OrbitLoRA: Learning Rotation-Aware Low-Data Adaptation of Vision Foundation Models

Yunlu Chen*, **Dominik Engel, Peter Wonka, and Ivan Viola**

King Abdullah University of Science and Technology (KAUST)

Abstract

Pretrained vision transformers provide strong semantic representations, but their token embeddings are not designed to transform predictably under image rotations. This limits their direct adaptation to domains such as satellite imagery, histopathology, microscopy, and other scientific imaging settings, where in-plane orientation is often arbitrary and no canonical pose exists. Existing equivariant adaptation methods for pretrained models typically rely on frame averaging or canonicalization, which can be effective on synthetically rotated natural images but do not teach the adapted model an internal rotation-aware representation. We introduce OrbitLoRA, a lightweight orbit-aware adapter for frozen visual foundation models. OrbitLoRA combines standard LoRA with a typed residual branch that maintains scalar and vector fields over ViT patch tokens. The typed branch is initialized from local steerable image features and harmonic coordinate fields, updated at selected transformer depths, and injected into the frozen token stream through zero-initialized residual gates. During finetuning, paired rotated views supervise the typed state through a transport-and-rotation consistency loss, while the semantic output is trained to remain invariant for the downstream task. This design preserves the rich features of pretrained ViTs while adding a learnable rotation-symmetry prior. Across classification and segmentation tasks, and across DINOv2-, CLIP-, SAM-, and LLaVA-style backbones, OrbitLoRA matches strong LoRA-with-augmentation baselines in high-data regimes and substantially improves adaptation in low-data regimes, where learning the symmetry prior is most valuable.

1 Introduction

Visual foundation models based on transformers have become a standard substrate for image recognition. Models such as ViT [11], CLIP [26], DINOv2 [23], and SAM [18] learn rich semantic representations from large-scale pretraining and can be adapted to many downstream tasks with only a small number of trainable parameters [15]. Their success, however, raises a practical question about geometric inductive biases. Vision transformers operate on discrete patch tokens with learned or fixed positional encodings, and self-attention is not tied to a continuous transformation law of the image plane. Under in-plane rotations, patches do not generally map to patches on the same grid, local orientations are not represented by steerable channels, and absolute positional encodings can change in ways unrelated to the underlying object. As a result, rotation symmetry is not enforced by the architecture and must instead be learned from data. This is unreliable under natural-image pretraining, where data contain strong upright, viewpoint, or photographer-induced pose biases.

This becomes problematic when adapting visual foundation models to scientific and geospatial domains. In satellite imagery, aerial mapping, histopathology, microscopy, cryo-electron microscopy, and other image-based measurement settings, the physical orientation of a pattern is often arbitrary. A rotated tissue structure, cell, material grain, or land-cover pattern should usually induce a predictable

*Correspondence: yunluchenxyz@gmail.com.

transformation of intermediate spatial features and, for classification, an invariant prediction. At the same time, these domains frequently operate in low-data regimes: expert annotations are expensive, acquisition is specialized, and the number of labeled examples is far smaller than in natural-image pretraining. A useful adaptation method should therefore preserve the semantic strength of pretrained transformers while adding the rotation inductive bias that the downstream domain actually requires.

A growing line of work seeks to make pretrained models equivariant without redesigning the entire architecture. EquiAdapt [21] uses canonicalization to map inputs into a preferred frame before applying a pretrained model. λ-Equitune [2] learns to weight features obtained from transformed inputs, generalizing group averaging. FoCal [29] performs test-time canonicalization by searching for a more typical view under foundation-model priors. These approaches are attractive because they treat the pretrained model as a black box. However, for rotation-equivariant scientific domains, canonicalization can be ambiguous: there may be no single correct upright orientation. Frame averaging and transformed-view ensembling can also be computationally costly and may not teach the internal representation itself to obey the symmetry and learn the inductive prior that benifits low-data regime. In practice, these methods are often evaluated on synthetically rotated natural images, where the original data distribution still contains a canonical pose. This setting differs from domains in which orientation is intrinsically arbitrary.

Another line of work builds rotation equivariance directly into transformer architectures, for example through group-equivariant positional encodings, harmonic self-attention, or equivariant patch embeddings [36, 17, 6]. These models provide a principled route to equivariant representation learning, but they require redesigning the backbone rather than adapting an existing pretrained ViT. Moreover, exact architectural equivariance in transformers remains difficult to scale: even recent equivariant ViTs improve rotation stability but still underperform simpler equivariant CNN backbones on general tasks [36, 6]. Replacing the backbone therefore forfeits much of the benefit of large-scale pretrained representations, especially when downstream labels are scarce, and pretraining a new foundation model architecture for each scientific domain is rarely realistic. This creates a practical gap: we want the data efficiency and semantic transfer of pretrained ViTs, while still encouraging the adapted model to internalize rotation symmetry during finetuning.

We introduce **OrbitLoRA**, a lightweight rotation-aware adapter for pretrained transformers. OrbitLoRA combines standard LoRA with a typed residual branch that maintains order-0 scalar fields and order-1 vector fields on the ViT patch grid. The typed state is initialized from local equivariant features produced by a small steerable stem and harmonic coordinate fields, providing a smooth rotation-aware reference without assuming a canonical pose. At selected reinjection stages, the typed state is updated from backbone tokens and modulates LoRA-adapted attention, coupling rotation information directly to the representation used by the pretrained model.

OrbitLoRA is not designed to enforce exact equivariance. Instead, it learns a *soft, representation-level rotation prior* that improves rotation consistency while preserving downstream accuracy. Paired rotated views provide transport-based supervision for the typed fields and invariance regularization for the global semantic feature. This design allows the adapter to reshape attention and token interactions in a rotation-aware manner, yielding stronger low-data performance than baselines that enforce invariance mechanically, such as canonicalization and frame averaging adapters.

To encapsulate our contributions:

- We study adaptation of pretrained foundation models in real rotation-symmetric domains without canonical orientation, highlighting the limitations of exact equivariance imposed externally.
- We introduce a typed residual branch for frozen ViTs, combining scalar and order-1 fields with standard LoRA adaptation.
- We propose a transport-and-rotation consistency objective on paired rotated views, enabling the adapter to learn a rotation-aware inductive prior.
- We demonstrate that OrbitLoRA improves the accuracy–consistency trade-off, particularly in low-data scientific imaging regimes, while remaining compatible with multiple pretrained transformer backbones and downstream tasks.

2 Related Work

Equivariance and rotation-aware vision. Equivariance provides an inductive bias that makes representations transform predictably under known symmetry groups [4]. It has motivated group-equivariant CNNs, steerable and harmonic networks, spherical CNNs, tensor field networks, and $E(n)$-equivariant graph models [9, 10, 35, 8, 30, 33, 28]. For images, rotation equivariance is often enforced through group convolutions, typed feature fields, or orientation channels [9, 10, 38, 33]. These models offer strong geometric guarantees, but typically require symmetry-specific architectures rather than direct adaptation of large pretrained backbones. Recent work also shows that non-equivariant models can learn partial equivariance from data, while explicit equivariance remains useful for data efficiency and matched-compute settings [13, 3].

Architectural rotation equivariance in transformers. Several recent works explore building rotation-equivariant representations directly into transformer architectures, including GEVIT [36], Equi-ViT [17], and HarmFormer [6], using group-equivariant positional encodings, harmonic attention, or equivariant patch embeddings. While principled, architectural equivariance in transformers remains challenging: these models are typically evaluated on low-resolution or synthetic datasets, and their generalization and task performance often lag behind simpler equivariant CNN backbones [17]. Consequently, scaling these methods to high-resolution scientific images or leveraging pretrained ViT backbones remains impractical, motivating lightweight adapter-based approaches such as OrbitLoRA.

Equivariant adaptation of pretrained models. Pretrained ViTs and CLIP-style encoders are powerful transfer learners [12, 27], but their patchification, positional encodings, and attention layers do not naturally respect continuous rotations. Prior work either redesigns transformers for equivariance [36] or adapts existing backbones through model-agnostic mechanisms such as frame averaging, transformed-view feature averaging, or canonicalization with foundation-model priors [25, 16, 22, 2, 24, 29]. These methods preserve pretrained models, but usually impose symmetry externally by selecting, predicting, or averaging over views.

Rotation symmetry is central in scientific domains such as remote sensing, histopathology, and microscopy, where a canonical pose may not exist and labeled data are limited [31, 1, 7]. OrbitLoRA targets this setting by combining frozen pretrained transformers and LoRA with a lightweight typed branch inspired by equivariant field representations [10, 35, 33, 30]. Unlike canonicalization or frame-averaging methods, OrbitLoRA learns a rotation-aware internal state inside the model via a typed branch conditioned on pretrained visual transformer features, using this inductive bias to improve downstream performance on rotation-symmetric data, especially in low-data regimes.

3 Method

Let f_Θ denote a pretrained ViT-style visual encoder, instantiated in our experiments as DINOv2 ViT-L [23]. We freeze the pretrained parameters Θ and attach standard low-rank adaptation layers [15] to the transformer projections. While LoRA enables parameter-efficient task adaptation, it does not prescribe how patch tokens should transform under rotations. Our objective is therefore to learn not only rotation-robust predictions, but also an internal rotation-aware state that guides the frozen token stream during finetuning.

Overview. OrbitLoRA augments a frozen transformer with two coupled adaptation paths: a standard LoRA branch and a lightweight typed branch. The LoRA branch adapts the pretrained token stream through low-rank updates, whereas the typed branch maintains fields with explicit rotation transformation laws. An overview of our method is shown in Figure 1. Let Ω denote the ViT patch lattice. At each typed stage, the branch stores

$$q_0(u) \in \mathbb{R}^{C_0}, \qquad q_1(u) \in \mathbb{R}^{C_1 \times 2}, \qquad u \in \Omega. \tag{1}$$

Here q_0 is a C_0-channel order-0 field, whose channels are scalars invariant to in-plane rotations up to spatial transport. The field q_1 is a C_1-channel order-1 field, whose channels are two-dimensional vectors transforming under the standard representation of $\mathrm{SO}(2)$. For a rotation g_θ, the desired field action is

$$[\mathcal{T}_\theta q_0](u) = q_0(R_{-\theta}u), \qquad [\mathcal{T}_\theta q_1](u) = \rho_1(\theta) q_1(R_{-\theta}u), \tag{2}$$

where R_θ is the planar rotation matrix and $\rho_1(\theta) = R_\theta$ acts on the two coordinates of each order-1 vector channel.

