# ARCHITECTURE-DRIVEN SHIFT: A LIGHTWEIGHT PROXY FOR LOGIT-SHIFT TRENDS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Continual Learning (CL) is a practical paradigm to utilize power of deep pre-trained neural networks, but which pre-trained model has a better ability to balance "Plasticity-Stability", deserving to be chosen? The logit shift serves as a natural proxy, which represents the logit shift in CL. However, obtaining the logit shift requires huge computational cost, which hinders large-scale model selection. Existing theoretical analyses fail to offer an efficient alternative because of the assumption of uniform hidden layer widths, which ignores the structural heterogeneity (variable width and depth) of real-world architectures. This raises a critical question: **what theoretical relationship can be identified between heterogeneous architecture and logit shift on prior tasks (that the model has been trained on)?** To answer the question, we decouple logit shift into architecture dependency and data dependency to establish our framework, which reveals that **the combination of two dependency, defined as Architecture-driven Shift (ADS), can capture the logit shift tendency well computable via a lightweight calibration using a few data samples and architectures.** Specifically, for a well-optimized model on prior tasks, higher ADS is associated with a larger logit shift after training on the current task, which derived based on three mechanistic components: (1) spectral norm scaling of weight matrix gradients with layer width, (2) the optimization path length of the new task, and (3) the asymptotic task conflict in wide networks. Extensive empirical results across more than 175 diverse architectures demonstrate a strong monotonic correlation (the weakest Spearman's $r_s = 0.731$) between ADS and logit shift on 5 datasets 12 scenarios. Practically, we demonstrate that ADS can serve as a lightweight proxy of the expected calibration error, which is a widely used metric for reliable model selection.

## 1 INTRODUCTION

With the growing capacity of pretrained deep neural networks (DNNs), Continual Learning (CL) is a practical paradigm that requires a pre-trained model sequentially adapt to a stream of new tasks (Plasticity) while maintaining performance on the original task as much as possible (Stability) (Li & Hoiem, 2017; Wu et al., 2019). For different applications, we can select a more suitable model from several pre-trained models according to the "Plasticity-Stability" trade-off. For instance, in scenarios where stability is prioritized, we select the model that has a higher stability. Logit shift is a natural and general proxy for quantifying "Plasticity-Stability" trade-off in CL (Benjamin et al., 2019; Guha & Lakshman, 2024), yet its computation is expensive because it requires a full training process on new tasks, forcing researchers to rely on CL metrics (e.g. average accuracy and forgetting) instead.

Since CL metrics based on output are computed directly from the model's logits, any architectural influence observed on these metrics can originate from an underlying architectural dependency within the logit shift itself. Prior studies have observed correlations between architectural variables and CL metrics, suggesting that that **logit shift contains an inherent architecture dependency that may be computed based on the architectural variables alone, enabling us to bypass the full training process.** Specifically, enlarging the width of hidden layers (Mirzadeh et al., 2022) or overall



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

model scale (Ramasesh et al., 2021) serves as a simple method for improving learning accuracy and mitigating catastrophic forgetting[1].

Yet these studies remain largely at the level of the CL metric, which is merely downstream consequences. To gain a more fundamental mechanistic understanding, it is crucial to analyze logit shift directly. Although there has been growing interest in modeling the relationship between architectures and logit shift (Guha & Lakshman, 2024; Kim et al., 2025), these studies rely on the assumption of a homogeneous architecture (uniform width of hidden layers), which is insufficient to provide practical guidance for real-world model selection. This motivates us to establish a more comprehensive theoretical analysis to cover heterogeneous architecture.

In this paper, we propose a theoretical framework that explicitly models the relationship between the heterogeneous architecture of the typical fully-connected neural network (FNN) and logit shift, which we define as Architecture-driven Shift (ADS). As mentioned above, logit shift contains an inherent architecture dependency that is directly determined by models' width and depth, which reduce computational resource costs. Therefore, we conceptually decouple logit shift into two components: (1) The architecture dependency and (2) the data dependency that is efficiently calibrated through a subset of the dataset.

To theoretically construct ADS, our analysis proceeds in both static structural properties and dynamic optimization behaviors. On the static structural level, we formulate the mathematical link between the width of hidden layers and the spectral norm of the weight gradients (Proposition 4.1). Specifically, we characterize how the spectral norm inherently scales with layer width, driven by the varying energy of forward activations. Concurrently, on the dynamic optimization level, we decouple the adaptation process into macro- and microscopic phenomena. Macroscopically, we establish how the width of hidden layers constrains the scale of the optimization trajectory (such as parameter displacement) during new task learning (Lemma 4.3). Microscopically, we reveal an interesting property of task interference: as networks grow wider, random high-dimensional noise asymptotically diminishes, purifying the cross-task conflicts into a deterministic rule governed by semantic task representations rather than random gradient conflicts (Eq. (12)).

By unifying static structural properties and dynamic optimization behaviors, we derive ADS, which can be efficiently computed using only a few calibration subset of data and architectures, enabling architectural variables to capture the logit shift tendency and paving the way for large-scale CL model selection. Specifically, for reliable CL model selection based on Expected Calibration Error (ECE), we show that ADS can serve as a lightweight coarse-grained selector to avoid training every candidate architecture on the full data.

The main contributions can be summarized as follows: (1) **Theoretical Framework for Heterogeneous Architecture of FNN.** To the best of our knowledge, we present the first theoretical framework to analyze a finer-grained object, i.e., logit shift, instead of the output-based metrics in CL with realistic, heterogeneous architectures (variable widths and depths). This bridges a critical gap in CL theory, as prior analyses were restricted to simplified, uniform-width models and thus could not provide practical guidance for real-world model selection. (2) **Mechanistic Decomposition of logit shift.** Our framework provides a novel mechanistic decomposition of the logit shift. We demonstrate that this complex phenomenon is governed by the synthesis of three distinct analytical components: a static scaling law of the weight gradients (Proposition 4.1), the macroscopic dynamics of the optimization trajectory (Lemma 4.3), and the microscopic purification of task semantic interference in wide networks (Eq. (12)). This decomposition offers an mechanistic insight into how architecture shapes logit-space dynamics. (3) **Empirical Validation and Application.** We provide comprehensive empirical evidence confirming the predictions of our theoretical framework across diverse architectures of FNN and Transformer. Through comprehensive experiments across a diverse suite of tasks, i.e., ranging from easy datasets like (Split/Rotated) MNIST (LeCun et al., 1998) and FashionMNIST (Xiao et al., 2017), to more complex datasets such as CIFAR-10/CIFAR-100 (Krizhevsky, 2009) and ImageNet (Deng et al., 2009) subsets, and both transfer and continual learning paradigms, we demonstrate that the ADS maintains a strong monotonic correspondence with the empirically observed logit shift. Crucially, we validate ADS as a lightweight selector for ECE, which is a commonly used metric for reliable CL model selection.

[1]Catastrophic Forgetting refers to the phenomenon where models sequentially adapt to new tasks, requiring parameter updates, which inadvertently disrupt the representations of previous tasks in CL. (McCloskey & Cohen, 1989)

# 2 Related Works

## 2.1 Methodological Approaches and Computational Bottlenecks in CL

Continual Learning (CL) aims to enable deep neural networks to sequentially adapt to a stream of new tasks. A fundamental challenge in this paradigm is catastrophic forgetting[2], where updating parameters for current tasks disrupts previously learned representations. Extensive research has concentrated on addressing this problem from three main perspectives: regularization-based, replay-based, and optimization-based methods.

Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017), one of the classic regularization-based methods, estimates parameter importance by calculating the diagonal of the Fisher Information Matrix after learning a task. It mitigates forgetting by penalizing changes to these critical parameters during subsequent sequential training. In contrast, replay-based approaches, such as Continual Learning with Experience And Replay (CLEAR) (Rolnick et al., 2019), focus on the data-level. They store a small subset of samples from previous tasks in an episodic memory and interleave them with current task data during training to preserve learned knowledge. Rather than manipulating parameters or datasets, Orthogonal Gradient Descent (OGD) (Farajtabar et al., 2020) directly intervenes in the optimization space. OGD projects the parameter gradients of new tasks onto the orthogonal complement of the subspace spanned by the gradients of prior tasks, mathematically ensuring that the predictions on earlier tasks remain invariant under gradient updates.

While these methodologies exhibit strong empirical performance, they all share a common limitation: they rely heavily on computationally expensive active training interventions (such as calculating high-dimensional Fisher matrices, managing memory buffers, or computing gradient projection matrices during training). This hinders large-scale, pre-training stage model selection.

## 2.2 Architectural Influence and Training Dynamics in CL

To address this bottleneck, recent research has explored the possibility of performing lightweight, pre-training model selection by analyzing the model's intrinsic architecture, rather than relying on full downstream training runs. For instance, Mirzadeh et al. (2022) demonstrated that simply enlarging the network width significantly mitigates catastrophic forgetting. They hypothesized that wider networks naturally exhibit greater gradient orthogonality and align with the "lazy training" regime (Chizat et al., 2019), where parameters move minimally from their initialization, thus preserving learned representations. Mathematically, this lazy training dynamic in overparameterized networks is closely characterized by the Neural Tangent Kernel (NTK) theory (Jacot et al., 2018; Lee et al., 2019). Under this infinite-width limit, several theoretical works have analyzed the generalization guarantees of gradient projection methods (Bennani et al., 2020) and have mathematically bounded catastrophic forgetting through the NTK overlap matrix (Doan et al., 2021). Extending beyond these idealized infinite-width regimes to practical settings, Lu et al. (2024) conducted a comprehensive empirical analysis of various finite-width architectural dimensions (such as depth, activation functions, and normalization layers) to map their impact on CL metrics.

These macroscopic studies successfully established the empirical link between architecture and downstream CL metrics (e.g. average accuracy and forgetting), which are too coarse to realize and analyze the trade-off between plasticity (the ability to acquire new knowledge) and stability (the preservation of historical knowledge). To theoretically understand how architecture intrinsically influences the trade-off, recent studies have moved beyond aggregate CL metrics toward finer-grained changes in model behavior, including function-/logit-space shifts (Benjamin et al., 2019; Guha & Lakshman, 2024) and hidden representation shifts (Kim et al., 2025), where logit shift captures representation change at the pre-softmax output layer.

The work most closely related to ours is Guha & Lakshman (2024), which initiated the formal theoretical analysis of these phenomena. They leveraged perturbation theory to analyze fully-connected neural networks (FNNs), proving that increasing network width directly bounds the logit shift on prior tasks, although with diminishing returns.

[2]First discussed by McCloskey & Cohen (1989).

Despite these valuable advancements, a major theoretical gap remains: existing studies Guha & Lakshman (2024); Kim et al. (2025) rely on the simplification of a **homogeneous architecture**, i.e., assuming uniform hidden layer widths across the network. However, real-world models are highly heterogeneous, featuring varying layer-wise widths and complex block depths. Our work directly addresses this gap by presenting a theoretical framework designed for heterogeneous architectures. By conceptually decoupling the logit shift into an architecture dependency and a data dependency, we propose the Architecture-driven Shift (ADS), which provides a theoretically grounded, computationally efficient proxy that captures the trajectory of logit shift with only a calibrated subset of architectures and data, facilitating large-scale model selection.

# 3 PRELIMINARY

## 3.1 NOTATION

Consider a fully-connected neural network (FNN) $f_t$ trained on the $t$-th task, with $L$ layers and hidden layer widths $\boldsymbol{w} = [w^{(1)}, \ldots, w^{(L)}]$. Let $w^{(0)} = \dim_{in}$ and $w^{(L+1)} = \dim_{out}$ denote the input and output dimensions, respectively. For any layer $l \in \{1, \ldots, L\}$, we define forward propagation and backpropagation as follows. During forward propagation, the input $\boldsymbol{x}$ is passed through each layer $l$ to compute the activation $\boldsymbol{a}^{(l)} = \sigma(\boldsymbol{z}^{(l)}) = \sigma(\Theta^{(l)}\boldsymbol{a}^{(l-1)})$, where $\Theta^{(l)}$ is the weight matrix of layer $l$, and $\sigma$ is the activation function. Following the convention in this area Guha & Lakshman (2024), we ignore the bias terms for simplicity. In the backpropagation phase, we define the error signal $\delta^{(l)}$ of the $l$-th layer, which represents the gradient of the loss $\mathcal{L}$ with respect to the layer's pre-activation $\boldsymbol{z}^{(l)}$. Let $\Theta = \mathrm{vec}(\{\Theta^{(l)}\}_{l=1}^{L}) \in \mathbb{R}^P$ means the vectorization of all model parameters.

Let $J_\Theta f_t(\boldsymbol{x}) \in \mathbb{R}^{C\times P}$ be the full-logit Jacobian, where $C$ is the number of classes and $P$ is the number of parameters. Defining $\boldsymbol{u}_t^* = \Delta f_t / \|\Delta f_t\|_2$ for $\Delta f_t = f_{t+1} - f_t$, we let $\boldsymbol{g}_{\text{old}} \triangleq J_\Theta f_t(\boldsymbol{x})^\top \boldsymbol{u}_t^* = \nabla_\Theta[\boldsymbol{u}_t^{*\top} f_t(\boldsymbol{x})]$. We further define the update direction at the $s$-th training step on task $t+1$ as $\boldsymbol{g}_{new}(s) \triangleq \nabla_{\Theta_{(t+1,s)}} \mathcal{L}_{t+1}$; $\|\cdot\|_F$ denotes the Frobenius norm.

## 3.2 PROBLEM SETUP

Formally, we formulate the process of continual learning (CL) as a sequential learning task. A heterogeneous ReLU FNN, parameterized by $\Theta$, with variable width $\boldsymbol{w}$ and depth $L$, is trained on a stream of datasets $D_1, \ldots, D_T$. Each task $t$ involves $S$ steps of supervised learning on $D_t = (\mathcal{X}_t, \mathcal{Y}_t)$, and we assume $S$ steps of training enable the model to perform well on $D_t$. Under this sequential setting, we aim to learn a ReLU FNN that requires navigating the fundamental "Plasticity-Stability" trade-off. Specifically, the model must exhibit the ability to adapt to each new task $D_t$ while retaining performance on previous tasks $D_{1:t-1}$. To mathematically quantify the trade-off, we adopt the definition of **logit shift** originally proposed by Benjamin et al. (2019); Guha & Lakshman (2024) as our core analytical objective, which is defined as the expected variation in the logit space:

$$\mathcal{S}_{t\to t+1}(\boldsymbol{x}) := \|f_{t+1}(\boldsymbol{x};\Theta_{t+1}) - f_t(\boldsymbol{x};\Theta_t)\|_2, \boldsymbol{x} \in \mathcal{D}_{1:t}, \tag{1}$$

where $f_t(\boldsymbol{x};\Theta_t)$ is simplified as $f_t(\boldsymbol{x})$ for brevity in the following discussion.

