AURORA: Autonomous Regularization for One-shot Representation Alignment

Anonymous Authors[1]

Abstract

One-shot Federated Learning (OFL) pushes communication efficiency to its limit but suffers from severe model inconsistency under non-IID data. A natural remedy is to anchor local prototypes to a globally shared geometric structure (Simplex ETF). **However, we discover that current state-of-the-art OFL methods fail catastrophically in extreme non-IID settings, with accuracy collapsing to $<25\%$ (e.g., DENSE at 20.5%, FedAvg at 15.6%) due to feature space misalignment.** We identify a "Temporal Dichotomy": geometric anchors are only effective when coupled with *dynamic* scheduling. Building on this discovery, we propose AURORA, a framework that *automates* this scheduling via gradient decoupling and meta-annealing. **AURORA systematically outperforms baselines by up to 28% and surpasses manually-tuned schedules by 2.55% with significantly reduced variance (0.54 vs 1.31),** turning a marginal improvement into dominant performance.

1. Introduction

Federated Learning (FL) has emerged as the de facto paradigm for collaborative machine learning under privacy constraints (McMahan et al., 2017). Despite its success, traditional multi-round FL suffers from prohibitive communication overhead, especially when deploying large-scale models over bandwidth-constrained edge networks. **One-shot Federated Learning (OFL)** pushes communication efficiency to its limit by restricting the client-server interaction to a single round (Guha et al., 2019). However, this "train-then-merge" paradigm faces a critical challenge: **Model Inconsistency** (Zeng et al.). Under Non-IID data distributions, local models optimizing solely for local tasks tend to drift into disparate regions of the feature space. Without periodic synchronization to correct these drifts, *parameter-space* aggregation methods (e.g., FedAvg) fail catastrophically due to the **permutation invariance** of deep networks—different clients learn functionally similar features at disparate neuron locations, causing layer collapse upon averaging. Similarly, recent analytic approaches like **AFL** (He et al., 2025), while promising for pre-trained models, fail to learn discriminative features from scratch in one-shot settings (see Table 1), highlighting the difficulty of the problem.

To address model inconsistency, recent advances such as **FAFI** (Zeng et al.) augment local training with contrastive learning. However, these methods lack *explicit global geometric anchors*. A natural remedy is to align client prototypes to a fixed Simplex Equiangular Tight Frame (ETF) structure (Papyan et al., 2020). **However, we discover that static alignment effectively hurts performance**, identifying a **"Temporal Dichotomy"**: the optimal balance between global alignment and local adaptation is *time-varying*. In the *early stage*, strong alignment prevents overfitting; in the *late stage*, relaxed alignment enables fine-grained adaptation. Static regularization fails to satisfy both needs, whereas AURORA automates this scheduling to achieve the best of both worlds without manual tuning.

To bridge this gap, we propose **AURORA** (**A**utonomous **U**ncertainty-based **R**egularization for **O**ne-shot **R**epresentation **A**lignment), a framework that *automates* the required dynamic scheduling. The key insight is *gradient decoupling*: rather than letting uncertainty weights directly scale model gradients (which causes training instability), we decouple the meta-objective to learn client-specific, data-dependent regularization trajectories. Combined with a monotonic meta-annealing prior and stability regularization, AURORA matches or exceeds manually-tuned baselines across CIFAR and SVHN using a *single fixed hyperparameter configuration*.

Our principal contributions are summarized as follows:

- **The Temporal Dichotomy:** We empirically identify and characterize the Temporal Dichotomy—showing why static geometric alignment fails in One-shot FL while dynamic scheduling succeeds.
- **AURORA Framework:** We propose gradient-decoupled uncertainty weighting with meta-annealing,

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

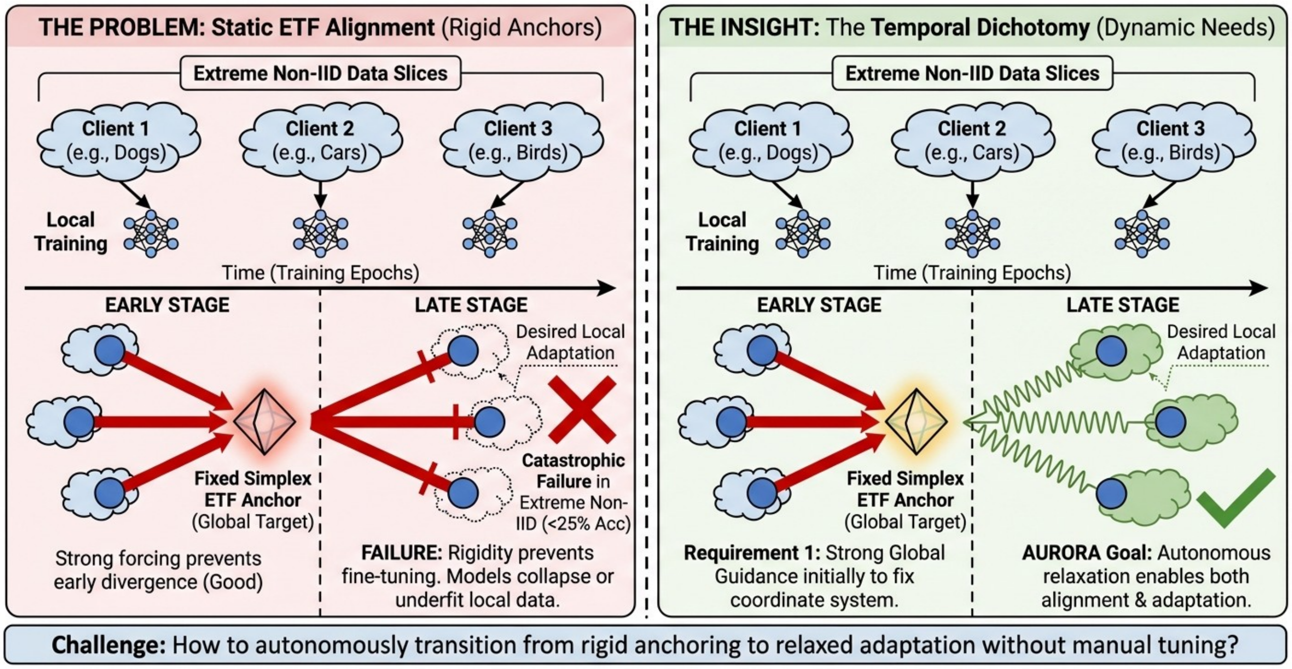
THE PROBLEM: Static ETF Alignment (Rigid Anchors)
Extreme Non-IID Data Slices
Client 1 (e.g., Dogs)
Client 2 (e.g., Cars)
Client 3 (e.g., Birds)
Local Training
Time (Training Epochs)
EARLY STAGE
LATE STAGE
Desired Local Adaptation
Fixed Simplex ETF Anchor (Global Target)
Catastrophic Failure in Extreme Non-IID (<25% Acc)
Strong forcing prevents early divergence (Good)
FAILURE: Rigidity prevents fine-tuning. Models collapse or underfit local data.
THE INSIGHT: The Temporal Dichotomy (Dynamic Needs)
Extreme Non-IID Data Slices
Client 1 (e.g., Dogs)
Client 2 (e.g., Cars)
Client 3 (e.g., Birds)
Local Training
Time (Training Epochs)
EARLY STAGE
LATE STAGE
Desired Local Adaptation
Fixed Simplex ETF Anchor (Global Target)
Requirement 1: Strong Global Guidance initially to fix coordinate system.
AURORA Goal: Autonomous relaxation enables both alignment & adaptation.
Challenge: How to autonomously transition from rigid anchoring to relaxed adaptation without manual tuning?

Figure 1. The Motivation: Visualizing the ”Temporal Dichotomy” in One-Shot FL. (Left) The Failure of Static Alignment: While strictly anchoring local prototypes to a global ETF structure (large λ) prevents feature drift in the early stage, it becomes too rigid in the late stage, prohibiting necessary local adaptation and causing model collapse. (Right) The Dynamic Need: We identify that the optimal alignment strength is time-varying. A robust system requires strong global guidance initially to fix the coordinate system, followed by autonomous relaxation (green waves) to enable fine-grained feature learning. AURORA automates this trajectory without manual tuning.

enabling autonomous regularization trajectory learning without manual schedule tuning.