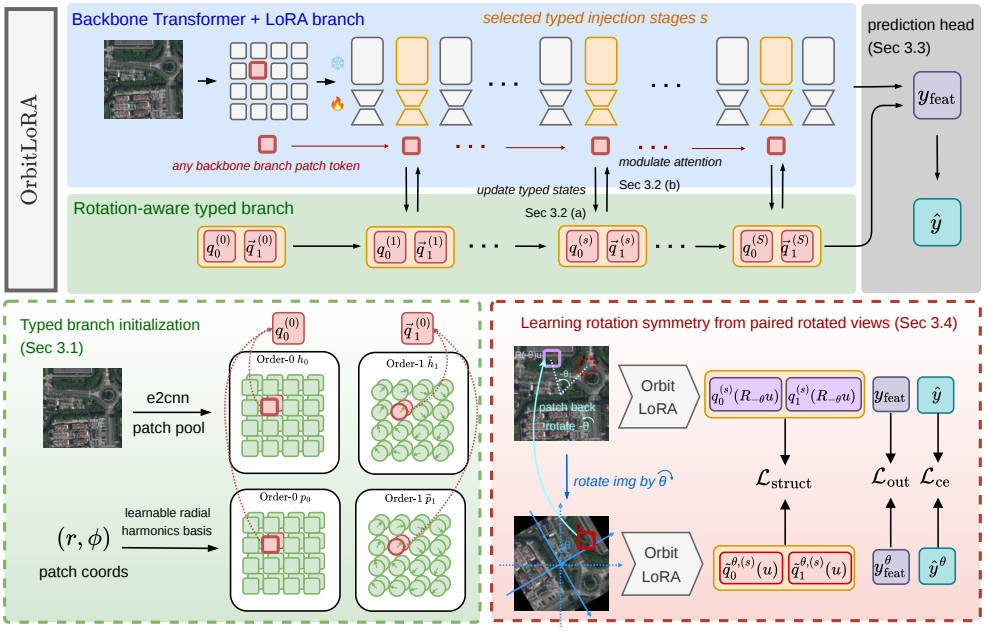

Figure 1: Overview of OrbitLoRA. A frozen ViT backbone is adapted with a standard LoRA branch and a rotation-aware typed branch. The typed state is initialized from `e2cnn` content fields and harmonic coordinate fields, then updated and reinjected into the backbone at selected stages to modulate the LoRA-adapted attention variables. The prediction head combines the final token stream with the updated scalar typed field. Training uses paired rotated views: patches in the rotated image are matched to the original image by inverse rotation, and consistency losses enforce equivariant typed fields and invariant global semantic features. As a result, the typed injection modules learn to guide the backbone toward rotation-robust adaptation.

The typed branch interacts with the transformer at a small set of stages. At each such stage, it modulates the query, key, and value tensors used by attention after the frozen backbone and LoRA contributions are combined. Consequently, the typed state can directly affect attention affinities through Q and K, and transported content through V, while all pretrained weights remain frozen. The modulation is residual and zero-initialized, so the model initially matches the frozen pretrained backbone equipped with ordinary LoRA.

3.1 Rotation-aware typed initialization: content fields and harmonic coordinate fields

The initial typed state combines local image content with a learnable geometric coordinate basis. For the content source, we use a three-layer $E(2)$-equivariant CNN implemented with `e2cnn` [34], retaining representations up to order 1. Given an image x, the stem produces patch-pooled order-0 and order-1 content fields on the ViT patch grid:

$$(H_0, H_1) = \texttt{e2cnn}(x), \qquad H_0 \in \mathbb{R}^{|\Omega| \times C_0}, \quad H_1 \in \mathbb{R}^{|\Omega| \times C_1 \times 2}. \tag{3}$$

For a patch $u \in \Omega$, we denote the corresponding local content features by $h_0(u) \in \mathbb{R}^{C_0}$ and $h_1(u) \in \mathbb{R}^{C_1 \times 2}$. This stem provides local equivariant evidence from the image. It is intentionally small and does not replace the pretrained transformer; instead, it supplies the adaptation module with features whose transformation laws are known.

The second source is a learnable radial harmonic coordinate basis. For a patch location $\xi = (r, \varphi)$ in polar coordinates relative to the image center, we define

$$p_0(\xi) = a_0(r), \qquad p_1(\xi) = a_1(r) \otimes \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix}, \tag{4}$$

where $a_0(r) \in \mathbb{R}^{C_0}$ and $a_1(r) \in \mathbb{R}^{C_1}$ are produced by small radial MLPs, and $\otimes$ denotes channel-wise multiplication by the two-dimensional unit direction. The scalar basis p_0 encodes distance to the

image center, while the order-1 basis p_1 provides a local orientation reference without choosing a global canonical pose. Under a rotated view, the basis is sampled at inverse-rotated source locations, and each order-1 vector is rotated forward by $\rho_1(\theta)$, matching Eq. 2.

The initial typed state is

$$q_0^{(0)} = \alpha_{\mathrm{cnt}} h_0 + \alpha_{\mathrm{pos}} p_0, \qquad q_1^{(0)} = \beta_{\mathrm{cnt}} h_1 + \beta_{\mathrm{pos}} p_1, \tag{5}$$

where $\alpha_{\mathrm{cnt}}, \beta_{\mathrm{cnt}}$ gate the content fields and $\alpha_{\mathrm{pos}}, \beta_{\mathrm{pos}}$ gate the coordinate fields. The coordinate gates are initialized zero, and the content gates are initialized to a small value 0.05, allowing the typed state to be used from the start of training without imposing a strong geometric bias before observing the downstream task.

3.2 Coupling the LoRA-adapted backbone with the rotation-aware typed branch

Let the transformer contain L blocks. OrbitLoRA inserts typed interactions at an ordered set of selected blocks

$$\mathcal{B} = (b_1, b_2, \ldots, b_S), \qquad 0 \le b_1 < b_2 < \cdots < b_S < L.$$

We choose evenly distributed stages $\mathcal{B} = (0, 3, 6, 9)$ for a 12-block ViT and $\mathcal{B} = (1, 7, 13, 19)$ for a 24-block ViT. We index the corresponding typed-interaction stages by $s \in \{1, \ldots, S\}$. Let $Z_s \in \mathbb{R}^{|\Omega| \times d}$ denote the patch-token input to the selected block at stage s. For a patch $u \in \Omega$, let $z_s(u) \in \mathbb{R}^d$ be its token representation; we omit u when it is clear from context. Let c_s denote the corresponding class token.

For each projection W in the frozen transformer, LoRA applies the standard low-rank update

$$W \quad \mapsto \quad W + \Delta W, \qquad \Delta W = BA, \qquad \mathrm{rank}(\Delta W) \ll d. \tag{6}$$

This branch provides parameter-efficient semantic adaptation while preserving the pretrained representation.

(a) Backbone tokens update typed states. At each typed stage s, the current patch tokens update the rotation-aware typed state. A small two-layer MLP g_s reads normalized patch tokens and predicts token-wise at patch $u \in \Omega$:

$$\left(\alpha_{\mathrm{cnt}}^{(s)}, \beta_{\mathrm{cnt}}^{(s)}, \alpha_{\mathrm{pos}}^{(s)}, \beta_{\mathrm{pos}}^{(s)}\right) = g_s(z_s). \tag{7}$$

The typed state is then updated from stage $s-1$ to stage s by injecting gated content and coordinate fields:

$$q_0^{(s)} = q_0^{(s-1)} + \alpha_{\mathrm{cnt}}^{(s)} \odot h_0 + \alpha_{\mathrm{pos}}^{(s)} \odot p_0, \tag{8}$$

$$q_1^{(s)} = q_1^{(s-1)} + \beta_{\mathrm{cnt}}^{(s)} \odot h_1 + \beta_{\mathrm{pos}}^{(s)} \odot p_1. \tag{9}$$

Here $\odot$ denotes token-wise channel gating, with scalar gates applied to order-0 channels and vector-channel gates shared across the two coordinates of each order-1 channel. This update allows the backbone tokens to selectively inject equivariant content and geometric information while preserving the prescribed transformation laws of the typed state.

(b) Typed branch modulates backbone attention. The updated typed state then modulates the attention computation at stage s. For each of the $P \in \{Q, K, V\}$, which combines the frozen backbone contribution and the LoRA update, OrbitLoRA applies a typed modulation to their patch-token rows. The modulation is conditioned on rotation-invariant information from the typed state together with the local patch token:$[z_s, q_0^{(s)}, \|q_1^{(s)}\|]$, where $\|q_1^{(s)}\| \in \mathbb{R}^{C_1}$ contains the per-channel magnitudes of the order-1 vectors for rotation-invariant feature. For each $P \in \{Q, K, V\}$ at local patch u, the typed branch applies a lightweight FiLM modulation to the LoRA-adapted projection from mixed conditioning source:

$$P_s(u) \leftarrow \mathrm{FiLM}\left(P_s(u) \mid [z_s, q_0^{(s)}, \|q_1^{(s)}\|]\right), \qquad P \in \{Q, K, V\}. \tag{10}$$

Here P_s denotes the projection after combining the frozen backbone and LoRA contributions. The FiLM layer applies token-wise affine modulation to the patch-token rows of P_s, while leaving the class-token row unchanged. It is initialized as the identity map, so the attention computation initially

matches ordinary LoRA. The resulting Q_s, K_s, V_s are used in the standard multi-head attention computation, $\mathrm{softmax}(Q_s K_s^\top / \sqrt{d_h}) V_s$.