**Our Objective.** Computing $\mathcal{S}_{t\to t+1}$ requires a full training on the complete dataset $D_{1:t}$, making it computationally prohibitive for large-scale model selection. Furthermore, we recognize that $\mathcal{S}_{t\to t+1}$ inherently contains an architectural dependency that may offer a theoretical pathway to capture the logit shift avoid the full training on the complete dataset.

However, since the training process is fundamentally driven by data, a pure decoupling of architecture and data is theoretically unattainable. Instead, our strategy is to explicitly extract the architecture dependency ($\boldsymbol{w}$, $l$ and $L$) while capturing the unavoidable data-dependent factors of optimization dynamics through a lightweight calibration process[3] on a subset of data and architectures.

Therefore, our primary objective is to formulate a proxy, Architecture-Driven Shift (ADS), that captures the tendency of $\mathcal{S}_{t\to t+1}$ primarily using the architectural variables ($\boldsymbol{w}$, $l$, and $L$). We formulate

[3]More details of this lightweight process are available in Table 2 and Appendix A.2.1.

ADS as $\mathcal{F}(\boldsymbol{w}, L, D_{\text{cal}})$, and empirically evaluate its monotonic correspondence with $S_{t\to t+1}$ in section 5. Here, $D_{\text{cal}} \subset D_{1:t}$ is a subset used to calibrate the data-dependent factors. Formalizing $\mathcal{F}$ enables lightweight evaluation of "Plasticity-Stability" for heterogeneous-architecture models.

### 3.3 Theoretical Groundings

Before presenting our main theoretical framework, we introduce the necessary theoretical groundings as follows. First, to ensure the network operates in a trainable regime where the aforementioned error signals do not vanish or explode exponentially with depth, we rely on the following prior:

**Assumption 3.1** (Non-degeneracy of Layer-wise Weights)**.** Consider an $L$-layer FNN with a Lipschitz continuous activation function $\sigma(\cdot)$ (where $\sigma(0) = 0$) and zero biases. The layer-wise weight matrices $\Theta^{(l)}$ for $l \in \{1, L\}$ follow a non-degenerate distribution with variance satisfying:

$$\text{Var}(\Theta^{(l)}) \gg 0 \iff \mathbb{E}[(\Theta^{(l)})^2] \gg \left(\mathbb{E}[\Theta^{(l)}]\right)^2. \tag{2}$$

Because microscopically analyzing CL optimization process requires realizing how sequential tasks gradients interact in high dimensions, we introduce next lemmas.

**Lemma 3.2** (Levy's Lemma on the Sphere (Vershynin, 2018))**.** *Let $\boldsymbol{v}_{noise}$ be a random vector uniformly distributed on the unit sphere in the effective parameter space $\mathbb{R}^{D_{eff}^{(l)}}$. For any 1-Lipschitz function $F$, such as the gradient projection $F(\boldsymbol{v}) = \boldsymbol{g}_{\text{old}}^{\top}\boldsymbol{v}$ (assuming $\|\boldsymbol{g}_{\text{old}}\|_2 = 1$), and for any scalar $\epsilon > 0$, the probability measure satisfies:*

$$P\left(|\boldsymbol{g}_{\text{old}}^{\top}\boldsymbol{v}_{noise}| \geq \epsilon\right) \leq 2\exp\left(-\frac{D_{eff}^{(l)}\epsilon^2}{2}\right). \tag{3}$$

**Lemma 3.3** (Asymptotic Vanishing of the Effective Dimension)**.** *Let the gradient space at layer $l$ admit the decomposition by the orthogonal direct sum: $\mathbb{R}^{N^{(l)}} = V_{\text{signal}} \oplus V_{\text{noise}}$, where $N^{(l)} = w^{(l)}w^{(l-1)}$. Suppose $\dim(V_{\text{signal}}) = k$ is fixed as the width grows. Then, in the limit $w^{(l)} \to \infty$,*

$$\lim_{w^{(l)}\to\infty} \sqrt{\log(L/\delta)/D_{\text{eff}}^{(l)}} = 0. \tag{4}$$

*Proof.* See Lemma A.9 for the full proof. □

Based on the groundings above, we now have enough theoretical tools to model three critical theoretical components: (1) the scaling of the spectral norm of weight gradients with layer width, (2) the relationship between architecture and the optimization path length, and (3) the asymptotic nature of task conflict in wide networks. Therefore, the expression of ADS can be derived in Section 4. [4]

## 4 Theoretical Framework Derivation

### 4.1 Formulation: Decomposing the logit shift

Our primary objective is to formulate a proxy, Architecture-Driven Shift (ADS), that captures the tendency of $\mathcal{S}_{t\to t+1}$ primarily using the architectural variables ($\boldsymbol{w}$ and $L$). Because we don't have any prior knowledge about ADS, we try to model it starting from the logit shift, which is clearly defined in Eq. (1). Furthermore, under the local linearization assumption commonly adopted in deep learning theory analyses Jacot et al. (2018); Lee et al. (2019), we analyze the logit shift using a first-order Taylor approximation around the parameters $\Theta_i$:

$$\|f_{i+1}(\boldsymbol{x}) - f_i(\boldsymbol{x})\|_2 \lesssim \left|\boldsymbol{u}_i^{*\top}(\boldsymbol{x})J_{\Theta}f_i(\boldsymbol{x})\Delta\Theta_i\right| \leq \sum_{l=1}^{L} \underbrace{\|\boldsymbol{g}_{\text{old}}^{(l)}\|_F \int_0^S \|\boldsymbol{g}_{\text{new}}^{(l)}(s)\|_F \cdot |\cos\theta_s^{(l)}| ds}_{\text{Term}^{(l)}}, \tag{5}$$

where $\theta_s^{(l)}$ is the angle between $\boldsymbol{g}_{\text{old}}^{(l)}$ and $\boldsymbol{g}_{\text{new}}^{(l)}(s)$. Next, we will analyze each term in $\text{Term}^{(l)}$ independently, with each term corresponding to a distinct theoretical component.

[4] We provide one roadmap of the derivation in Appendix Figure 4.

### 4.2 Static Structural Property: Width-wise Gradient Norm Scaling

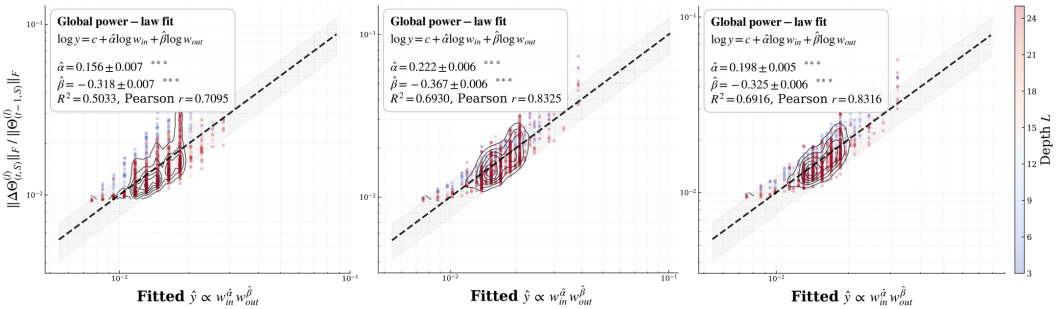
**Proposition 4.1.** *(Gradient Scaling for Weight Matrix) Let* $f_{i,\boldsymbol{u}^*}(\boldsymbol{x}) \triangleq \boldsymbol{u}_i^{*\top} f_i(\boldsymbol{x})$. $\boldsymbol{g}_{\text{old}}$ *satisfies:*

$$\boldsymbol{g}_{\text{old}}^{(l)} = \nabla_{\Theta^{(l)}} f_{i,\boldsymbol{u}^*}(\boldsymbol{x}) = \delta^{(l|\boldsymbol{u}^*)} (a^{(l-1)})^\top, \tag{6}$$

$$\|\boldsymbol{g}_{\text{old}}^{(l)}\|_F = \|\boldsymbol{g}_{\text{old}}^{(l)}\|_2 \propto \sqrt{w^{(l-1)}}. \tag{7}$$

*Proof.* See details in Proposition A.7 of Appendix. □

### 4.3 Macroscopic Optimization Dynamics: Scaling of the Optimization Path




(a) FashionMNIST → MNIST (b) MNIST → CIFAR-10 (c) FashionMNIST → CIFAR-10

Figure 1: Empirical Validation of the Width Term Assumption in FashionMNIST → CIFAR-10. More details are available in Appendix A.2.2.

**Empirical Observation (Width Scaling):** Inspired by the empirical observation in Figure 1, we extend the width scaling assumption in (Guha & Lakshman, 2024) to satisfies the variable width of real-world model as follows.

**Assumption 4.2.** Let $\Theta_{(t,S)}^{(l)} \in \mathbb{R}^{w^{(l)} \times w^{(l-1)}}$ be the $l$-th layer weight matrix that has been updated $S$ steps on the $t$-th task.

$$\frac{\|\Theta_{(t,S)}^{(l)} - \Theta_{(t-1,S)}^{(l)}\|_F}{\|\Theta_{(t-1,S)}^{(l)}\|_F} \propto (w^{(l-1)})^\alpha (w^{(l)})^\beta, \tag{8}$$

where $\alpha$ and $\beta$ means task-specific factor that are calibrated from a calibration dataset $D_{\text{cal}}$.

**Lemma 4.3.** *To quantify the distance of the optimization trajectory, we define* $\text{PathLen}^{(l)} \triangleq \int_0^S \|\nabla_{\Theta_{(i+1,r_k)}^{(l)}} \mathcal{L}_{i+1}\|_F dr_k$*, i.e., the integral of the Frobenius norm over* $S$ *training steps. Under Assumptions 4.2 and A.1, the path length yields the stated proportionality:*

$$\text{PathLen}^{(l)} \propto (w^{(l-1)})^\alpha (w^{(l)})^{\beta+\frac{1}{2}}. \tag{9}$$

*Proof.* See in Appendix A.10. □

### 4.4 Microscopic Optimization Dynamics: Asymptotic Purification of Task Conflicts

**Empirical Observation (Architecture-Dependent Depth Profiles):** Motivated by observations in Figure 2, we introduce an assumption characterizing the relationship between depth $l$ and semantic task conflict signals $\rho_{task}$.

**Assumption 4.4.** Let $\cos\theta$ be the cosine similarity of the output sensitivity $\boldsymbol{g}_{old}$ and update direction $\boldsymbol{g}_{new}(\tau)$. Its deterministic task component satisfies:

$$\rho_{task}^{(l)} \propto l^b \cdot e^{-cl}, b > 0, c \geq 0, \tag{10}$$

where $b, c$ are parameters calibrated on $D_{\text{cal}}$. The boundary case $c = 0$ describes monotonic depth amplification, which is a common phenomenon in prior works (Guha & Lakshman, 2024; Kim et al., 2025) and is observed in the right panel of Figure 2.

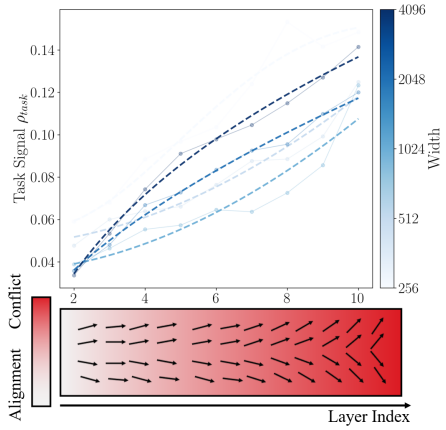



Figure 2: Empirical Evidence of the Architecture-Dependent Depth Profiles (Assumption 4.4). More empirical evidence see in Appendix A.2.3. Bottom arrows depict alignment/conflict levels.

Table 1: Full-pool verification of statistical correlation between **ADS** and **logit shift** over 175 FNN architectures per scenario. These results, averaged over 3 random seeds, verify the statistical correspondence predicted by ADS. Out-of-sample predictive results under disjoint architecture and data splits are reported in Table 2. For simplicity, we refer to MNIST, Fashion-MNIST, and CIFAR-10 as M, F, and C, and abbreviate Direction Consistency as DC. *** indicates $p < 0.001$.

| SCENARIO | SPEARMAN $r_s$ | KENDALL $r_k$ | DC |
|---|---|---|---|
| M → F | 0.920*** | 0.761*** | 88.06% |
| M → C | 0.881*** | 0.683*** | 84.14% |
| F → M | 0.955*** | 0.810*** | 90.52% |
| F → C | 0.731*** | 0.526*** | 76.32% |
| SPLIT M | 0.738*** | 0.560*** | 78.00% |
| SPLIT F | 0.814*** | 0.631*** | 82.20% |
| ROTATED M | 0.802*** | 0.619*** | 80.96% |
| ROTATED F | 0.960*** | 0.833*** | 91.67% |

**Proposition 4.5.** *(Signal-Noise Decomposition under High Dimensionality) Under the Signal-Noise Decomposition hypothesis, the cosine similarity between output sensitivity $\boldsymbol{g}_{old}$ and update direction $\boldsymbol{g}_{new}(\tau)$ at layer $l$ can be decomposed into a deterministic task-alignment term and a stochastic high-dimensional noise term. Let $\rho^{(l)}_{task}$ represents the semantic conflict component. With probability at least $1-\delta$, the cosine similarity satisfies the following scaling law:*

$$\cos\theta^{(l)} \propto \left( |\rho^{(l)}_{task}| + \sqrt{\frac{\log(L/\delta)}{D^{(l)}_{eff}}} \right). \tag{11}$$

*Proof.* See in Appendix A.8. □

Based on Proposition 4.5 and Lemma 3.3, the gradient cosine similarity is dominated by the deterministic signal component, which satisfy Assumption 4.4. The relationship simplifies to:

$$\cos\theta^{(l)} \approx \rho^{(l)}_{\text{task}} \propto l^b \cdot e^{-cl}. \tag{12}$$

**Remark:** This implies that in wide layers, any significant gradient interference ($\cos\theta^{(l)} \neq 0$) stems mainly from semantic task conflict ($\rho^{(l)}_{\text{task}}$), not random parameter collisions.

### 4.5 SYNTHESIS: DERIVING THE ARCHITECTURE-DRIVEN SHIFT

Incorporating Lemma 4.3 and Eq. (12), the total contribution of layer $l$ satisfies:

$$\text{Term}^{(l)} \lesssim \sqrt{w^{(l-1)}} \cdot w^{(l-1)\,\alpha} \cdot w^{(l)\,\beta} \cdot |l^b e^{-cl}| = (w^{(l-1)})^{\alpha+\frac{1}{2}} (w^{(l)})^{\beta+\frac{1}{2}} \cdot |l^b e^{-cl}|. \tag{13}$$

Substituting it into Eq. (5):

$$\underbrace{\|f_{i+1}(\boldsymbol{x}) - f_i(\boldsymbol{x})\|_2}_{\mathcal{S}_{t\to t+1}(\boldsymbol{x})} \lesssim \underbrace{\sum_{l=1}^{L} (w^{(l-1)})^{\alpha+\frac{1}{2}} (w^{(l)})^{\beta+\frac{1}{2}} |l^b e^{-cl}|}_{\mathcal{F}\left(\boldsymbol{w}, L, D_{\text{cal}}\right)}. \tag{14}$$

In conclusion, by abstracting the left- and right-hand side of Eq. (14) into $\mathcal{S}_{t\to t+1}$ and $\mathcal{F}\left(\boldsymbol{w}, L, D_{\text{cal}}\right)$, respectively, we fulfill the objective set forth in Section 3.2 and enables direct navigation of the "Plasticity-Stability" trade-off, without requiring full new-task training for each candidate architecture.

