight classification as a case study

In Section 3, we have shown that the IFT yields a smooth correspondence between samples and latent codes in the neighborhood of an individual sample. To further investigate how this local property can benefit downstream tasks, we investigate the problem of multi-class classification in weight spaces. Depending on how latent codes of the entire classes of data are arranged, we examine the conditions, under which the local implicit functions can be extended to a single smooth map over each class.



(a) UDF (b) κ (c) MSE (d) k-NN Jaccard

(e) Ground-truth (3D) (f) Learned (3D) (g) Ground-truth (2D) (h) Learned (2D)

Figure 3: Structural preservation and diagnostics on the Fourier-curve testbed.

We allow $\lambda \geq 0$ in the regularized critical point condition $\xi^{\lambda}_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X}) = \mathbf{0}$ of Section 3.3, so that $\lambda = 0$ recovers the unregularized setting. Each fitted latent code $\boldsymbol{z}^*_j$ is a critical point for its sample $\boldsymbol{X}_j$. Hence, the fitted pairs of a class lie in the critical set $\mathcal{S}_k$ defined below. Since the loss is non-convex in $\boldsymbol{z}$, a sample can be associated with several latent codes, and $\mathcal{S}_k$ can split into several connected components, which we call *branches*.

Proposition 1 (Component-wise global smooth extension)**.** *Let the data space decompose into K disjoint classes, $\mathcal{X} = \bigsqcup_{k=1}^{K} \mathcal{X}_k$, where each class $\mathcal{X}_k$ is a connected and simply connected smooth submanifold. Let $\mathcal{S}^*_k$ be a connected component of the critical set $\mathcal{S}_k := \{(\boldsymbol{z}, \boldsymbol{X}) \in \mathcal{Z} \times \mathcal{X}_k \mid \xi^{\lambda}_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X}) = \mathbf{0}\}$. Assume that:*

1. ***Non-degeneracy**: $\sigma_{\min}(\mathrm{D}_1\, \xi^{\lambda}_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})) > 0$ for all $(\boldsymbol{z}, \boldsymbol{X}) \in \mathcal{S}^*_k$.*
2. ***Bounded latent codes**: there exists $R < \infty$ such that $\|\boldsymbol{z}\|_2 \leq R$ for all $(\boldsymbol{z}, \boldsymbol{X}) \in \mathcal{S}^*_k$.*

Then there exists a unique smooth map $g_k \colon \mathcal{X}_k \to \mathcal{Z}$ such that $\mathcal{S}^*_k = \{(g_k(\boldsymbol{X}), \boldsymbol{X}) \mid \boldsymbol{X} \in \mathcal{X}_k\}$, that is, every sample of the class has exactly one latent code on this branch. Consequently, the latent codes $g_k(\mathcal{X}_k)$ of this branch form a connected set.

The proof is given in Appendix B.2. Although the IFT by itself guarantees structural preservation only locally, Proposition 1 shows that, when $\mathrm{D}_1\, \xi^{\lambda}_{\boldsymbol{v}^*}$ has full rank along a branch and the latent codes stay bounded, the local implicit functions join into a single smooth map g_k over the whole class. If the fitted codes of a class lie in one branch, the class therefore occupies one connected region of the latent space instead of fragments scattered across disconnected optimization basins, and so do its generated weights $\phi(\boldsymbol{v}^*, g_k(\mathcal{X}_k))$ since ϕ is continuous. This per-class connectedness is how the local IFT analysis benefits downstream classification on latent codes and generated weights (Section 5), although it does not imply that different classes are separated in $\mathcal{Z}$.

5 Experiments

In order to systematically verify the results in Sections 3, we first evaluate a synthetic Swiss-roll testbed with known structure (Section 5.1). We then compare downstream classification with WSL baselines on 2D and 3D benchmarks (Section 5.2) and finally check the IFT conditions numerically (Section 5.3). All experiments fit training and test samples as in Figure 1 and train downstream classifiers on the resulting latent codes $\boldsymbol{z}_j$ and generated weights $\boldsymbol{w}_j$.

Table 1: Classification accuracy (%) on 2D image benchmarks. Baseline results are taken from original publications; "–" indicates not reported. Ours: mean ± std over 5 independent runs. Values in parentheses indicate the parameter count of the downstream classifier.

Method	MNIST	F-MNIST	CIFAR10	CIFAR10-aug
DWSNet	85.71 ± 0.57	67.06 ± 0.29	–	–
NFN	92.9 ± 0.218	75.6 ± 1.07	–	46.6 ± 0.072
Inr2Array	98.5 ± 0.00	79.3 ± 0.00	–	63.4 ± 0.00
MWT-L	98.8 ± 0.06	90.43 ± 0.23	59.57 ± 0.52	64.7
HMoE-MWT	$\mathbf{99.06 \pm 0.15}$	90.72 ± 0.25	$\mathbf{65.01 \pm 0.63}$	69.11
Ours-z (11.5k classifier)	97.69 ± 0.20	87.77 ± 0.48	51.07 ± 0.34	55.01 ± 0.55
Ours-w (11.5k classifier)	97.66 ± 0.16	88.00 ± 0.36	50.11 ± 0.35	51.83 ± 0.63
Ours-z (2.15M classifier)	98.52 ± 0.12	$\mathbf{90.73 \pm 0.29}$	59.07 ± 0.29	$\mathbf{69.73 \pm 0.32}$
Ours-w (2.15M classifier)	98.47 ± 0.14	90.53 ± 0.34	56.54 ± 0.20	63.80 ± 0.66