This interface differs from generic token-level modulation. Because the typed branch acts after the frozen and LoRA QKV contributions have been combined, it conditions the actual attention variables used by the transformer. Intuitively, modulating Q and K allows the typed state to influence attention affinities, and hence which patches attend to one another under a rotated configuration. Modulating V allows the typed state to influence the content transported by those affinities. The pretrained weights remain fixed, and the FiLM heads are zero-initialized so that OrbitLoRA starts exactly from the frozen backbone with ordinary LoRA. Rotation-aware attention modulation is therefore learned only when supported by the downstream data.

For transformer blocks not included in $\mathcal{B}$, the model uses the ordinary LoRA-adapted attention and MLP computations without typed modulation. Thus, the typed branch provides sparse geometric interventions rather than replacing the pretrained transformer computation.

3.3 Learning equivariance from paired rotated views

During training, we sample a rotation angle θ and form a label-preserving rotated view x^θ. The original and rotated images are passed through the same model, producing typed states and semantic features

$$\left(\{q_0^{(s)}, q_1^{(s)}\}_{s=1}^S, y_{\mathrm{feat}}, \hat{y}\right) \quad \text{and} \quad \left(\{q_0^{\theta,(s)}, q_1^{\theta,(s)}\}_{s=1}^S, y_{\mathrm{feat}}^\theta, \hat{y}^\theta\right). \tag{11}$$

The supervised task loss is applied to both views:

$$\mathcal{L}_{\mathrm{ce}} = \frac{1}{2}\left[\mathrm{CE}(\hat{y}, y) + \mathrm{CE}(\hat{y}^\theta, y)\right]. \tag{12}$$

The central geometric supervision is applied to the typed branch at every typed interaction stage. To compare the original typed state with the rotated-view typed state, we transport the original field onto the rotated patch grid. For each target patch $u \in \Omega$ in the rotated view, we compute the corresponding source coordinate

$$\xi_\theta(u) = R_{-\theta} u.$$

If $\xi_\theta(u)$ lies inside the image domain, we bilinearly sample the original field at $\xi_\theta(u)$; otherwise, the patch is marked invalid by a mask and excluded from loss terms. The transported fields are therefore

$$\widetilde{q}_0^{\theta,(s)}(u) = q_0^{(s)}(\xi_\theta(u)), \qquad \widetilde{q}_1^{\theta,(s)}(u) = \rho_1(\theta) q_1^{(s)}(\xi_\theta(u)), \tag{13}$$

where sampling at non-grid coordinates is implemented by bilinear interpolation. Thus $\mathcal{T}_\theta$ includes both spatial resampling and representation rotation.

We enforce equivariance with a masked cosine loss between the transported field and the field produced directly from the rotated image:

$$\mathcal{L}_{\mathrm{struct}} = \frac{1}{2S} \sum_{s=1}^{S} \sum_{\tau \in \{0,1\}} \left[1 - \cos\left(q_\tau^{\theta,(s)}, \widetilde{q}_\tau^{\theta,(s)}\right)\right]. \tag{14}$$

The cosine similarity is computed over valid transported patches only, using the mask M_θ to exclude locations whose inverse-rotated source coordinates fall outside the image. For q_0, the similarity is computed over scalar channels; for q_1, it is computed over the flattened vector-channel components. This loss does not ask the model to discover a canonical orientation. Instead, it teaches the auxiliary typed branch to produce fields whose transformations under rotation are predictable.

Finally, we regularize the semantic representation to be invariant under the label-preserving rotation:

$$\mathcal{L}_{\mathrm{out}} = 1 - \cos\left(y_{\mathrm{feat}}, y_{\mathrm{feat}}^\theta\right). \tag{15}$$

The full training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{struct}} \mathcal{L}_{\mathrm{struct}} + \lambda_{\mathrm{out}} \mathcal{L}_{\mathrm{out}}. \tag{16}$$

Table 1: Low-data DINOv2 ViT-L adaptation on EuroSAT. Each entry reports standard test accuracy / C_{24} rotation-averaged accuracy across training-data regimes. For LoRA, we report results both without and with rotation augmentation; all other baselines use rotation augmentation.

Method	**0.5% (n=108)**	**1% (n=216)**	**5% (n=1080)**	**10% (n=2160)**	**100% (n=21600)**
DINOv2 linear probe	0.612±.392 / 0.621±.274	0.632±.311 / 0.658±.214	0.803 / 0.816	0.844 / 0.846	0.914 / 0.915
LoRA (without aug)	0.795±.064 / 0.732±.048	0.882±.020 / 0.835±.013	0.964 / 0.906	0.969 / 0.917	0.986 / 0.958
LoRA (with rotation aug)	0.851±.028 / 0.861±.030	0.889±.024 / 0.889±.017	0.962 / 0.963	0.967 / 0.973	0.984 / 0.986
LoRA + aug + consist. loss	0.877±.013 / 0.874±.014	0.908±.011 / 0.906±.009	0.964 / 0.963	0.971 / 0.972	0.983 / 0.981
LoRA + λ-EquiTune	0.864±.017 / 0.870±.011	0.919±.012 / 0.912±.008	0.968 / 0.968	0.974 / **0.977**	**0.987** / **0.987**
LoRA + EquiAdapt	0.867±.018 / 0.871±.012	0.896±.017 / 0.903±.010	0.964 / 0.964	0.968 / 0.972	0.982 / 0.984
LoRA + FoCal	0.864±.014 / 0.864±.009	0.882±.014 / 0.880±.008	0.964 / 0.965	0.975 / 0.974	0.981 / 0.978
OrbitLoRA (ours)	**0.909**±.007 / **0.902**±.011	**0.937**±.012 / **0.936**±.008	**0.970** / **0.969**	**0.977** / **0.977**	0.984 / 0.982

Table 2: DINOv2 ViT-L adaptation on intrinsically orientation-ambiguous medical images. PathMNIST and DermaMNIST are evaluated under low-data and full-data regimes. Each entry reports validation accuracy / C_{24} rotation-averaged accuracy.

Method	**PathMNIST**			**DermaMNIST**		
	n=100	**n=200**	**Full (n = 89996)**	**n=100**	**n=200**	**Full (n=7007)**
Linear probe	0.5778±.870 / 0.3977±1.682	0.5735±.722 / 0.3578±2.356	0.9188 / 0.8589	0.6421±.242 / 0.6155±.227	0.6471±.213 / 0.6124±.240	0.6899 / 0.6774
LoRA	0.7310±.068 / 0.7385±.048	0.8149±.052 / 0.7562±.037	0.9929 / 0.9936	0.6710±.075 / 0.6773±.054	0.6700±.066 / 0.6680±.047	0.7896 / 0.7610
+ EquiAdapt	0.7836±.029 / 0.7776±.020	0.8519±.023 / 0.8013±.016	0.9937 / 0.9924	0.6959±.039 / 0.6780±.028	0.6929±.034 / 0.6753±.024	0.7976 / 0.7546
+ λ-EquiTune	0.7864±.027 / 0.7888±.019	0.8596±.021 / 0.8398±.015	**0.9946** / **0.9944**	0.6929±.037 / 0.6846±.026	0.6909±.032 / 0.6751±.023	**0.8096** / 0.7748
+ FoCal	0.7450±.046 / 0.7535±.032	0.8311±.036 / 0.7772±.025	0.9920 / 0.9919	0.6742±.056 / 0.6760±.040	0.6721±.049 / 0.6692±.035	0.7910 / 0.7622
OrbitLoRA	**0.7960**±.030 / **0.8111**±.021	**0.8716**±.024 / **0.8485**±.017	0.9917 / 0.9942	**0.7178**±.041 / **0.6918**±.029	**0.7129**±.035 / **0.7095**±.025	0.7976 / **0.7900**

What differs from canonicalization and frame averaging. Prior adaptation methods often make a pretrained model rotation-aware by evaluating transformed inputs, averaging or weighting features across transformed views, or mapping the input to a canonical orientation [21, 2, 29]. These strategies are attractive because they avoid redesigning the backbone. However, they rely on an external transformation procedure at the image or feature level. In domains without a natural canonical frame, canonicalization can be ambiguous, and averaging over many frames can be computationally expensive or may suppress orientation-dependent evidence.

OrbitLoRA instead learns an internal typed state that is coupled to the pretrained attention stream. The frozen transformer remains a semantic expert; standard LoRA provides low-rank task adaptation; and the typed branch supplies a learned rotation-aware state that is explicitly trained to transport according to $\mathrm{SO}(2)$. Rather than only correcting the final representation, the typed branch modulates the actual QKV tensors used by attention after the frozen and LoRA contributions have been combined. This design lets geometric information influence both attention correspondence and value aggregation while preserving the pretrained model at initialization. Because the mechanism only assumes access to token features and linear projections, it can be attached not only to DINOv2, but also to other ViT-like vision, segmentation, or vision-language foundation models.

4 Experiments

Backbones and baselines. We evaluate OrbitLoRA on four pretrained foundation-model families: DINOv2 ViT-L [23], CLIP ViT-B [27], LLaVA-1.5-7B [20], and SAM 3 [5]. For classification, adapters are applied to the visual encoder; for LLaVA, classification is formulated as multiple-choice VQA; for segmentation, adapters are inserted into the SAM 3 image encoder. We compare with vanilla LoRA [15], EquiAdapt [21], λ-EquiTune [2], and FoCal [29]. These baselines cover standard PEFT, learned canonicalization, learned frame averaging, and test-time canonicalization.