Table 2: Spearman correlations $r_s$ between ADS and the measured logit shift under the architecture-level and data-level split protocol across scenarios and model families (175 FNNs, 35 CNNs with skip-connection mechanism and 30 ViTs). For simplicity, we refer to MNIST, FashionMNIST, CIFAR-10/100 and ImageNet-1k as M, F, C-10/100 and I, respectively. ADS parameters are fitted only on the calibration architectures $A_{\text{cal}}$ using logit shifts measured on the calibration data subset $D_{\text{cal}}$, and are evaluated on held-out architectures $A_{\text{val}}$ using the held-out data subset $D_{\text{test}}$. In all experiments, we maintain $D_{\text{cal}} : D_{\text{test}} = 3 : 7$. The first, second and third header rows under "Architecture Calibration Ratio" correspond to FNNs, CNNs with skip connections and ViTs, respectively. ADS saves computation by measuring logit shifts only on $A_{\text{cal}} \times D_{\text{cal}}$, instead of $(A_{\text{cal}} \cup A_{\text{test}}) \times (D_{\text{cal}} \cup D_{\text{test}})$. More details of calibration are available in Appendix A.2.1.

| Model | Scenarios | Architecture Calibration Ratio (%) | | | | | Average$_{\pm\text{Std}}$ |
|---|---|---|---|---|---|---|---|
| | | **5**<br>**25.71**<br>**26.67** | **10**<br>**31.43**<br>**30** | **15**<br>**34.29**<br>**36.67** | **20**<br>**40**<br>**40** | **25**<br>**45.71**<br>**46.67** | |
| FNN | M → F | 0.762 | 0.875 | 0.881 | 0.879 | **0.882** | $0.856_{\pm 0.053}$ |
| | F → M | 0.890 | 0.932 | 0.931 | 0.932 | **0.939** | $0.925_{\pm 0.020}$ |
| | Split M (0–4 → 5–9) | 0.689 | 0.858 | **0.880** | 0.846 | 0.878 | $0.830_{\pm 0.080}$ |
| | Split M (5–9 → 0–4) | 0.794 | **0.894** | 0.879 | 0.856 | 0.879 | $0.860_{\pm 0.039}$ |
| | Rotated M (5 tasks) | 0.752 | 0.616 | **0.756** | 0.600 | 0.618 | $0.668_{\pm 0.078}$ |
| | Rotated F (5 tasks) | 0.965 | 0.938 | **0.961** | 0.959 | 0.959 | $0.956_{\pm 0.011}$ |
| CNN | M → C | **0.934** | 0.882 | 0.874 | 0.890 | 0.895 | $0.895_{\pm 0.023}$ |
| | C → M | 0.334 | 0.342 | 0.423 | 0.465 | **0.570** | $0.427_{\pm 0.097}$ |
| | F → C | **0.643** | 0.570 | 0.510 | 0.507 | 0.453 | $0.537_{\pm 0.072}$ |
| | C → F | 0.949 | 0.915 | 0.949 | 0.948 | **0.954** | $0.943_{\pm 0.016}$ |
| ViT | C-100 → I | 0.647 | 0.694 | 0.722 | **0.813** | 0.806 | $0.736_{\pm 0.072}$ |
| | I → C-100 | 0.609 | **0.788** | 0.729 | 0.692 | 0.700 | $0.704_{\pm 0.065}$ |

## 5 Experiments

We organize our experiments around three questions: (1) **Effectiveness:** Does Architecture-Driven Shift (ADS) consistently capture logit shift trends across learning paradigms and model families? (See Table 1; more validations are provided in Appendix Figure 6.) (2) **Calibration Efficiency:** Can ADS, calibrated using small subsets of candidate architectures and data samples, generalize to held-out architectures and samples? (See Table 2; the detailed calibration protocol is available in Appendix A.2.1.) (3) **Practical Utility:** Can ADS efficiently prune candidate architectures for a specific application: ECE-based model selection? (See Figure 3 and Appendix Figure 7.)

**Evaluation Metrics.** To evaluate how well ADS captures the trend of the logit shift, we employ classic statistical metrics (Spearman's coefficient $r_s$, Kendall's coefficient $r_k$) along with the 95% confidence interval (CI), and introduce a new metric, *direction consistency* (DC):

$$DC = \frac{2\sum_{1\le i<j\le |A_{\text{test}}|}}{|A_{\text{test}}| \cdot (|A_{\text{test}}| - 1)} \mathbb{I}\left[\Delta ADS_{ij} \cdot \Delta \widehat{S}_{ij} > 0\right], \tag{15}$$

where $|A_{\text{test}}|$ is the number of architectures, $\mathbb{I}[\cdot]$ is the indicator function, and $\Delta(\cdot)_{ij}$ denotes the difference between the values of the corresponding variable for the $i$-th and $j$-th architectures. For empirical evaluation, we aggregate $S$ in Eq. 1 over the samples, i.e., measured logit shift $\widehat{S}$.

In reliable continual learning, ECE is a traditional metric to evaluate the reliability of models' output in neural network calibration literature Guo et al. (2017); Li et al. (2024).

$$\text{ECE} = \sum_{b=1}^{15} \frac{|B_b|}{|D_{\text{test}}|} \cdot |\text{conf}(B_b) - \text{acc}(B_b)|, \tag{16}$$

where $B_b$ denotes the set of sample indices whose predicted confidence falls into the $b$-th bin and $|D_{\text{test}}|$ means the number of data samples. Here, $\text{acc}(B_b) = \frac{1}{|B_b|}\sum_{i\in B_b} \mathbb{I}(\hat{y}_i = y_i)$ represents the average accuracy of the samples in bin $B_b$, and $\text{conf}(B_b) = \frac{1}{|B_b|}\sum_{i\in B_b} \hat{p}_i$ represents the average predicted confidence within that bin.

### 5.1 Theoretical Framework Verification

As shown in Table 1, across transfer learning and sequence CL, ADS exhibits a consistently significant monotonic correlation with logit shift and high DC, indicating it captures the logit shift trend. Appendix Figure 6 shows that ADS strongly correlates with logit shift in Vision Transformer, proposed by Dosovitskiy et al., suggesting ADS is effective across different model families.

As shown in Table 2, calibrated on only a small proportion of data (30%) and architectures, the fitted parameters are highly transferable across scenarios and model families, demonstrating that ADS can be efficiently adapted in practice with low computational requirements. For CNNs, we use a larger architecture calibration ratio because the candidate pool contains only 35 architectures. The absolute calibration cost remains small: the CNN calibration subsets contain only 10–20 architectures, and only 30% of the data. Although ADS performs robustly in most settings, we observe a notable failure case in the CIFAR-10 $\rightarrow$ MNIST for CNNs with skip connections, which may be associated with a larger discrepancy between nominal/truth block depth; see Appendix A.3.2 for details.

### 5.2 Model Selection based on ECE

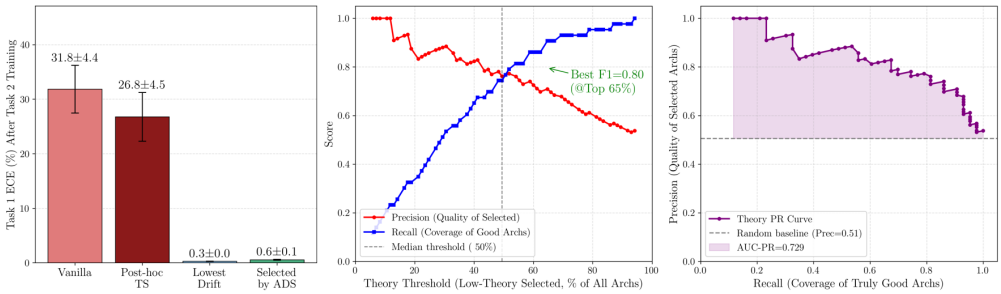



Figure 3: Quantitative evaluation of the ADS-based selector: Calibration efficacy, filtering trade-offs, and statistical robustness on Split MNIST. More validation scenarios see in Appendix Figure 7.

In reliable CL, ECE is a widely used calibration metric. We define the ECE drift below the median as the positive class to analyze the performance of ADS-based selector as shown in Figure 3.

As shown in Figure 3 (left), we apply Post-hoc Temperature Scaling (TS) (Guo et al., 2017) to a poorly-calibrated architecture (Vanilla), yielding a marginal reduction in ECE (31.8% $\rightarrow$ 26.8%). In contrast, the model selected by ADS achieves 0.6% ECE, statistically comparable to the theoretical optimum (0.3%). These results suggest that architectural selection complement post-hoc calibration methods, and that ADS performs this selection efficiently. Figure 3 (middle) shows that ADS enables reliable pruning of the candidate model search space. Specifically, selecting few top-ranked architectures yields high precision, which means that the selected candidates are more likely to exhibit low ECE drift. In comparison, a larger threshold brings high Recall, preserving mostly stable architectures and discarding likely high ECE drift ones. Figure 3 (right) evaluates the ranking capability and robustness of ADS through a Precision-Recall (PR) curve. Under diverse settings, Area Under the PR Curve (AUC-PR) ranges from 0.61 to 0.73, significantly outperforming the 0.51 random baseline. This threshold-independent performance validates that ADS captures intrinsic property of logit shift, rather than relying on superficial, dataset-specific statistical correlations.

## 6 Conclusion

In this paper, we propose a lightweight proxy, Architecture-driven Shift (ADS), computed from heterogeneous architectural variables using a few calibration samples and candidate architectures, which captures the logit shift trend in continual learning (CL). Our theoretical framework, which is based on both static structural properties and dynamic optimization behaviors, provides a mechanistic explanation of how architecture shapes logit-space dynamics. Experiments across diverse model families and scenarios show that ADS is strongly and monotonically correlated with logit shift. Moreover, an ADS-based selector enables pruning the model search space in terms of Expected Calibration Error (ECE), which decreases the huge computational requirements.

### AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript organization, English drafting, and language editing. AI tools were also used to help prepare plotting code and visualization scripts for the figures. The research design, theoretical framework, experimental protocols, and data analysis were developed by the authors. AI tools were used for discussion and consistency checks between the implementation and the manuscript. The experimental data and curves come from actual experiments, and all plots were generated by scripts based on data manually checked by the authors. We ran and tested the code and checked the reported numerical results, references, and equations. We have not used generative AI tools for theoretical derivations, experimental design, or result interpretation. We have reviewed all AI-assisted work. The authors take responsibility for the final content of this work, including all AI-assisted text, code, and figures.

### REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the complete theoretical framework in Appendix A.1, including the relevant assumptions and theoretical derivations. Additional experimental results and analyses are reported in Appendix A.2. The evaluation metrics used in our experiments are described in the main text (see 5). We also provide an anonymous repository at `https://anonymous.4open.science/r/Architecture-Driven-Shift-42`, which currently contains a fairly complete set of experimental results. We refer readers to these parts for the details needed to reproduce and verify our theoretical and empirical results.

## REFERENCES

Ari Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SkMwpiR9Y7`.

Mehdi Abbana Bennani, Thang Doan, and Masashi Sugiyama. Generalisation guarantees for continual learning with orthogonal gradient descent. *arXiv preprint arXiv:2006.11942*, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. URL `https://api.semanticscholar.org/CorpusID:57246310`.

Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A theoretical analysis of catastrophic forgetting through the ntk overlap matrix. In *International Conference on Artificial Intelligence and Statistics*, pp. 1072–1080. PMLR, 2021.

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. *ICLR*, 2021.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

Etash Kumar Guha and Vihan Lakshman. On the diminishing returns of width for continual learning. In *International Conference on Machine Learning*, pp. 16706–16730. PMLR, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Chirag Gupta, Sivaraman Balakrishnan, and Aaditya Ramdas. Path length bounds for gradient descent and flow. *Journal of Machine Learning Research*, 22(68):1–63, 2021.

Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Yankun Han. Weight initialization and variance dynamics in deep neural networks and large language models. *arXiv preprint arXiv:2510.09423*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Joonkyu Kim, Yejin Kim, and Jy yong Sohn. Measuring representational shifts in continual learning: A linear transformation perspective. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Kz1zCJRr1r.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324, 1998. URL https://api.semanticscholar.org/CorpusID:14542261.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Lanpei Li, Elia Piccoli, Andrea Cossu, Davide Bacciu, and Vincenzo Lomonaco. Calibration of continual learning models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4160–4169, 2024.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Aojun Lu, Tao Feng, Hangjie Yuan, Xiaotian Song, and Yanan Sun. Revisiting neural networks for continual learning: An architectural perspective. In *IJCAI*, pp. 4651–4659, 2024. URL https://www.ijcai.org/proceedings/2024/514.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Huiyi Hu, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Wide neural networks forget less catastrophically. In *International conference on machine learning*, pp. 15699–15717. PMLR, 2022.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International conference on learning representations*, 2021.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Sidak Pal Singh, Bobby He, Thomas Hofmann, and Bernhard Schölkopf. Hallmarks of optimization trajectories in neural networks: Directional exploration and redundancy. *arXiv preprint arXiv:2403.07379*, 2024.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 374–382, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017. URL `https://api.semanticscholar.org/CorpusID:702279`.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International conference on machine learning*, pp. 5393–5402. PMLR, 2018.

# A Appendix

## A.1 Theoretical Framework

### A.1.1 The Roadmap of the Theoretical Framework

To clarify the theoretical framework, we provide one roadmap as shown in Figure 4.

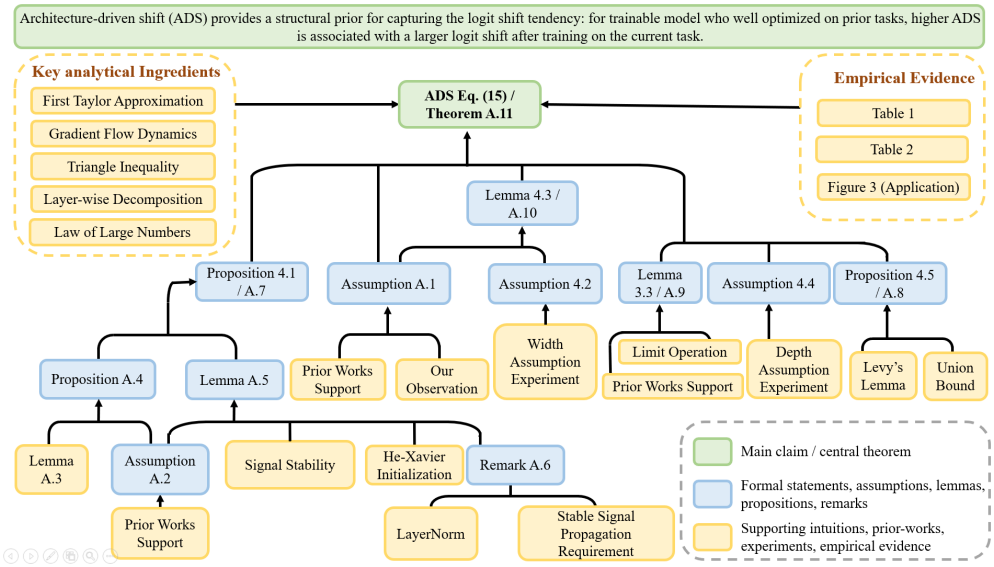



Figure 4: Roadmap of the theoretical framework for deriving Architecture-driven Shift (ADS).