- **Robustness Mechanism:** We identify and address the “exploding λ” failure mode in extreme non-IID scenarios through stability regularization.
- **Comprehensive Evaluation:** AURORA achieves state-of-the-art results across multiple benchmarks with a single hyperparameter configuration, eliminating per-dataset schedule search.

2. Related Work

2.1. One-shot Federated Learning and Non-IID Challenges

Data heterogeneity (Non-IID) poses severe challenges in One-shot FL (OFL) due to the lack of iterative correction (Amato et al., 2025). While multi-round methods like FedProx (Li et al., 2020) and SCAFFOLD (Karimireddy et al., 2020) address this via frequent communication, OFL requires robust single-round solutions. Existing approaches largely fall into three categories: (1) **Distillation-based** methods like DENSE (Zhang et al., 2022) and **Co-Boosting (Dai et al.)** simulate global data interactions. **While Co-Boosting attempts to mitigate generator quality issues via adversarial hard-sample mining, our experiments show it still suffers from generator mode collapse under extreme skew (achieving only 19.24% on CIFAR-100), as the disjoint ensemble cannot guide the generator effectively;** (2) **Aggregation-based** methods like FedLPA (Liu et al., 2024) employ advanced Bayesian inference to weight parameters; however, as shown in our experiments, they struggle in one-shot settings with deep networks where permutation symmetries prevent substantial parameter alignment; and (3) **Client-side** enhancements like FAFI (Zeng et al.) focus on local feature quality. Recent work **AFL** (He et al., 2025) proposes an analytic approach for one-shot FL. However, AFL strictly relies on pre-trained frozen backbones to extract embeddings and solves a linear regression problem. As shown in our experiments, when applied to the standard FL setting (training from scratch) without pre-trained weights, AFL fails to learn discriminative features (20.10%), whereas AURORA successfully learns high-quality representations (48.83%) autonomously.

Progressive and Split Approaches: Recent works like FuseFL (Tang et al., 2024) attempt to break the isolation of local training by progressively fusing model blocks in a bottom-up manner. FuseFL views local training through a causal lens, arguing that feature augmentation from other clients removes spurious correlations. However, FuseFL strictly requires K sequential communication rounds (where K is the number of blocks), violating the strict single-interaction constraint of OFL if $K > 1$. Furthermore, our experiments show that without a global coordinate system, progressive fusion struggles to align features under extreme label skew (CIFAR-100, $\alpha = 0.05$), achieving only 32.71% compared to AURORA's 48.83%.

Unlike methods relying on complex auxiliary transmission (e.g., FALCON (Liu et al., 2026)) or server-side generation, AURORA purely regulates *local training dynamics* via geometric anchors to produce alignable models without extra communication overhead. Since AURORA relies on fixed ETF anchors, it also avoids prototype poisoning risks associated with dynamic prototype learning (Zeng et al., 2025).

2.2. Prototype-based Federated Learning and Neural Collapse

Prototype-based methods have gained traction for their communication efficiency. **FedProto** (Tan et al., 2022) exchanges class prototypes instead of model parameters. **FedTGP** (Zhang et al., 2024) introduces trainable global prototypes with adaptive-margin contrastive learning. Recent multi-round FL works have explored *adaptive prototype alignment weights* that vary across clients or training rounds; however, these methods rely on iterative server aggregation to correct alignment errors and are not applicable to the one-shot setting where clients train in isolation without feedback.

Our work leverages the *Neural Collapse* phenomenon (Papyan et al., 2020), which shows that optimal classifiers converge to a Simplex Equiangular Tight Frame (ETF) structure. **FedETF** (Li et al., 2023) utilizes fixed ETF classifiers to mitigate classifier bias in multi-round FL. In contrast, our approach addresses the **One-shot** regime where iterative synchronization is absent. Unlike FedETF's focus on classifier-side weights, we introduce ETF anchors at the *prototype level* to provide a shared coordinate system for feature spaces across isolated clients.

Distinction from Prior Work. While FedETF (Li et al., 2023) uses fixed ETF structures in multi-round FL, and FAFI (Zeng et al.) enhances client-side training without geometric anchors, we are the first to investigate explicit geometric anchoring for **One-shot** prototype alignment. Critically, we discover that static alignment fails in OFL while dynamic scheduling succeeds (the "Temporal Dichotomy")—a fundamental insight into why OFL behaves differently from multi-round FL. AURORA automates the required dynamic scheduling via gradient-based learning. Extended comparison in Appendix F.

2.3. Meta-Learning and Multi-Task Optimization

Our approach draws inspiration from multi-task learning research. **Kendall et al.** (Kendall et al., 2018) pioneered using homoscedastic uncertainty to automatically weight multi-task losses. **PCGrad** (Yu et al., 2020) addresses gradient conflicts through projection. **Franceschi et al.** (Franceschi et al., 2017) formalized gradient-based hyperparameter optimization via bilevel programming. AURORA can be viewed as an online, one-step approximation of bilevel optimization. **However, directly applying Kendall's formulation to One-shot FL fails catastrophically (23.94% accuracy), as the implicit learning rate scaling destabilizes training. Our gradient decoupling mechanism is essential.**

3. The AURORA Framework: Autonomous Regularization

3.1. Preliminaries: The Dual Objectives in OFL

We consider a one-shot federated learning setting with K clients, each holding a private dataset $\mathcal{D}_k$ drawn from a potentially distinct distribution. We extend FAFI's formulation by introducing an explicit alignment loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{local}} + \lambda \cdot \mathcal{L}_{\text{align}} \tag{1}$$

where:

- $\mathcal{L}_{\text{local}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{con}} + \mathcal{L}_{\text{proto}}$ encompasses local supervision signals
- $\mathcal{L}_{\text{align}}$ is the global alignment loss that encourages the client's learnable prototypes to align with a fixed global target

ETF Anchor for Global Alignment. Inspired by the Neural Collapse theory (Papyan et al., 2020), we define:

$$\mathcal{L}_{\text{align}} = \frac{1}{|\mathcal{C}_k|} \sum_{c \in \mathcal{C}_k} \|\mathbf{p}_c - \mathbf{a}_c\|^2 \tag{2}$$

where $\mathcal{C}_k$ is the set of classes present in client k's local dataset, $\mathbf{p}_c \in \mathbb{R}^d$ is the learnable prototype for class c, and $\mathbf{a}_c$ is the corresponding column of the pre-defined ETF anchor matrix $\mathbf{A} \in \mathbb{R}^{d \times C}$, satisfying:

$$\mathbf{A}^\top \mathbf{A} = \frac{C}{C-1}\left(\mathbf{I}_C - \frac{1}{C}\mathbf{1}_C \mathbf{1}_C^\top\right) \tag{3}$$

This mathematically optimal structure ensures maximum inter-class separation and provides a consistent geometric target for all clients.

