This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Basis-Oriented Low-rank Transfer for Few-Shot and Test-Time Adaptation

Junghwan Park[1] Woojin Cho[1] Junhyuk Heo[1] Darongsae Kwon[1] Kookjin Lee[2]
[1]TelePIX, [2]Arizona State University
{junghwan, woojin, hjh1037, darong.kwon}@telepix.net, kookjin.lee@asu.edu

Abstract

Adapting large pre-trained models to unseen tasks under tight data and compute budgets remains challenging. Meta-learning approaches explicitly learn good initializations, but they require an additional meta-training phase over many tasks, incur high training cost, and can be unstable. At the same time, the number of task-specific pre-trained models continues to grow, yet the question of how to transfer them to new tasks with minimal additional training remains relatively underexplored. We propose BOLT (Basis-Oriented Low-rank Transfer), a framework that reuses existing fine-tuned models not by merging weights, but instead by extracting an orthogonal, task-informed spectral basis and adapting within that subspace. In the offline phase, BOLT collects dominant singular directions from multiple task vectors and orthogonalizes them per layer to form reusable bases. In the online phase, we freeze these bases and train only a small set of diagonal coefficients per layer for the new task, yielding a rank-controlled update with very few trainable parameters. This design provides (i) a strong, training-free initialization for unseen tasks, obtained by pooling source-task coefficients—along with a lightweight rescaling step—while leveraging the shared orthogonal bases, and (ii) a parameter-efficient fine-tuning (PEFT) path that, in our experiments, achieves robust performance compared to common PEFT baselines as well as a representative meta-learned initialization. Our results show that constraining adaptation to a task-informed orthogonal subspace provides an effective alternative for unseen-task transfer.

1. Introduction

Adapting large pre-trained models to new tasks and shifted data distributions is critical for real-world deployment, especially when facing unseen tasks with limited data [30, 66, 69]. In such few-shot or test-time adaptation scenarios, the model's initialization often plays a decisive role [4, 30] in whether learning will succeed.

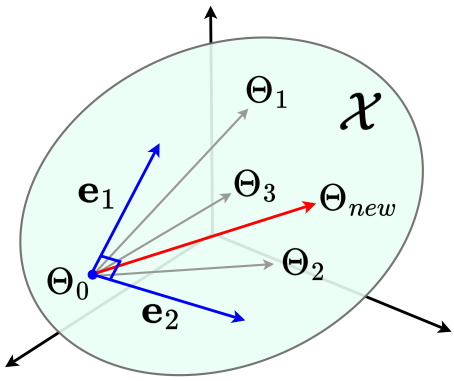


Figure 1. **Conceptual view of BOLT**: Task vectors $\{\Theta_i - \Theta_0\}_{i=1}^N$ can be represented in a common subspace $\mathcal{X}$ formed by orthogonal bases $\{\mathbf{e}_d\}_{d=1}^r$. By reusing the same subspace, the model can adapt to an unseen task vector $\{\Theta_{\text{new}} - \Theta_0\}$ more quickly.

Gradient-based meta-learning methods such as Model-Agnostic Meta-Learning (MAML) [12] were among the first to emphasize the importance of a good initialization by explicitly training models to rapidly adapt to novel tasks. However, these approaches involve expensive bi-level optimization and can be unstable to train [17], and require a dedicated meta-training phase over many tasks, which may be impractical in settings where new tasks continually emerge. This motivates developing methods that offer strong initialization for unseen tasks without requiring an additional meta-training stage, especially in settings where a library of fine-tuned models is already available.

At the same time, the ecosystem of pre-trained models is rapidly expanding, covering a diverse array of domains and tasks [11, 14, 40]. This abundance presents an opportunity: instead of additional heavy training, one might transfer knowledge directly from existing models to new tasks. Prior work has explored merging or composing multiple fine-tuned models to combine their capabilities [13, 22, 33]. For example, a fine-tuned model can be represented as a task vector [21], and adding or averaging such vectors can blend learned behaviors. While these

model-merging techniques can perform well on tasks that were part of their training set, they are not designed for entirely unseen tasks. A merged model essentially interpolates between known task solutions and tends to overfit to the training tasks, lacking a mechanism to generalize beyond them [52]. In fact, naïve task arithmetic often entangles disparate update directions, leading to interference and unpredictable behavior when tasks are misaligned. These limitations motivate a different approach to model transfer. In this work, we pursue an alternative paradigm for adapting models to unseen tasks. If we have a library of models fine-tuned on diverse source tasks, the weight-space directions in which these models moved during fine-tuning may span a rich subspace for future tasks. Our insight is that these task-specific directions can be extracted and re-used as a basis for new tasks. Recent studies have shown that the dominant singular directions characterize the principal axes of the fine-tuning update, and that orthogonalizing such directions across tasks can mitigate interference.

Building on this insight, we propose Basis-Oriented Low-Rank Transfer (BOLT), which adapts a pre-trained model to a new task without any explicit meta-training [17] or model merging [59]. Instead of combining weights from different models, BOLT constructs an orthogonal spectral basis from existing fine-tuned models and uses this basis as the backbone for adapting to an unseen target task. A conceptual overview is shown in Figure 1.