5.1 Controlled synthetic testbed with known structure

We construct a synthetic testbed based on closed Fourier-curves to evaluate our theoretical results in an idealized setting that matches their assumptions. Each sample represents a two-dimensional closed curve parameterized by a combination of the first three cosine harmonics,

$$r(\theta; \boldsymbol{z}_j^{\mathrm{gt}}) = 0.5 + z_{j,1}^{\mathrm{gt}} \cos(\theta) + z_{j,2}^{\mathrm{gt}} \cos(2\theta) + z_{j,3}^{\mathrm{gt}} \cos(3\theta), \quad \theta \in [0, 2\pi), \tag{20}$$

where $\boldsymbol{z}_j^{\mathrm{gt}} \in \mathbb{R}^3$ denotes the ground-truth generating vector. Each curve is converted into an observation $\boldsymbol{X}_j \in \mathcal{X}$ by evaluating its UDF across a grid of 1000 coordinates. As shown in Figure 3a, colors encode UDF distance values. While the observation data $\boldsymbol{X}_j$ is high-dimensional ($n = 10^3$ coordinate-based UDF values) and the curves lie in two dimensions, the generating parameters $\boldsymbol{z}_j^{\mathrm{gt}}$ are sampled on a Swiss-roll shape in $\mathbb{R}^3$ (Figure 3e, g), which induces a low-dimensional data manifold $\mathcal{X}$. This controlled testbed satisfies our data manifold hypothesis by construction, allowing us to evaluate whether the hyper-network and latent weight representations recover the true underlying structure. Since the ground-truth structure of the data is known, we set $l = 3$ to match the dimension of the generating parameters and implement the hyper-network as a linear map $\phi(\boldsymbol{v}, \boldsymbol{z}) = \boldsymbol{V}\boldsymbol{z} + \boldsymbol{b}$ with $\boldsymbol{v} = (\boldsymbol{V}, \boldsymbol{b})$ (details in Appendix Table 5). For the classification purpose, we partition the Swiss roll along its spiral into four connected segments, shown in distinct colors in Figure 3e–h, with 1000 training and 500 test samples.

As shown in Figure 3b, the condition number κ of the derivative $\mathrm{D}_1\, \xi_{\boldsymbol{v}^*}^{\lambda}(\boldsymbol{z}_j^*, \boldsymbol{X}_j)$ remains uniformly low across all samples, illustrating that this derivative has full rank and is even well conditioned. This provides direct empirical evidence for Section 3.3, demonstrating that non-degeneracy holds under non-zero reconstruction residuals (Figure 3c) even without applying L_2 regularization ($\lambda = 0$). We next examine the learned latent representations $\boldsymbol{z}^*$. From the visualization of all samples (Figure 3e–h), we see that the spiral structure and segment ordering are faithfully recovered. Here, the three-dimensional plots (Figure 3e, f) display the raw, unprojected coordinates directly in $\mathbb{R}^3$, whereas the two-dimensional panels (Figure 3g, h) show a PCA projection. Figure 3d demonstrates that local neighborhoods are well preserved across the entire shape, yielding an average k-NN Jaccard index of 0.732 with $k = 10$ (higher indicates stronger neighborhood consistency). For classification, the learned latent codes $\boldsymbol{z}^*$ achieve 91.0% accuracy, closely matching the 93.2% obtained on the ground-truth $\boldsymbol{z}^{\mathrm{gt}}$.

5.2 Classification on 2D and 3D benchmarks

We evaluate a downstream classification task on several benchmarks, i.e., MNIST (Yann, 2010), FashionMNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky et al., 2009) for 2D images, alongside ModelNet40 (Wu et al., 2015), ShapeNet10 (Chang et al., 2015), and ScanNet10 (Qin et al., 2019) for 3D shapes (Tables 1 and 2). For the CIFAR10-aug column in Table 1, we apply ten-fold image augmentation to the CIFAR-10 training set. Compared with baselines such as HMoE (Janik

Table 2: Classification accuracy (%) on 3D shape benchmarks. Conventions follow Table 1.

Method	ModelNet40	ShapeNet10	ScanNet10
NFN	–	88.7 ± 0.461	65.9 ± 1.10
inr2vec	87.0	93.3	72.1
Ours-z (11.5k classifier)	84.78 ± 0.63	93.52 ± 0.18	67.08 ± 0.74
Ours-w (11.5k classifier)	84.55 ± 0.77	93.54 ± 0.40	67.71 ± 0.76
Ours-z (114k classifier)	$\mathbf{87.84 \pm 0.44}$	$\mathbf{94.37 \pm 0.17}$	72.04 ± 0.63
Ours-w (114k classifier)	87.81 ± 0.34	94.30 ± 0.20	$\mathbf{72.27 \pm 0.46}$

Table 3: Verification of IFT conditions across benchmarks (mean $\pm$ std over 5 independent fits).