### A.1.2 Detailed Theoretical Framework

**Assumption A.1** (Trajectory Regularity)**.** The optimization trajectory in parameter space is free from sharp directional reversals, implying that the ratio of path length to displacement remains bounded.

$$\mathrm{PathLen}^{(l)} \approx C_{\mathrm{traj}} \cdot \mathrm{Disp}^{(l)}, \tag{17}$$

where $C_{\mathrm{traj}}$ is constant. There are a growing body of theoretical and empirical work supports this assumption. **Theoretical Motivation.** From optimization theory, Gupta et al. (2021) show that under standard regularity conditions, such as linear convergence or the Polyak–Kurdyka–Łojasiewicz (PKL) condition, the optimization trajectory admits a bounded length relative to the distance to the solution set $\mathbf{X}^*$, i.e.,

$$\mathrm{PathLen}^{(l)} \;\le\; C_{\mathrm{traj}} \cdot \mathrm{Disp}^{(l)},$$

where $C_{\mathrm{traj}}$ depends on problem-specific parameters. Notably, theoretical bounds for PKL functions suggest $C_{\mathrm{traj}} \propto \sqrt{\kappa}$ (where $\kappa$ is the condition number). These results establish that long, highly winding trajectories are theoretically unnecessary under commonly assumed training regimes.

**Empirical Evidence.** Empirically, the work Goodfellow et al. (2014) demonstrates that, across a wide range of neural architectures, the loss evaluated along the straight line connecting initialization $\mathbf{x}_0$ and the final solution is typically smooth, monotonic, and nearly convex. Their linear interpolation experiments indicate that the 1-D linear subspace captures the majority of parameter variation, suggesting limited deviation from a straight path during optimization. More recently, Singh et al. (2024) analyze actual optimization trajectories and introduce Mean Directional Similarity (MDS) to quantify directional alignment. They report consistently high MDS values (e.g., $\omega \approx 0.76$ for ResNet-50 on ImageNet), indicating strong directional redundancy in parameter updates. Furthermore, their trajectory visualizations show that increasing model scale leads to progressively darker (i.e., higher cosine similarity) trajectory maps, reflecting increasingly straight and directionally aligned optimization paths.

**Our Observation.** We do simple experiments and find this regularity holds consistently, with the ratio stabilizing around $C_{\mathrm{traj}} \in (1, 1.11)$ for our models and tasks, further validating the assumption.

**Assumption A.2** (Non-degeneracy of Layer-wise Weight)**.** Consider an $L$-layer fully-connected neural network with activation function $\sigma(\cdot)$ (Lipschitz continuous, $\sigma(0) = 0$) and zero biases. The network is initialized in a trainable regime where gradients do not vanish exponentially with depth. Specifically, the layer-wise weights $\Theta^{(l)}$ follow a non-degenerate distribution with variance satisfying:

$$\mathrm{Var}_{\Theta^{(l)}} \gg 0, \qquad \forall\, l = 1, \ldots, L \iff \mathbb{E}[\Theta^{(l)2}] \gg \left(\mathbb{E}[\Theta^{(l)}]\right)^2. \tag{18}$$

This assumption is both theoretically grounded and standard in practice, ensuring the network operates in a trainable regime where gradient norms do not vanish exponentially during backpropagation.

**Theoretical Basis:** Standard initialization schemes He et al. (2015); Glorot & Bengio (2010), are designed specifically to satisfy this variance condition. **Lazy training regime:** Since the He initialization satisfies the aforementioned properties, they are naturally preserved throughout training under the lazy training regime. (Chizat et al., 2019) **Empirical Support:** The necessity of this non-degeneracy condition is validated by extensive large-scale experiments. As shown in Figure. 2 within Xiao et al. (2018) and Figure. 1 and Figure. 6 within Han (2025), training dynamics collapse immediately when the variance drops below the critical threshold, whereas satisfying this assumption enables stable training across hundreds of layers.

**Lemma A.3** (The Recursive Formula for the Error Signal)**.** *We define $\delta^{(l)} = \frac{\partial \mathcal{L}}{\partial z^{(l)}}$, which is the backpropagated error signal of the $l$-th layer. The recursive formula for $\delta^{(l)}$ satisfiies:*

$$\delta^{(l)} = \Theta^{(l+1)\top} \delta^{(l+1)} \odot \sigma'(z^{(l)}), \tag{19}$$

*where $\odot$ represents the Hadamard product (element-wise multiplication).*

*Proof.* We utilize the relationship between the total differential of the loss function $\mathcal{L}$ and its gradient. The differential $d\mathcal{L}$ with respect to the pre-activation vectors $z^{(l)}$ and $z^{(l+1)}$ is given by:

$$d\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial z^{(l)}}\right)^\top dz^{(l)} = \delta^{(l)\top} dz^{(l)}, \tag{20}$$

$$d\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial z^{(l+1)}}\right)^\top dz^{(l+1)} = \delta^{(l+1)\top} dz^{(l+1)}. \tag{21}$$

Recall the forward propagation formula $z^{(l+1)} = \Theta^{(l+1)}\sigma(z^{(l)})$. To find the partial derivative with respect to $z^{(l)}$, we treat the weights $\Theta^{(l+1)}$ as constant (i.e., $d\Theta^{(l+1)} = 0$). Taking the differential of $z^{(l+1)}$ yields:

$$dz^{(l+1)} = d\left(\Theta^{(l+1)}\sigma(z^{(l)})\right) \tag{22}$$

$$= \underbrace{(d\Theta^{(l+1)})}_{0}\sigma(z^{(l)}) + \Theta^{(l+1)}d\left(\sigma(z^{(l)})\right) \tag{23}$$

$$= \Theta^{(l+1)}\left(\sigma'(z^{(l)}) \odot dz^{(l)}\right). \tag{24}$$

Substituting this expression into Eq. (21):

$$d\mathcal{L} = \delta^{(l+1)\top}\left[\Theta^{(l+1)}\left(\sigma'(z^{(l)}) \odot dz^{(l)}\right)\right] = \left(\Theta^{(l+1)\top}\delta^{(l+1)}\right)^{\top}\left(\sigma'(z^{(l)}) \odot dz^{(l)}\right). \tag{25}$$

We apply the vector identity involving the Hadamard product, $\mathbf{u}^{\top}(\mathbf{v} \odot \mathbf{w}) = (\mathbf{u} \odot \mathbf{v})^{\top}\mathbf{w}$. Letting $\mathbf{u} = \Theta^{(l+1)\top}\delta^{(l+1)}$, $\mathbf{v} = \sigma'(z^{(l)})$, and $\mathbf{w} = dz^{(l)}$, we obtain:

$$d\mathcal{L} = \left(\Theta^{(l+1)\top}\delta^{(l+1)} \odot \sigma'(z^{(l)})\right)^{\top} dz^{(l)}. \tag{26}$$

By comparing this result with Eq. (20), we identify the gradient $\delta^{(l)}$ as:

$$\delta^{(l)} = \Theta^{(l+1)\top}\delta^{(l+1)} \odot \sigma'(z^{(l)}).$$

□

**Proposition A.4** (Stability of the $\ell_2$ Norm of Error Signal)**.** *For any layer $1 \le l \le L$, let $\delta^{(l-1)}$ be the backpropagated error signal of the $l$-th layer. The second moment of $\delta^{(l-1)}$ satisfies the property that the error norm remains constant with respect to the layer width $w^{(l)}$:*

$$\mathbb{E}\left[\|\delta^{(l-1)}\|_2^2\right] \sim \mathbb{E}\left[\|\delta^{(l)}\|_2^2\right]. \tag{27}$$

*Proof.* Recall the recursive formula for the error signal during backpropagation (Lemma A.3): $\delta^{(l-1)} = \sigma'(z^{(l-1)}) \odot \Theta^{(l)\top}\delta^{(l)}$, The element-wise representation for the $j$-th component is given by:

$$\delta_j^{(l-1)} = \sigma'(z_j^{(l-1)})\sum_{i=1}^{w^{(l)}}\Theta_{ij}^{(l)}\delta_i^{(l)}. \tag{28}$$

Consequently, the squared $\ell_2$ norm of $\delta^{(l-1)}$ can expanded as:

$$\|\delta^{(l-1)}\|_2^2 = \sum_{j=1}^{w^{(l-1)}}(\delta_j^{(l-1)})^2 = \sum_{j=1}^{w^{(l-1)}}\left(\sigma'(z_j^{(l-1)})\sum_{i=1}^{w^{(l)}}\Theta_{ij}^{(l)}\delta_i^{(l)}\right)^2. \tag{29}$$

We define the conditional expectation operator $\mathbb{E}_{\Theta}[\cdot \mid \delta^{(l)}, z^{(l-1)}]$, treating $\delta^{(l)}$ and $z^{(l-1)}$ as fixed quantities and taking the expectation solely over the weights $\Theta^{(l)}$. Leveraging the linearity of expectation, we obtain:

$$\mathbb{E}_{\Theta}\left[\|\delta^{(l-1)}\|_2^2 \mid \delta^{(l)}, z^{(l-1)}\right] = \sum_{j=1}^{w^{(l-1)}}(\sigma'(z_j^{(l-1)}))^2 \cdot \mathbb{E}_{\Theta}\left[\left(\sum_{i=1}^{w^{(l)}}\Theta_{ij}^{(l)}\,\delta_i^{(l)}\right)^2 \middle| \delta^{(l)}\right]. \tag{30}$$

The quadratic term inside the expectation can be expanded as:

$$\left(\sum_{i=1}^{w^{(l)}}\Theta_{ij}^{(l)}\delta_i^{(l)}\right)^2 = \sum_{i=1}^{w^{(l)}}\sum_{p=1}^{w^{(l)}}\Theta_{ij}^{(l)}\delta_i^{(l)}\Theta_{pj}^{(l)}\delta_p^{(l)}. \tag{31}$$

Thus, the expectation over the weights becomes:

$$\mathbb{E}_\Theta\left[\left(\sum_{i=1}^{w^{(l)}}\Theta_{ij}^{(l)}\,\delta_i^{(l)}\right)^2\middle|\,\delta^{(l)}\right]=\sum_{i=1}^{w^{(l)}}\sum_{p=1}^{w^{(l)}}\delta_i^{(l)}\delta_p^{(l)}\mathbb{E}_\Theta\left[\Theta_{ij}^{(l)}\Theta_{pj}^{(l)}\right]. \tag{32}$$

Let $\Theta_{\cdot j}^{(l)}=[\Theta_{1j}^{(l)},\dots,\Theta_{w^{(l)}j}^{(l)}]^\top$ denote the column vector of weights connecting all neurons in layer $l$ to the $j$-th neuron in layer $l-1$. The expression above can be rewritten in vector form as $\delta^{(l)\top}\mathbb{E}_\Theta[\Theta_{\cdot j}^{(l)}\Theta_{\cdot j}^{(l)\top}]\delta^{(l)}$.

We assume the weights satisfy **second-order isotropy**. Specifically, $\mathbb{E}_\Theta[\Theta_{\cdot j}^{(l)}\Theta_{\cdot j}^{(l)\top}]=v_\Theta^{(l)}I_{w^{(l)}}$, where $v_\Theta^{(l)}=\mathbb{E}[(\Theta_{ij}^{(l)})^2]$ represents the second moment of the weights. This assumption is weaker than requiring weights to be independent and identically distributed (i.i.d.), as it imposes constraints only on the covariance structure (uncorrelatedness) and the second moment, without specifying the underlying distribution (e.g., Gaussian vs. heavy-tailed). Under this assumption, the cross-terms vanish ($\mathrm{Cov}(\Theta_{ij},\Theta_{pj})=0$ for $i\neq p$), yielding:

$$\mathbb{E}_\Theta\left[\left(\sum_{i=1}^{w^{(l)}}\Theta_{ij}^{(l)}\,\delta_i^{(l)}\right)^2\middle|\,\delta^{(l)}\right]=v_\Theta^{(l)}\delta^{(l)\top}I_{w^{(l)}}\delta^{(l)}=v_\Theta^{(l)}\|\delta^{(l)}\|_2^2. \tag{33}$$

Substituting this back into Eq. (30):

$$\mathbb{E}_\Theta\left[\|\delta^{(l-1)}\|_2^2\mid\delta^{(l)},z^{(l-1)}\right]=v_\Theta^{(l)}\|\delta^{(l)}\|_2^2\cdot\sum_{j=1}^{w^{(l-1)}}(\sigma'(z_j^{(l-1)}))^2. \tag{34}$$

Assuming the pre-activations $z_j^{(l-1)}$ within the same layer are identically distributed, we apply the Law of Large Numbers. As the width $w^{(l-1)}$ becomes sufficiently large:

$$\frac{1}{w^{(l-1)}}\sum_{j=1}^{w^{(l-1)}}(\sigma'(z_j^{(l-1)}))^2\longrightarrow\mathbb{E}_z\left[(\sigma'(z^{(l-1)}))^2\right]. \tag{35}$$

For ReLU activation functions, where $\sigma'(z)\in\{0,1\}$, it holds that $(\sigma'(z))^2=\sigma'(z)$. Letting $c_\sigma=\mathbb{E}[\sigma'(z^{(l-1)})]$, the summation converges to $w^{(l-1)}c_\sigma$.