Concretely, BOLT operates in two phases. In an offline phase, we collect the task vectors from multiple source models and perform singular value decomposition (SVD) to extract their major directions. We then orthogonalize these directions to form a shared spectral basis for each layer. In the subsequent online phase, the new task is learned by freezing this orthogonal basis and training only a small set of coefficients. The resulting weight update for the unseen task is constrained to lie in the span of those basis directions – equivalently, the update is diagonal in the shared spectral basis. This design keeps the number of trainable parameters extremely low and confines the update to a subspace spanned by prior tasks, providing explicit control over the update's effective rank and reducing the risk of overfitting to spurious patterns. We validate BOLT on challenging few-shot image classification benchmarks and label-free test-time adaptation scenarios. Without any additional meta-training, BOLT achieves more robust performance than conventional fine-tuning or parameter-efficient tuning baselines. As an additional comparison, Figure 2 reports few-shot adaptation results on 17 unseen tasks using a ViT-B/32 backbone, where BOLT's initialization outperforms meta-learned initialization. In summary, our contributions are:

- **Orthogonal multi-task bases**: We build task-informed, layer-wise orthogonal bases by collecting the top singular directions from multiple fine-tuned models and orthogonalizing them. This provides a stable shared coordinate system for later adaptation.
- **Strong initialization without training**: By projecting task vectors into the shared spectral basis and pooling their coefficients—with a lightweight rescaling step—we obtain a training-free initialization that already performs well on unseen tasks.
- **Generalization on few-shot and test-time adaptation**: Comprehensive evaluation on few-shot, out-of-domain (OOD), and label-free test-time adaptation across general and remote-sensing datasets, showing that BOLT consistently matches or surpasses strong PEFT with far fewer task-specific parameters.

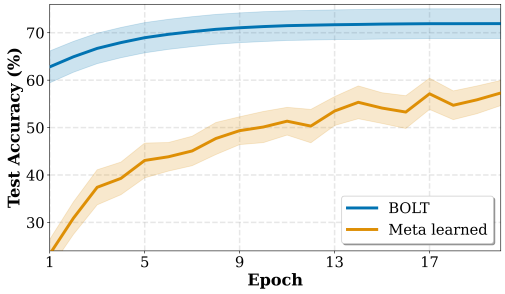


Figure 2. **Meta-learning vs. BOLT:** few-shot adaptation curves comparing meta-learned initialization and BOLT. Our method reaches higher accuracy in fewer epochs, showing faster convergence than meta-learned initialization.

2. Related Work

Parameter-efficient adaptation. When data and compute are limited, a central challenge is enabling adaptation with minimal changes to the backbone. Recent work addresses this by constraining where learnable parameters are introduced. Low-rank modules such as LoRA [18] and its decomposed variants [31] inject compact update factors into linear layers, while modulation- and adapter-based methods [3, 29] adjust activations or insert small bottlenecks to provide controlled capacity. Prompt-based tuning [23] instead steers the backbone using a few learnable tokens. In the CLIP [40] setting, feature-space approaches—including linear probes, TIP-Adapter [63], and LP++ [20]—avoid weight updates entirely. Despite differing in where capacity is added, these methods all restrict adaptation to a lightweight and well-structured subspace.

Task vectors and composition. A complementary line of work edits models directly in weight space by representing fine-tuned checkpoints as task vectors, enabling arithmetic operations such as averaging, interpolation, and tangent-space editing [21, 37]. While this reveals that indepen-

dently fine-tuned models often lie in shared basins, naïve combinations can entangle incompatible update directions, prompting structure-aware approaches that use curvature-based weighting [33] or conflict-aware sparsification [57]. More recent analyses decompose task vectors into layer-wise singular directions [13], providing a means to diagnose interference and merge updates along decorrelated axes, while anisotropic scaling methods [62] learn how to emphasize or suppress specific components. Collectively, these methods expose the compositional structure of fine-tuning updates, though they primarily reweight existing directions rather than building new coordinate systems for unseen tasks.

Subspace methods and orthogonalization. A related line of work controls where learning occurs by shaping the subspace of allowable updates. In continual learning, projecting gradients away from previously spanned directions mitigates forgetting [44, 61], and later studies analyze how the choice and frequency of such projections influence robustness [58, 65]. Orthogonalization also arises in representation analysis, where similarity metrics, whitening, and decorrelated normalization help disentangle feature directions and reduce redundancy [19, 24, 41]. In parameter space, permutation or orthogonal alignment reconciles symmetries before model averaging, improving merge fidelity [1]. Overall, these approaches demonstrate that enforcing orthogonality—on gradients, representations, or weights—can reduce interference and motivate learning within a shared, decorrelated subspace.

Initialization for Unseen Tasks. A central determinant of performance on unseen tasks is the starting point from which adaptation proceeds. While gradient-based meta-learning popularized learning such initializations, it typically requires a separate meta-training phase over many tasks [6, 12, 35]. A complementary line of work emphasizes structure in the update space, using low-rank or orthogonalized coordinates to stabilize early learning. Building on this perspective, we propose a training-free initializer and show that pooling and lightly rescaling the spectral coefficients of source-task updates provides a compact starting point that performs well on unseen tasks. Conceptually, our approach is related to meta-learning in that both aim to construct useful initializations for new tasks, but BOLT obtains such initial states analytically from a collection of fine-tuned models instead of introducing a separate meta-optimization stage.

3. Preliminaries

Before introducing our framework, we formalize the problem setting, and clarify how task-specific updates are represented and combined.

3.1. Setting and Task Vectors

We consider a pre-trained model with weights Θ_0, and a set of N related source tasks for which fine-tuned models $\{\Theta_i\}_{i=1}^N$ are available. Each Θ_i is obtained by fine-tuning Θ_0 on task i. We define the task vector for task i as the difference between the fine-tuned and original weights:

$$\Delta_i \;=\; \Theta_i - \Theta_0. \tag{1}$$

This Δ_i represents the full update that adapts the pre-trained model to task i. For each layer ℓ, suppose the weight parameters in Θ_0 form a matrix of dimensions $m_\ell \times n_\ell$ and let $M_i^{(\ell)} = \Theta_i^{(\ell)} - \Theta_0^{(\ell)}$ be the corresponding layer update for task ℓ. This layer-wise view aligns with prior analyses that use singular vectors to characterize task-specific changes and allows us to directly apply standard linear algebra tools.