Dataset	Training MSE	κ (99th percentile)	$\kappa > 10^3$	$\kappa > 10^4$
MNIST	$2.57 \pm 0.07 \times 10^{-3}$	$2.50 \pm 0.42 \times 10^{4}$	$21.59 \pm 2.75\%$	$2.49 \pm 0.43\%$
F-MNIST	$4.89 \pm 0.09 \times 10^{-3}$	$1.09 \pm 0.45 \times 10^{4}$	$10.87 \pm 4.16\%$	$1.09 \pm 0.49\%$
CIFAR10	$3.88 \pm 0.05 \times 10^{-3}$	$2.96 \pm 0.49 \times 10^{4}$	$31.63 \pm 1.91\%$	$2.91 \pm 0.48\%$
CIFAR10-aug	$2.15 \pm 0.02 \times 10^{-3}$	$7.39 \pm 0.24 \times 10^{5}$	$99.97 \pm 0.02\%$	$59.33 \pm 1.10\%$
ModelNet40	$5.20 \pm 0.20 \times 10^{-4}$	$2.72 \pm 2.00 \times 10^{3}$	$2.59 \pm 1.74\%$	$0.26 \pm 0.17\%$
ShapeNet10	$4.77 \pm 0.25 \times 10^{-4}$	$2.11 \pm 2.28 \times 10^{3}$	$2.07 \pm 2.17\%$	$0.21 \pm 0.23\%$
ScanNet10	$5.41 \pm 0.40 \times 10^{-4}$	$2.06 \pm 1.51 \times 10^{2}$	$0.16 \pm 0.21\%$	$0.01 \pm 0.02\%$

Table 4: Ablation on latent dimension l and regularization λ under off-shape noise.

	Latent dim. l ($\lambda = 0$)				L_2 **reg.** λ ($l = 5$)			
	2	**3**	4	5	0.00	0.02	0.04	0.08
MSE	9.25e−4	9.52e−5	**7.25e−5**	7.52e−5	**7.52e−5**	7.85e−5	8.19e−5	8.83e−5
κ	4.58e3	**4.46**	19.3	302	302	169	122	**87.8**
k-NN Jacc.	0.120	**0.751**	0.522	0.573	0.573	0.576	0.577	**0.577**
Acc. (%)	77.0	89.6	90.8	90.2	90.2	91.0	91.0	**91.6**

& Byra, 2026) that require Transformer backbones with Mixture-of-Experts architecture, our framework uses much smaller networks (Table 5 in the Appendix). Notably, an 11.5k-parameter classifier already yields competitive accuracy on both $\boldsymbol{z}$ and $\boldsymbol{w}$. These results are consistent with Proposition 1, showing that the local structural preservation from the IFT also benefits downstream classification.

5.3 IFT CONDITION VERIFICATION

We further verify the IFT conditions on the real-world 2D and 3D benchmarks, where the reconstruction residuals do not vanish (Table 3). The training MSE stays at the order of 10^{-3} or below on all datasets, so the residuals are small but non-zero. To assess the non-degeneracy of $\mathrm{D}_1\, \xi^{\lambda}_{\boldsymbol{v}^*}(\boldsymbol{z}^*_j, \boldsymbol{X}_j)$ across the training set, we compute its condition number κ for every sample and report the 99th percentile together with the fractions of values above 10^3 and 10^4. Except for CIFAR10-aug, this percentile stays below 3×10^4 and the fraction above 10^4 is at most 3%. CIFAR10-aug shows larger values, yet its ill-conditioned tail remains small, with only 7.3% above 10^5 and 0.73% above 10^6. These results confirm that, as long as training keeps the MSE small, $\mathrm{D}_1\, \xi^{\lambda}_{\boldsymbol{v}^*}$ remains non-degenerate on real data, which supports the analysis in Section 3.3 and the non-degeneracy assumption of Proposition 1. Figure 4 further verifies this behavior at the sample level on MNIST, showing that smaller MSE values suppress the curvature perturbation $E^*_{\boldsymbol{X}_j}(\boldsymbol{z}^*_j)$ as $\lambda_{\min}(E^*_{\boldsymbol{X}_j}(\boldsymbol{z}^*_j))$ rises toward zero, which directly substantiates the spectral floor analysis in Section 3.3.

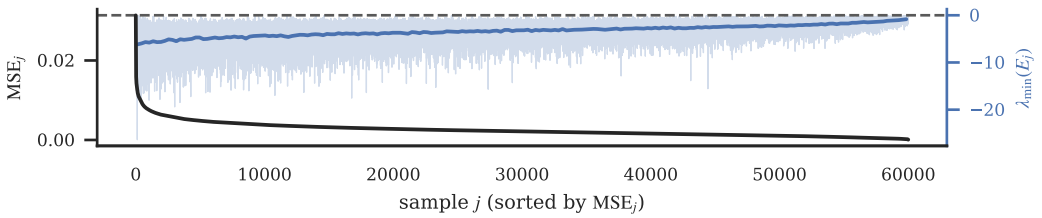


Figure 4: Per-sample reconstruction error MSE_j (bottom, black) and $\lambda_{\min}(E^*_{\boldsymbol{X}_j}(\boldsymbol{z}^*_j))$ (top, blue) on the 100 MNIST training samples with the largest error, sorted by descending MSE_j.