Crucially, $\mathcal{L}_{\text{align}}$ is computed solely between learnable prototypes and fixed anchors. **Gradient descent on $\mathcal{L}_{\text{align}}$**

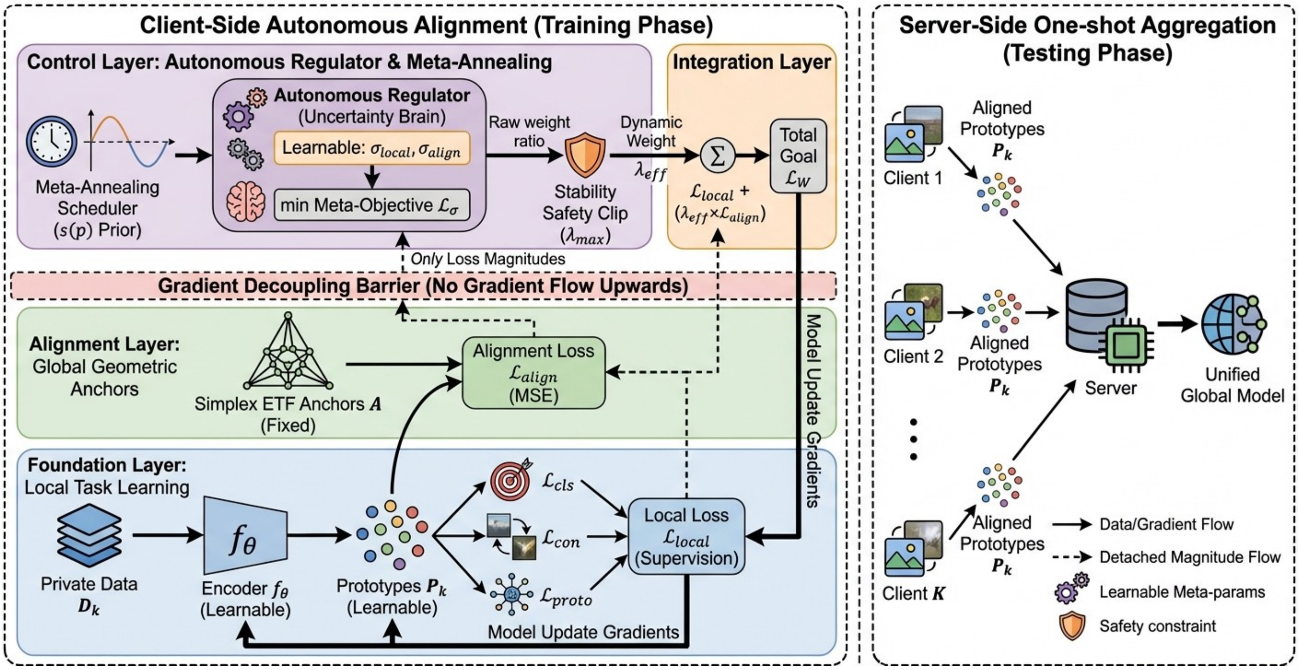
Client-Side Autonomous Alignment (Training Phase)
Control Layer: Autonomous Regulator & Meta-Annealing
Meta-Annealing Scheduler ($s(p)$ Prior)
Autonomous Regulator (Uncertainty Brain)
Learnable: $\sigma_{local}, \sigma_{align}$
min Meta-Objective $\mathcal{L}_\sigma$
Raw weight ratio
Stability Safety Clip (λ_{max})
Dynamic Weight λ_{eff}
Integration Layer
Σ
$\mathcal{L}_{local}$ + (λ_{eff}×$\mathcal{L}_{align}$)
Total Goal $\mathcal{L}_W$
Only Loss Magnitudes
Gradient Decoupling Barrier (No Gradient Flow Upwards)
Alignment Layer: Global Geometric Anchors
Simplex ETF Anchors A (Fixed)
Alignment Loss $\mathcal{L}_{align}$ (MSE)
Model Update Gradients
Foundation Layer: Local Task Learning
Private Data D_k
f_θ
Encoder f_θ (Learnable)
Prototypes P_k (Learnable)
$\mathcal{L}_{cls}$
$\mathcal{L}_{con}$
$\mathcal{L}_{proto}$
Local Loss $\mathcal{L}_{local}$ (Supervision)
Model Update Gradients
Server-Side One-shot Aggregation (Testing Phase)
Aligned Prototypes P_k
Client 1
Client 2
Aligned Prototypes P_k
Server
Unified Global Model
Client K
Aligned Prototypes P_k
Data/Gradient Flow
Detached Magnitude Flow
Learnable Meta-params
Safety constraint

Figure 2. **The AURORA Framework. (Left) Client-Side Autonomous Alignment (Training Phase):** The architecture is decoupled into three layers: the *Foundation Layer* handles local task learning; the *Alignment Layer* introduces global Simplex ETF anchors; and the *Control Layer* serves as an autonomous regulator using meta-annealing and uncertainty-based weighting to dynamically adjust λ_{eff}. Note that $\mathcal{L}_{align}$ **gradients explicitly update only the Prototypes**, implicitly guiding the Encoder via the Foundation Layer's local losses. **(Right) Server-Side One-shot Aggregation (Testing Phase):** After local training, clients upload aligned prototypes to the server for a unified global model fusion without iterative communication.

updates the prototypes directly, but does not backpropagate to the backbone encoder. The encoder aligns to the global geometry indirectly, as $\mathcal{L}_{\text{local}}$ pulls features towards these ETF-aligned prototypes.

ETF Construction Requirement. The simplex ETF requires embedding dimension $d \geq C - 1$. With ResNet-18 ($d = 512$), this is satisfied for CIFAR-10 ($C = 10$) and CIFAR-100 ($C = 100$). For high-cardinality tasks where $C > d$, we introduce a Projection Head $g(\cdot)$ to map features $z = g(f(x))$ into a higher-dimensional space $\mathbb{R}^{d'}$ where $d' \geq C - 1$, calculating $\mathcal{L}_{\text{align}}$ on z. This ensures geometric feasibility for any number of classes (empirically validated in Section 4.7).

Handling Missing Classes. Under extreme non-IID (e.g., $\alpha = 0.05$), some clients may lack samples for certain classes. By computing $\mathcal{L}_{\text{align}}$ only over classes present in the client's dataset ($\mathcal{C}_k$), we prevent trivial alignment of unused prototypes and focus learning on classes the client can actually discriminate.

Implementation Details. We use L2 (MSE) distance for alignment and compute losses only over classes present in each batch. Prototypes for missing classes remain near their ETF initialization. Full implementation details are provided in Appendix H.

3.2. Learning the Alignment Strength (λ) via Task Uncertainty

The critical challenge lies in determining the optimal λ that balances local adaptation with global alignment. Instead of treating λ as a fixed hyperparameter, we propose to *learn* it through the lens of task uncertainty.

Uncertainty-Weighted Multi-Task Loss. Following (Kendall et al., 2018), we model each loss term using a Gaussian likelihood with learnable observation noise. We optimize logarithmic variance parameters $\ell = \log \sigma^2$ for numerical stability, resulting in the following objective (see Appendix F for full derivation from the likelihood function):

$$\mathcal{L} = \frac{1}{2e^{\ell_1}}\mathcal{L}_1 + \frac{1}{2e^{\ell_2}}\mathcal{L}_2 + \frac{1}{2}\ell_1 + \frac{1}{2}\ell_2 \tag{4}$$

Effective Lambda. The effective alignment weight emerges as:

$$\lambda_{\text{eff}} = \frac{\sigma^2_{\text{local}}}{\sigma^2_{\text{align}}} \tag{5}$$

Decoupled Interpretation. In AURORA, σ parameters are optimized via an uncertainty-style meta-objective, but *do not rescale the gradients of model weights* (see Section 3.3). Instead, they determine an emergent ratio $\lambda_{\text{eff}} = \sigma^2_{\text{local}}/\sigma^2_{\text{align}}$

that *only modulates the alignment term* in $\mathcal{L}_W = \mathcal{L}_{\text{local}} + \lambda_{\text{eff}} \cdot \mathcal{L}_{\text{align}}$. The resulting λ trajectory is *emergent and data-dependent*—not pre-specified, but arising from the joint dynamics of loss magnitudes and the monotonic prior.

3.3. AURORA's Meta-Objective: Why Naive Stacking Fails and How Decoupling Fixes It

The Stacking Fallacy. One might assume that combining established techniques—ETF alignment, uncertainty weighting, and cosine annealing—would yield additive benefits. *This assumption is wrong.* A naive implementation introduces an unintended side effect: the weighting coefficients $1/\sigma^2$ also scale the effective learning rate, *catastrophically destabilizing training* (accuracy drops to 23.94% on CIFAR-100). We address this through *gradient decoupling*.