3.2. Task Arithmetic and Its Limitations

A simple way to combine knowledge from multiple tasks is to merge their weight updates. We first define the merged model Θ_{agg} as follows:

$$\Theta_{agg} \;=\; \Theta_0 + \sum_{i=1}^{N} \alpha_i \Delta_i. \tag{2}$$

Here, Δ_i is the task vector of task i, and α_i are blending coefficients. This task arithmetic directly composes the fine-tuning updates from different tasks. While easy to compute, naïve weight merging as in Eq. (2) can suffer from interference between tasks. If the dominant update directions of tasks are not mutually aligned, the $\sum_i \alpha_i \Delta_i$ will entangle disparate directions and degrade performance. In practice, merged models often exhibit unpredictable behavior unless the tasks are very compatible. More importantly, merging alone is not sufficient for entirely new target tasks. The combination in Eq. (2) can only blend the behaviors of existing task updates Δ_i, and when the target task departs from these sources, there is no principled way to choose coefficients α_i that produce the required behavior. In particular, once a merge is chosen, there is no explicit adaptation step for the new task. In contrast, our approach treats these directions as a shared basis and learns task-specific coefficients from the target data rather than relying on a single static merge.

3.3. Singular Directions and Low-Rank Structure

Although a task update Δ_i has the same dimensionality as the original parameters Θ_0, in practice the change it induces is much more structured. Empirically, most layers do not need to move in all directions of the weight space at once; instead, the update tends to concentrate its energy in a few dominant directions. To make this precise, for a given layer

ℓ, we look at the matrix-shaped update $M_i^{(\ell)}$ corresponding to task i at that layer and apply a thin SVD:

$$M_i^{(\ell)} \;=\; U_i^{(\ell)} \Sigma_i^{(\ell)} V_i^{(\ell)\top}, \tag{3}$$

where $U_i^{(\ell)}$ and $V_i^{(\ell)}$ collect the left and right singular directions of the update, and $\Sigma_i^{(\ell)}$ contains the singular values sorted in decreasing order. When only a few singular values in $\Sigma_i^{(\ell)}$ are large, Eq. (3) shows that $M_i^{(\ell)}$ is well-approximated by a low-dimensional subspace spanned by those dominant singular directions. This is the key property we later exploit when we aggregate singular directions from multiple source tasks and orthogonalize them into the shared bases.

3.4. Orthogonalization and Basis Construction

As shown in Section 3.3, task-specific layer updates $M_i^{(\ell)}$ often concentrate their energy in a few singular directions (Eq. (3)). This suggests that, across many source tasks, we can collect those dominant directions and use them to span a common subspace for future adaptation.

For each layer ℓ, suppose we have N source tasks and their matrix-shaped updates $\{M_i^{(\ell)}\}_{i=1}^N$. We first compute a thin SVD for each and keep only the top k_i singular directions. We then stack these directions across tasks to make

$$\begin{aligned} U_{\text{stack}}^{(\ell)} &= \Big[U_1^{(\ell)}(:,1{:}k_1)\;\cdots\;U_N^{(\ell)}(:,1{:}k_N)\Big] \in \mathbb{R}^{m_\ell \times r}, \\ V_{\text{stack}}^{(\ell)} &= \Big[V_1^{(\ell)}(:,1{:}k_1)\;\cdots\;V_N^{(\ell)}(:,1{:}k_N)\Big] \in \mathbb{R}^{n_\ell \times r}. \end{aligned} \tag{4}$$

So that $U_{\text{stack}}^{(\ell)} \in \mathbb{R}^{m_\ell \times r}$ and $V_{\text{stack}}^{(\ell)} \in \mathbb{R}^{n_\ell \times r}$, where $r = \sum_{i=1}^N k_i$ is the total number of collected directions.

Because these stacked directions may still be correlated across tasks, we orthogonalize them to obtain the final task-informed, layer-wise bases $U_{\text{orth}}^{(\ell)}$ and $V_{\text{orth}}^{(\ell)}$, defined as

$$U_{\text{orth}}^{(\ell)} \;=\; \mathcal{O}\big(U_{\text{stack}}^{(\ell)}\big), \qquad V_{\text{orth}}^{(\ell)} \;=\; \mathcal{O}\big(V_{\text{stack}}^{(\ell)\top}\big)^\top, \tag{5}$$

where $\mathcal{O}(\cdot)$ denotes the SVD-based orthogonalization operator. These matrices serve as the shared spectral coordinates in which all subsequent target-task updates are represented. Concretely, we apply a single whitening step

$$\begin{aligned} U_{\text{orth}}^{(\ell)} &= U_{\text{stack}}^{(\ell)} \big({U_{\text{stack}}^{(\ell)}}^\top U_{\text{stack}}^{(\ell)} + \varepsilon I\big)^{-\frac{1}{2}}, \qquad \varepsilon > 0, \\ V_{\text{orth}}^{(\ell)} &= V_{\text{stack}}^{(\ell)} \big({V_{\text{stack}}^{(\ell)}}^\top V_{\text{stack}}^{(\ell)} + \varepsilon I\big)^{-\frac{1}{2}}, \qquad \varepsilon > 0. \end{aligned} \tag{6}$$

with $\varepsilon > 0$ for numerical stability. Following prior work [13], we compute U_{stack}^{ℓ} and V_{stack}^{ℓ} via the thin SVD $U_{\text{stack}}^{\ell} = \Psi_u \Sigma_u \Phi_u^\top$ and $V_{\text{stack}}^{\ell} = \Psi_v \Sigma_v \Phi_v^\top$. We then set $U_{\text{orth}} = \Psi_u \Phi_u^\top$ and $V_{\text{orth}} = \Psi_v \Phi_v^\top$.

All subsequent adaptation in Section 4 will keep $U_{\text{orth}}^{(\ell)}$ and $V_{\text{orth}}^{(\ell)}$ fixed and will learn only how much to scale each basis direction.

4. Proposed Method

In this section, we now describe how BOLT performs adaptation using the shared orthogonal bases. Figure 3 summarizes the overall pipeline, and the following subsections describe each step mathematically.