By the Law of Total Expectation:

$$\mathbb{E}\left[\|\delta^{(l-1)}\|_2^2\right]=\mathbb{E}\left[v_\Theta^{(l)}\|\delta^{(l)}\|_2^2\cdot w^{(l-1)}c_\sigma\right] \tag{36}$$

$$=v_\Theta^{(l)}\cdot w^{(l-1)}c_\sigma\cdot\mathbb{E}\left[\|\delta^{(l)}\|_2^2\right]. \tag{37}$$

Invoking Assumption A.2, we have $\mathbb{E}[(\Theta_{ij}^{(l)})^2]\gg(\mathbb{E}[\Theta_{ij}^{(l)}])^2$, implying $v_\Theta^{(l)}=\mathrm{Var}(\Theta_{ij})+\left(\mathbb{E}[\Theta_{ij}^{(l)}]\right)^2\approx\mathrm{Var}(\Theta_{ij}^{(l)})$. Because of the He initialization, $\mathrm{Var}(\Theta_{ij}^{(l)})=\frac{2}{w^{(l-1)}}$ is satisfied. Furthermore, assuming $z$ is symmetric about 0, for ReLU we have $c_\sigma=1/2$. Substituting these values:

$$\mathbb{E}\Big[\|\delta^{(l-1)}\|_2^2\Big]=v_\Theta^{(l)}\cdot w^{(l-1)}\cdot c_\sigma\cdot\mathbb{E}\Big[\|\delta^{(l)}\|_2^2\Big] \tag{38}$$

$$=\frac{2}{w^{(l-1)}}\cdot w^{(l-1)}\cdot\frac{1}{2}\cdot\mathbb{E}\Big[\|\delta^{(l)}\|_2^2\Big] \tag{39}$$

$$=\mathbb{E}\Big[\|\delta^{(l)}\|_2^2\Big]. \tag{40}$$

Accounting for potential noise and finite-width effects, we conclude that the error norms are of the same order:

$$\mathbb{E}\left[\|\delta^{(l-1)}\|_2^2\right]\sim\mathbb{E}\left[\|\delta^{(l)}\|_2^2\right].$$

□

**Lemma A.5** (Activation Scaling Law)**.** *The $\ell_2$ norm of the activation vector at the $l$-th layer scales proportionally with the square root of the layer width:*

$$\|a^{(l-1)}\|_2 \propto \sqrt{w^{(l-1)}}. \tag{41}$$

*Proof.* Consider the pre-activation $z_i^{(l)}$ for an arbitrary neuron $i \in \{1, \dots, w^{(l)}\}$ in the current layer $l$. It is defined as the linear transformation of the activations from the previous layer:

$$z_i^{(l)} = \sum_{j=1}^{w^{(l-1)}} \Theta_{ij}^{(l)} a_j^{(l-1)}. \tag{42}$$

We analyze the variance of $z_i^{(l)}$ conditioned on the input activations $a^{(l-1)}$. Utilizing the linearity of variance for independent variables (invoking the weak isotropy assumption where weights are uncorrelated, i.e., $\mathrm{Cov}(\Theta_{ij}, \Theta_{ik}) \approx 0$ for $j \neq k$), the cross-terms vanish.

Furthermore, applying assumption A.2, we have $\mathbb{E}[\Theta^2] \gg (\mathbb{E}[\Theta])^2$, i.e., $\mathrm{Var}(\Theta_{ij}^{(l)}) \approx \mathbb{E}[(\Theta_{ij}^{(l)})^2] \triangleq v_\Theta^{(l)}$. Thus, the conditional variance simplifies to:

$$\mathrm{Var}\left(z_i^{(l)} \mid a^{(l-1)}\right) = \sum_{j=1}^{w^{(l-1)}} \mathrm{Var}\left(\Theta_{ij}^{(l)} a_j^{(l-1)} \mid a^{(l-1)}\right) \tag{43}$$

$$= \sum_{j=1}^{w^{(l-1)}} (a_j^{(l-1)})^2 \cdot \mathrm{Var}(\Theta_{ij}^{(l)}) \tag{44}$$

$$\approx v_\Theta^{(l)} \sum_{j=1}^{w^{(l-1)}} (a_j^{(l-1)})^2 \tag{45}$$

$$= v_\Theta^{(l)} \|a^{(l-1)}\|_2^2. \tag{46}$$

To ensure stable signal propagation during training (preventing vanishing or exploding gradients), two conditions are typically imposed: (1) **Signal Stability.** The variance of the pre-activations should remain on the order of a constant, i.e., $\mathrm{Var}(z_i^{(l)}) \sim \mathcal{O}(1)$. (2) **Initialization Scaling.** Following He initialization, the variance of the weights must be inversely proportional to the input dimension (fan-in): $v_\Theta^{(l)} \propto \frac{1}{w^{(l-1)}}$. Substituting these conditions into the variance equation yields:

$$\frac{1}{w^{(l-1)}} \cdot \|a^{(l-1)}\|_2^2 \sim \mathcal{O}(1). \tag{47}$$

Solving for the squared norm, we obtain $\|a^{(l-1)}\|_2^2 \propto w^{(l-1)}$. Taking the square root of both sides leads to the final conclusion:

$$\|a^{(l-1)}\|_2 \propto \sqrt{w^{(l-1)}}.$$

□

*Remark* A.6. While the derivation explicitly relies on the independence assumptions valid at initialization, the scaling law $\|a^{(l-1)}\|_2 \propto \sqrt{w^{(l-1)}}$ holds throughout training in practical settings. This is ensured either by the use of normalization layers (e.g., LayerNorm), which strictly enforce feature variance, or implicitly by the requirement of **stable signal propagation** in deep networks.

**Proposition A.7** (Gradient Scaling for Weight Matrix)**.** *Let $f_{i,\boldsymbol{u}^*}(\boldsymbol{x}) := \boldsymbol{u}_i^{*\top} f_i(\boldsymbol{x})$. Its layer-wise parameter sensitivity satisfies:*

$$g_{\text{old}}^{(l)} = \nabla_{\Theta^{(l)}} f_{i,\boldsymbol{u}^*}(\boldsymbol{x}) = \delta^{(l|\boldsymbol{u}^*)} (a^{(l-1)})^\top, \tag{48}$$

$$\|g_{\text{old}}^{(l)}\|_F = \|g_{\text{old}}^{(l)}\|_2 \propto \sqrt{w^{(l-1)}}. \tag{49}$$

*Proof.* We treat the scalarized full-logit map $f_{i,\boldsymbol{u}^*}(\boldsymbol{x}) = \boldsymbol{u}_i^{*\top} f_i(\boldsymbol{x})$ as the objective function, where $\boldsymbol{u}_i^*$ is held fixed during differentiation. By the chain rule,

$$\nabla_{\Theta^{(l)}} f_{i,\boldsymbol{u}^*} = \delta^{(l|\boldsymbol{u}^*)} (a^{(l-1)})^\top, \tag{50}$$

where $\delta^{(l|\boldsymbol{u}^*)}$ denotes the backward signal initialized by $\boldsymbol{u}_i^*$ at the output layer..

This is a rank-1 matrix. For any rank-1 matrix formed by the outer product of two vectors $\mathbf{u}$ and $\mathbf{v}$, the spectral norm satisfies $\|\mathbf{u}\mathbf{v}^\top\|_F = \|\mathbf{u}\mathbf{v}^\top\|_2 = \|\mathbf{u}\|_2 \cdot \|\mathbf{v}\|_2$. Applying this property yields:

$$\|\nabla_{\Theta^{(l)}} f_{i,\boldsymbol{u}^*}\|_F = \|\nabla_{\Theta^{(l)}} f_{i,\boldsymbol{u}^*}\|_2 = \|\delta^{(l|\boldsymbol{u}^*)}\|_2 \|a^{(l-1)}\|_2. \tag{51}$$

**(1) The Error Term (Proposition A.4).** The total energy of the backpropagated error is conserved, i.e., $\|\delta^{(l|\boldsymbol{u}^*)}\|_2 \sim 1$.

**(2) The Activation Term (Lemma A.5).** The activation norm scales with the width's square root: $\|a^{(l-1)}\|_2 \propto \sqrt{w^{(l-1)}}$.

Combining these results, the norm of the gradient scales as:

$$\|\nabla_{\Theta^{(l)}} f_{i,\boldsymbol{u}^*}\|_2 \propto 1 \cdot \sqrt{w^{(l-1)}} = \sqrt{w^{(l-1)}}.$$

□

**Proposition A.8** (Signal-Noise Decomposition under High Dimensionality)**.** *Under the Signal-Noise Decomposition hypothesis, the cosine similarity between full-logit sensitivity* $\boldsymbol{g}_{old} \triangleq \nabla_{\Theta_{(i,S)}} f_{i,\boldsymbol{u}^*}(\boldsymbol{x})$ *and update direction* $\boldsymbol{g}_{new}(s) \triangleq \nabla_{\Theta_{(i+1,r_k)}} \mathcal{L}_{i+1}$ *at layer* $l$ *can be decomposed into a deterministic task-alignment term and a stochastic high-dimensional noise term. With probability at least* $1-\delta$*, the cosine similarity satisfies the following scaling law:*

$$\cos\theta^{(l)} \propto \left( |\rho_{\textit{task}}^{(l)}| + \sqrt{\frac{\log(L/\delta)}{D_{\textit{eff}}^{(l)}}} \right), \tag{52}$$

*where* $D_{\textit{eff}}^{(l)}$ *denotes the effective dimension of the parameter space at layer* $l$*, and* $\rho_{\textit{task}}^{(l)}$ *represents the semantic conflict component.*

*Proof.* We analyze the gradient interaction in the high-dimensional parameter space. Let $\bar{\boldsymbol{g}}_{\text{old}}^{(l)} = \text{vec}(\boldsymbol{g}_{\text{old}}^{(l)})/\|\boldsymbol{g}_{\text{old}}^{(l)}\|_F \in \mathbb{S}^{N-1}$ and $\bar{\boldsymbol{g}}_{\text{new}}^{(l)}(s) = \text{vec}(\boldsymbol{g}_{\text{new}}^{(l)}(s))/\|\boldsymbol{g}_{\text{new}}^{(l)}(s)\|_F \in \mathbb{S}^{N-1}$ denote the normalized, vectorized gradients of the $l$-th layer, where $N = w^{(l)}w^{(l-1)}$. The Frobenius-based cosine similarity is strictly equivalent to the vector inner product of these normalized vectors, i.e., $\cos\theta_s^{(l)} = (\bar{\boldsymbol{g}}_{\text{old}}^{(l)})^\top \bar{\boldsymbol{g}}_{\text{new}}^{(l)}(s)$. Inspired by Gur-Ari et al., we model the normalized gradient $\bar{\boldsymbol{g}}_{\text{new}}^{(l)}(s)$ using the Signal-Noise Decomposition:

$$\bar{\boldsymbol{g}}_{\text{new}}^{(l)}(s) = \alpha \boldsymbol{u}_{\text{signal}} + \beta \boldsymbol{v}_{\text{noise}}, \tag{53}$$

where $\boldsymbol{u}_{\text{signal}}$ resides in a low-dimensional subspace capturing semantic task conflicts, and $\boldsymbol{v}_{\text{noise}}$ represents the residual stochastic component, which is assumed to be uniformly distributed on the unit sphere $\mathbb{S}^{D_{\text{eff}}^{(l)}-1}$ of the residual effective subspace based on the **Conditional Isotropy**.

Since $\|\bar{\boldsymbol{g}}_{\text{old}}^{(l)}\|_2 = 1$ by construction, the cosine similarity expands as:

$$\cos\theta_s^{(l)} = (\bar{\boldsymbol{g}}_{\text{old}}^{(l)})^\top \bar{\boldsymbol{g}}_{\text{new}}^{(l)}(s) = \underbrace{\alpha \left( (\bar{\boldsymbol{g}}_{\text{old}}^{(l)})^\top \boldsymbol{u}_{\text{signal}} \right)}_{\text{Deterministic: } \rho_{\text{task}}^{(l)}} + \underbrace{\beta \left( (\bar{\boldsymbol{g}}_{\text{old}}^{(l)})^\top \boldsymbol{v}_{\text{noise}} \right)}_{\text{Stochastic: } \epsilon_s^{(l)}}. \tag{54}$$

To bound the stochastic term $\epsilon_s^{(l)} = \beta(\bar{\boldsymbol{g}}_{\text{old}}^{(l)})^\top \boldsymbol{v}_{\text{noise}}$, we employ **Levy's Lemma (Concentration of Measure on the Sphere)**. Consider the projection function $F(\boldsymbol{v}) = (\bar{\boldsymbol{g}}_{\text{old}}^{(l)})^\top \boldsymbol{v}$. Since $F$ is 1-Lipschitz (as $|F(\boldsymbol{v}_1) - F(\boldsymbol{v}_2)| \le \|\bar{\boldsymbol{g}}_{\text{old}}^{(l)}\|_2 \|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_2 = \|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_2$) and $\boldsymbol{v}_{\text{noise}}$ is isotropically distributed on $\mathbb{S}^{D_{\text{eff}}^{(l)}-1}$, for any $\epsilon > 0$:

$$P\left( |(\bar{\boldsymbol{g}}_{\text{old}}^{(l)})^\top \boldsymbol{v}_{\text{noise}}| \ge \epsilon \right) \le 2\exp\left( -\frac{D_{\text{eff}}^{(l)} \epsilon^2}{2} \right). \tag{55}$$

We aim to establish a uniform upper bound, denoted as $\epsilon_{\text{orth}}$, such that the gradient orthogonality condition holds across all $L$ layers simultaneously with high probability (e.g., $1-\delta$).

The probability of the entire network satisfying this condition ("Global Success") can be lower-bounded by the complement of the union of failure events at individual layers. By applying the **Union Bound**, we derive:

$$P(\text{Global Success}) = 1 - P\left(\bigcup_{l=1}^{L}\{\text{Failure at layer } l\}\right) \tag{56}$$

$$\geq 1 - \sum_{l=1}^{L} P(\text{Failure at layer } l) \tag{57}$$

$$= 1 - L \cdot p_{\text{fail}}, \tag{58}$$

where $p_{\text{fail}} = P(|\boldsymbol{g}_{\text{old}}^{\top}\boldsymbol{v}_{\text{noise}}| \geq \epsilon_{\text{orth}})$ represents the failure probability for a single layer.