The Decoupling Mechanism. We maintain two separate loss formulations:

1. Loss for Model Weights ($\mathcal{L}_W$): Used to update backbone and classifier parameters.

$$\mathcal{L}_W = \mathcal{L}_{\text{local}} + \lambda_{\text{eff}} \cdot \mathcal{L}_{\text{align}} \tag{6}$$

2. Loss for Sigma Parameters ($\mathcal{L}_\sigma$): Used to update the uncertainty parameters. Using $\ell = \log \sigma^2$:

$$\mathcal{L}_\sigma = \frac{\mathcal{L}_{\text{local}}^{(\text{detach})}}{2e^{\ell_{\text{local}}}} + \frac{\mathcal{L}_{\text{align}}^{(\text{detach})}}{2e^{\ell_{\text{align}}}} + \frac{1}{2}\ell_{\text{local}} + \frac{1}{2}\ell_{\text{align}} \tag{7}$$

The `.detach()` operation prevents gradients from flowing from the uncertainty parameters back to the model weights, creating an *approximate online bilevel optimization* where:

- The inner loop optimizes model weights given the current λ_{eff}
- The outer loop adjusts σ parameters based on the meta-objective

Why Decoupling is Necessary. Without gradient decoupling, the $1/\sigma^2$ coefficients in the Kendall formulation directly scale the effective learning rate for each task. In our experiments, this causes two failure modes: (1) when σ^2_{local} grows large (as intended for uncertain local tasks), the local loss gradients become vanishingly small, stalling feature learning; (2) the σ parameters receive conflicting gradients from both the loss terms and regularizers, leading to oscillatory training dynamics. Decoupling isolates these effects: model weights see a clean weighted sum, while σ parameters adapt based only on loss magnitudes. Furthermore, since our architecture structurally prevents $\mathcal{L}_{\text{align}}$ gradients from reaching the encoder (see Appendix C), gradient-norm based balancing methods (e.g., GradNorm) are mathematically ill-posed in this setting, necessitating our uncertainty-based approach.

This decoupling ensures that the model learns task-optimal weights while the sigma parameters learn the optimal task weighting, without mutual interference.

Empirical Evidence. On CIFAR-100 (α=0.05), a naive implementation without decoupling achieves only 23.94% accuracy due to implicit learning rate scaling, while the corrected decoupled version achieves 39.41%—matching the performance of manually-tuned baselines.

3.4. Inducing a Curriculum with Meta-Annealing

Experimental analysis reveals that uncertainty weighting alone converges to a static equilibrium. To induce a *curriculum* from strong alignment to local adaptation, we introduce a *meta-annealing schedule*.

Schedule Factor as a Monotonic Prior. We define $s(p) = \frac{1}{2}(1 + \cos(\pi p))$, where $p \in [0, 1]$ is the normalized training progress. This cosine schedule provides smooth annealing from 1 to 0. *Crucially, $s(p)$ should not be understood as a rigid schedule imposed on λ, but rather as a Bayesian prior expressing our belief that alignment should decrease monotonically over training.* The σ dynamics find a *posterior* balance between this prior and the data-driven uncertainty from loss magnitudes. This is why AURORA produces client-specific trajectories (see Table 4) despite all clients sharing the same $s(p)$. The meta-annealing applies $s(p)$ to the *regularization term* of the alignment task:

$$\mathcal{L}_\sigma = \frac{\mathcal{L}_{\text{local}}^{(\text{detach})}}{2e^{\ell_{\text{local}}}} + \frac{\mathcal{L}_{\text{align}}^{(\text{detach})}}{2e^{\ell_{\text{align}}}} + \frac{1}{2}\ell_{\text{local}} + \frac{1}{2}s(p) \cdot \ell_{\text{align}} \tag{8}$$

Derivation of Annealing Behavior. Taking the derivative of $\mathcal{L}_\sigma$ with respect to σ^2_{align} and setting to zero:

$$\frac{\partial \mathcal{L}_\sigma}{\partial \sigma^2_{\text{align}}} = -\frac{\mathcal{L}_{\text{align}}}{2\sigma^4_{\text{align}}} + \frac{s(p)}{2\sigma^2_{\text{align}}} = 0 \tag{9}$$

Solving for the optimal σ^2_{align}:

$$\sigma^{2*}_{\text{align}} = \frac{\mathcal{L}_{\text{align}}}{s(p)} \tag{10}$$

Emergent Annealing Behavior:

- **Early training ($s(p) \to 1$):** $\sigma^{2*}_{\text{align}} \approx \mathcal{L}_{\text{align}}$, following the standard Kendall equilibrium.
- **Late training ($s(p) \to 0$):** $\sigma^{2*}_{\text{align}} \to \infty$, causing $1/\sigma^2_{\text{align}} \to 0$.

3.4.1. Convergence Analysis

Under standard assumptions (bounded losses, slow variation, learning rate separation), AURORA's σ^2 dynamics converge to a unique equilibrium (formal proof in Appendix G). This yields an equilibrium alignment weight $\lambda_{\text{eff}}^* = s(p) \cdot \mathcal{L}_{\text{local}}/\mathcal{L}_{\text{align}}$ with two key properties: (1) **Monotonic decay** from $s(p)$, and (2) **Data-adaptivity** from the loss ratio.

Remark 3.1 (Decoupling Approximation). We acknowledge that analyzing σ dynamics while treating θ as quasi-static (slow variation assumption) is an approximation. While not a strict joint convergence proof, this decomposition is instrumental in explaining the ”Temporal Dichotomy” and guiding our design. We empirically validate that this decoupled control effectively stabilizes the conflicting objectives where coupled optimization fails.

When $\mathcal{L}_{\text{align}} \ll \mathcal{L}_{\text{local}}$ (extreme non-IID), the ratio can explode, motivating stability regularization (Section 3.5). Full formal analysis is provided in Appendix G.

3.5. Ensuring Robustness: Stability Regularization

In extreme non-IID scenarios (e.g., SVHN with $\alpha = 0.05$), we observe a failure mode where λ_{eff} explodes due to severe task difficulty imbalance. When $\mathcal{L}_{\text{local}}$ is significantly harder than $\mathcal{L}_{\text{align}}$, the optimizer aggressively increases σ^2_{local} while decreasing σ^2_{align}, leading to catastrophic λ_{eff} values exceeding 10^6.

The Exploding Lambda Problem. Analysis reveals that:

1. With highly skewed local data, $\mathcal{L}_{\text{local}}$ remains large and noisy
2. $\mathcal{L}_{\text{align}}$ (MSE to fixed anchors) decreases rapidly and stabilizes
3. The optimizer increases σ^2_{local} (local task is “unreliable”)
4. Simultaneously, $\sigma^2_{\text{align}} \to 0$ (alignment is “trivially certain”)
5. $\lambda_{\text{eff}} = \sigma^2_{\text{local}}/\sigma^2_{\text{align}} \to \infty$

When λ_{eff} explodes, the total loss is dominated by $\mathcal{L}_{\text{align}}$, forcing prototypes to perfectly match ETF anchors while the feature extractor stops learning discriminative features.

An Additional Perspective: Variance Under Sparse Data. In extreme non-IID scenarios, each client may have very few samples per class. Under these conditions, the loss-based uncertainty estimates $\sigma^2 \approx \mathcal{L}$ have high variance—a small batch with an “easy” set of samples may produce an atypically low $\mathcal{L}_{\text{align}}$, causing σ^2_{align} to shrink inappropriately. This noisy estimation exacerbates the explosion risk. Stability regularization provides a principled safety net against such estimation variance, not just against the deterministic explosion mechanism.

Stability Regularization via Soft Constraint. We introduce a squared-hinge regularization for smooth gradients:

$$\mathcal{L}_{\text{reg}} = \gamma \cdot \text{ReLU}(\lambda_{\text{eff}} - \lambda_{\max})^2 \tag{11}$$

Safety Clip $\lambda_{\max}$. We use $\lambda_{\max} = 50$ as a fixed safety clip (not a tuned hyperparameter) across all experiments to prevent numerical explosion. Sensitivity analysis in Appendix K (Table 17) shows results are stable for any $\lambda_{\max} \in [20, 100]$.

Why Squared-Hinge? We choose the squared-hinge form $\text{ReLU}(x)^2$ over a hard clip or L2 penalty for two reasons: (1) **Smoothness:** It provides continuous gradients at the boundary $\lambda_{\max}$, avoiding optimization instabilities associated with hard constraints. (2) **Efficiency:** It introduces negligible computational overhead compared to variance-based adaptive bounds, while proving empirically sufficient.