4.1. Projection to the Shared Coordinates

Let $M_i^{(\ell)}$ denote the parameter update of source task i at layer ℓ, defined as in Sections 1–3. For layer ℓ, we omit the index for clarity and denote $M := M_i^{(\ell)}$, $U := U_{\text{orth}}^{(\ell)}$, $V := V_{\text{orth}}^{(\ell)}$. Because U and $V^\top$ are column- and row-orthonormal, respectively, any update M can be projected onto the task-informed subspace by

$$S \;=\; U^\top M V, \quad S \in \mathbb{R}^{r \times r}, \tag{7}$$

which measures how strongly M activates each pair of shared directions obtained in Eqs. (4)-(6). Note that this step does not re-define the bases; it only re-expresses M in the already-fixed coordinates.

4.2. Diagonal Reconstruction Problem

Our goal in Section 4 is not to keep the full $r \times r$ matrix in Eq. (7), but to find the best diagonal matrix inside this subspace that reconstructs M when mapped back to the original space. We therefore consider the least-squares problem

$$\min_D \; \|M - UDV^\top\|_F^2, \tag{8}$$

where $D \in \mathbb{R}^{r \times r}$ is constrained to be diagonal, i.e., $D_{ij} = 0$ for $i \neq j$. Eq. (8) asks for the diagonal coefficients in the shared basis that best match the offline update M. To make Eq. (8) explicit, insert and subtract $USV^\top$ inside the Frobenius norm:

$$\|(M - USV^\top) + U(S - D)V^\top\|_F^2 \tag{9}$$

$$= \|M - USV^\top\|_F^2 + \|U(S - D)V^\top\|_F^2, \tag{10}$$

The cross term in Eq. (9) vanishes because the two components lie in orthogonal subspaces:

$$U^\top (M - USV^\top) V \;=\; S - S \;=\; 0. \tag{11}$$

The first term in Eq. (10) is independent of D, so minimizing Eq. (8) is equivalent to

$$\min_D \; \|U(S - D)V^\top\|_F^2. \tag{12}$$

At this point we use the orthonormality already established in Section 3. When a matrix already lives in the span of U and V, the Frobenius norm is preserved. Thus, the term in Eq. (12) can be expressed as follows:

$$\|U(S - D)V^\top\|_F^2 \;=\; \|S - D\|_F^2. \tag{13}$$

Therefore, the diagonal extraction problem further reduces to the following form:

$$\min_D \; \|S - D\|_F^2. \tag{14}$$

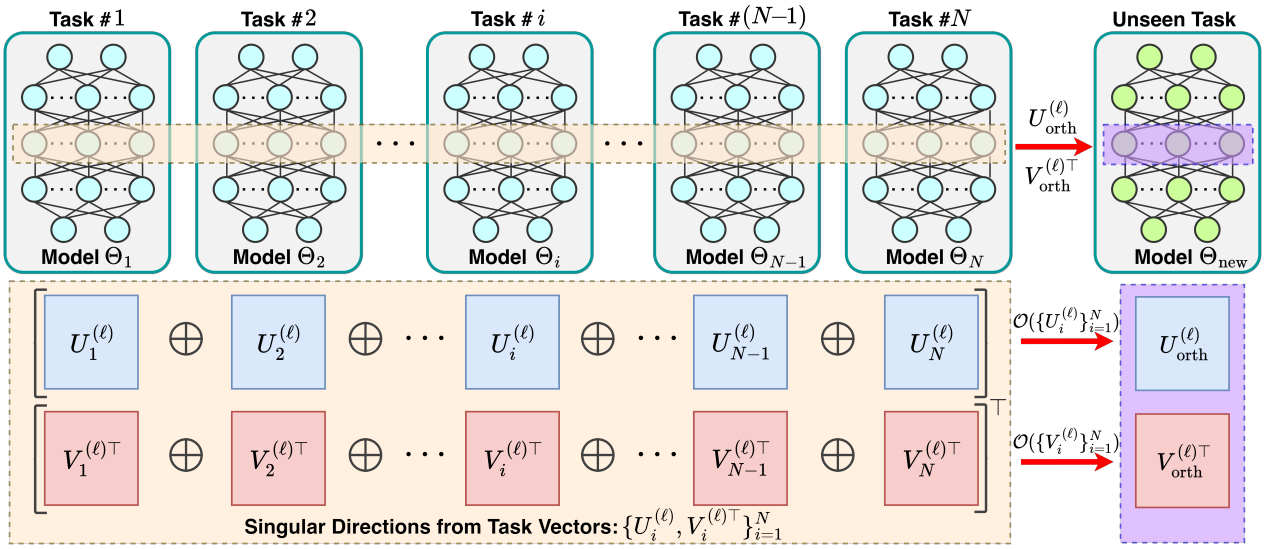


Figure 3. **Overall BOLT pipeline**: for task vectors $\{\Theta_i\}_{i=1}^N$, we extract layer-wise SVDs and orthogonalize them to obtain shared bases $\{U_{\text{orth}}, V_{\text{orth}}\}$. These fixed bases are later reused to construct weights for an unseen task with only small diagonal parameters.

4.3. Closed-form Diagonal and Coefficient Vector

Expanding the Frobenius norm in Eq. (14) yields Eq. (15):

$$\|S - D\|_F^2 = \sum_{j \neq k} S_{jk}^2 + \sum_j (S_{jj} - D_{jj})^2. \tag{15}$$

The off-diagonal part $\sum_{j \neq k} S_{jk}^2$ does not depend on D, so the minimizer is obtained by matching every diagonal entry. In other words, the optimal diagonal simply copies the diagonal of S:

$$D_{jj}^\star = S_{jj} \quad \forall j. \tag{16}$$

Consequently, we can write

$$D^\star = \text{diag}(\mathbf{s}), \qquad \mathbf{s} := \text{diag}(S) \in \mathbb{R}^r. \tag{17}$$

The r-dimensional vector $\mathbf{s}$ is the only information from M that we will carry over into the online phase.