To isolate these theoretical levers under controlled conditions, we evaluate the synthetic Swiss-roll testbed with 5% off-shape noise, ablating latent dimension l and regularization λ (Table 4). Table 4 matches the dimension requirement of Lemma 1. The dimension of latent codes $l = 3$ gives the smallest κ and the best neighborhood preservation. Larger latent dimensions absorb off-shape noise and lower the reconstruction error, which explains why complex datasets such as CIFAR-10 require larger latent dimensions, but they also inflate κ. Increasing λ reduces κ from 302 to 87.8 at $l = 5$ without hurting accuracy, as predicted by the $\lambda \boldsymbol{I}_l$ shift in equation 19. Hence, when the intrinsic dimension is unknown, a latent dimension $l \geq m^*$ combined with a tuned L_2 regularization keeps $\mathrm{D}_1\, \xi^{\lambda}_{\boldsymbol{v}^*}$ non-degenerate and preserves the data structure.

6 Related work and discussions

Our theoretical framework provides a unifying mathematical lens to examine existing WSL paradigms in the context of HN-INR. Extended discussion of additional related literature is provided in Appendix A. Post-hoc symmetry quotienting approaches (Navon et al., 2023; Zhou et al., 2023a;b) operate on independently fitted models. From an IFT perspective, the data-to-weight correspondence can easily shatter across different local minima, since unconstrained optimization lacks Jacobian rank-regularity guarantees. Permutation-equivariant architectures essentially act as a post-hoc compensation for this optimization dispersion. Other approaches like Functa (Dupont et al., 2022) rely on a shared base model. Similarly, MWT (Gielisse & van Gemert, 2025) uses meta-learning to seek an aligned initialization, incorporating downstream classification loss directly, so that the resulting initialization is naturally suited for classification. However, these anchor-based methods rigidly restrict representations to a local neighborhood around the fixed anchor during learning. In contrast, our framework directly guarantees a non-degenerate derivative $\mathrm{D}_1\, \xi_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})$. This yields a flexible neighborhood structure, where semantically related instances smoothly cluster without rigid anchor constraints, while distinct instances remain free to separate. More broadly, many WSL tasks, including meta-learning and continual learning, also require analyzing the relationship between data and weights (Han et al., 2026). While this work employs the HN-INR as a minimal and controllable setting for rigorous theoretical analysis, our formulation can be transferred to broader architectures by designing task-specific implicit functions and verifying their derivative non-degeneracy through the IFT.

7 Conclusion

In this work, we investigate the HN-INR as a concise, effective framework for structural preservation in weight space. Within this framework, we construct the implicit function from the derivative of the INR reconstruction loss. By analyzing its derivative, we examine the non-degeneracy conditions required for the IFT to establish a correspondence between the data space and the latent weight space, avoiding representation shattering. Empirical evaluations on a synthetic Fourier-curve testbed, such as the Swiss roll, and 2D/3D classification benchmarks further support our theoretical findings. Our IFT modeling offers a potential mathematical paradigm for future research in WSL.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the writing of the paper, verifying the mathematical proofs, and aiding programming. We have not used generative AI tools for generating research ideas or designing experiments, and data generation and data annotation are not applicable to this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

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

Emilien Dupont, Hyunjik Kim, S. M. 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 Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5694–5725. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/dupont22a.html.

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

Alexander Gielisse and Jan van Gemert. End-to-end implicit neural representations for classification. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18728–18737, 2025.

Cameron Gordon, Lachlan E MacDonald, Hemanth Saratchandran, and Simon Lucey. D'oh: Decoder-only random hypernetworks for implicit neural representations. In *Proceedings of the Asian Conference on Computer Vision*, pp. 2507–2526, 2024.

Jeffrey Gu and Serena Yeung-Levy. Foundation models secretly understand neural network weights: Enhancing hypernetwork architectures with foundation models. *The Thirteenth International Conference on Learning Representations*, 2025.

Jeffrey Gu, Kuan-Chieh Wang, and Serena Yeung. Generalizable neural fields as partially observed neural processes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5330–5339, 2023.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. In *The Fifth International Conference on Learning Representations*, 2017.

Xiaolong Han, Zehong Wang, Bo Zhao, Binchi Zhang, Jundong Li, Damian Borth, Rose Yu, Haggai Maron, Yanfang Ye, Lu Yin, et al. A survey of weight space learning: Understanding, representation, and generation. *arXiv preprint arXiv:2603.10090*, 2026.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Suizhi Huang, Xingyi Yang, Hongtao Lu, and Xinchao Wang. Few-shot implicit function generation via equivariance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16262–16272, 2025.

Stanislaw Janik and Michal Byra. Weight-space mixture-of-experts for implicit neural representation classification. In *European Conference on Computer Vision*, pp. 690–706. Springer, 2026.

Jonathan Kahana, Eliahu Horwitz, Imri Shuval, and Yedid Hoshen. Deep linear probe generators for weight space learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XoYdD3m0mv.

Patrick Kidger and Terry Lyons. Universal approximation with deep narrow networks. In *Conference on learning theory*, pp. 2306–2327. PMLR, 2020.

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*, pp. 11808–11817, 2023.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Luca De Luigi, Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi di Stefano. Deep learning on implicit neural representations of shapes. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OoOIW-3uadi.

Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. Signing the unsigned: Robust surface reconstruction from raw pointsets. In *Computer Graphics Forum*, volume 29, pp. 1733–1741. Wiley Online Library, 2010.

Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *International Conference on Machine Learning*, pp. 25790–25816. PMLR, 2023.