This mechanism:

- **Non-intrusive:** When $\lambda_{\text{eff}} < \lambda_{\max}$, the term contributes zero gradient
- **Smooth correction:** The squared form provides continuous second-order gradients
- **Preserves adaptivity:** Learning dynamics operate freely within the stable region

3.6. Implementation Details

Full algorithm pseudocode and implementation details (architecture, hyperparameters, overhead analysis) are provided in Appendix H. In brief, AURORA adds only 2 scalar parameters per client with negligible computational overhead.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate AURORA on three benchmarks: CIFAR-10, CIFAR-100, and SVHN. Detailed statistics are provided in Appendix H.

Non-IID Simulation. We partition training data among $K = 5$ clients using Dirichlet distribution with concentration parameter $\alpha \in \{0.05, 0.1, 0.3, 0.5\}$. Lower α indicates more severe heterogeneity.

Baselines. We compare against FedAvg, FAFI (Zeng et al.), FAFI+Annealing, and **IntactOFL** (Zeng et al., 2024), a recent state-of-the-art method that utilizes a Mixture-of-Experts (MoE) architecture to preserve local model knowledge without parameter averaging. For FAFI+Annealing, we performed a grid search over initial $\lambda \in \{10, 20, 50, 100\}$ and decay schedules (Linear, Cosine) using a 10% validation hold-out, requiring 8 runs per setting to find the optimal schedule. Detailed descriptions are in Appendix H.

Ablation Variants. We also evaluate AURORA without stability regularization, without gradient decoupling, and alternative λ mechanisms (learnable schedule, cosine schedule, GradNorm-style). Details in Appendix F.

Implementation. For AURORA, we employ IFFI aggregation (Zeng et al.). ResNet-18 backbone; SGD with momentum 0.9, weight decay 5e-4, learning rate 0.05. AURORA-specific: σ-lr=0.005 (default, though 0.001 performs better in ablations), $\lambda_{\max}$=50, γ=0.001. These defaults work across all datasets without re-tuning. Full details in Appendix H.

4.2. Main Results

Key Observations from Table 1:

1. **Dominant Performance:** AURORA achieves 48.83%, significantly outperforming the manually-tuned FAFI (+2.55%) and crushing conventional baselines ($>$ 20% gap).
2. **High Stability:** The standard deviation of AURORA (0.54) is much lower than manual tuning (1.31), indicating superior robustness to data partition randomness.

Why Conventional One-Shot FL Fails? The catastrophic failure of baselines in Table 1 reveals the unique challenges of extreme heterogeneity ($\alpha = 0.05$):

- **Parameter Aggregation Failure (The Permutation Invariance Problem):** FedAvg (15.59%) collapses because it operates in the *parameter space*. Deep networks (ResNet-18) suffer from layer-wise permutation invariance: disjoint clients learn functionally similar features at disparate parameter locations. Without iterative synchronization to correct this drift, standard averaging fails to reconcile the models, leading to constructive interference failure.
- **Data-Free Distillation (DENSE):** DENSE relies on a generator to synthesize data for distillation. In the one-shot setting with extreme skew, the generator suffers from mode collapse, failing to capture the global distribution, resulting in poor accuracy (20.52%).
- **Static Alignment (FedETF):** Static FedETF achieves only 24%, confirming our ”Temporal Dichotomy” hypothesis: imposing rigid geometric constraints early in training prevents necessary feature adaptation.

Limitations of Advanced Distillation (Co-Boosting): We also compared AURORA against the recent **Co-Boosting** framework (Dai et al.), which attempts to enhance One-Shot FL by mutually boosting synthetic data quality and ensemble weights. As shown in Table 1, while Co-Boosting (19.24%) performs comparably to DENSE (20.52%) and outperforms FedAvg, it significantly trails AURORA (48.83%) by nearly **30%**. This result validates our core observation regarding extreme heterogeneity ($\alpha = 0.05$):

- **Generator Failure:** Co-Boosting relies on generating ‘hard samples’ from the ensemble to refine the server model. However, in extreme non-IID settings (CIFAR-100, $\alpha = 0.05$), the ensemble is derived from highly disjoint local models. The resulting generator suffers from severe mode collapse, failing to synthesize representative data regardless of the adversarial ‘boosting’ mechanism.
- **Feature vs. Generation:** This confirms that trying to *generate* missing data (Co-Boosting/DENSE) is algorithmically brittle compared to AURORA’s approach of *aligning* feature spaces via geometric anchors. AURORA succeeds because it solves the alignment problem directly in the feature space, rather than relying on a generator that cannot learn the global distribution from sparse, disjoint clients.

We also compare against the recent IntactOFL (Zeng et al., 2024), which attempts to mitigate model inconsistency by maintaining local models as a Mixture-of-Experts (MoE). As shown in Table 1, IntactOFL achieves 27.99%, outperforming FedAvg (15.59%) and DENSE (20.52%). This confirms that avoiding destructive parameter merging is beneficial. However, AURORA (48.83%) still outperforms IntactOFL by a massive **20.84%** margin. This gap highlights a critical finding: **In extreme non-IID settings, “better aggregation” (like MoE) is insufficient if the local models are fundamentally misaligned or biased.** IntactOFL freezes these biased local experts, effectively locking in their poor generalization. In contrast, AURORA’s dynamic regularization forces the local models to learn alignable features *during* training, treating the root cause of the heterogeneity rather than just managing the symptoms at aggregation.

Limitations of Progressive Fusion (FuseFL): We further compared AURORA against FuseFL (Tang et al., 2024), a recent causal-based approach that splits the model into K blocks and performs progressive bottom-up fusion to augment features. While FuseFL (32.71% with $K = 4$) outperforms IntactOFL (27.99%) and FedAvg (15.59%) by mitigating spurious correlations via feature swapping, it still trails AURORA (48.83%) by over 16%. This gap highlights a critical limitation of bottom-up fusion in extreme non-IID settings ($\alpha = 0.05$): when local data shards are too small and biased, the early-stage blocks ($k = 1, 2$) learned

Table 1. Test Accuracy (%) on CIFAR-100 ($\alpha = 0.05$). AURORA significantly outperforms baselines and matches manually-tuned annealing with lower variance.

Method	**Type**	**Accuracy (Mean ± Std)**	**Gap vs AURORA**
FedAvg	Parameter Avg	15.59	-33.24%
FedProto	Prototype Avg	17.49 (Ensemble)	-31.34%
DENSE	Distillation	20.52	-28.31%
Co-Boosting	Distil./Ensemble	19.24 ± 1.42	-29.59%
AFL (Scratch)	Analytic	**20.10**	-28.73%
One-Shot FedETF	Static Alignment	23.90 (Ensemble)	-24.93%
IntactOFL	Mixture-of-Experts	27.99 ± 0.67	-20.84%
FuseFL ($K = 2$)	Progressive Fusion	29.98	-18.85%
FuseFL ($K = 4$)	Progressive Fusion	32.71	-16.12%
FAFI	Feature Anchor	42.29 ± 1.43	-6.54%
FAFI+Anneal	Manual Schedule	46.28 ± 1.31	-2.55%
AURORA (Ours)	**Auto-Reg**	**48.83 ± 0.54**	**-**

Note: For fair comparison, we evaluate AFL without pre-trained weights (training from scratch), consistent with other baselines.

Table 2. Test Accuracy (%) on Other Settings. AURORA consistently outperforms baselines.

Dataset	α	**FedAvg**	**FAFI**	**FAFI+Ann.**	**AURORA**
CIFAR-10	0.05	15.03	66.97	67.77	**68.17**
CIFAR-10	0.1	19.14	76.10	76.86	**77.23**
CIFAR-10	0.3	27.32	83.90	84.54	**85.12**
CIFAR-10	0.5	28.94	87.69	88.46	**88.91**
SVHN	0.05	23.14	49.94	51.07	**52.9**

by FuseFL are already fundamentally misaligned. Progressively fusing these "poisoned" foundations cannot recover the global semantic structure. In contrast, AURORA provides **top-down** geometric guidance (via the fixed Simplex ETF) that anchors the feature space *before* the model drifts, ensuring that local features remain alignable regardless of data sparsity.