4.4. Pooling Across Multiple Source Tasks

When we have N tasks, we simply repeat the projection Eq. (7) and the diagonal extraction Eq. (17) for each task:

$$S_i = U^\top M_i V, \ \mathbf{s}_i = \text{diag}(S_i) \in \mathbb{R}^r, \ i = 1, \ldots, N, \tag{18}$$

These $\{\mathbf{s}_i\}_{i=1}^N$ are all expressed in the same orthonormal coordinates because U and V were fixed offline (i.e. Eqs. (4)-(6)). A simple, data-free initializer for a new/unseen task is then the component-wise mean

$$\mathbf{s}_{\text{pool}} = \frac{1}{N} \sum_{i=1}^N \mathbf{s}_i, \tag{19}$$

Figure 4 summarizes pooling, where each source update is projected onto the shared bases to form S_i, and then the resulting diagonals $\{\mathbf{s}_i\}_{i=1}^N$ are averaged to obtain $\mathbf{s}_{\text{pool}}$ for the online phase.

Scalar rescaling of the diagonal. After the data-free mean initializer in Eq. (19), we apply a single global scaling factor to all diagonal coefficients, selecting it via a brief sweep on the training set. For each candidate $\alpha \in \mathcal{A}$ (a small finite set, e.g., $\{1, 3, 5, 7, 10\}$), we form

$$\tilde{\mathbf{s}}^{(\ell)}(\alpha) = \alpha \, \mathbf{s}_{\text{pool}}^{(\ell)} \quad \text{for all layers } \ell, \tag{20}$$

compose the merged parameters

$$\Theta_0 + \sum_\ell U_{\text{orth}}^{(\ell)} \, \text{diag}(\tilde{\mathbf{s}}^{(\ell)}(\alpha)) V_{\text{orth}}^{(\ell)\top}, \tag{21}$$

and evaluate accuracy on a few mini-batches of the training data. We then select

$$\hat{\alpha} = \arg\max_{\alpha \in \mathcal{A}} \text{Acc}\big(\Theta_0, \{\tilde{\mathbf{s}}^{(\ell)}(\alpha)\}; \mathcal{D}_{\text{train}}\big), \tag{22}$$

and set the initialization to $\mathbf{s}_0^{(\ell)} = \hat{\alpha} \, \mathbf{s}_{\text{pool}}^{(\ell)}$ for all ℓ. This step does not modify the orthogonal bases and adds negligible cost, yet often stabilizes early adaptation by matching the overall scale of the diagonal update to the target task.

4.5. Online Adaptation in the Shared Subspace

At adaptation time, we keep the task-informed bases from Section 3 fixed and learn only the spectral coefficients. For a new task, the admissible update at layer ℓ is

$$\Delta_{\text{new}}^{(\ell)}(\mathbf{s}^{(\ell)}) = U_{\text{orth}}^{(\ell)} \, \text{diag}(\mathbf{s}^{(\ell)}) V_{\text{orth}}^{(\ell)\top}. \tag{23}$$

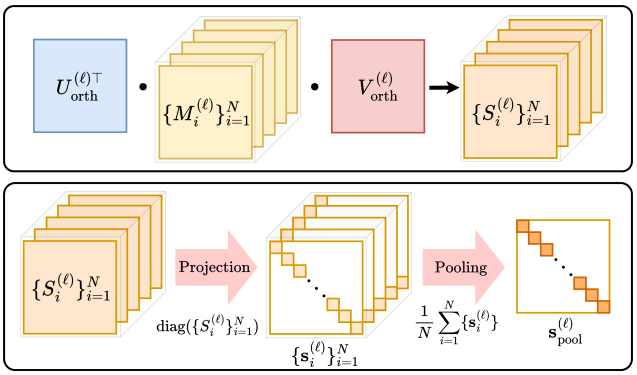


Figure 4. **Initialization of the diagonal coefficients**: each source task is projected onto $\{U_{\text{orth}}, V_{\text{orth}}\}$ to obtain its best diagonal $\{\mathbf{s}_i^{\ell}\}_{i=1}^{N}$, and these per-task diagonals are pooled to form $\mathbf{s}_{\text{pool}}^{\ell}$ for a new task. This provides a data-free, task-informed starting point.

where $\mathbf{s}^{(\ell)}$ is the only learnable parameter at that layer. This parameterization restricts the update to the spectral diagonal of the shared basis, so that $\text{rank}(\Delta_{\text{new}}^{(\ell)}(\mathbf{s}^{(\ell)})) \leq r$ and forces the target-task update to remain inside the task-informed subspace spanned by $\{U_{\text{orth}}^{(\ell)}, V_{\text{orth}}^{(\ell)}\}$. The adapted model for the new task can then be written as

$$\Theta(\mathbf{s}) = \Theta_0 + \sum_{\ell} \Delta_{\text{new}}^{(\ell)}(\mathbf{s}^{(\ell)}), \tag{24}$$

where all $\mathbf{s}^{(\ell)}$ are initialized from the pooled and rescaled coefficients in Eq. (20). In this way, BOLT performs online adaptation by optimizing only a small number of spectral coefficients per layer, dramatically reducing the number of task-specific parameters while still allowing effective adaptation in the shared orthogonal subspace.

5. Experiments

We evaluate BOLT across few-shot, OOD, and test-time adaptation settings to validate its generality and efficiency. This section outlines the benchmarks and implementation details, followed by quantitative comparisons and ablations.

5.1. Benchmarks and setup

Tracks. We evaluate three complementary tracks: (i) few-shot classification with $k \in \{1, 2, 4, 8, 16\}$ labeled samples per class, (ii) OOD robustness on ImageNet-V2/A/R/Sketch, and (iii) label-free test-time adaptation (TTA) under distribution shift.