We constrain the total failure probability to a small constant $\delta$. Setting $L \cdot p_{\text{fail}} = \delta \to 0$, and substituting Eq, 55, we have:

$$p_{\text{fail}} = 2\exp\left(-\frac{D_{\text{eff}}^{(l)}\epsilon_{\text{orth}}^2}{2}\right) = \frac{\delta}{L}. \tag{59}$$

Solving this equation for $\epsilon_{\text{orth}}$ yields the uniform bound:

$$\epsilon_{\text{orth}} = \sqrt{\frac{2\log\left(\frac{2L}{\delta}\right)}{D_{\text{eff}}^{(l)}}}. \tag{60}$$

Consequently, for any layer $l \in \{1, \ldots, L\}$, the stochastic projection noise $|\epsilon_s|$ is bounded with probability $1-\delta$. The noise term scales as:

$$|\epsilon_s| \leq \sqrt{\frac{2\log\left(\frac{2L}{\delta}\right)}{D_{\text{eff}}^{(l)}}} \propto \sqrt{\frac{\log(L/\delta)}{D_{\text{eff}}^{(l)}}}. \tag{61}$$

Substituting this scaling behavior back into the cosine decomposition (Eq. (54)), we obtain:

$$\cos\theta^{(l)} \propto \left(|\rho_{\text{task}}^{(l)}| + \sqrt{\frac{\log(L/\delta)}{D_{\text{eff}}^{(l)}}}\right).$$

□

**Lemma A.9** (Asymptotic Vanishing of the Effective Dimension)**.** *Let the gradient space at layer $l$ admit the orthogonal decomposition $\mathbb{R}^{N^{(l)}} = V_{\text{signal}} \oplus V_{\text{noise}}$, where $N^{(l)} = w^{(l)}w^{(l-1)}$ denotes the nominal parameter dimension. Suppose the signal subspace dimension $\dim(V_{\text{signal}}) = k$ is independent of model width. Then in the wide-network limit $w^{(l)} \to \infty$,*

$$\lim_{w^{(l)}\to\infty} \sqrt{\frac{\log(L/\delta)}{D_{\mathit{eff}}^{(l)}}} = 0. \tag{62}$$

*Proof.* By Gur-Ari et al. (2018), gradient updates during training reside predominantly in a low-dimensional subspace spanned by the top eigenvectors of the Hessian, whose dimension equals the number of output classes $k$, a quantity determined solely by the task and invariant to model width. Since $V_{\text{noise}} = V_{\text{signal}}^{\perp}$, the effective noise dimension satisfies

$$D_{\text{eff}}^{(l)} = N^{(l)} - k = w^{(l)}w^{(l-1)} - k.$$

For fixed $k$ and $w^{(l-1)}$, we have $D_{\text{eff}}^{(l)} \to \infty$ as $w^{(l)} \to \infty$. Since $\log(L/\delta)$ is independent of $w^{(l)}$, it follows immediately that

$$\lim_{w^{(l)}\to\infty}\sqrt{\frac{\log(L/\delta)}{D_{\text{eff}}^{(l)}}}=\lim_{w^{(l)}\to\infty}\sqrt{\frac{\log(L/\delta)}{w^{(l)}w^{(l-1)}-k}}=0.$$

□

**Lemma A.10.** *To quantify the distance of the optimization trajectory, we define* $\mathrm{PathLen}^{(l)} \triangleq \int_0^S \|\nabla_{\Theta^{(l)}_{(i+1,r_k)}}\mathcal{L}_{i+1}\|_F dr_k$*, i.e., the integral of the Frobenius norm over* $S$ *training steps. Under Assumptions 4.2 and A.1, the path length yields the stated proportionality:*

$$\mathrm{PathLen}^{(l)} \propto (w^{(l-1)})^{\alpha}(w^{(l)})^{\beta+\frac{1}{2}}. \tag{63}$$

*Proof.* We define $\mathrm{Disp}^{(l)}$ as $\|\Delta\Theta\|_F = \left\|\int_0^S \nabla_{\Theta^{(l)}_{(i+1,s)}}\mathcal{L}_{i+1}ds\right\|_F$ represents the displacement during model optimization. Based on Assumptions 4.2, $\mathrm{Disp}^{(l)}$ satisfies:

$$\mathrm{Disp}^{(l)} = \|\Theta_t^{(l)} - \Theta_{t-1}^{(l)}\|_F \propto \|\Theta_{t-1}^{(l)}\|_F (w^{(l-1)})^{\alpha}(w^{(l)})^{\beta}. \tag{64}$$

Considering He initialization, there is $\|\Theta^{(l)}\|_F \propto (w^{(l)})^{1/2}$. Under Assumption A.1, we have:

$$\mathrm{PathLen}^{(l)} \approx C_{\mathrm{traj}} \cdot \mathrm{Disp}^{(l)} \propto (w^{(l-1)})^{\alpha}(w^{(l)})^{\beta+\frac{1}{2}}, \tag{65}$$

where $C_{\mathrm{traj}}$ is constant close to 1 ($\in (1, 1.1)$ in our experiments), associating with the trajectory. □

**Theorem A.11** (Scaling Law of Output Shift)**.** *Let* $f_i(\boldsymbol{x})$ *and* $f_{i+1}(\boldsymbol{x})$ *denote the model output before and after learning task* $i+1$*, respectively. Under the assumption that second-order terms are negligible (dominating* $< 20\%$ *of the variation in our small-scale experiments), the shift in the output function for a previous task sample* $x$ *scales as:*

$$\|f_{i+1}(\boldsymbol{x}) - f_i(\boldsymbol{x})\|_2 \lesssim \sum_{l=1}^{L}(w^{(l-1)})^{\alpha+\frac{1}{2}}(w^{(l)})^{\beta+\frac{1}{2}} \cdot |l^b e^{-cl}|, \tag{66}$$

*Proof.* We analyze the output shift using a first-order Taylor approximation around the parameters $\Theta_i$. Following the local linearization perspective widely adopted in deep learning theory analyses Jacot et al. (2018); Lee et al. (2019), we have:

$$f_{i+1}(\boldsymbol{x}) - f_i(\boldsymbol{x}) \approx J_\Theta f_i(\boldsymbol{x})\Delta\Theta_i. \tag{67}$$

Substituting the gradient flow dynamics $\Delta\Theta_i = -\int_0^S \nabla_\Theta \mathcal{L}_{i+1}(\Theta_s)ds$, we obtain:

$$\begin{aligned}
\|f_{i+1}(\boldsymbol{x}) - f_i(\boldsymbol{x})\|_2 &= \left|\boldsymbol{u}_i^{*\top}(\boldsymbol{x})\big(f_{i+1}(\boldsymbol{x}) - f_i(\boldsymbol{x})\big)\right| && (68)\\
&\lesssim \left|\boldsymbol{u}_i^{*\top}(\boldsymbol{x}) J_\Theta f_i(\boldsymbol{x})\Delta\Theta_i\right| && (69)\\
&= \left|\boldsymbol{g}_{\text{old}}^\top \Delta\Theta_i\right| && (70)\\
&= \left|\boldsymbol{g}_{\text{old}}^\top \left(-\int_0^S \nabla_\Theta \mathcal{L}_{i+1}(\Theta_s)ds\right)\right| && (71)\\
&= \left|\boldsymbol{g}_{\text{old}}^\top \int_0^S \boldsymbol{g}_{\text{new}}(s)ds\right| && (72)\\
&= \left|\int_0^S \langle \boldsymbol{g}_{\text{old}}, \boldsymbol{g}_{\text{new}}(s)\rangle ds\right| && (73)\\
&\le \int_0^S \left|\langle \boldsymbol{g}_{\text{old}}, \boldsymbol{g}_{\text{new}}(s)\rangle\right| ds && (74)\\
&= \int_0^S \left|\sum_{l=1}^L \langle \boldsymbol{g}_{\text{old}}^{(l)}, \boldsymbol{g}_{\text{new}}^{(l)}(s)\rangle_F\right| ds && (75)\\
&\le \int_0^S \sum_{l=1}^L \|\boldsymbol{g}_{\text{old}}^{(l)}\|_F \cdot \|\boldsymbol{g}_{\text{new}}^{(l)}(s)\|_F \cdot |\cos\theta_s^{(l)}| ds && (76)\\
&= \sum_{l=1}^L \underbrace{\|\boldsymbol{g}_{\text{old}}^{(l)}\|_F \int_0^S \|\boldsymbol{g}_{\text{new}}^{(l)}(s)\|_F \cdot |\cos\theta_s^{(l)}| ds}_{\text{Term}^{(l)}}, && (77)
\end{aligned}$$

where $\theta_s^{(l)}$ is the angle between the fixed old task gradient $\boldsymbol{g}_{\text{old}}^{(l)}$ and the time-varying new task gradient $\boldsymbol{g}_{\text{new}}^{(l)}(s)$. Here, $\boldsymbol{u}_i^*$ is an analytical dual witness that is treated as fixed when taking derivatives with respect to $\Theta_i$; it is not required for computing ADS.

We analyze the bound layer by layer:

1. **Considering $\boldsymbol{g}_{\text{old}}^{(l)}$.** From Proposition A.7, the gradient norm scales as $\|\boldsymbol{g}_{\text{old}}^{(l)}\|_2 = \|\nabla_{\Theta^{(l)}} f_{i,\boldsymbol{u}^*\cdot}(\boldsymbol{x})\|_2 \propto \sqrt{w^{(l-1)}}$.

2. **Considering $\int_0^S \|\boldsymbol{g}_{\text{new}}^{(l)}(s)\|_F \cdot |\cos\theta_s^{(l)}| ds$.** By Assumption A.1, the optimization trajectory exhibits high directional stability, implying that the angle $\theta_s^{(l)}$ remains nearly constant throughout the update. We thus approximate the instantaneous alignment by its characteristic value $|\cos\theta^{(l)}|$.

$$\int_0^S \|\boldsymbol{g}_{\text{new}}^{(l)}(s)\|_F \cdot |\cos\theta_s^{(l)}| ds \lesssim \left(\int_0^S \|\boldsymbol{g}_{\text{new}}^{(l)}(s)\|_F ds\right) \cdot |\cos\theta^{(l)}| = \text{PathLen}^{(l)} \cdot |\cos\theta^{(l)}| \tag{78}$$

3. **Combination.** Incorporating Lemma A.10 ($\text{PathLen}^{(l)} \propto (w^{(l-1)})^\alpha (w^{(l)})^{\beta+\frac{1}{2}}$) and $\cos\theta \propto l^b e^{-cl}$, the total contribution of layer $l$ satisfies:

$$\text{Term}^{(l)} \lesssim \sqrt{w^{(l-1)}} \cdot w^{(l-1)\,\alpha} \cdot w^{(l)\,\beta} \cdot |l^b e^{-cl}| = (w^{(l-1)})^{\alpha+\frac{1}{2}} (w^{(l)})^{\beta+\frac{1}{2}} \cdot |l^b e^{-cl}|. \tag{79}$$

Summing over all layers $l = 1, \ldots, L$ yields the final result. □

## A.2 Detailed Validation of Theoretical Framework

### A.2.1 Experimental Setup

**Architectural Diversity.** Unlike prior studies that typically assume uniform widths, we constructed a comprehensive pool of Fully-Connected Neural Networks (FNNs) with highly heterogeneous topologies. The network depth was varied ($L \in \{3, 5, 10, 15, 20, 25\}$), and hidden layer widths were sampled from a candidate pool $\mathcal{W} \in \{256, 512, 1024, 2048, 4096\}$. To ensure structural generality, we generated five distinct topological categories: (1) Uniform: Constant width across all layers. (2) Monotonic: strictly Increasing or Decreasing widths. (3) Non-Monotonic: Bottleneck (compression followed by expansion), Spindle (expansion followed by compression), and fully Randomized configurations. This resulted in a total of 175 unique architecture instances per task scenario.

What's more, our experiments further evaluate on convolutional neural networks (CNNs) with skip connections and vision transformers (ViTs) proposed by Dosovitskiy et al.. Specifically, the detailed architecture of the CNN is available in Figure 5c. We alter "width" and "depth" by varying the output channels and number of CleanConvBlock layers. For ViTs, we adjust the dimension of matrix $Q, K, V$ and the number of Transformer blocks to obtain variable "width" and "depth" (depth $L$: [6, 8, 12, 16]; width $\mathcal{W}$: [128, 192, 256, 384]). To enable comprehensive statistical analysis across diverse architectures, we additionally vary the expansion ratio of the feed-forward network (FFN)[5] as ratio $= d_{\text{mid}}/d_{\text{in}}$, where $d_{\text{mid}}$ is the hidden dimension of the FFN and $d_{\text{in}}$ is its input dimension, which is equal to the dimension of $Q, K, V$.

**Training Protocol.** We performed supervised learning on standard benchmark datasets (e.g., MNIST, FashionMNIST). The networks were initialized using Kaiming Normal initialization for ReLU activation. In our experiments, FNNs are trained for 15 epochs per task and CNNs for 10 epochs per task, using SGD with momentum 0.9, weight decay $5 \times 10^{-4}$, batch size 256, and a cosine learning-rate schedule. We set the random seed to 42 for all experiments to ensure reproducibility. Experiments were conducted on a workstation with a GeForce RTX 4090 D (22 GB) GPU and an AMD Ryzen 9 9950X CPU.

**Calibration Set Origination.** In practice, $D_{\text{cal}}$ is sampled from $D_{1:t}$ based on a manually specified calibration ratio, as detailed in Table 2 of the main paper and Table 4 in the Appendix.

**Calibration Protocol.** For reproducibility, we further specify the calibration protocol used for the data-dependent factor in ADS. For each continual transition $t \to t+1$, we first train a candidate architecture $a \in A$ on task $t$ to obtain $f_t^a(\cdot; \Theta_t^a)$, and then continue training a copied model on task $t+1$ to obtain $f_{t+1}^a(\cdot; \Theta_{t+1}^a)$.[6] The calibration set is always sampled from the prior-task data $D_{1:t}$: we randomly split it into $D_{\text{cal}}$ and $D_{\text{test}}$, with $D_{\text{cal}} : D_{\text{test}} = 3 : 7$ in the strict-split experiments. The measured logit shift used for fitting is

$$\widehat{S}^{\text{cal}}_{t\to t+1}(a) = \left( \frac{1}{|D_{\text{cal}}|} \sum_{x \in D_{\text{cal}}} \left\| f_{t+1}^a(x; \Theta_{t+1}^a) - f_t^a(x; \Theta_t^a) \right\|_2^2 \right)^{1/2}, \tag{80}$$

where $\widehat{S}^{\text{test}}_{t\to t+1}(a)$ is computed analogously on $D_{\text{test}}$. Architectures are independently split into $A_{\text{cal}}$ and $A_{\text{val}}$ by stratified sampling according to the proxy cost $\sum_l 1/w^{(l)}$. The ADS parameters are fitted only on $A_{\text{cal}}$ by nonlinear least squares:

$$(\widehat{A}, \widehat{\alpha}, \widehat{\beta}, \widehat{b}, \widehat{c}) = \arg\min_{A,\alpha,\beta,b,c} \sum_{a \in A_{\text{cal}}} \left[ A \sum_{l=1}^{L_a} \left( w_a^{(l-1)} \right)^{\alpha+\frac{1}{2}} \left( w_a^{(l)} \right)^{\beta+\frac{1}{2}} |l^b e^{-cl}| - \widehat{S}^{\text{cal}}_{t\to t+1}(a) \right]^2. \tag{81}$$

[5] FFN is also known as a fully-connected neural network (FNN).

[6] To reduce stochastic variance, we repeat the experiment on task $t+1$ three times using different random seeds $\{0, 1, 2\}$.