Aviv Navon, Aviv Shamsian, Ethan Fetaya, Gal Chechik, Nadav Dym, and Haggai Maron. Equivariant deep weight space alignment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

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*, pp. 165–174, 2019.

Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32, 2019.

Konstantin Schürholt, Boris Knyazev, Xavier Giró-i Nieto, and Damian Borth. Hyper-representations as generative models: Sampling unseen neural network weights. *Advances in Neural Information Processing Systems*, 35:27906–27920, 2022.

Konstantin Schürholt, Giorgos Bouritsas, Eliahu Horwitz, Derek Lim, Yoav Gelberg, Bo Zhao, Allan Zhou, Damian Borth, and Stefanie Jegelka. Neural network weights as a new data modality. In *ICLR 2025 Workshop Proposals*, 2024. URL https://openreview.net/forum?id=Bz6wEdobY7.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in neural information processing systems*, 32, 2019.

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.

Przemysław Spurek, Sebastian Winczowski, Jacek Tabor, Maciej Zamorski, Maciej Zieba, and Tomasz Trzcinski. Hypernetwork approach to generating point clouds. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9099–9108. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/spurek20a.html.

Hermann Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024.

LeCun Yann. Mnist handwritten digit database. *ATT Labs.*, 2010.

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. *Advances in neural information processing systems*, 36:24966–24992, 2023a.

Allan Zhou, Kaien Yang, Yiding Jiang, Kaylee Burns, Winnie Xu, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Neural functional transformers. *Advances in neural information processing systems*, 36:77485–77502, 2023b.

A Extended related work

This section provides a broader survey of related work in WSL and INR, complementing the IFT-centric discussion in Section 6.

INR weight classification. Using INR weights directly for downstream tasks, such as classification, can be challenging, since the optimized INRs can vary significantly depending on their initializations, even for the same signal (Huang et al., 2025). Permutation-equivariant classifiers, such as DWSNet (Navon et al., 2023; 2024) and NFN (Zhou et al., 2023a), address these parameter symmetry issues by ensuring that models are invariant to weight reordering. Similarly, Transformer-based architectures (Zhou et al., 2023b) can also produce permutation-equivariant representations of weights for downstream tasks. More recently, MWT (Gielisse & van Gemert, 2025) employs meta-learning to optimize INR initializations end-to-end for Transformer classification, while HMoE (Janik & Byra, 2026) incorporates a Mixture-of-Experts (MoE) architecture to route weight tokens across specialized sub-networks. Although these approaches achieve strong performance, they integrate downstream task labels directly into the weight representations during optimization. While these approaches make progress in handling permutation-induced symmetry, a more fundamental challenge remains unresolved. Namely, optimizing neural networks even for the same data samples cannot be guaranteed to converge to the same set of weights. This reflects a fundamental non-uniqueness of over-parameterized INR weights that arises from the optimization process itself, beyond permutation symmetry. In practice, this often leads symmetry-aware methods to depend on more intensive sampling of equivalent weight configurations, or increased model capacity to achieve invariance.

Weight space learning and its latent representation. Functa (Dupont et al., 2022) is a pioneering work that formalizes INR weights as data. Namely, a shared base network captures common semantics, while per-sample modulation vectors encode variations. These low-dimensional modulation vectors are shown to be effective representations for downstream tasks. Schürholt et al. (2022) further learn hyper-representations via self-supervised training on the weights of INR model zoos, yielding embeddings that generalize across tasks. Similarly, inr2vec (Luigi et al., 2023) maps INR weights into latent embeddings for classification and generation, showing that meaningful semantic structure is preserved in weight space representations. Another related work, ProbeGen (Kahana et al., 2025), optimizes latent embeddings to generate probes for classifying trained INRs. These works indicate a growing interest and challenge in directly analyzing neural network weights and in discovering compact latent representations that capture their underlying semantics.

Auto-decoder and HN-INR architecture. Early work on INRs, such as SIREN (Sitzmann et al., 2020), already explores the use of hyper-networks (Ha et al., 2017), in which network weights are generated from latent embeddings conditioned on inputs. Spurek et al. (2020) use a hyper-network to generate 3D point clouds. DeepSDF (Park et al., 2019) introduces an auto-decoder architecture that learns instance-specific latent representations. Similarly, SRNs (Sitzmann et al., 2019) combine jointly trainable latent embeddings with a hyper-network to reconstruct 3D scenes and objects. More recently, D'OH (Gordon et al., 2024) proposes a decoder-only hyper-network formulation, where instance embeddings are directly optimized from random initializations, without imposing structural assumptions on the latent space. From an architectural perspective, a growing body of work adopts HN-INR formulations to improve generalization and reconstruction quality across instances and tasks (Gu et al., 2023; Gu & Yeung-Levy, 2025; Chen & Wang, 2022; Kim et al., 2023).

B PROOFS

B.1 PROOF OF LEMMA 1

This subsection provides the mathematical proof of Lemma 1, showing that the two subspace conditions guarantee the strict positive definiteness of $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})$, and that exact reconstruction along the implicit function yields a local embedding.

Proof. The proof establishes the positive definiteness of $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})$ before deriving the local embedding.