Datasets. We evaluate our method on two disjoint families of tasks. **(i) General-domain tasks** include DTD [7], GTSRB [47], MNIST [26], SVHN [34], STL10 [8], OxfordIIITPet [38], Flowers102 [36], CIFAR100 [25], PCAM [48], CIFAR10 [25], Food101 [2], FashionMNIST [55], RenderedSST2 [45], EMNIST [9], FGVCAircraft [32], CUB200 [49], and Country211 [40]. **(ii) Remote-sensing tasks** include AID [54], CLRS [28], EuroSAT_RGB [15], MLRSNet [39], NWPU-RESISC45 [5], Optimal-31 [51], PatternNet [67], RS_C11 [64], RSD46-WHU [56], RSI-CB128 [27], RSSCN7 [70], SAT-4 [68], SIRI-WHU [68], UC_Merced [60], and WHU-RS19 [53].

We report results separately for these two families because they differ markedly in their visual statistics and transfer behavior. General-domain datasets contain natural or object-centric imagery with relatively homogeneous texture and illumination, whereas remote-sensing datasets consist of aerial scenes characterized by greater spatial complexity, scale variation, and distribution shifts induced by environmental or atmospheric factors. Evaluating them jointly would obscure domain-specific effects, while treating them separately allows us to examine how adaptation behaves under distinct visual regimes. Notably, remote-sensing imagery forms a particularly challenging stress test for few-shot and low-rank adaptation, since its domain gap from pre-trained CLIP representations is substantially larger than in conventional vision benchmarks.

Baselines. We compare BOLT against a range of adaptation methods. Zero-shot CLIP [40] classifies images directly from text prompts without any task-specific training, while a linear probe [40] fits a linear classifier on frozen CLIP features and serves as a lightweight adaptation baseline. Tip-Adapter [63] performs training-free few-shot adaptation using a key–value cache built from support examples, and LP++ [20] strengthens linear probing through improved feature normalization and optimization. Among parameter-efficient tuning methods, LoRA [18] injects low-rank updates into weight matrices, and aTLAS [62] learns anisotropic scaling coefficients over task vectors to improve compositional transfer.

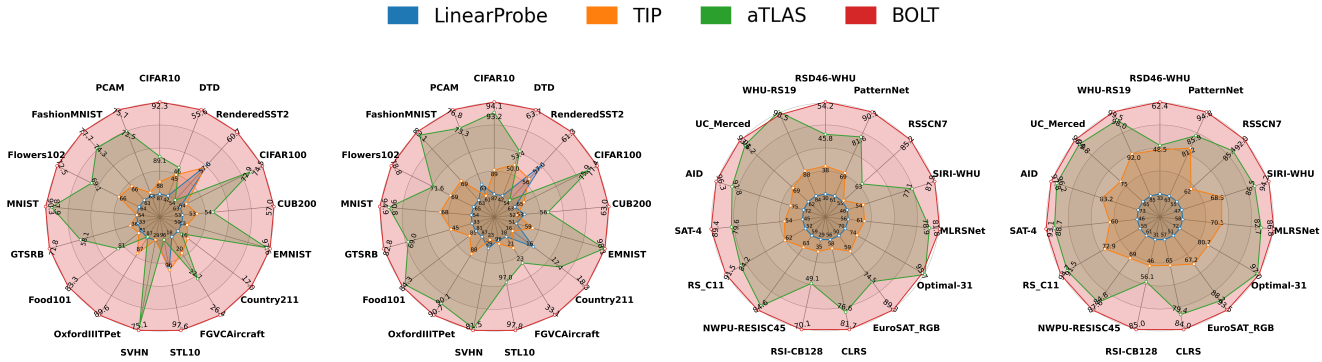
Training and evaluation. We benchmark BOLT on three CLIP backbones—ViT-B/32, ViT-B/16, and ViT-L/14—across few-shot and test-time adaptation settings to assess robustness under diverse data regimes. In all experiments, BOLT performs layer-wise retention of up to 12 singular directions, capturing dominant update structure while maintaining a highly compact model (about 8k learnable parameters). The spectral bases are constructed by aggregating all source-task vectors within each domain. Implementation details, hyperparameters, and extended results are provided in the supplementary material.

5.2. Few-Shot Adaptation

Few-shot learning aims to adapt a pre-trained model to a new task using only a handful of labeled samples—k im-

Table 1. Few-shot accuracy (%) averaged over three CLIP backbones (ViT-B/32, ViT-B/16, ViT-L/14) and all datasets within each domain. Left: general-domain; right: remote-sensing. Best and second-best per column are in **bold** and underlined.

	General datasets					Remote-sensing datasets				
Method	1-shot	2-shot	4-shot	8-shot	16-shot	1-shot	2-shot	4-shot	8-shot	16-shot
Zero-shot CLIP			60.74					58.59		
Linear Probe	61.08	61.09	61.11	61.16	61.23	58.87	58.95	59.19	59.67	60.70
LoRA	60.97	61.40	64.86	65.34	69.57	70.12	73.91	81.14	86.31	90.76
TIP	61.49	61.74	62.44	63.05	63.78	60.36	62.01	64.94	68.49	73.26
LP++	43.34	54.11	60.88	67.83	72.09	51.85	67.59	77.36	84.00	88.47
aTLAS	66.34	67.37	71.31	73.41	74.44	69.87	73.68	77.86	81.38	86.24
BOLT (ours)	**71.30**	**72.67**	**74.90**	**76.72**	**78.46**	**80.77**	**83.86**	**87.59**	**89.92**	**91.86**



(a) General, 4-shot (b) General, 16-shot (c) Remote sensing, 4-shot (d) Remote sensing, 16-shot

Figure 5. Few-shot accuracy at 4 and 16 shots using the ViT-B/32 backbone across general and remote-sensing datasets.