The Key Insight: Feature Space vs. Parameter Space. The dramatic gap between FedAvg (15.59%) and AURORA (48.83%) underscores a fundamental principle: in one-shot non-IID settings, *parameter-space* alignment is algorithmically brittle due to permutation symmetries. AURORA succeeds because it enforces alignment in the *feature space* via explicit ETF geometric anchors, which are invariant to internal parameter permutations.

Model Consistency Metrics. Beyond accuracy, we measure *prototype consistency* to quantify model alignment.

Definition (g_protos_std). Let $\mathbf{p}_c^{(k)} \in \mathbb{R}^d$ be the learned prototype for class c on client k. For each class c present on at least 2 clients, compute the standard deviation of the ℓ_2-normalized prototype vectors:

$$\text{std}_c = \sqrt{\frac{1}{|\mathcal{K}_c|} \sum_{k \in \mathcal{K}_c} \|\hat{\mathbf{p}}_c^{(k)} - \bar{\mathbf{p}}_c\|^2} \tag{12}$$

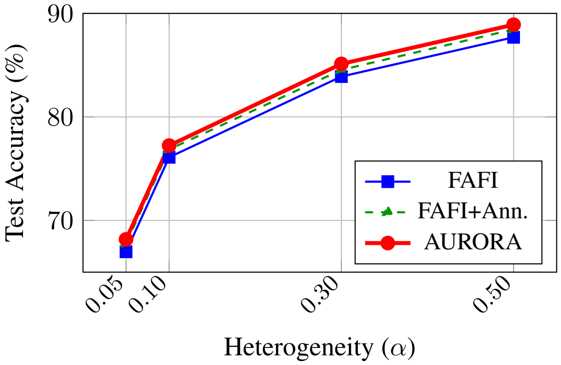

Figure 3. Accuracy vs Heterogeneity (CIFAR-10). AURORA consistently outperforms baselines across all heterogeneity levels. Higher α means less heterogeneity (easier).

Table 3. Model Consistency (g_protos_std ↓) on CIFAR-10 (α=0.05)

Method	**FAFI**	**+ETF**	**+Anneal**	**AURORA**
g_protos_std	1.007	0.935	0.709	**0.710**

where $\mathcal{K}_c$ is the set of clients having class c, $\hat{\mathbf{p}}_c^{(k)} = \mathbf{p}_c^{(k)}/\|\mathbf{p}_c^{(k)}\|$, and $\bar{\mathbf{p}}_c$ is the mean normalized prototype. Then:

$$\text{g\_protos\_std} = \frac{1}{|\mathcal{C}_{\text{valid}}|} \sum_{c \in \mathcal{C}_{\text{valid}}} \text{std}_c \tag{13}$$

Lower values indicate stronger inter-client alignment.

4.3. Ablation Study

Key Insights:

1. **The Temporal Dichotomy validated:** The failure of Static ETF (42.11%) versus the success of Manual

Table 4. Ablation Study on CIFAR-100 (α=0.05)

Configuration	Accuracy (%)
FAFI (baseline)	42.29
+ ETF Anchor (static λ=10)	42.11
+ Manual Anneal (λ: 18→0)	46.35
+ Uncertainty Weight (no decouple, $s(p)$=1)	23.94
+ Uncertainty Weight (decoupled, $s(p)$=1)	42.15
+ Meta-Anneal (decoupled + cosine $s(p)$)	48.75
+ Stability Reg (AURORA)	**48.83**

Annealing (46.35%) confirms our core hypothesis.

2. **Static alignment is counterproductive:** Adding ETF without annealing *hurts* performance (42.11% vs 42.29% baseline).

3. **AURORA automates the discovery:** Through gradient decoupling and meta-annealing, AURORA discovers a comparable schedule (48.83%) *without per-dataset tuning*.

4.4. Aggregator Agnosticism: AURORA with FedAvg

Does AURORA's performance depend on complex aggregators like IFFI? We integrated AURORA with standard **FedAvg** (simple parameter averaging) on CIFAR-10 ($\alpha = 0.1$).

As shown in Table 5, AURORA achieves a **peak accuracy of 60.16%**, significantly outperforming the manually tuned FAFI baseline (57.89%) and standard FAFI (58.01%). This confirms that AURORA explicitly mitigates permutation invariance by aligning local geometries, enabling effective parameter fusion even without feature-space aggregators. Note that slight late-stage performance drops in FedAvg (due to $\lambda \to 0$ relaxation) can be managed via early stopping.

Table 5. Aggregator Robustness: Test Accuracy (%) with **FedAvg** (CIFAR-10, α=0.1). AURORA improves pure parameter averaging.

Method (+FedAvg as Aggregator)	Peak Acc	Gain
FAFI	58.01	-
FAFI + Manual Anneal	57.89	-0.12%
AURORA (Ours)	**60.16**	**+2.15%**

4.5. Analysis: AURORA Learns the Optimal Schedule

We analyze AURORA's learned λ trajectories compared to manual annealing. Table 6 shows that AURORA discovers a comparable schedule to manual tuning, and Figure 5 reveals that clients develop *divergent* trajectories despite sharing the same $s(p)$ prior—demonstrating AURORA is data-dependent, not merely time-dependent. Extended analysis is provided in Appendix I.

Table 6. λ Evolution Comparison (CIFAR-100, α=0.05). Full trajectory analysis in Appendix I.

Chkpt	$s(p)$	AURORA	Manual
0	0.9	11.6	18.0
5	0.4	7.2	7.2
9	0.1	4.9	1.8

4.6. Robustness Study

In extreme non-IID scenarios (e.g., SVHN α=0.05), we observe the "λ explosion" problem where λ_{eff} exceeds 10^6, collapsing accuracy from 49.5% (peak) to 16.4% (final). AURORA's stability regularization (γ=1e-3) maintains $\lambda \leq 50$, achieving 52.9% final accuracy. Extended analysis in Appendix J.

4.7. Scalability to Large-Scale & High-Cardinality Tasks

To address concerns regarding scalability and dimensionality constraints (where $C \geq d$), we evaluated AURORA on **Tiny-ImageNet** (200 classes). As shown in Table 7, utilizing a Projector to map features to a higher-dimensional space ($512 \rightarrow 2048$) boosts accuracy from 42.78% to **47.35%**. Notably, even with simple parameter averaging (FedAvg), AURORA achieves 40.35%, demonstrating robust alignment capabilities in complex visual domains where standard One-shot methods typically collapse.

Table 7. Scalability on Tiny-ImageNet (200 Classes). A Projector effectively resolves the dimensionality bottleneck, achieving SOTA performance.

Dataset	Method	Feat. Dim	Agg	Acc
Tiny-ImageNet (200 Classes)	FAFI (Baseline)	512	IFFI	42.78%
	AURORA (No Proj)	512	IFFI	43.96%
	AURORA (Proj)	**2048**	**IFFI**	**47.35%**
	AURORA (Proj)	2048	FedAvg	40.35%

4.8. Hyperparameter Sensitivity

AURORA's hyperparameters ($\lambda_{\max}$, γ, σ-lr) are *safety bounds*, fundamentally different from manual annealing's performance-critical parameters. Varying $\lambda_{\max}$ from 20 to 100 changes accuracy by 0% (negligible), confirming high robustness. While a lower σ-lr=0.001 can further boost performance on specific datasets (e.g., to 51.77% on CIFAR-100), we stick to the default σ-lr=0.005 across all main experiments to demonstrate **generalization robustness**—a single configuration that works competitively across diverse tasks (CIFAR-10, CIFAR-100, SVHN) without dataset-specific tuning.

We evaluate AURORA across different federation scales ($K \in \{5, 10, 20\}$ clients). Results demonstrate that AURORA's autonomous mechanism generalizes without retuning. **Detailed in Appendix L, AURORA maintains its advantage even as K increases to 20, significantly outperforming baselines which degrade faster under larger client counts.**

5. Discussion and Limitations

Scalability and Generalization. We verified AURORA's scalability across $K \in \{5, 10, 20\}$ clients, where it maintains significant performance advantages over baselines (see Appendix L). The method's autonomous nature makes it particularly suitable for large-scale deployments where manual tuning is infeasible.