Table 3: Statistical Verification of the Width Scaling Law (Assumption 4.2).

| **Network Depth** | **Sample Size**$(n)$ | $\hat{\boldsymbol{\alpha}}$ for $w_{\text{in}}$ | $\hat{\boldsymbol{\beta}}$ for $w_{\text{out}}$ | **Pearson** $r$ | **Spearman** $r_s$ | $p$**-value** |
|---|---|---|---|---|---|---|
| 3 | 50 | $0.214_{\pm 0.023}$ | $-0.486_{\pm 0.021}$ | 0.962 | 0.961 | $< 0.001$ |
| 5 | 120 | $0.201_{\pm 0.017}$ | $-0.426_{\pm 0.017}$ | 0.917 | 0.897 | $< 0.001$ |
| 10 | 270 | $0.146_{\pm 0.016}$ | $-0.351_{\pm 0.016}$ | 0.818 | 0.858 | $< 0.001$ |
| 15 | 420 | $0.109_{\pm 0.014}$ | $-0.286_{\pm 0.014}$ | 0.752 | 0.813 | $< 0.001$ |
| 20 | 570 | $0.124_{\pm 0.013}$ | $-0.299_{\pm 0.013}$ | 0.740 | 0.812 | $< 0.001$ |
| 25 | 720 | $0.101_{\pm 0.012}$ | $-0.279_{\pm 0.012}$ | 0.726 | 0.801 | $< 0.001$ |

*Note:* This table empirically validates the width-scaling component of Assumption 4.2. The reported $\hat{\alpha}$ and $\hat{\beta}$ are log-log regression estimates of the width exponents $\alpha$ and $\beta$ in Eq. (8), corresponding to $w_{\text{in}} = w^{(l-1)}$ and $w_{\text{out}} = w^{(l)}$, respectively. The intercept $\eta_0$ is only an auxiliary regression constant and is not an ADS parameter. The ADS parameters $b$ and $c$ are not fitted in this width-scaling validation.

The same fitted parameters are then applied without refitting to held-out architectures $a \in A_{\text{val}}$, and the final reported correlation is computed between $\text{ADS}(a; \widehat{A}, \widehat{\alpha}, \widehat{\beta}, \widehat{b}, \widehat{c})$ and $\widehat{S}^{\text{test}}_{t \to t+1}(a)$. Each scenario and model family has its own fitted ADS parameters; parameters are not fitted separately for each architecture and are not reused across different task transitions or model families.

### A.2.2 Additional Evidence of Width Scaling Assumption

Beyond visual evidence of Assumption 4.2 as shown in main paper, we performed a log-log linear regression $\log\left(\frac{\|\Delta\Theta\|_F}{\|\Theta\|_F}\right) = \eta_0 + \hat{\alpha}\log(w_{\text{in}}) + \hat{\beta}\log(w_{\text{out}})$ across thousands of layer-wise configurations sampled from networks of varying depths (from 3 to 25 layers). The quantitative results are summarized in Table 3.

### A.2.3 Additional Evidence of Architecture-Dependent Depth Profiles

As demonstrated in Figure 5, the architecture-dependent depth profiles exist across scenarios (different combinations of various datasets) and model families (FNN, VGG-family networks and CNNs with skip-connection mechanism).

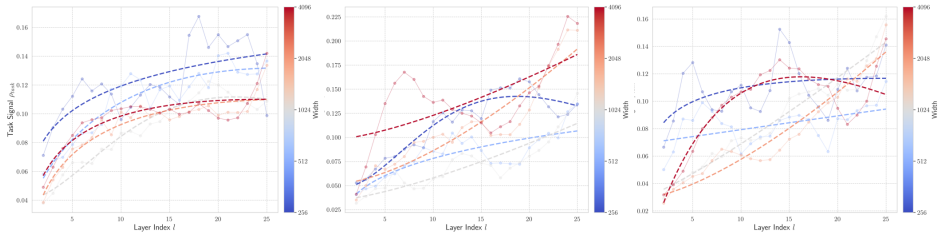
Specifically, FNNs are predominantly described by the zero-attenuation power-law submodel, i.e., $l^b \cdot e^{-cl}$ with $c = 0$. Notably, the monotonic depth amplification trend derived from this submodel aligns with the empirical findings in prior works Guha & Lakshman (2024); Kim et al. (2025), which also observe that increasing depth exacerbates forgetting. In contrast, CNNs (both VGG-family networks and those with skip connections) consistently favor a positive attenuation parameter ($c > 0$) and exhibit a "inverted-U-shaped" curve as shown in Figures 5b and 5c.

### A.2.4 Strong correlation between ADS and logit shift in Vision Transformer (ViT)

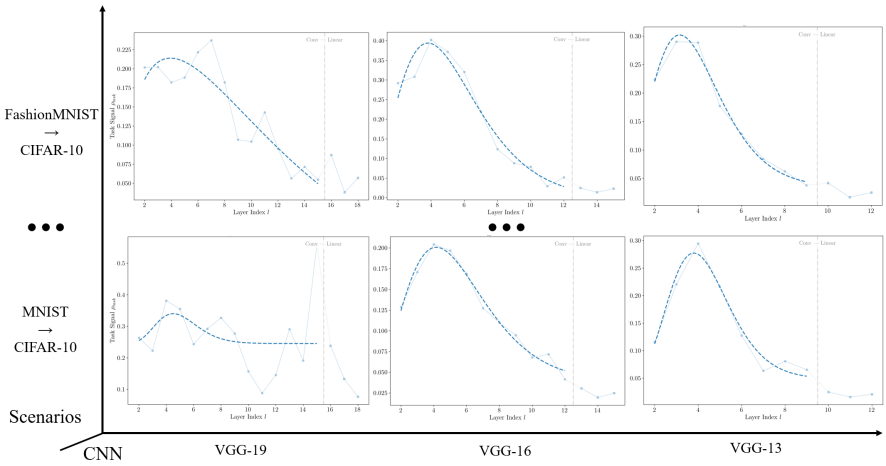
We validate the scope of ADS in Vision Transformer (ViT), which proposed by Dosovitskiy et al.. Considering the huge quantity requirement of ViT, we choose CIFAR-100 and ImageNet as our benchmarks, and we resize ImageNet's data to $32 \times 32$ to align with CIFAR-100. As shown in Figure 6, the proposed ADS metric exhibits a strong Spearman rank correlation ($0.926$ and $0.821$) with the logit shift, indicating that ADS's ability to capture logit shifts generalizes well to the ViTs.

### A.2.5 Comparison of Calibrated Models Based on Different Methods

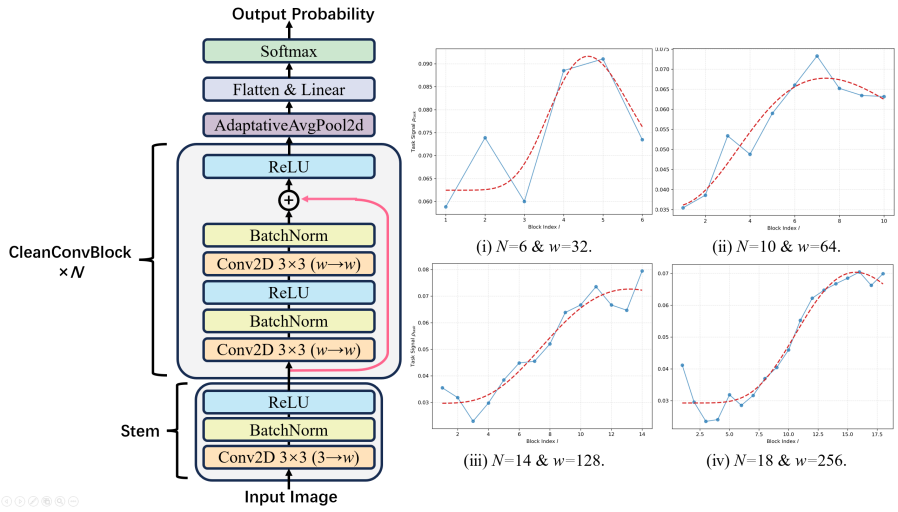
To determine whether ADS captures architectural information beyond coarse model scale and depth, we compare it with three architecture-only baselines. For an architecture $a$ with $L(a)$ hidden layers, we define layer widths $\{w^{(l)}\}_{l=0}^{L(a)+1}$, input dimension $w^{(0)}$, and output dimension $w^{(L(a)+1)}$.



(a) Validation on FNNs. Specifically, the subfigures from left to right depict the monotonic depth amplification phenomenon under the transfer scenarios FashionMNIST → MNIST, MNIST → CIFAR-10, and FashionMNIST → CIFAR-10, respectively.

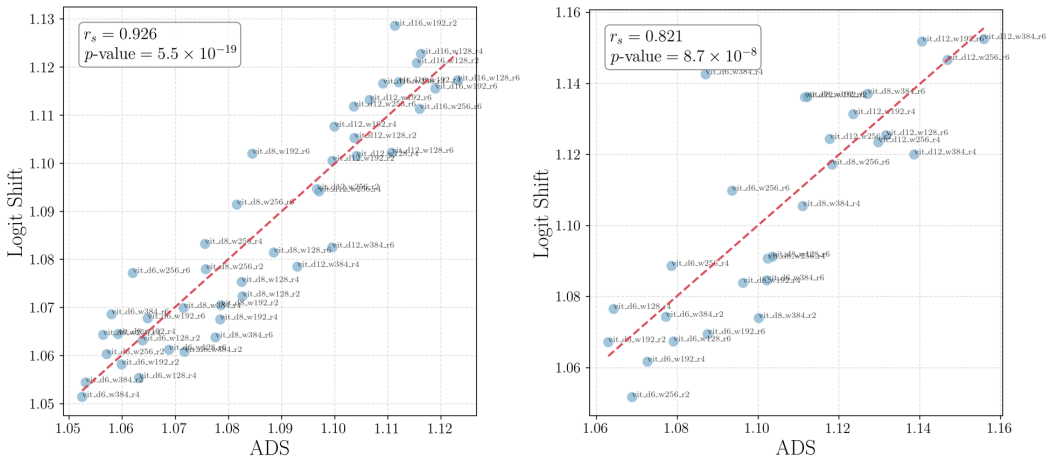



(b) Validation on VGG-family models.




(c) Validation on CNNs with skip connections. Left: the model architecture used in our experiments, with the red arrow indicating the skip connection. We define the model width $w$ as the output channels of each $3 \times 3$ convolutional layer, and the depth $N$ as the number of CleanConvBlocks. Right: middle-layer vulnerability on MNIST → CIFAR-10 across different model architectures $(N, w)$. More results are available in our codebase.

Figure 5: Empirical Validation of the Architecture-Dependent Depth Profile across several scenarios and model-families.




(a) CIFAR-10 → ImageNet (b) ImageNet → CIFAR-100

Figure 6: Correlation between empirically measured logit shift and ADS across heterogeneous ViT variants on CIFAR-100 and ImageNet, with skip-connections removed. Since ADS is derived for FNNs and residual paths can confound logit-shift measurements, skip-connections are removed only in this controlled analysis. All other ViT experiments use standard ViTs, which also support the same ADS trend, as shown in Table 2 in the main paper. See Appendix A.2.1 for details.

**Parameter-Count.** The parameter-count baseline is defined as:

$$\widehat{S}_{\text{Params}}(a) = P(a) = \sum_{l=1}^{L(a)+1} \left( w^{(l-1)} w^{(l)} + w^{(l)} \right). \tag{82}$$

$\widehat{S}_{\text{Params}}(a)$ counts the weights and biases of all hidden layers and the output classifier.

**Depth-Only.** The depth baseline is defined as:

$$\widehat{S}_{\text{Depth}}(a) = L(a), \tag{83}$$

where $L(a)$ counts only hidden layers and excludes the fixed output classifier.

**Simple Regression.** To rule out that the strong correlation originated from fitting on $A_{\text{cal}} \cup D_{\text{cal}}$, we introduce the simple architecture regression as follows:

$$\widehat{S}_{\text{Reg}}(a) = \gamma_0 + \gamma_1 \log P(a) + \gamma_2 L(a) + \gamma_3 \log \bar{w}(a), \tag{84}$$

where $\bar{w}(a) := \frac{1}{L(a)} \sum_{l=1}^{L(a)} w^{(l)}$ represents the mean hidden-layer width. The regression coefficient tensor $\boldsymbol{\gamma} = [\gamma_0, \gamma_1, \gamma_2, \gamma_3]$ is estimated by ordinary least squares using only $A_{\text{cal}} \cup D_{\text{cal}}$:

$$\widehat{\boldsymbol{\gamma}} = \arg\min_{\boldsymbol{\gamma}} \sum_{a \in A_{\text{cal}}} \left[ S_{\text{cal}}(a) - \widehat{S}_{\text{Reg}}(a) \right]^2, \tag{85}$$

where $S_{\text{cal}}(a)$ means the measured logit shift in the calibration set $D_{\text{cal}}$ of a specific architecture $a$.

As shown in Table 4, we report the Spearman correlation $r_s$ between the predicted score $\widehat{S}(a)$ of these baselines and the seed-averaged measured logit shift $S_{\text{test}}(a)$ over the held-out architectures $A_{\text{val}}$. These controls test whether ADS captures heterogeneous layer-wise structure beyond model size, depth, or a generic linear combination of simple architectural statistics.

### A.2.6 Validation of Expected Calibration Error

**Experimental Setup.** See details in Appendix A.2.1.

Table 4: Spearman correlations $r_s$ between ADS and the measured logit shift for different baselines across various scenarios in FNNs. We abbreviate Simple Architecture Regression as SAR. The bold values indicate the highest correlation within each scenario and calibration ratio.