ages per class for a k-shot setting. Following standard practice, we adapt a frozen CLIP encoder to each target dataset using only $k \in \{1, 2, 4, 8, 16\}$ class-balanced examples while keeping the spectral basis fixed. As summarized in Table 1, BOLT achieves the best performance across both the general and remote-sensing domains, showing clear advantages even under extremely limited supervision. In few-shot settings, the improvement margin is particularly pronounced, highlighting the effectiveness of spectral basis adaptation for data-scarce scenarios. While aTLAS learns anisotropic scaling coefficients on existing task vectors, reweighting known directions within a linear subspace, our method instead constructs a compact diagonal spectral basis derived from multi-task singular directions. By decorrelating these directions into an orthogonal basis, BOLT mitigates interference during adaptation and achieves stable, data-efficient performance, as shown in Table 1. In addition, aTLAS requires all task vectors to be stored and optimized jointly, which increases memory usage, whereas BOLT builds a shared orthogonal basis once and reuses it across tasks, achieving much higher memory efficiency without sacrificing performance.

Figure 5 illustrates per-dataset results for the ViT-B/32 backbone at 4- and 16-shot highlighting dataset-specific performance. BOLT consistently ranks highest across diverse datasets in both general and remote-sensing domains, confirming its strong adaptability across heterogeneous distributions.

Table 2. OOD accuracy (%) using ViT-B/32 backbone at 16-shot. Results are reported on ImageNet OOD variants.

Method	ImageNet-A	ImageNet-R	ImageNet-S	ImageNet-V2
Zero-shot CLIP	14.76	50.95	38.92	52.91
Linear Probe	15.13	51.49	39.48	54.72
LoRA	10.28	42.16	36.76	54.51
TIP	14.64	50.56	38.62	52.00
LP++	14.66	51.92	39.01	54.22
aTLAS	14.87	52.21	39.83	55.29
BOLT (ours)	**15.88**	**53.85**	**41.26**	**55.69**

5.3. OOD Robustness

We evaluate robustness under distribution shift using the ViT-B/32 backbone with 16-shot supervision. All methods are trained on ImageNet [43] and then evaluated under distribution shift on ImageNet-A [10], ImageNet-R [16],

Table 3. TTA results on general-domain datasets, reporting accuracy (%) averaged over all datasets, where the best and second-best methods are highlighted in **bold** and underlined, respectively.

Method	ViT-B/32	ViT-B/16	ViT-L/14
Zero-shot CLIP	56.87	61.55	67.90
Layer Norm	62.68	67.63	74.08
aTLAS	56.87	61.56	67.92
BOLT (ours)	**67.74**	**69.92**	**74.11**

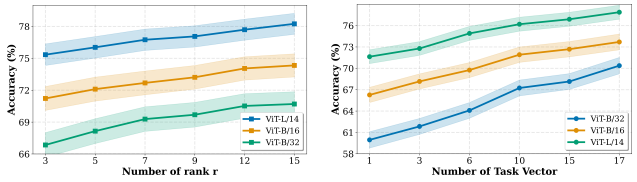
ImageNet-S (Sketch) [50], and ImageNet-V2 [42]. Table 2 shows that BOLT delivers the strongest overall performance, achieving the highest accuracy on all four OOD datasets.

While linear probing remains a competitive baseline, lightweight adaptation methods such as LP++ and TIP exhibit notable degradation under these challenging shifts. LoRA performs particularly poorly, likely due to its large number of learnable parameters, which makes it more susceptible to overfitting in low-data OOD scenarios. In contrast, BOLT maintains stable accuracy across all benchmarks, suggesting that its layer-wise spectral parameterization—derived from task vector subspaces—transfers effectively even when the input distribution changes substantially. These results indicate that BOLT provides strong OOD generalization without introducing heavy task-specific modules or increasing model capacity.

5.4. Test-Time Adaptation (TTA)

We evaluate BOLT in a label-free test-time adaptation (TTA) [30] setting using an offline transductive protocol. Unlike strict streaming TTA, which requires single-pass sequential processing, we assume access to the full unlabeled target split for stable optimization. We adopt a BOLT-tailored variant of unsupervised FixMatch (UFM) [46]: for each input, weak and strong augmentations are generated, sharpened pseudo-labels are produced from the weak view with high-confidence predictions treated as fixed targets, and the remaining samples are optimized using a confidence-masked consistency loss over a few epochs with a cosine learning-rate schedule.

Table 3 reports the resulting TTA performance on general-domain datasets for three CLIP backbones. Across all backbones, BOLT consistently improves over the zero-shot CLIP model and existing baselines, with the largest gains on the smaller ViT-B/32 backbone, suggesting that spectral diagonal adaptation helps compensate limited capacity by exploiting unlabeled target data more effectively. Overall, coupling BOLT's spectral basis with a UFM-style objective yields strong label-free adaptation under distribution shift while keeping the encoder architecture fixed and updating only low-dimensional spectral coefficients instead of introducing new task-specific modules.



(a) Effect of # rank r. (b) Effect of # task vectors.

Figure 6. **Ablation studies on spectral basis construction and adaptation.** (a) Accuracy vs. # rank r, showing that performance saturates after a small subspace size. (b) Accuracy vs. # task vectors used to build the shared basis, indicating that a compact yet diverse set of sources suffices.

5.5. Ablation Study

We study two factors in constructing the spectral basis: the layer-wise rank r and the number of source task vectors. All ablations use the 16-shot setting, average over three seeds, and train only the diagonal spectral coefficients; source vectors are selected at random.

Effect of the rank r. Using all source task vectors, we vary the number of singular vectors kept per layer. As shown in Figure 6a, all backbones benefit from increasing r, with the smaller ViT-B/32 showing the strongest sensitivity at low ranks. Larger models (ViT-B/16, ViT-L/14) remain stable even with small r, and performance generally saturates around $r \approx 12$.