Datasets. We evaluate image classification on EuroSAT [14], Osteosarcoma Tumor Assessment [19], PathMNIST, and DermaMNIST [37], and semantic segmentation on LoveDA [32]. EuroSAT contains Sentinel-2 land-use images; we use a stratified 80/10/10 split, giving 21,600 training images. PathMNIST and DermaMNIST use the official MedMNIST splits with 89,996 and 7,007 training images, respectively. LoveDA contains high-resolution remote-sensing images of seven semantic classes with 2,522 training images. We additionally report cross-model results on a high-resolution osteosarcoma pathology dataset in Appendix G. These datasets span remote sensing, histopathology, dermatology, and dense land-cover prediction, where in-plane orientation is not tied to a unique canonical pose.

Table 3: Rotation-aware adaptation of CLIP ViT-B/16 on EuroSAT. Each entry reports validation accuracy / C_{24} rotation-averaged accuracy. The zero-shot row gives the frozen text-prompt baseline, while adapted rows compare LoRA and OrbitLoRA under the same data regimes.

Method	Aug.	Data Regime			
		0.5% (n=108)	**1% (n=216)**	**10%(n=2160)**	**100% (n=21600)**
CLIP-B/16 zero-shot text		0.4235 / 0.3040			
LoRA	×	0.5604$\pm$.098 / 0.5336$\pm$.069	0.6689$\pm$.074 / 0.6588$\pm$.053	0.8267 / 0.8179	0.9472 / **0.9401**
	✓	0.5802$\pm$.090 / 0.5543$\pm$.063	0.6837$\pm$.067 / 0.6714$\pm$.047	0.8357 / 0.8336	0.9448 / 0.9467
OrbitLoRA	–	**0.6128**$\pm$.055 / **0.5964**$\pm$.039	**0.7006**$\pm$.044 / **0.6904**$\pm$.031	**0.8535 / 0.8499**	**0.9531 / 0.9501**

Metrics. For classification, we report top-1 accuracy; for segmentation, we report mIoU. We also report C_{24} rotation-averaged metrics over $\{0^\circ, 15^\circ, \ldots, 345^\circ\}$. We introduce additional metrics to evaluate rotation robustness in the Appendix B.

4.1 Results

Data-efficient adaptation in orientation-ambiguous domains. Tables 1 and 2 evaluate DINOv2 ViT-L adaptation on EuroSAT, PathMNIST, and DermaMNIST. These domains provide a natural testbed for rotation-aware adaptation because remote-sensing images, tissue patches, and lesion images generally do not have a unique semantic upright orientation. Across these datasets, OrbitLoRA is most effective when supervision is scarce. On EuroSAT, it outperforms LoRA, λ-EquiTune, EquiAdapt, and FoCal at 0.5% and 1% training data in both standard accuracy and C_{24} rotation-averaged accuracy. The matched consistency baseline in Table 1 helps attribute these gains: at 0.5% data, rotation augmentation improves LoRA from 0.795 to 0.851, adding the paired-view consistency objective further improves accuracy to 0.877, and OrbitLoRA reaches 0.909. At 1% data, the corresponding accuracies are 0.882, 0.889, 0.908, and 0.937 for LoRA without augmentation, LoRA with rotation augmentation, LoRA with augmentation and consistency, and OrbitLoRA, respectively. Thus, paired-view regularization contributes to the improvement, while the typed rotation-aware branch provides an additional gain in the low-data regime. Consistent cross-entropy on both rotated views is already used by the augmented LoRA baseline because both views share the same class label; it is therefore not treated as a separate controlled component. The same trend holds on MedMNIST: OrbitLoRA gives the strongest low-data results, with particularly consistent gains in the rotation-averaged metric.

We further evaluate cross-model transfer on the high-resolution Osteosarcoma Tumor Assessment pathology dataset [19] using DINOv2, CLIP, and LLaVA, in both 50-image and 750-image training regimes. OrbitLoRA consistently improves over augmented LoRA, with the largest gains in the 50-image regime; detailed results are reported in Appendix G, Table 10.

These results support the central role of the typed rotation branch as a data-efficient inductive prior. Rotation augmentation improves robustness for most methods, but it does not remove OrbitLoRA's advantage; in the lowest-data regimes, OrbitLoRA remains stronger than augmented LoRA and the canonicalization-based baselines. This suggests that the gain does not come only from observing more rotated samples, but from internalizing rotation structure through the typed state and its attention modulation. As the amount of labeled data increases, the pretrained representation approaches saturation and the gap naturally narrows. Nevertheless, OrbitLoRA remains competitive or best in most full-data settings, indicating that the proposed geometric adaptation improves low-data robustness without sacrificing canonical-view performance.

Generality across foundation-model families. Tables 3 and 4 test whether the same adaptation principle transfers beyond a pure visual encoder. For CLIP ViT-B/16, OrbitLoRA improves over vanilla LoRA in nearly all data regimes, with the largest gains at 0.5% and 1% training data. This indicates that the typed branch remains useful when the visual encoder is pretrained through image–text contrastive supervision rather than self-supervised visual learning.

For LLaVA-1.5-7B, we cast classification as multiple-choice visual question answering: the image is paired with a class-choice question, and the model predicts the answer option corresponding to the class. OrbitLoRA improves over LoRA in both standard and rotation-averaged accuracy across the low- and medium-data regimes. This result shows that the proposed adaptation is not tied to a linear

Table 4: Rotation-aware adaptation of LLaVA-1.5-7B on EuroSAT with a multiple-choice VQA interface. Each entry reports validation accuracy / C_{24} rotation-averaged accuracy. This setting tests whether OrbitLoRA transfers from visual encoders to multimodal language models.

Method	**Aug.**	**Data Regime**			
		0.5% (n=108)	**1% (n=216)**	**10%(n=2160)**	**100% (n=21600)**
llava zero-shot	–	0.261 / 0.290			
LoRA	×	0.840 / 0.700	0.881 / 0.763	0.953 / 0.862	**0.974** / 0.910
	✓	0.848 / 0.829	0.904 / 0.889	0.949 / 0.939	0.961 / **0.965**
OrbitLoRA	–	**0.875 / 0.853**	**0.922 / 0.903**	**0.957 / 0.944**	0.959 / 0.964

classification head; it can also improve the visual representations consumed by a language-model decoder in a VQA-style interface.

Extension to dense semantic segmentation prediction. Table 9 in the appendix evaluates SAM3 adaptation on LoveDA semantic segmentation. We report standard validation mIoU and the C_{24} rotation-averaged mIoU. With 10% data, OrbitLoRA gives a clear improvement over LoRA and all rotation-adaptation baselines, both with and without rotation augmentation. In the full-data setting, the absolute differences are smaller because the task is closer to saturation, but OrbitLoRA still obtains the best rotation-averaged mIoU. These results show that the typed attention modulation is applicable not only to image-level classification, but also to dense prediction, where rotation-consistent spatial correspondence is directly relevant.

Ablation experiments are presented in the Appendix F to validate our method design choices.

5 Discussion

OrbitLoRA is designed to learn a rotation-aware inductive prior inside the adapter rather than to impose exact equivariance externally. This distinction is important when adapting pretrained foundation models to rotation-symmetric domains. Canonicalization methods such as EquiAdapt and FoCal enforce invariance by learning or selecting a reference frame, which can be effective when a stable upright orientation exists. However, in remote sensing, histopathology, dermatology, and dense land-cover prediction, there is often no unique semantic canonical pose. Selecting one frame may therefore discard valid orientation-dependent evidence or introduce errors when the canonicalizer is trained from limited data. Frame-averaging methods such as λ-EquiTune avoid committing to one canonical frame, but they enforce consistency by aggregating transformed views, which can increase inference cost and dilute discriminative orientation-dependent features.

OrbitLoRA follows a softer representation-level approach. It learns order-0 and order-1 typed states from paired rotated views and uses these states to modulate the attention variables of the frozen backbone after the LoRA update. This lets the adapter organize pretrained features according to rotation structure while preserving the backbone's discriminative capacity. The consistent gains in the low-data regimes suggest that this mechanism provides a useful geometric prior for transfer: fewer labeled examples are needed to learn how task-relevant evidence should transform under rotation. The results across DINOv2, CLIP, LLaVA, and SAM 3 further indicate that this prior is not specific to a single architecture or output format, but applies across visual encoders, vision-language models, multimodal VQA, and dense segmentation.

Limitations. OrbitLoRA is not strictly equivariant. Unlike canonicalization or group-averaging methods, it can exhibit a small nonzero equivariance defect. Thus, if the application requires exact mathematical equivariance by construction, hard-coded equivariant mechanisms may be preferable. However, our experiments show that exact rotation consistency alone is not sufficient for low-data transfer performance. The hard-coded equivariant baselines can achieve perfect consistency under the diagnostic metric by construction, yet they do not consistently achieve the best task accuracy in the low-data regimes. OrbitLoRA substantially improves rotation consistency over vanilla LoRA while also achieving stronger standard and rotation-averaged task performance. This suggests that, for foundation-model adaptation, learning a useful rotation-aware prior inside the representation can be



Figure 2: We provide qualitative segmentation results on the LoveDA validation set. Ordinary LoRA (d, f) is trained with rotation augmentation and the percentage in (c-f) refers to the amount of training samples used. Labels: building road water barren forest agriculture ignore

more effective than enforcing invariant predictions externally. We analyze this accuracy–consistency trade-off in detail in Appendix B.

6 Conclusion

We presented OrbitLoRA, a rotation-aware adapter that learns a soft geometric prior inside pretrained foundation models. Across classification, VQA, and segmentation, OrbitLoRA improves low-data transfer and rotation robustness, showing that learned rotation priors can be more effective than externally imposed invariance.

Acknowledgement

This research was supported by King Abdullah University of Science and Technology (KAUST) (BAS/1/1680-01-01) and CISCO - KAUST AI research center.

References

[1] Saghir Alfasly, Abubakr Shafique, Peyman Nejat, Jibran Khan, Areej Alsaafin, Ghazal Alabtah, and H. R. Tizhoosh. Rotation-agnostic image representation learning for digital pathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[2] Sourya Basu, Pulkit Katdare, Prasanna Sattigeri, Vijil Chenthamarakshan, Katherine Driggs-Campbell, Payel Das, and Lav R. Varshney. Efficient equivariant transfer learning from pretrained models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[3] Johann Brehmer, Sönke Behrends, Pim De Haan, and Taco Cohen. Does equivariance matter at scale? *Transactions on Machine Learning Research*, 2025.