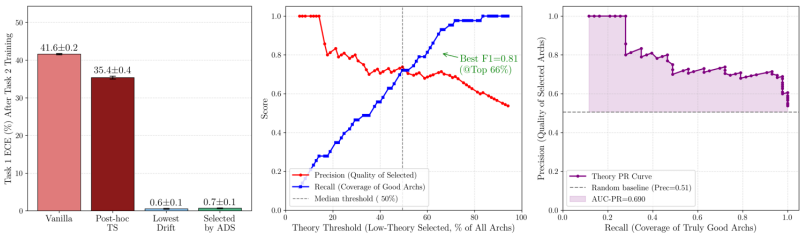
| SCENARIO | BASELINE | 5% | 10% | 15% | 20% | 25% | AVERAGE$_{\pm\text{STD}}$ |
|---|---|---|---|---|---|---|---|
| M → F | #PARAMS | 0.698 | 0.683 | 0.697 | 0.698 | 0.702 | $0.696_{\pm 0.007}$ |
| | DEPTH | 0.710 | 0.715 | 0.714 | 0.715 | 0.723 | $0.716_{\pm 0.004}$ |
| | SAR | 0.754 | 0.722 | 0.737 | 0.708 | 0.736 | $0.731_{\pm 0.017}$ |
| | ADS | **0.762** | **0.875** | **0.881** | **0.879** | **0.882** | $\mathbf{0.856}_{\pm 0.053}$ |
| F → M | #PARAMS | 0.742 | 0.725 | 0.737 | 0.726 | 0.734 | $0.733_{\pm 0.008}$ |
| | DEPTH | 0.790 | 0.792 | 0.796 | 0.784 | 0.804 | $0.793_{\pm 0.007}$ |
| | SAR | 0.772 | 0.794 | 0.808 | 0.772 | 0.799 | $0.789_{\pm 0.016}$ |
| | ADS | **0.890** | **0.932** | **0.931** | **0.932** | **0.939** | $\mathbf{0.925}_{\pm 0.020}$ |
| SPLIT M (0–4 → 5–9) | #PARAMS | 0.163 | 0.120 | 0.246 | 0.187 | 0.183 | $0.180_{\pm 0.046}$ |
| | DEPTH | 0.041 | 0.002 | 0.140 | 0.078 | 0.060 | $0.064_{\pm 0.051}$ |
| | SAR | -0.114 | 0.188 | 0.006 | 0.263 | 0.312 | $0.131_{\pm 0.179}$ |
| | ADS | **0.689** | **0.858** | **0.880** | **0.846** | **0.878** | $\mathbf{0.830}_{\pm 0.080}$ |
| SPLIT M (5–9 → 0–4) | #PARAMS | 0.121 | 0.098 | 0.206 | 0.182 | 0.167 | $0.155_{\pm 0.044}$ |
| | DEPTH | -0.061 | -0.090 | 0.025 | -0.013 | -0.019 | $-0.032_{\pm 0.045}$ |
| | SAR | 0.048 | 0.378 | 0.117 | 0.319 | 0.320 | $0.236_{\pm 0.145}$ |
| | ADS | **0.794** | **0.894** | **0.879** | **0.856** | **0.879** | $\mathbf{0.860}_{\pm 039}$ |
| SPLIT F (0–4 → 5–9) | #PARAMS | 0.246 | 0.239 | 0.295 | 0.280 | 0.276 | $0.267_{\pm 0.024}$ |
| | DEPTH | 0.005 | -0.011 | 0.079 | 0.029 | 0.041 | $0.029_{\pm 0.035}$ |
| | SAR | -0.154 | 0.424 | 0.240 | 0.454 | 0.471 | $0.287_{\pm 0.263}$ |
| | ADS | **0.649** | **0.831** | **0.852** | **0.785** | **0.813** | $\mathbf{0.786}_{\pm 0.080}$ |
| SPLIT F (5–9 → 0–4) | #PARAMS | 0.114 | 0.102 | 0.167 | 0.173 | 0.155 | $0.142_{\pm 0.032}$ |
| | DEPTH | -0.105 | -0.136 | -0.042 | -0.061 | -0.068 | $-0.082_{\pm 0.038}$ |
| | SAR | 0.223 | 0.470 | 0.282 | 0.408 | 0.435 | $0.364_{\pm 0.106}$ |
| | ADS | **0.748** | **0.750** | **0.854** | **0.818** | **0.839** | $\mathbf{0.802}_{\pm 0.050}$ |

**Experimental Results.** The empirical validation of the ADS-based selector's performance in reliable continual learning model selection is shown in Figure 7. Although the selected architectures do not achieve the ideal lowest Expected Calibration Error (ECE), they are more reliable than both the vanilla architecture and those calibrated with a post-hoc temperature scaling mechanism. As illustrated in the middle column of Figure 7, the ADS-based selector serves as an effective coarse-grained selector. It achieves high precision at low thresholds and high recall at high thresholds, indicating that it can identify the better-performing models or filter out the poorly performing ones with high probability. Furthermore, the high AUC-PR values across various scenarios demonstrate that the ADS-based selector consistently outperforms the random baseline, which typically yields an AUC-PR of 0.51.

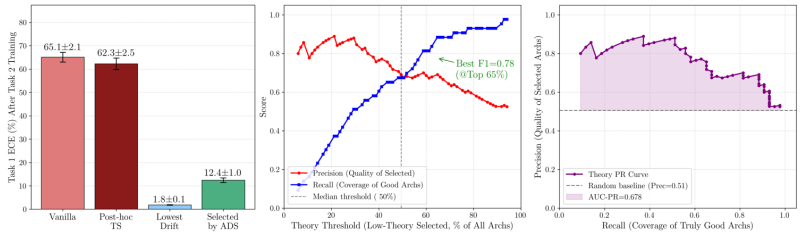
## A.3 DISCUSSION AND FUTURE WORK

### A.3.1 PRACTICAL USAGE AND RISK MITIGATION

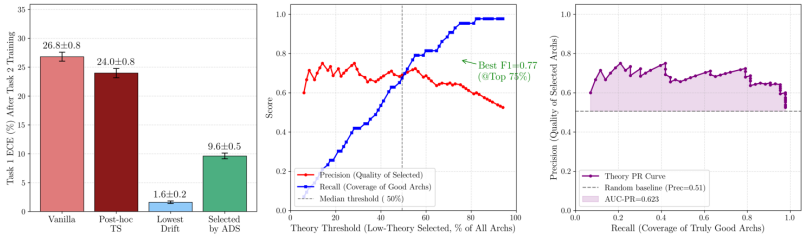
The reported instability in specific failure modes (e.g., CNNs with deep residual paths and rotation-heavy tasks) raises a valid concern: an unreliable selector can be detrimental if deployed blindly. To mitigate this, we emphasize that ADS is designed as a coarse-grained pre-filter rather than a final, absolute decision-maker. In realistic model selection pipelines, ADS can be employed to prune 70%-90% of the candidate architecture space at low cost. Only the top-ranked models selected by ADS need to undergo full downstream training for final validation. This hybrid strategy ensures the computational efficiency of ADS while hedging against its failure cases, as the final selection is still grounded in true CL metrics. When a scenario falls into known failure modes (e.g., heavy skip connections), we suggest supplementing ADS with simple heuristic baselines.



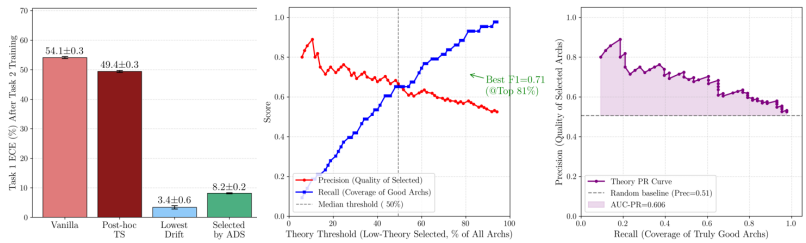

(a) MNIST (classes 5-9 → classes 0-4)



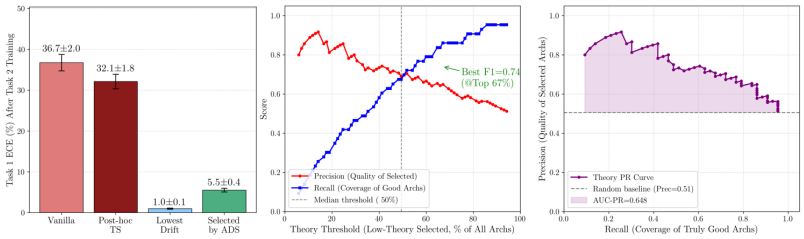

(b) FashionMNIST (classes 0-4 → classes 5-9)




(c) FashionMNIST (classes 5-9 → classes 0-4)




(d) CIFAR-10 (classes 0-4 → classes 5-9)




(e) CIFAR-10 (classes 5-9 → classes 0-4)

Figure 7: Quantitative evaluation of the ADS-based selector across all benchmarks: calibration efficacy (left), filtering trade-offs via precision/recall curves (middle), and PR curve with AUC-PR (right). Each row corresponds to a different scenario.

Table 5: Discrepancy between nominal and effective depth in CNNs with skip connections. Mean gaps and confidence intervals are obtained by bootstrapping the matched architectures. The third column reports the paired mean difference between C→M and each comparison scenario, i.e., $\mathbb{E}_{a\in A}\big[G_{(C\to M),a} - G_{(\text{other scenarios}),a}\big]$. The final column gives the Holm-adjusted one-sided paired Wilcoxon $p$-value.

| Scenario | Mean $G$ [95% CI] | C→M difference [95% CI] | $p_{\text{adj}}$ |
|---|---|---|---|
| C→M | 0.1775 [0.1708, 0.1839] | – | – |
| M→C | 0.1580 [0.1523, 0.1636] | 0.0195 [0.0149, 0.0242] | $4.58\times10^{-5}$ |
| F→C | 0.1626 [0.1571, 0.1675] | 0.0149 [0.0105, 0.0191] | $7.63\times10^{-5}$ |
| C→F | 0.1513 [0.1453, 0.1570] | 0.0261 [0.0226, 0.0298] | $4.58\times10^{-5}$ |

### A.3.2 Failure Case Exploratory Analysis of CNNs with Skip-Connection Mechanisms

**A Possible Contributor to the Failure Case.** We observe a failure case for CIFAR-10→MNIST in the CNN setting with skip connections: the average Spearman correlation between ADS and the measured logit shift drops to 0.433, compared with 0.893 for MNIST→CIFAR-10. To explore this failure, we examine the discrepancy between nominal block depth and effective depth along the Task-2 gradient path. Consider a network with $D$ blocks. For block $j$, we first measure its accumulated optimization activity during Task-2 training as

$$s_j = \sum_{t\in\mathcal{T}} \left\| \nabla_{\Theta^{(j)}} \mathcal{L}_2^{(t)} \right\|_F, \tag{86}$$

where $\Theta^{(j)}$ denotes the parameters of block $j$, and $\mathcal{T}$ is the set of recorded training steps. We then define its gradient-path effective depth as

$$\widetilde{d}_l = D\,\frac{\sum_{j=1}^{l} s_j}{\sum_{j=1}^{D} s_j}. \tag{87}$$

This construction allocates the total depth $D$ according to where the Task-2 gradient activity is concentrated. If all blocks contribute equally, then $\widetilde{d}_l = l$, recovering the nominal depth coordinate. In contrast, if skip connections redirect gradient propagation toward particular parts of the network, the cumulative gradient activity becomes non-uniform and $\widetilde{d}_l$ deviates from $l$. Because $\widetilde{d}_D = D$ by construction, the proxy captures the distribution of gradient activity across depth rather than its overall magnitude.

We quantify the discrepancy between nominal and effective depth by

$$G = \sqrt{\frac{1}{D}\sum_{l=1}^{D}\left(\frac{\widetilde{d}_l - l}{D}\right)^2}, \tag{88}$$

where a larger $G$ indicates a stronger departure from the nominal depth coordinate. Table 5 reports the results over CNNs with skip connections in all four scenarios. Here, M, C, and F denote MNIST, CIFAR-10, and FashionMNIST, respectively.

As shown in Table 5, C→M has the largest discrepancy. Its mean $G$ is larger than other scenarios, with paired differences ranging from 0.0149 to 0.0261. All corresponding 95% confidence intervals exclude zero, and all comparisons remain significant after Holm correction.

Therefore, C→M coincides with the strongest mismatch between nominal depth and the effective depth realized in the skip-connected network. This provides evidence that the mismatch may be a contributor: ADS indexes blocks using nominal depth, whereas the Task-2 gradient path assigns them substantially different effective coordinates. We nevertheless regard this result as exploratory rather than causal.

The F→C scenario exhibits a different degradation pattern. Across the six calibration ratios, its calibration-set Spearman correlation remains high (0.846–0.933), whereas its held-out correlation

Table 6: Low-shift architecture screening in the C→M failure setting. ADS is fitted on $A_{\mathrm{cal}}/D_{\mathrm{cal}}$ and evaluated on $A_{\mathrm{val}}/D_{\mathrm{test}}$ over 200 cost-stratified splits. Low-shift architectures are the bottom 30% by logit shift. Values in parentheses denote random-selection baselines.

| Retained (%) | Pruned (%) | Low-shift precision (%) | Low-shift recall (%) | Best-architecture retention (%) |
|---|---|---|---|---|
| 30 | 70 | 49.2 (32.0) | 49.2 (32.0) | 85.0 (32.0) |
| 40 | 60 | 46.2 (32.0) | 57.8 (40.0) | 88.5 (40.0) |
| 50 | 50 | 44.0 (32.0) | 66.1 (48.0) | 89.5 (48.0) |
| 60 | 40 | 43.3 (32.0) | 81.2 (60.0) | 91.0 (60.0) |

decreases from $0.643$ to $0.346$. Its nominal-to-proxy depth discrepancy is also smaller than that of C→M. Thus, the gradient-path depth mismatch should be interpreted as a scenario-specific correlate of the C→M failure rather than a universal explanation for all moderate-correlation settings. The F→C results instead suggest a distinct generalization issue, whose source—such as split sensitivity or depth–width interactions—requires further investigation.

**Practical utility of ADS for candidate prioritization.** Although ADS exhibits weak global ranking performance in the C→M setting, it retains a moderate average signal for identifying low-shift architectures, with a low-shift identification AUC of $0.706$. As shown in Table 6, increasing the retained fraction from 30% to 60% improves low-shift recall from 49.2% to 81.2%, while the probability of retaining the lowest-shift architecture in $A_{\mathrm{val}}$ increases from 85.0% to 91.0%.

At the 60% operating point, ADS prunes 40% of the candidate pool while achieving higher low-shift recall than matched random selection (81.2% vs. 60.0%). It also retains the lowest-shift held-out architecture in 91.0% of the splits, compared with 60.0% under random selection.

These average improvements indicate that ADS can enrich a reduced candidate pool for downstream validation. However, because performance remains sensitive to the particular architecture split, we interpret ADS as a conservative prioritization mechanism rather than a reliable global ranker or an aggressive hard-pruning rule. This result does not establish reliable high-shift detection or localization of ranking errors to the middle-shift region.

### A.3.3 Future Work

While the theoretical framework proposed in this paper demonstrates the effectiveness of ADS in capturing the logit shift trend, several promising directions warrant further investigation.

**Effective Depth Modeling.** Current ADS theory relies on nominal layer depth; however, in networks with skip connections as demonstrated in Appendix A.3.2, the actual effective propagation path of the gradient flow is different from the nominal depth. Future work will focus on deriving the "effective path" of residual networks from a signal propagation perspective and integrating it into the ADS framework.

**Analytical Modeling of Architecture-Dependent Depth Profile.** Currently, the quantification of architecture-dependent depth (i.e., the term $l^b \cdot e^{-cl}$ in Eq. 12 of main paper) relies on empirical fitting. In subsequent work, we will attempt to derive a theoretical closed-form solution for this profile, thereby completely eliminating the dependency on data calibration.

**Model-Family Diversity Modeling.** While ADS captures logit shifts across FNNs, CNNs, and ViTs, its theoretical foundation stems from fully-connected scaling laws. Extending it to modern architectures demands adapting to unique primitives, such as spatial dimensions/receptive fields for CNNs and attention heads/FFN expansions for ViTs. Future work will derive explicit gradient flow scaling laws for these modular blocks and integrate them into a composite ADS metric.

**Modeling Higher-Order Terms in the Taylor Expansion.** The current local linear approximation (Eq. 5 of main paper) may introduce bias under high learning rates or large parameter updates.

Future work will incorporate second-order perturbation theory to quantify the contribution of higher-order Taylor terms to the logit shift.

**Large-Scale Validation.** We will validate ADS on larger-scale real-world benchmarks (e.g., full ImageNet, large-scale multimodal tasks) and larger model families (e.g., larger Vision Transformers and Large Language Model fine-tuning scenarios) to further test the robustness and generality of ADS under diverse hardware and computational constraints.