Positive definiteness of $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})$. From equation 15, under exact reconstruction $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X}) = \frac{2}{nc}\tilde{\boldsymbol{J}}^\top\tilde{\boldsymbol{J}}$, which is strictly positive definite, if and only if the composite system Jacobian $\tilde{\boldsymbol{J}} \in \mathbb{R}^{nc\times l}$ has full column rank. We prove that $\mathrm{rank}(\tilde{\boldsymbol{J}}) = l$ by contradiction. Suppose $\tilde{\boldsymbol{J}}$ is rank-deficient. Then there exists a non-zero vector $\boldsymbol{c} \in \mathbb{R}^l \setminus \{\boldsymbol{0}\}$ such that

$$\tilde{\boldsymbol{J}}\boldsymbol{c} = \mathrm{D}_1\,F(\boldsymbol{w}^*)\,(\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*)\boldsymbol{c}) = \boldsymbol{0} \in \mathbb{R}^{nc}. \tag{21}$$

Define the weight perturbation $\boldsymbol{y} \coloneqq \mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*)\boldsymbol{c} \in \mathbb{R}^d$. By Condition 1, $\mathrm{rank}(\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*)) = l$, which ensures that $\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*)$ has a trivial null space. Because $\boldsymbol{c} \neq \boldsymbol{0}$, we have $\boldsymbol{y} \neq \boldsymbol{0}$ and $\boldsymbol{y} \in \mathrm{col}(\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*))$. Simultaneously, equation 21 dictates that $\mathrm{D}_1\,F(\boldsymbol{w}^*)\boldsymbol{y} = \boldsymbol{0}$, meaning $\boldsymbol{y} \in \mathrm{null}(\mathrm{D}_1\,F(\boldsymbol{w}^*))$. Consequently, $\boldsymbol{y}$ belongs to the subspace intersection

$$\boldsymbol{y} \in \mathrm{null}(\mathrm{D}_1\,F(\boldsymbol{w}^*)) \cap \mathrm{col}(\mathrm{D}_2\,\phi(\boldsymbol{v}^*, \boldsymbol{z}^*)). \tag{22}$$

However, Condition 2 establishes that this intersection contains only the zero vector. This forces $\boldsymbol{y} = \boldsymbol{0}$, which directly contradicts $\boldsymbol{y} \neq \boldsymbol{0}$. Therefore no such non-zero vector $\boldsymbol{c}$ can exist, confirming that $\mathrm{rank}(\tilde{\boldsymbol{J}}) = l$. For any $\boldsymbol{c} \neq \boldsymbol{0}$, the quadratic form satisfies $\boldsymbol{c}^\top\,\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})\boldsymbol{c} = \frac{2}{nc}\|\tilde{\boldsymbol{J}}\boldsymbol{c}\|_2^2 > 0$. Hence $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})$ is symmetric and strictly positive definite, with $\sigma_{\min}(\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})) = \lambda_{\min}(\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})) > 0$ and $\det(\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})) \neq 0$.

Local embedding. Since f and ϕ are C^∞, so is $\xi_{\boldsymbol{v}^*}$. At $(\boldsymbol{z}^*, \boldsymbol{X})$, we have $\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X}) = \boldsymbol{0}$, and $\mathrm{D}_1\,\xi_{\boldsymbol{v}^*}(\boldsymbol{z}^*, \boldsymbol{X})$ has full rank by the first part. The IFT therefore provides open neighborhoods $U(\boldsymbol{X}) \subset \mathcal{X}$ and $V(\boldsymbol{z}^*) \subset \mathcal{Z}$ and a unique C^∞ mapping $g\colon U(\boldsymbol{X}) \to V(\boldsymbol{z}^*)$ satisfying equation 12, which completes (a).

Let $h\colon \mathcal{Z} \to \mathbb{R}^{nc}$, $h(\boldsymbol{z}) \coloneqq F(\phi(\boldsymbol{v}^*, \boldsymbol{z}))$, denote the forward evaluation map. Exact reconstruction along g states that $h(g(\boldsymbol{X}')) = \boldsymbol{X}'$ for all $\boldsymbol{X}' \in U(\boldsymbol{X})$, so h is a left inverse of g on $U(\boldsymbol{X})$. Three consequences follow. First, g is injective. Second, the inverse of g on its image $g(U(\boldsymbol{X}))$ is the restriction of the continuous map h, so g is a homeomorphism onto its image. Third, differentiating $h(g(\boldsymbol{X}')) = \boldsymbol{X}'$ along the data manifold gives

$$\mathrm{D}\,h(g(\boldsymbol{X}'))\,\mathrm{D}\,g(\boldsymbol{X}')\boldsymbol{u} = \boldsymbol{u}, \qquad \forall \boldsymbol{u} \in T_{\boldsymbol{X}'}\mathcal{X}, \tag{23}$$

so $\mathrm{D}\,g(\boldsymbol{X}')$ is injective on $T_{\boldsymbol{X}'}\mathcal{X}$ and g is a smooth immersion. A smooth immersion that is a homeomorphism onto its image is a smooth embedding, and its image $g(U(\boldsymbol{X}))$ is an embedded submanifold of $\mathcal{Z}$ of dimension m^*. Finally, the injective linear map $\mathrm{D}\,g(\boldsymbol{X})$ from the m^*-dimensional space $T_{\boldsymbol{X}}\mathcal{X}$ into $\mathbb{R}^l$ forces $m^* \leq l$, which completes (b). □