[4] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[5] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[6] Fuyao Chen, Yuexi Du, Elèonore V Lieffrig, Nicha C Dvornek, and John A Onofrey. Equi-vit: Rotational equivariant vision transformer for robust histopathology analysis. *arXiv preprint arXiv:2601.09130*, 2026.

[7] Benjamin Chidester, Tianming Zhou, Minh N. Do, and Jian Ma. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14):i530–i537, 2019.

[8] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

[9] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2990–2999, 2016.

[10] Taco S. Cohen and Max Welling. Steerable CNNs. In *International Conference on Learning Representations*, 2017.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[13] Nate Gruver, Marc Anton Finzi, Micah Goldblum, and Andrew Gordon Wilson. The lie derivative for measuring learned equivariance. In *International Conference on Learning Representations*, 2023.

[14] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[15] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[16] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *Proceedings of the 40th International Conference on Machine Learning*, pages 15546–15566, 2023.

[17] Tomáš Karella, Adam Harmanec, Jan Kotera, Jan Blažek, and Filip Šroubek. Harmformer: Harmonic networks meet transformers for continuous roto-translation equivariance. *arXiv preprint arXiv:2411.03794*, 2024.

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[19] Patrick Leavey, Anita Sengupta, Dinesh Rakheja, Ovidiu Daescu, Harish Babu Arunachalam, and Rashika Mishra. Osteosarcoma data from UT Southwestern/UT Dallas for viable and necrotic tumor assessment (Osteosarcoma-Tumor-Assessment), 2019. Dataset.

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[21] Arnab Kumar Mondal, Siba Smarak Panigrahi, Oumar Kaba, Sai Rajeswar Mudumba, and Siamak Ravanbakhsh. Equivariant adaptation of large pretrained models. In *Advances in Neural Information Processing Systems*, volume 36, pages 50293–50309, 2023.

[22] Arnab Kumar Mondal, Siba Smarak Panigrahi, Sékou-Oumar Kaba, Sai Rajeswar Mudumba, and Siamak Ravanbakhsh. Equivariant adaptation of large pretrained models. In *Advances in Neural Information Processing Systems*, volume 36, pages 50293–50309, 2023.

[23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[24] Siba Smarak Panigrahi and Arnab Kumar Mondal. Improved canonicalization for model agnostic equivariance. In *CVPR 2024 Workshop on Equivariant Vision: From Theory to Practice*, 2024.

[25] Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021.

[28] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9323–9332, 2021.

[29] Utkarsh Singhal, Ryan Feng, Stella X. Yu, and Atul Prakash. Test-time canonicalization by foundation models for robust perception. In *International Conference on Machine Learning*, 2025. arXiv:2507.10375.

[30] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[31] Di Wang, Qiming Zhang, Yufei Xu, Jing Zhang, Bo Du, Dacheng Tao, and Liangpei Zhang. Advancing plain vision transformer towards remote sensing foundation model. *arXiv preprint arXiv:2208.03987*, 2022.

[32] Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. *arXiv preprint arXiv:2110.08733*, 2021.

[33] Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] Maurice Weiler and Gabriele Cesa. General $E(2)$-equivariant steerable CNNs. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[35] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[36] Renjun Xu, Kaifan Yang, Ke Liu, and Fengxiang He. E(2)-equivariant vision transformer. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 2356–2366. PMLR, 2023.

[37] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific data*, 10(1):41, 2023.

[38] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

A Broader Impacts

This work aims to improve the adaptation of visual foundation models in domains where image orientation is inherently arbitrary, such as satellite imagery, microscopy, and histopathology. By encouraging internal rotation-aware representations while keeping the pretrained backbone frozen, OrbitLoRA may reduce the amount of labeled data and computation required to deploy strong vision models in scientific and medical imaging settings. Potential benefits include more robust analysis under acquisition variability and improved accessibility of foundation-model adaptation for resource-constrained research groups. At the same time, the method inherits the risks of the underlying pretrained model and downstream data, including dataset bias, spurious correlations, and possible overconfidence in high-stakes applications. In medical or scientific use, OrbitLoRA should therefore be evaluated under domain-specific validation protocols and used as an assistive component rather than as a substitute for expert judgment. More broadly, rotation robustness does not guarantee semantic reliability, fairness, or safety, and we encourage careful auditing before deployment in operational settings such as clinical decision support, environmental monitoring, or geospatial surveillance.

B Evaluating Rotation Consistency: Discussion on Imposed Hard-Coded Equivariance vs. Learned Rotation Priors

Beyond task performance, we evaluate two complementary measures of rotational invariance. The first is the **equivariance defect** δ, defined as the average squared change in the pre-softmax logits under group transformations:

$$\delta_G(f) = \mathbb{E}_{\mathbf{x}} \left[\frac{1}{|G|} \sum_{g \in G} \left\| f(g \cdot \mathbf{x}) - f(\mathbf{x}) \right\|^2 \right], \tag{17}$$

where $G = C_{24}$ denotes rotations by $0°, 15°, 30°, \ldots, 345°$, and $f(\mathbf{x})$ is the logit vector. A perfectly invariant classifier has $\delta_G(f) = 0$.

The second metric is **prediction consistency** ρ, which measures invariance at the decision level:

$$\rho = \Pr_{g \neq e} \left[\arg\max f(g \cdot \mathbf{x}) = \arg\max f(\mathbf{x}) \right].$$

Thus, $\rho = 1$ indicates that the predicted label is unchanged for all non-identity rotations. The two metrics capture different failure modes: δ detects continuous logit drift even when the top-1 prediction is stable, whereas ρ captures discrete label flips.

Table 5 evaluates rotation consistency using equivariance defect and prediction consistency under C_{24}. This diagnostic is useful for checking whether predictions remain stable under discrete rotations, but it should not be read as a complete measure of whether a method learns a useful symmetry prior. Several baselines are structurally designed to satisfy this exact criterion: λ-EquiTune and EquiAdapt use a C_{24}-GCNN in their adaptation pipeline, while FoCal enforces invariance through test-time canonicalization. Their zero defect and perfect consistency are therefore expected by construction. **These methods solve the measured invariance constraint mechanically; the more important question is whether the imposed equivariance improves transfer performance.**

The answer from the low-data results is mixed for the hard-coded equivariant adapters. Although canonicalization and frame-averaging methods achieve perfect C_{24} consistency, they do not obtain the strongest accuracy in the regimes where an inductive prior should matter most. This suggests that enforcing invariance at the output or inference level is not equivalent to learning a transferable rotation-aware representation. A canonicalizer can make predictions invariant by selecting a frame, and a group-averaging method can make predictions invariant by pooling over frames, but neither mechanism necessarily teaches the pretrained visual features how task evidence should transform. **Hard-coded equivariance can guarantee consistency, but it can still be a weak inductive prior for low-data adaptation if it is not integrated into the representation used by the foundation model.**

OrbitLoRA is designed for this transfer setting. It does not impose exact equivariance and therefore has a small nonzero defect. However, relative to vanilla LoRA, it sharply reduces equivariance

Table 5: DINOv2 ViT-L rotation consistency on EuroSAT across training-data regimes. Each entry reports **equivariance defect $\delta_{C_{24}}$ / prediction consistency** $\rho_{C_{24}}$. Lower defect and higher consistency are better. Hard-coded equivariant baselines achieve perfect C_{24} consistency by construction, whereas OrbitLoRA is not exactly equivariant but substantially improves consistency over LoRA while achieving stronger low-data transfer performance in Table 1.

Method	Aug.	Data Regime				
		0.5% (n=108)	**1% (n=216)**	**5% (n=1080)**	**10% (n=2160)**	**100% (n=21600)**
DINOv2 linear probe	×	0.1681 / 0.7872	0.1717 / 0.7860	0.1113 / 0.8917	0.0970 / 0.9070	0.0637 / 0.9483
	✓	0.1632 / 0.7663	0.1699 / 0.7540	0.1181 / 0.8553	0.1056 / 0.8803	0.0694 / 0.9290
LoRA	×	0.1172 / 0.8753	0.0724 / 0.9247	0.0229 / 0.9808	0.0195 / 0.9882	0.0164 / 0.9908
	✓	0.0394 / 0.9338	0.0342 / 0.9430	0.0168 / 0.9848	0.0148 / 0.9890	0.0093 / 0.9940
LoRA + λ-EquiTune	×	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**
	✓	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**
LoRA + EquiAdapt	×	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**
	✓	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**
LoRA + FoCal	×	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**
	✓	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**	**0.0000 / 1.0000**
OrbitLoRA (ours)	×	0.0035 / 0.9812	0.0030 / 0.9858	0.0021 / 0.9893	0.0023 / 0.9930	0.0019 / 0.9937
	✓	0.0033 / 0.9805	0.0026 / 0.9867	0.0022 / 0.9878	0.0018 / 0.9897	0.0014 / 0.9948

defect and improves prediction consistency across data regimes, showing that the adapter does learn rotation structure. At the same time, OrbitLoRA achieves the best standard and rotation-averaged task performance in the low-data settings. **This is the key advantage: OrbitLoRA converts rotation symmetry into a useful representation-level inductive prior, rather than only enforcing invariant predictions by construction.**

This distinction is especially important for foundation-model adaptation. The pretrained backbone already contains rich visual features, but those features are not necessarily organized for rotation-symmetric downstream domains such as remote sensing or medical imaging. A hard-coded equivariant wrapper can regularize the final prediction, but it may leave the internal adaptation problem mostly unchanged or even suppress useful orientation-dependent evidence. OrbitLoRA instead injects the rotation prior into the attention modulation itself through the typed state. This allows the adapter to reshape how the pretrained representation is used while preserving the backbone's discriminative capacity.