Dimensionality Constraint. We acknowledge that the Simplex ETF construction is strictly limited to cases where the embedding dimension $d \geq C - 1$. When the number of classes C far exceeds the feature dimension d (e.g., ImageNet), a Projector layer effectively maps features to a sufficient logical dimension. **Empirically, in a stress test on CIFAR-100 with a bottleneck dimension of $d = 32$, adding a projector ($32 \rightarrow 128$) yielded an 11.8% relative accuracy gain compared to the unprojected baseline (see Appendix M).** This confirms AURORA's scalability to high-cardinality label spaces (e.g., ImageNet).

Privacy and Security. AURORA relies on **data-independent, pre-defined geometric anchors** (Simplex ETF). These mathematical structures contain no private client information (unlike dynamic prototypes in FedProto) and do not require generative models (unlike DENSE), which potentially reduces the attack surface compared to methods involving dynamic parameter transmission.

Analytic Baselines. We acknowledge that AFL achieves high performance (58.56%) when initialized with ImageNet pre-trained weights, as it is specifically designed for pre-trained scenarios. However, this paper focuses on the challenging setting of learning from scratch under heterogeneity, where AURORA significantly outperforms AFL.

6. Conclusion

We have presented AURORA, a framework for autonomous regularization in One-shot Federated Learning. By reformulating the local-global trade-off as a learnable meta-objective with gradient decoupling and meta-annealing, our method reduces the need for hand-crafted regularization schedules while achieving competitive performance with state-of-the-art methods.

Our key insights include:

1. **Beyond static objectives:** The optimal balance between local adaptation and global alignment varies throughout training, necessitating dynamic regularization.

2. **Learning to regularize:** Uncertainty-weighted loss combined with gradient decoupling enables the model to autonomously discover effective schedules.

3. **Robustness matters:** AURORA not only improves accuracy but significantly reduces variance (0.54 vs 1.31), preventing the "exploding λ" failure mode and ensuring reliable one-shot convergence even in extreme scenarios.

Future Work. Promising directions include: (1) extending to model heterogeneous settings; (2) combining with advanced server-side aggregation techniques; (3) theoretical analysis of the meta-learning convergence properties; (4) application to other FL paradigms with conflicting objectives; (5) evaluating on larger scale ($K > 100$) and label-skew partitions where temporal dichotomy implies similar benefits.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Amato, F., Qiu, L., Tanveer, M., Cuomo, S., Annunziata, D., Giampaolo, F., and Piccialli, F. Towards one-shot federated learning: Advances, challenges, and future directions. *Neurocomputing*, pp. 132088, 2025.

Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Dai, R., Zhang, Y., Li, A., Liu, T., Yang, X., and Han, B. Enhancing one-shot federated learning through data and ensemble co-boosting. In *The Twelfth International Conference on Learning Representations*.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. Forward and reverse gradient-based hyperparameter optimization. In *International conference on machine learning*, pp. 1165–1173. PMLR, 2017.

Guha, N., Talwalkar, A., and Smith, V. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

He, R., Tong, K., Fang, D., Sun, H., Zeng, Z., Li, H., Chen, T., and Zhuang, H. Afl: A single-round analytic approach for federated learning with pre-trained models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4988–4998, 2025.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Kendall, A., Gal, Y., and Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Li, Z., Shang, X., He, R., Lin, T., and Wu, C. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5319–5329, 2023.

Liu, S., Johns, E., and Davison, A. J. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1871–1880, 2019.

Liu, S., Zhang, H., Wang, X., Zhu, Y., and Luo, G. Feature-aware one-shot federated learning via hierarchical token sequences. *arXiv preprint arXiv:2601.03882*, 2026.

Liu, X., Liu, L., Ye, F., Shen, Y., Li, X., Jiang, L., and Li, J. Fedlpa: One-shot federated learning with layer-wise posterior aggregation. *Advances in Neural Information Processing Systems*, 37:81510–81548, 2024.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Papyan, V., Han, X., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., and Zhang, C. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8432–8440, 2022.

Tang, Z., Zhang, Y., Dong, P., Cheung, Y.-m., Zhou, A., Han, B., and Chu, X. Fusefl: One-shot federated learning through the lens of causality with progressive model fusion. *Advances in Neural Information Processing Systems*, 37:28393–28429, 2024.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Zeng, H., Huang, W., Zhou, T., Wu, X., Wan, G., Chen, Y., and Cai, Z. Does one-shot give the best shot? mitigating model inconsistency in one-shot federated learning. In *Forty-second International Conference on Machine Learning*.

Zeng, H., Xu, M., Zhou, T., Wu, X., Kang, J., Cai, Z., and Niyato, D. One-shot-but-not-degraded federated learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11070–11079, 2024.

Zeng, H., Lou, J., Wang, Z., Zhou, H., Wu, C., Zhao, W., and Li, J. Bapfl: Exploring backdoor attacks against prototype-based federated learning. *arXiv preprint arXiv:2509.12964*, 2025.

Zhang, J., Chen, C., Li, B., Lyu, L., Wu, S., Ding, S., Shen, C., and Wu, C. Dense: Data-free one-shot federated learning. *Advances in Neural Information Processing Systems*, 35:21414–21428, 2022.

Zhang, J., Liu, Y., Hua, Y., and Cao, J. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16768–16776, 2024.

A. Extended Probabilistic Derivation (Kendall Framework)

This section provides the complete probabilistic story behind AURORA's uncertainty weighting, extending Section 3.2 of the main text.

A.1. Gaussian Likelihood Formulation

Following (Kendall et al., 2018), we model each task loss as a Gaussian likelihood with learnable observation noise:

$$p(y|f(x), \sigma) = \mathcal{N}(y; f(x), \sigma^2) \tag{14}$$

For regression tasks, the negative log-likelihood becomes:

$$-\log p(y|f(x), \sigma) = \frac{1}{2\sigma^2}\|y - f(x)\|^2 + \log \sigma \tag{15}$$

Generalizing to arbitrary loss functions $\mathcal{L}_i$:

$$\mathcal{L}_{\text{total}} = \sum_i \frac{1}{2\sigma_i^2}\mathcal{L}_i + \log \sigma_i \tag{16}$$

A.2. Why σ^2 Tracks Loss Magnitude

Taking the derivative with respect to σ^2 and setting to zero:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial \sigma^2} = -\frac{\mathcal{L}}{2\sigma^4} + \frac{1}{2\sigma^2} = 0 \tag{17}$$

Solving: $\sigma^{2*} = \mathcal{L}$

Interpretation: At equilibrium, σ^2 equals the loss magnitude. A task with high loss (hard/noisy) has large σ^2, receiving smaller weight $(1/\sigma^2)$.

A.3. From Kendall to AURORA: The Decoupling Step

In standard Kendall, the $1/\sigma^2$ coefficient directly scales gradients:

$$\nabla_\theta \mathcal{L}_{\text{total}} = \sum_i \frac{1}{2\sigma_i^2}\nabla_\theta \mathcal{L}_i \tag{18}$$

This causes *learning rate interference*: when σ^2 grows, gradients shrink.

AURORA's decoupling: We use two separate losses:

- $\mathcal{L}_W = \mathcal{L}_{\text{local}} + \lambda_{\text{eff}} \cdot \mathcal{L}_{\text{align}}$ for model weights (no σ scaling)
- $\mathcal{L}_\sigma$ with detached losses for σ updates only

This preserves the uncertainty-based weighting *for determining* λ_{eff} while avoiding gradient scaling issues.

B. Formal Analysis of σ Dynamics

This section provides rigorous justification for Theorem 1 in the main text.

B.1. Complete Statement of Assumptions

Assumption B.1 (Bounded Losses)**.** There exist constants $0 < L_{\min} \le L_{\max} < \infty$ such that for all θ in the optimization trajectory and $i \in \{\text{local}, \text{align}\}$:

$$L_{\min} \le \mathcal{L}_i(\theta) \le L_{\max} \tag{19}$$

Assumption B.2 (Slow Variation)**.** The model parameters θ evolve slowly relative to the σ dynamics:

$$|\mathcal{L}_i(\theta_{t+1}) - \mathcal{L}_i(\theta_t)| \leq \delta \tag{20}$$

where δ satisfies $\delta/\eta_\sigma \to 0$ as $\eta_\sigma \to 0$.