Discussion of Condition 2. The first part of the proof also shows that, at a given $(\boldsymbol{z}^*, \boldsymbol{X})$, Conditions 1 and 2 hold together if and only if $\operatorname{rank}(\tilde{\boldsymbol{J}}) = l$, since a failure of either condition yields a non-zero $\boldsymbol{c}$ with $\tilde{\boldsymbol{J}}\boldsymbol{c} = \boldsymbol{0}$. Condition 2 holds automatically when $\mathrm{D}_1 F(\boldsymbol{w}^*)$ has full column rank d, which requires $nc \geq d$. For over-parameterized INRs with $d > nc$, $\operatorname{null}(\mathrm{D}_1 F(\boldsymbol{w}^*))$ has dimension at least $d - nc$, and Condition 2 becomes a genuine requirement on how the hyper-network places the l-dimensional subspace $\operatorname{col}(\mathrm{D}_2 \phi(\boldsymbol{v}^*, \boldsymbol{z}^*))$. In practice, we monitor the non-degeneracy of $\mathrm{D}_1 \xi^{\lambda}_{\boldsymbol{v}^*}(\boldsymbol{z}, \boldsymbol{X})$ directly (Section 5.3) instead of checking the two conditions separately.

B.2 Proof of Proposition 1

This subsection proves Proposition 1, showing that non-degeneracy along a branch and bounded latent codes extend the local implicit functions of the IFT to a single smooth map on each class. We write $\bar{B}_R(\boldsymbol{0}) \coloneqq \{\boldsymbol{z} \mid \|\boldsymbol{z}\|_2 \leq R\}$ for the ball in the bounded-latent-code assumption. If $\mathcal{Z}$ is a proper open subset of $\mathbb{R}^l$, the radius R is taken such that $\bar{B}_R(\boldsymbol{0}) \subset \mathcal{Z}$, which holds for every R when $\mathcal{Z} = \mathbb{R}^l$.

Proof. Fix a class k and let $\pi_k \colon \mathcal{S}^*_k \to \mathcal{X}_k$, $(\boldsymbol{z}, \boldsymbol{X}) \mapsto \boldsymbol{X}$, denote the projection. We show that π_k is a bijection whose inverse $\boldsymbol{X} \mapsto (g_k(\boldsymbol{X}), \boldsymbol{X})$ is smooth. The proof proceeds in three steps.

Local graph structure. Let $(\boldsymbol{z}_0, \boldsymbol{X}_0) \in \mathcal{S}^*_k$. The map $\xi^{\lambda}_{\boldsymbol{v}^*}$ is affine in $\boldsymbol{X}$, so its defining formula gives a C^∞ map for every sample array $\boldsymbol{X}$, not only for $\boldsymbol{X} \in \mathcal{X}_k$. Since $\mathrm{D}_1 \xi^{\lambda}_{\boldsymbol{v}^*}(\boldsymbol{z}_0, \boldsymbol{X}_0)$ has full rank by the non-degeneracy assumption, the IFT in the ambient space of sample arrays provides open neighborhoods $\tilde{U}$ of $\boldsymbol{X}_0$ and $V \subset \mathcal{Z}$ of $\boldsymbol{z}_0$, together with a C^∞ map $\tilde{g} \colon \tilde{U} \to V$ whose graph is the zero set of $\xi^{\lambda}_{\boldsymbol{v}^*}$ in $V \times \tilde{U}$. Let $U \subset \tilde{U} \cap \mathcal{X}_k$ be a connected open neighborhood of $\boldsymbol{X}_0$ in $\mathcal{X}_k$ and $g_U \coloneqq \tilde{g}|_U$. Then $\mathcal{S}_k \cap (V \times U)$ is the graph of g_U. This graph is connected and contains $(\boldsymbol{z}_0, \boldsymbol{X}_0)$, so it lies in $\mathcal{S}^*_k$ and equals $\mathcal{S}^*_k \cap (V \times U)$. Hence π_k is a local homeomorphism with smooth local inverses $\boldsymbol{X} \mapsto (g_U(\boldsymbol{X}), \boldsymbol{X})$.

Properness. The critical set $\mathcal{S}_k$ is closed in $\mathcal{Z} \times \mathcal{X}_k$ as a zero set, and so is its connected component $\mathcal{S}^*_k$. For a compact set $A \subset \mathcal{X}_k$, the bounded-latent-code assumption gives $\pi_k^{-1}(A) \subset \bar{B}_R(\boldsymbol{0}) \times A$, which is a compact subset of $\mathcal{Z} \times \mathcal{X}_k$. As a closed subset of this compact set, $\pi_k^{-1}(A)$ is compact, so π_k is proper.