Thus, OrbitLoRA's nonzero defect should not be interpreted as a disadvantage relative to perfectly invariant baselines. The relevant comparison is the accuracy–consistency trade-off. OrbitLoRA is far more rotation-consistent than LoRA, but unlike hard-coded equivariant adapters, this consistency comes with stronger low-data task performance. **The results indicate that learning a soft rotation-aware prior inside the adapter is more effective for low-data transfer than imposing exact equivariance outside the representation.**

C Representation-consistency properties of the typed branch

This appendix formalizes the representation-consistency properties of the OrbitLoRA typed branch. These results concern the auxiliary typed path only. They should not be interpreted as a claim that the complete frozen ViT encoder is exactly rotation-equivariant. The pretrained ViT token stream is learned from natural images and is not assumed to obey an exact $\mathrm{SO}(2)$ transformation law. OrbitLoRA instead introduces typed fields with prescribed transformation rules and uses paired rotated views to encourage the interaction between these fields and the frozen token stream to follow the desired transported behavior.

C.1 Field action and discretization

Let Ω denote the ViT patch lattice. For an in-plane rotation by angle θ, let R_θ be the planar rotation matrix. We define the order-0 and order-1 representations as

$$\rho_0(\theta) = I, \qquad \rho_1(\theta) = R_\theta.$$

For an order-τ field q_τ, with $\tau \in \{0, 1\}$, the intended field action is

$$[\mathcal{T}_\theta q_\tau](u) = \rho_\tau(\theta) q_\tau(R_{-\theta} u). \tag{18}$$

For $\tau = 0$, this is spatial transport of scalar channels. For $\tau = 1$, this combines spatial transport with rotation of the two vector coordinates in each channel.

In a continuous domain, Eq. equation 18 defines an exact $\mathrm{SO}(2)$ action. In the implementation, fields are sampled on the discrete ViT patch lattice, and off-grid source locations are evaluated by bilinear interpolation. Therefore the realized transformation is subject to discretization and interpolation error. All statements below should be read as representation-consistency statements for the typed branch under this field action.

C.2 Consistency of the harmonic coordinate fields

OrbitLoRA uses a learnable radial harmonic coordinate basis

$$p_0(\xi) = a_0(r), \qquad p_1(\xi) = a_1(r) \otimes \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix},$$

where $\xi = (r, \varphi)$ denotes the polar coordinate of a patch location relative to the image center, and $a_0(r)$ and $a_1(r)$ are radial functions.

Lemma 1 (Consistency of the harmonic coordinate basis)**.** *The coordinate fields satisfy*

$$\mathcal{T}_\theta p_0 = p_0, \qquad \mathcal{T}_\theta p_1 = p_1.$$

Thus the harmonic coordinate basis is consistent with the field action in Eq. equation 18.

Proof. For the scalar field, rotation preserves radius. Hence

$$[\mathcal{T}_\theta p_0](u) = p_0(R_{-\theta} u) = a_0(r) = p_0(u).$$

For the order-1 field, let $u = (r, \varphi)$. Then $R_{-\theta} u$ has polar coordinates $(r, \varphi - \theta)$, and

$$p_1(R_{-\theta} u) = a_1(r) \otimes \begin{bmatrix} \cos(\varphi - \theta) \\ \sin(\varphi - \theta) \end{bmatrix}.$$

Applying the order-1 representation gives

$$\begin{aligned} [\mathcal{T}_\theta p_1](u) &= R_\theta p_1(R_{-\theta} u) \\ &= a_1(r) \otimes R_\theta \begin{bmatrix} \cos(\varphi - \theta) \\ \sin(\varphi - \theta) \end{bmatrix} \\ &= a_1(r) \otimes \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix} = p_1(u). \end{aligned}$$

This proves the claim. □

C.3 Closure of the typed update

At typed stage s, the update for an order-τ field can be written as

$$q_\tau^{(s)} = q_\tau^{(s-1)} + \gamma_{\tau,\mathrm{cnt}}^{(s)} \odot h_\tau + \gamma_{\tau,\mathrm{pos}}^{(s)} \odot p_\tau, \tag{19}$$

where h_τ is the equivariant content field, p_τ is the harmonic coordinate field, and $\gamma_{\tau,\mathrm{cnt}}^{(s)}$ and $\gamma_{\tau,\mathrm{pos}}^{(s)}$ are scalar gates. For $\tau = 0$, these gates correspond to $\alpha_{\mathrm{cnt}}^{(s)}$ and $\alpha_{\mathrm{pos}}^{(s)}$. For $\tau = 1$, they correspond to $\beta_{\mathrm{cnt}}^{(s)}$ and $\beta_{\mathrm{pos}}^{(s)}$, and each gate is shared across the two coordinates of the corresponding vector channel.

Proposition 1 (Closure of the typed update under transported scalar gates)**.** *Assume that the content and coordinate fields transform according to*

$$h_\tau^\theta = \mathcal{T}_\theta h_\tau, \qquad p_\tau^\theta = \mathcal{T}_\theta p_\tau, \qquad \tau \in \{0, 1\}.$$

Assume also that the gates are transported scalar fields:

$$\gamma_{\tau,\mathrm{cnt}}^{\theta,(s)}(u) = \gamma_{\tau,\mathrm{cnt}}^{(s)}(R_{-\theta} u), \qquad \gamma_{\tau,\mathrm{pos}}^{\theta,(s)}(u) = \gamma_{\tau,\mathrm{pos}}^{(s)}(R_{-\theta} u).$$

If

$$q_\tau^{\theta,(s-1)} = \mathcal{T}_\theta q_\tau^{(s-1)},$$

then

$$q_\tau^{\theta,(s)} = \mathcal{T}_\theta q_\tau^{(s)}.$$

Therefore, under these sufficient conditions, the typed update preserves the prescribed rotation law at each typed stage.

Proof. For the rotated input, the typed update is

$$q_\tau^{\theta,(s)}(u) = q_\tau^{\theta,(s-1)}(u) + \gamma_{\tau,\mathrm{cnt}}^{\theta,(s)}(u) \odot h_\tau^\theta(u) + \gamma_{\tau,\mathrm{pos}}^{\theta,(s)}(u) \odot p_\tau^\theta(u).$$

Using the induction assumption, the transformation laws of h_τ and p_τ, and the scalar transport of the gates, we obtain

$$\begin{aligned} q_\tau^{\theta,(s)}(u) = \rho_\tau(\theta) q_\tau^{(s-1)}(R_{-\theta}u) + \gamma_{\tau,\mathrm{cnt}}^{(s)}(R_{-\theta}u) \odot \rho_\tau(\theta) h_\tau(R_{-\theta}u) \\ + \gamma_{\tau,\mathrm{pos}}^{(s)}(R_{-\theta}u) \odot \rho_\tau(\theta) p_\tau(R_{-\theta}u). \end{aligned}$$

The gates are scalar channels and therefore commute with $\rho_\tau(\theta)$. For $\tau = 0$ this is immediate, and for $\tau = 1$ each gate multiplies both vector coordinates of the corresponding order-1 channel equally. Thus

$$\begin{aligned} q_\tau^{\theta,(s)}(u) &= \rho_\tau(\theta) \Big[q_\tau^{(s-1)}(R_{-\theta}u) + \gamma_{\tau,\mathrm{cnt}}^{(s)}(R_{-\theta}u) \odot h_\tau(R_{-\theta}u) \\ &\qquad + \gamma_{\tau,\mathrm{pos}}^{(s)}(R_{-\theta}u) \odot p_\tau(R_{-\theta}u) \Big] \\ &= \rho_\tau(\theta) q_\tau^{(s)}(R_{-\theta}u) \\ &= [\mathcal{T}_\theta q_\tau^{(s)}](u). \end{aligned}$$

This proves closure for one typed stage. Repeating the argument over stages gives the result. □

Interpretation. Proposition 1 gives a sufficient condition for the typed branch to preserve its intended rotation law. In OrbitLoRA, the content fields are produced by an equivariant stem, and the coordinate fields satisfy Lemma 1. The remaining condition concerns the gates. Because the gates are predicted from pretrained ViT tokens, exact scalar transport is not assumed. Instead, the structural loss encourages the learned gates and typed states to follow this transported behavior on paired rotated views.

C.4 Invariant typed conditioning for FiLM modulation

OrbitLoRA modulates the LoRA-adapted attention variables using FiLM coefficients conditioned on

$$[z_s(u), q_0^{(s)}(u), \|q_1^{(s)}(u)\|].$$

The token $z_s(u)$ comes from the pretrained ViT and is not assumed to be exactly equivariant. The typed component of the conditioning, however, is invariant to the internal rotation of order-1 vector coordinates up to spatial transport.

Lemma 2 (Invariant typed conditioning)**.** *If $q_1^\theta = \mathcal{T}_\theta q_1$, then for each patch u,*

$$\|q_1^\theta(u)\| = \|q_1(R_{-\theta}u)\|.$$

Consequently, the typed conditioning vector

$$[q_0(u), \|q_1(u)\|]$$

transforms as a transported scalar field.

Proof. For each order-1 channel c,

$$q_{1,c}^\theta(u) = R_\theta q_{1,c}(R_{-\theta}u).$$

Since R_θ is orthogonal,

$$\|q_{1,c}^\theta(u)\|_2 = \|R_\theta q_{1,c}(R_{-\theta}u)\|_2 = \|q_{1,c}(R_{-\theta}u)\|_2.$$

Thus the magnitude of each order-1 channel is invariant to the vector-coordinate rotation and changes only by spatial transport. The order-0 field q_0 is also a transported scalar field. Therefore $[q_0, \|q_1\|]$ is transported as scalar typed information. □

Consequence. Lemma 2 justifies feeding $q_0^{(s)}$ and $\|q_1^{(s)}\|$ into scalar FiLM heads. The modulation head does not consume raw order-1 coordinates; it consumes their magnitudes, which are invariant to the internal vector-coordinate rotation. The full conditioning also includes the ViT token z_s, whose rotation behavior is learned and regularized rather than assumed.