Effect of the number of source task vectors. Fixing $r = 12$, we vary the number of task vectors used to form the basis. Figure 6b shows rapid gains when adding the first few sources—again most pronounced for ViT-B/32—after which performance plateaus as the subspace becomes sufficiently well spanned.

6. Conclusion

We presented BOLT, a spectral adaptation framework that constructs layer-wise orthogonal bases from task vectors and adapts new tasks by learning only diagonal coefficients in this shared subspace. This design removes the reliance on any meta-training stage and instead provides a compact, controllable update space grounded in task-informed spectral coordinates. Empirically, BOLT delivers strong few-shot performance, robust out-of-domain generalization, and effective label-free test-time adaptation across multiple CLIP backbones, consistently matching or surpassing strong parameter-efficient tuning baselines. These results demonstrate that orthogonal, task-informed spectral coordinates offer a simple and scalable mechanism for low-parameter transfer in large pre-trained models.

7. Acknowledgments

K. Lee acknowledges partial support from the U.S. National Science Foundation under grant IIS 2338909.

References

[1] Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022. 3

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 6

[3] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 2

[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019. 1

[5] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 6

[6] Woojin Cho, Kookjin Lee, Donsub Rim, and Noseong Park. Hypernetwork-based meta-learning for low-rank physics-informed neural networks. *Advances in Neural Information Processing Systems*, 36:11219–11231, 2023. 3

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 6

[8] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6

[9] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 6

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[11] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 1, 3

[13] Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18695–18705, 2025. 1, 3, 4

[14] Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, et al. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *Advances in Neural Information Processing Systems*, 36:29343–29371, 2023. 1

[15] 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. 6

[16] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 7

[17] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021. 1, 2

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 2, 6

[19] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 791–800, 2018. 3

[20] Yunshi Huang, Fereshteh Shakeri, Jose Dolz, Malik Boudiaf, Houda Bahig, and Ismail Ben Ayed. Lp++: A surprisingly strong linear probe for few-shot clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23773–23782, 2024. 2, 6

[21] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022. 1, 2

[22] Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. Model stock: All we need is just a few fine-tuned models. In *European Conference on Computer Vision*, pages 207–223. Springer, 2024. 1

[23] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022. 2

[24] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019. 3

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[26] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 6

[27] Haifeng Li, Xin Dou, Chao Tao, Zhixiang Wu, Jie Chen, Jian Peng, Min Deng, and Ling Zhao. Rsi-cb: A large-scale

remote sensing image classification benchmark using crowdsourced data. *Sensors*, 20(6):1594, 2020. 6

[28] Haifeng Li, Hao Jiang, Xin Gu, Jian Peng, Wenbo Li, Liang Hong, and Chao Tao. Clrs: Continual learning benchmark for remote sensing image scene classification. *Sensors*, 20 (4):1226, 2020. 6

[29] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022. 2

[30] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025. 1, 8

[31] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024. 2

[32] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6

[33] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022. 1, 3

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, page 7. Granada, 2011. 6

[35] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 3

[36] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 6

[37] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36:66727–66754, 2023. 2

[38] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 6

[39] Xiaoman Qi, Panpan Zhu, Yuebin Wang, Liqiang Zhang, Junhuan Peng, Mengfan Wu, Jialong Chen, Xudong Zhao, Ning Zang, and P Takis Mathiopoulos. Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:337–350, 2020. 6

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1, 2, 6

[41] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017. 3

[42] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 8

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 7

[44] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021. 3

[45] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013. 6

[46] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 8

[47] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011. 6

[48] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pages 210–218. Springer, 2018. 6

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[50] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019. 8

[51] Qi Wang, Shaoteng Liu, Jocelyn Chanussot, and Xuelong Li. Scene classification with recurrent attention of vhr remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):1155–1167, 2018. 6

[52] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022. 2

[53] Gui-Song Xia, Wen Yang, Julie Delon, Yann Gousseau, Hong Sun, and Henri Maître. Structural high-resolution satellite image indexing. In *ISPRS TC VII Symposium-100 Years ISPRS*, pages 298–303, 2010. 6

[54] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017. 6

[55] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[56] Zhifeng Xiao, Yang Long, Deren Li, Chunshan Wei, Gefu Tang, and Junyi Liu. High-resolution remote sensing image retrieval based on cnns from a dimensional perspective. *Remote Sensing*, 9(7):725, 2017. 6

[57] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023. 3

[58] Enneng Yang, Li Shen, Zhenyi Wang, Shiwei Liu, Guibing Guo, and Xingwei Wang. Data augmented flatness-aware gradient projection for continual learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5630–5639, 2023. 3

[59] Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024. 2

[60] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010. 6

[61] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019. 3

[62] Frederic Z Zhang, Paul Albert, Cristian Rodriguez-Opazo, Anton van den Hengel, and Ehsan Abbasnejad. Knowledge composition using task vectors with learned anisotropic scaling. *Advances in Neural Information Processing Systems*, 37:67319–67354, 2024. 3, 6

[63] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pages 493–510. Springer, 2022. 2, 6

[64] Lijun Zhao, Ping Tang, and Lianzhi Huo. Feature significance-based multibag-of-visual-words model for remote sensing image scene classification. *Journal of Applied Remote Sensing*, 10(3):035004–035004, 2016. 6

[65] Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3718–3727, 2023. 3

[66] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 1

[67] Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS journal of photogrammetry and remote sensing*, 145:197–209, 2018. 6

[68] Qiqi Zhu, Yanfei Zhong, Bei Zhao, Gui-Song Xia, and Liangpei Zhang. Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, 13(6):747–751, 2016. 6

[69] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2605–2615, 2023. 1

[70] Qin Zou, Lihao Ni, Tong Zhang, and Qian Wang. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and remote sensing letters*, 12(11):2321–2325, 2015. 6