Global inversion. The image $\pi_k(\mathcal{S}^*_k)$ is open in $\mathcal{X}_k$ because π_k is a local homeomorphism, and closed because a proper map into a manifold is closed. Since $\mathcal{X}_k$ is connected, π_k is surjective. A proper surjective local homeomorphism onto a connected manifold is a covering map. Because $\mathcal{S}^*_k$ is connected and $\mathcal{X}_k$ is simply connected, this covering has a single sheet, so π_k is a bijection. Every $\boldsymbol{X} \in \mathcal{X}_k$ thus has exactly one $\boldsymbol{z}$ with $(\boldsymbol{z}, \boldsymbol{X}) \in \mathcal{S}^*_k$, and setting $g_k(\boldsymbol{X}) \coloneqq \mathrm{pr}_{\mathcal{Z}}(\pi_k^{-1}(\boldsymbol{X}))$ gives $\mathcal{S}^*_k = \{(g_k(\boldsymbol{X}), \boldsymbol{X}) \mid \boldsymbol{X} \in \mathcal{X}_k\}$. Any map with this graph equals g_k, so g_k is unique. By Step 1, g_k coincides with the smooth map g_U on each neighborhood U, so g_k is C^∞. Finally, $g_k(\mathcal{X}_k)$ is connected as the continuous image of the connected set $\mathcal{X}_k$. □

The argument uses neither compactness of $\mathcal{X}_k$ nor the absence of a boundary. Properness follows from the bounded-latent-code assumption alone, and Step 1 also holds at boundary points of $\mathcal{X}_k$ because the IFT is applied in the ambient space of sample arrays. The proposition therefore also covers classes with a boundary, such as the Swiss-roll segments of Section 5.1.

C Architectures and hyper-parameters

Bracketed integer lists in Table 5 denote the sequence of hidden layer widths for multilayer perceptrons, such that $[256]^2$ indicates two consecutive hidden layers with 256 units each, while dashes denote omitted or inapplicable components. All neural hyper-networks employ a single dense projection trunk without separate prediction heads. The synthetic Swiss roll serves as a controlled baseline implemented with a purely linear hyper-network and unregularized latent codes ($\lambda_z = 0$), directly verifying Condition 1 in the absence of nonlinear optimization artifacts.

The implementation penalizes the mean squared latent entry, so the reported weight λ_z differs by a factor of $2/l$ from the weight λ of the penalty $\frac{\lambda}{2}\|\boldsymbol{z}\|_2^2$ in Section 3.3. For a mini-batch of sample

Table 5: Architectures and training settings.

Parameter	Swiss roll	MNIST	F-MNIST	CIFAR10	CIFAR10-aug	ModelNet40	ShapeNet10	ScanNet10
HNet	linear	$[256]^2$	$[384]^2$	$[512]^2$	$[512]^2$	$[256]^2$	$[256]^2$	$[256]^2$
MNet	$[32]^3$	$[64]^3$	$[64]^3$	$[192]^3$	$[192]^3$	[64, 128, 64]	$[128]^3$	$[128]^3$
dim($\boldsymbol{z}$)	3	20	32	192	256	32	24	24
HNet LR	$1e^{-3}$	$1e^{-4}$	$1e^{-4}$	$1e^{-5}$	$5e^{-5}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$
$\boldsymbol{z}$ LR	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$
batch size	4096	2048	2048	2048	2048	512	512	512
small cls	$[32]^2$	$[64]^3$	$[64]^3$	$[64]^3$	$[64]^3$	$[64]^3$	$[64]^3$	$[64]^3$
big cls	–	$[1024]^3$	$[1024]^3$	$[1024]^3$	$[1024]^3$	$[256]^3$	$[256]^3$	$[256]^3$
λ_z	0	0.04	0.03	1.0	10.0	0.05	0.04	0.001
$\lambda = 2\lambda_z/l$	0	4.00e−3	1.88e−3	1.04e−2	7.81e−2	3.13e−3	3.33e−3	8.33e−5

indices $\mathcal{I}$, the training objective reads

$$\frac{1}{|\mathcal{I}|}\sum_{j\in\mathcal{I}}\Big(\ell(\boldsymbol{v},\boldsymbol{z}_j,\boldsymbol{X}_j)+\frac{\lambda_z}{l}\|\boldsymbol{z}_j\|_2^2\Big), \tag{24}$$

where $\frac{1}{l}\|\boldsymbol{z}_j\|_2^2$ is the mean of the l squared entries of $\boldsymbol{z}_j$. Each latent code $\boldsymbol{z}_j$ enters only the j-th summand, so at the converged hyper-network weights $\boldsymbol{v}^*$ its critical point condition, after dropping the common factor $1/|\mathcal{I}|$, is

$$\xi_{\boldsymbol{v}^*}(\boldsymbol{z}_j,\boldsymbol{X}_j)+\frac{2\lambda_z}{l}\boldsymbol{z}_j=\boldsymbol{0}. \tag{25}$$

This is the regularized condition $\xi^{\lambda}_{\boldsymbol{v}^*}(\boldsymbol{z}_j,\boldsymbol{X}_j)=\boldsymbol{0}$ of Section 3.3 with

$$\lambda=\frac{2\lambda_z}{l}. \tag{26}$$

Differentiating once more with respect to $\boldsymbol{z}_j$ gives the shift $\frac{2\lambda_z}{l}\boldsymbol{I}_l=\lambda\boldsymbol{I}_l$ of equation 18, so the condition numbers κ in Section 5.3 are those of $\mathrm{D}_1\,\xi^{\lambda}_{\boldsymbol{v}^*}(\boldsymbol{z}_j,\boldsymbol{X}_j)$ in the convention of the main text. Table 5 lists both λ_z and the equivalent λ, and the ablation in Table 4 reports $\lambda\in\{0,0.02,0.04,0.08\}$, which corresponds to $\lambda_z\in\{0,0.05,0.10,0.20\}$ at $l=5$.