C.5 Recovery of ordinary LoRA at initialization

OrbitLoRA is initialized to recover ordinary LoRA. This ensures that the typed branch does not perturb the pretrained representation before learning from downstream data.

Proposition 2 (LoRA recovery at initialization)**.** *Suppose that the FiLM modulation is parameterized residually as*

$$\mathrm{FiLM}(P_s(u) \mid c_s(u)) = (1 + \delta a_s(u)) \odot P_s(u) + \delta b_s(u),$$

where $P \in \{Q, K, V\}$ and $c_s(u)$ is the conditioning vector. If

$$\delta a_s(u) = 0, \qquad \delta b_s(u) = 0$$

at initialization for all typed stages and all attention projections, then the typed modulation initially leaves the LoRA-adapted attention variables unchanged. If the typed contribution to the prediction head is also zero-initialized, then OrbitLoRA computes the same function as ordinary LoRA at initialization.

Proof. At initialization, $\delta a_s(u) = 0$ and $\delta b_s(u) = 0$. Therefore

$$\mathrm{FiLM}(P_s(u) \mid c_s(u)) = P_s(u)$$

for every patch token and every projection $P \in \{Q, K, V\}$. Thus the attention variables are exactly those obtained from the frozen backbone plus the ordinary LoRA updates. Blocks without typed interactions already perform ordinary LoRA computations. If the typed contribution to the prediction head is zero-initialized, the head also does not initially depend on the typed state. Hence the complete model recovers ordinary LoRA at initialization. □

C.6 Implications and limitations

The above results show that the typed branch is representation-consistent under the prescribed order-0 and order-1 field actions when its inputs and scalar gates transform appropriately. They do not imply exact equivariance of the full OrbitLoRA model. The pretrained ViT tokens, the token-conditioned gates, and the final attention computation are not constrained to be exactly equivariant. OrbitLoRA therefore uses paired rotated views to learn a soft rotation-aware prior: the structural loss encourages typed fields to follow the transported field action, while the output loss encourages the global semantic feature to be invariant. This is the intended regime of the method: improved accuracy–consistency trade-off for foundation-model adaptation, rather than a guarantee of zero equivariance defect.

D Implementation Details

All task-performance experiments are implemented in PyTorch and optimized with AdamW using learning rate 10^{-4} and batch size 32. Unless otherwise stated, we use LoRA rank 16 for all adapted models. Experiments were conducted on a server with $4\times$ NVIDIA A100-PCIE 40GB GPUs, dual Intel Xeon Gold 6226R CPUs, and 1TB system RAM. The separate efficiency comparison in Appendix E uses batch size 16 for all methods to ensure a hardware-matched comparison.

For the DINOv2 ViT-L experiments, images are resized to 224×224. The ViT patch size is 14, producing a 16×16 grid of 256 patch tokens with token dimension 1024. The OrbitLoRA typed branch maintains an order-0 invariant state of dimension 128 and an order-1 equivariant state of dimension 32. The semantic feature dimension used by the classification head is 256. The typed state is reinjected into the backbone at transformer blocks 1, 7, 13, and 19, corresponding to early, mid, and late depths of the network.

The image-conditioned typed branch is implemented with an `escnn/e2cnn` steerable convolutional stem. It takes the input image and applies three equivariant convolutional stages with strides 2, 2, and 2, followed by group normalization and norm nonlinearities. The resulting feature map is adaptively

pooled to the ViT patch grid, yielding image-conditioned order-0 and order-1 typed features. The content branch is initialized with small gates, so that its initial contribution to the typed state is weak rather than dominant.

In addition to the image-conditioned typed features, OrbitLoRA uses a fixed coordinate harmonic basis defined on the 16×16 patch grid. Patch coordinates are normalized to $[-1, 1]^2$ and scaled by $1/\sqrt{2}$. The invariant positional component depends only on the radius r, while the order-1 component is represented by sine–cosine pairs parameterized by the polar angle. Thus the order-0 positional features are invariant by construction, and the order-1 positional features transform equivariantly under in-plane rotation. Positional gates are initialized to zero, so the model initially relies only on the weak image-conditioned typed state and learns to use the harmonic positional basis during training.

At each reinjection block, a lightweight gating MLP reads the current DINOv2 patch tokens and predicts additive gates for both the image-conditioned typed features and the coordinate harmonic features. The gating MLP uses a low-rank bottleneck $1024 \to 16 \to 320$, where the output is split into gates for the order-0 and order-1 typed states. The typed state is then updated by combining the current state, the image-conditioned features, the harmonic positional features, and learned typed-state mixing modules. This allows the backbone tokens to control how much rotation-structured information is accumulated at each depth.

The updated typed state is injected back into the backbone through an invariant projection. Specifically, we concatenate the order-0 state with the norm of the order-1 state, $\mathrm{concat}(q_0, \|q_1\|)$, and project the result to the DINOv2 token dimension. This produces a residual token update that is added to the backbone stream. Each reinjection site has a learned scalar injection gate initialized to zero, so the adapted model is functionally identical to the frozen backbone at initialization and gradually opens the typed residual path during training.

OrbitLoRA also conditions the LoRA bottleneck with the typed state. For each token, the conditioning input concatenates the backbone token, the order-0 typed state, and the order-1 magnitude, giving a feature of dimension $1024 + 128 + 32$. This feature predicts FiLM-style scale and shift parameters for the rank-16 LoRA bottleneck. In this way, the low-rank update is modulated by both semantic token content and the learned rotation-aware state, rather than using a fixed LoRA update for all orientations.

For classification, the final prediction head pools patch-level semantic features. The last backbone tokens are normalized and projected to a 256-dimensional semantic space, then fused with the final invariant typed state. The pooled patch representation is used for classification. A separate CLS-token projection is included with a learned mixing gate initialized to zero, so the classifier begins as a patch-average head and can learn to use the CLS token when beneficial. For segmentation experiments, OrbitLoRA is inserted into the SAM 3 image encoder, while the downstream segmentation interface follows the native SAM 3 prediction pipeline.

The rotation-structure losses use weights $\lambda_{\mathrm{struct}} = 5.0$ and $\lambda_{\mathrm{output}} = 15.0$. These losses supervise the internal typed state and the task output under paired rotated views, encouraging the order-0 features to remain invariant and the order-1 features to transform consistently with the rotation action.

E Training and Inference Cost

We measure trainable parameters, peak memory, throughput, and inference FLOPs under the same hardware and software setup, using a batch size of 16 for all methods. Due to time constraints, we do not report end-to-end wall-clock training time for every method; instead, training throughput (images/s) provides a hardware-matched runtime comparison. The task- performance experiments described in Appendix D use batch size 32; the batch size of 16 here is specific to this efficiency benchmark.

OrbitLoRA's main overhead occurs during training because it processes paired views and retains the typed state. At inference, it remains a single-view forward pass: compared with unmerged LoRA (the LoRA + aug row), it adds approximately 1.3% GFLOPs and 11.6% evaluation memory, while throughput decreases from 91 to 88 images/s.

For clarity, our computational-cost concern applies specifically to (weighted) frame-averaging methods such as λ-EquiTune. The cost of frame-canonicalization methods varies: EquiAdapt uses a

Table 6: Training and inference cost under a matched batch size of 16. Train/evaluation memory denotes peak memory, throughput is measured in images per second, and inference FLOPs are reported per image.

Method	Trainable params	Train mem (MB)	Train img/s	Eval mem (MB)	Eval img/s	Eval GFLOPs/img
LoRA + aug	6.30M	7,585	43	1,441	91	165.3
LoRA (merged)	6.30M	7,585	43	1,376	102	162.0
LoRA + EquiAdapt	6.37M	7,783	40	1,514	86	188.6
LoRA + λ-EquiTune	6.41M	38,575	4.8	1,842	13	1,322.0
LoRA + FoCal	6.30M	7,585	43	1,441	12	1,296.5
OrbitLoRA	8.16M	19,510	32	1,608	88	167.4
OrbitLoRA (merged)	8.16M	19,510	32	1,554	95	164.6

Table 7: Component ablations on EuroSAT with 0.5% training data. All variants use the same frozen DINOv2 ViT-L backbone and LoRA configuration.

Variant	Accuracy ↑	Rotation defect ↓
Full OrbitLoRA	0.902	0.0035
without structural equivariance loss	0.878	0.0068
without output invariance loss	0.861	0.0364
without harmonic coordinate fields	0.874	0.0060
without equivariant content fields	0.866	0.0064
e2cnn replaced by standard convolutional content stem	0.881	0.0052
typed branch without backbone coupling	0.870	0.0102

lightweight learned frame-selection module, whereas FoCal incurs an $8\times$ test-time inference cost. Our main concern with canonicalization is not computational but conceptual: in rotation-symmetric domains without a canonical orientation, selecting a single reference frame is inherently ambiguous and can lead to inferior performance.

OrbitLoRA is only partially mergeable. Standard LoRA weights outside the four typed-interaction blocks can be merged into the frozen backbone. The LoRA components at those four blocks, together with the typed branch and its input-dependent FiLM conditioning, must remain dynamic. Accordingly, the OrbitLoRA (merged) row reports the partially merged configuration.

F Ablation Experiments

We conduct ablation studies on EuroSAT using 0.5% of the training data to isolate the contribution of each component in OrbitLoRA. All variants use the same frozen DINOv2 ViT-L backbone, the same LoRA configuration, and the same training protocol unless otherwise specified. We report classification accuracy and rotation defect.

Component ablations. We first ablate the losses and typed-state sources used by OrbitLoRA. The structural equivariance loss is removed by setting $\lambda_{\mathrm{struct}} = 0$, and the output invariance loss is removed by setting $\lambda_{\mathrm{out}} = 0$. To study the role of the initial typed state, we remove either the harmonic coordinate fields or the equivariant content fields from Eq. equation 5. We also replace the equivariant content stem with a standard convolutional stem of comparable capacity. Finally, we evaluate a variant in which the typed branch is computed but does not update or modulate the pretrained token stream.

From Table 7, we observe that the full OrbitLoRA achieves the highest accuracy and the lowest rotation defect, demonstrating that both the typed branch and the geometric supervision are critical for learning a robust rotation-aware representation in low-data regimes. Removing the structural equivariance loss has a minimal effect on accuracy but slightly increases rotation defect, indicating that the typed state is still partially guided by the other losses. In contrast, removing the output invariance loss leads to a significant drop in accuracy and a higher rotation defect, highlighting its importance for task-specific supervision. Ablating the harmonic coordinate fields or the equivariant content fields reduces both accuracy and rotation consistency, showing that both sources contribute

Table 8: Ablation of typed modulation targets on EuroSAT with 0.5% training data. The full model modulates Q, K, and V after the frozen and LoRA contributions have been combined.

Typed modulation target	Accuracy ↑	Rotation defect ↓
Q only	0.872	0.0073
K only	0.877	0.0082
V only	0.884	0.0068
QKV	0.902	0.0035

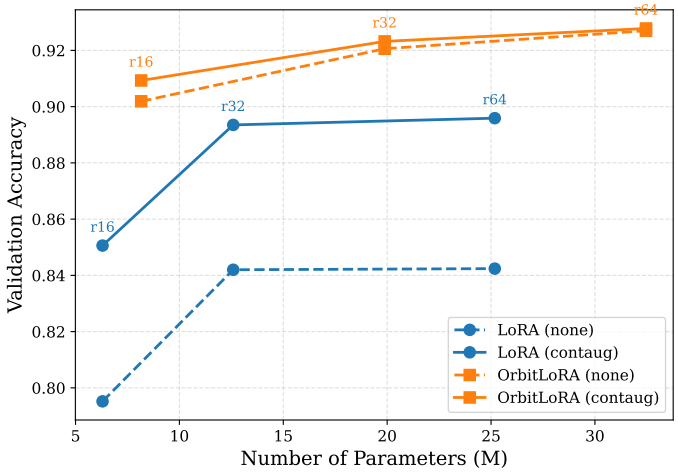

Figure 3: Validation accuracy on EuroSAT with 0.5% training data under different LoRA ranks and trainable parameter counts. Ordinary LoRA is evaluated at ranks 16, 32, and 64, corresponding to 6.30M, 12.59M, and 25.18M trainable parameters. OrbitLoRA is evaluated at the same ranks, corresponding to 8.16M, 19.88M, and 32.46M trainable parameters. We compare training without continuous rotation augmentation and training with continuous rotation augmentation.

complementary geometric information. Replacing the content stem with a standard convolution leads to a small degradation, indicating the benefit of explicitly equivariant local features. Finally, disabling the typed branch's influence on the backbone causes the largest drop among the ablations, confirming that modulating the pretrained token stream is a key mechanism for OrbitLoRA.

Attention modulation ablations. We next study where the typed branch should interact with the transformer attention computation. In the full model, the typed branch modulates the patch-token rows of the LoRA-adapted query, key, and value projections. We compare this design against variants that modulate only one projection, as well as a variant that modulates query and key but not value.

Table 8 shows that modulating only one of the projections provides a small-to-moderate improvement over the baseline, with V modulation contributing more to accuracy and Q/K modulation slightly more to rotation consistency. Combining Q and K improves rotation robustness significantly, while the full QKV modulation achieves the best tradeoff between accuracy and rotation defect. This confirms that the typed branch benefits the model both by guiding attention affinities through Q and K and by influencing the content aggregated through V.

Effect of rank and parameter count. We further study the effect of the LoRA rank on both ordinary LoRA and OrbitLoRA. Although we use rank 16 in the main experiments for efficiency, we also evaluate ranks 32 and 64 to understand whether the performance saturates with additional trainable parameters. Figure 3 plots validation accuracy as a function of trainable parameter count for ordinary LoRA and OrbitLoRA, with and without continuous rotation augmentation.

At rank 16, ordinary LoRA uses 6.30M trainable parameters and obtains 0.7952 accuracy without continuous rotation augmentation and 0.8506 with continuous rotation augmentation. In comparison, OrbitLoRA uses 8.16M trainable parameters at the same rank and achieves 0.9019 without continuous

Table 9: Rotation-aware SAM 3 adaptation on LoveDA semantic segmentation. Each entry reports validation mIoU / C_{24} rotation-averaged mIoU under 10% and full-data training. This setting tests whether OrbitLoRA extends from image-level classification to dense prediction.

Method	**Aug.**	**Data Regime**	
		10% (n=252)	**100% (n=2522)**
SAM3 zero-shot text		0.2803 / 0.2659	
LoRA	×	0.4829 / 0.4625	0.5465 / 0.5249
	✓	0.4894 / 0.4719	<u>0.5473</u> / <u>0.5489</u>
LoRA + FoCal	×	0.4832 / <u>0.4697</u>	0.5422 / 0.5333
	✓	0.4850 / 0.4723	0.5441 / 0.5412
LoRA + EquiAdapt	×	0.4742 / 0.4548	**0.5474** / <u>0.5400</u>
	✓	0.4758 / 0.4628	0.5404 / 0.5468
LoRA + λ-EquiTune	×	<u>0.4869</u> / 0.4566	0.5465 / 0.5222
	✓	<u>0.4897</u> / <u>0.4760</u>	0.5453 / 0.5475
OrbitLoRA (ours)	×	**0.5145 / 0.4972**	<u>0.5473</u> / **0.5496**
	✓	**0.5164 / 0.5011**	**0.5498 / 0.5543**

rotation augmentation and 0.9093 with continuous rotation augmentation. Thus, even at the lowest rank considered, OrbitLoRA substantially outperforms ordinary LoRA while adding only a modest number of trainable parameters.

Increasing the rank improves both methods, but the two models behave differently. Ordinary LoRA benefits strongly from continuous rotation augmentation: its accuracy increases from 0.7952 to 0.8506 at rank 16, from 0.8420 to 0.8935 at rank 32, and from 0.8424 to 0.8959 at rank 64. Without such augmentation, ordinary LoRA saturates after rank 32, suggesting that additional low-rank capacity alone does not resolve the rotation mismatch. OrbitLoRA is less dependent on continuous rotation augmentation. Without continuous rotation augmentation, its accuracy increases from 0.9019 at rank 16 to 0.9206 at rank 32 and 0.9270 at rank 64. With continuous rotation augmentation, it obtains 0.9093, 0.9172, and 0.9248 at ranks 16, 32, and 64, respectively.

These results indicate that the gain of OrbitLoRA is not simply due to adding more trainable parameters. At rank 16, OrbitLoRA with 8.16M parameters already outperforms ordinary LoRA at rank 64 with 25.18M parameters, even when ordinary LoRA is trained with continuous rotation augmentation. The rank sweep also shows that OrbitLoRA can further benefit from increased adaptation capacity, reaching the best validation accuracy at rank 64 without continuous rotation augmentation. This supports the view that the typed branch provides a more effective use of adaptation capacity by coupling rotation-aware fields to the pretrained attention stream.

G Cross-Model Results on High-Resolution Osteosarcoma

We further evaluate OrbitLoRA on the high-resolution (1024×1024) digital pathology and microscopic histology dataset Osteosarcoma Tumor Assessment [19], which contains three classes: viable tumor, necrotic tumor, and non-tumor tissue. We report results using both a 50-image low-data training split and the full 750-image training set. The table reports classification accuracy in the order DINO / CLIP / LLaVA.

OrbitLoRA consistently improves over augmented LoRA across DINO, CLIP, and LLaVA, with the largest improvements in the 50-image regime. These results extend the cross-model evaluation beyond EuroSAT and provide evidence in a high-resolution pathology setting where in-plane orientation is intrinsically ambiguous.

Table 10: Cross-model classification results on the high-resolution Osteosarcoma Tumor Assessment dataset. Each entry reports accuracy for DINO, CLIP, and LLaVA, respectively.

	Reduced training set (50 images)	Full training set (750 images)
Method	DINO / CLIP / LLaVA	DINO / CLIP / LLaVA
LoRA	.836/.793/.805	.934/.876/.899
LoRA + aug	.843/.798/.811	.941/.879/.903
LoRA + aug + consistency loss	.851/.804/–	.943/.880/–
LoRA + aug + EquiAdapt	.854/.803/–	.940/.878/–
LoRA + aug + λ-EquiTune	.860/.806/–	.944/.895/–
LoRA + aug + FoCal	.849/.802/–	.943/.880/–
OrbitLoRA	**.870/.808/.833**	**.954/.901/.915**

H Semantic Segmentation Qualitative Results

In Figure 2 we provide qualitative results for semantic segmentation with SAM3 adaptation. We show how our method performs compared to ordinary LoRA trained with continuous rotation augmentation at two training sample budgets, 10% and 100%. As demonstrated in the Figure, zero-shot segmentation of ambiguously rotated satelite imagery shows severe misclassifications, while adaptation generally improves the results. In line with our quantitative evaluation (Table 9), OrbitLoRA performs best with a more significant gap when trained on very few samples.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations discussed in the appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Proof of proposition 1 and additional theoretical results in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Additional implementation details are provided in the appendix. We will release code on acceptance.

   Guidelines:

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

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

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

   Answer: [No]

   Justification: We will release the code in future but at the time of submission it is not enough time to collect all code and data.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Implementation details are in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We have standard deviation reported from three seeds for low data regime experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: Implementation details in the appendix

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: This work follows NeurIPS Code of Ethics.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Broader impacts stated in the appendix.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: This paper is based on public data and base models without unsafe risks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Appendix specified the license.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: No new assets.

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

Answer: [N/A]

Justification: No crowdsourcing and research with human subjects involved in the research.

Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

Answer: [N/A]

Justification: No such risks.

Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.