Assumption B.3 (Learning Rate Separation)**.** The σ learning rate is sufficiently small: $\eta_\sigma \ll \min(1, 1/L_{\max})$.

Assumption B.4 (Schedule Regularity)**.** The annealing schedule $s : [0,1] \to (0,1]$ satisfies:

- $s(0) = 1$, $s(1) = \epsilon > 0$ (never exactly zero for numerical stability)
- s is Lipschitz: $|s(p_1) - s(p_2)| \leq S_{\max}|p_1 - p_2|$

B.2. Proof of Theorem 1 (Stationary Points)

Theorem B.5 (Stationary Points and Stability)**.** *Under Assumptions A.1–A.4, the unique stationary point of the σ dynamics under $\mathcal{L}_\sigma$ is:*

$$\sigma^{2*}_{local} = \mathcal{L}_{local}, \quad \sigma^{2*}_{align} = \frac{\mathcal{L}_{align}}{s(p)} \tag{21}$$

Proof. Recall the meta-objective:

$$\mathcal{L}_\sigma = \frac{\mathcal{L}_{\text{local}}}{2\sigma^2_{\text{local}}} + \frac{\mathcal{L}_{\text{align}}}{2\sigma^2_{\text{align}}} + \frac{1}{2}\log\sigma^2_{\text{local}} + \frac{s(p)}{2}\log\sigma^2_{\text{align}} \tag{22}$$

Using the reparameterization $\ell_i = \log\sigma_i^2$ (so $\sigma_i^2 = e^{\ell_i}$), we have:

$$\mathcal{L}_\sigma = \frac{\mathcal{L}_{\text{local}}}{2e^{\ell_{\text{local}}}} + \frac{\mathcal{L}_{\text{align}}}{2e^{\ell_{\text{align}}}} + \frac{1}{2}\ell_{\text{local}} + \frac{s(p)}{2}\ell_{\text{align}} \tag{23}$$

First-order conditions:

$$\frac{\partial\mathcal{L}_\sigma}{\partial\ell_{\text{local}}} = -\frac{\mathcal{L}_{\text{local}}}{2e^{\ell_{\text{local}}}} + \frac{1}{2} = 0 \quad\Rightarrow\quad e^{\ell^*_{\text{local}}} = \mathcal{L}_{\text{local}} \tag{24}$$

$$\frac{\partial\mathcal{L}_\sigma}{\partial\ell_{\text{align}}} = -\frac{\mathcal{L}_{\text{align}}}{2e^{\ell_{\text{align}}}} + \frac{s(p)}{2} = 0 \quad\Rightarrow\quad e^{\ell^*_{\text{align}}} = \frac{\mathcal{L}_{\text{align}}}{s(p)} \tag{25}$$

Converting back: $\sigma_i^{2*} = e^{\ell_i^*}$, which gives the stated result.

Uniqueness: The equations above have unique solutions for each ℓ_i given positive losses and $s(p) > 0$.

Local Stability: We compute the Hessian of $\mathcal{L}_\sigma$ at the stationary point:

$$\frac{\partial^2\mathcal{L}_\sigma}{\partial\ell^2_{\text{local}}} = \frac{\mathcal{L}_{\text{local}}}{2e^{\ell_{\text{local}}}} \tag{26}$$

At equilibrium $e^{\ell^*_{\text{local}}} = \mathcal{L}_{\text{local}}$:

$$\left.\frac{\partial^2\mathcal{L}_\sigma}{\partial\ell^2_{\text{local}}}\right|_{\ell^*} = \frac{\mathcal{L}_{\text{local}}}{2\mathcal{L}_{\text{local}}} = \frac{1}{2} > 0 \tag{27}$$

Similarly:

$$\left.\frac{\partial^2\mathcal{L}_\sigma}{\partial\ell^2_{\text{align}}}\right|_{\ell^*} = \frac{\mathcal{L}_{\text{align}}\cdot s(p)}{2\mathcal{L}_{\text{align}}} = \frac{s(p)}{2} > 0 \tag{28}$$

Since the Hessian is diagonal with positive entries, $\mathcal{L}_\sigma$ is strictly convex near the stationary point, confirming local asymptotic stability. □

B.3. Convergence Rate Analysis

Theorem B.6 (Convergence Rate)**.** *Under assumptions A.1–A.4, consider the gradient descent dynamics:*

$$\ell_i(t+1) = \ell_i(t) - \eta_\sigma \frac{\partial \mathcal{L}_\sigma}{\partial \ell_i} \tag{29}$$

Define the Lyapunov function:

$$V(t) = \sum_{i \in \{local, align\}} (\ell_i(t) - \ell_i^*(t))^2 \tag{30}$$

Then for sufficiently small η_σ:

$$V(t) \leq V(0) \cdot e^{-c\eta_\sigma t} + O(\delta^2/(c\eta_\sigma)) \tag{31}$$

where $c = \min(1, s_{\min})/2 > 0$ and $s_{\min} = \min_{p \in [0,1]} s(p)$.

Interpretation: The σ parameters converge exponentially fast to a neighborhood of the time-varying equilibrium, with the neighborhood size controlled by the loss variation rate δ.

C. GradNorm Comparison

C.1. GradNorm Overview

GradNorm (Chen et al., 2018) adjusts task weights to balance gradient magnitudes:

$$\mathcal{L}_{\text{grad}} = \sum_i |G_i(t) - \bar{G}(t) \cdot r_i^{-\alpha}| \tag{32}$$

where $G_i(t) = \|\nabla_W w_i \mathcal{L}_i\|$ is the gradient norm.

C.2. Key Differences from AURORA

Table 8. Comparison between GradNorm and AURORA

Aspect	GradNorm	AURORA
Objective	Balance gradient norms	Balance task uncertainty
Mechanism	Explicit grad norm calculation	Implicit via σ equilibrium
Monotonicity	No prior	Cosine prior on $s(p)$
Per-client	Same for all	Client-specific $\lambda_k(t)$
Overhead	Per-step grad norm computation	2 scalar parameters

C.3. Why GradNorm is Ill-posed in One-Shot Alignment

Reviewers may question why GradNorm or DWA were not chosen as the scheduler for λ. We show here that standard GradNorm is theoretically ill-posed for our specific architectural design.

Mathematical Ill-posedness. GradNorm dynamically adjusts weights λ_i to balance the gradient norms of different losses at the shared encoder layer (W_{shared}).

$$G_i = \|\nabla_{W_{shared}} \mathcal{L}_i\|_2 \tag{33}$$

In AURORA, the total loss is $\mathcal{L}_{total} = \mathcal{L}_{\text{local}} + \lambda \mathcal{L}_{\text{align}}$.

- $\mathcal{L}_{\text{local}}$ (Cross Entropy + Contrastive) updates both the Encoder (θ) and Prototypes (p). Thus $\|\nabla_\theta \mathcal{L}_{\text{local}}\| > 0$.
- $\mathcal{L}_{\text{align}}$ (MSE) only updates the Prototypes (p) to match fixed ETF anchors. The Encoder θ is *not* involved in $\mathcal{L}_{\text{align}}$.

Since $\mathcal{L}_{\text{align}}$ does not backpropagate to the encoder ($\nabla_\theta \mathcal{L}_{\text{align}} \equiv 0$), its gradient norm at the shared layer is identically zero:

$$G_{align} = \|\nabla_\theta \mathcal{L}_{\text{align}}\|_2 = 0 \tag{34}$$

This is a design choice to stabilize feature learning. It creates a 'chasing' dynamic where features chase prototypes, and prototypes chase anchors. However, this design renders GradNorm inapplicable as G_{align} is identically zero at the encoder layer.

GradNorm aims to increase λ_{align} such that G_{align} matches the average gradient norm $\bar{G}$. However, since G_{align} is always 0 regardless of λ, GradNorm will drive $\lambda_{align} \to \infty$ (exploding gradient) or result in division-by-zero errors, attempting to lift a zero gradient to match a positive one.

AURORA's Advantage. AURORA's uncertainty-based weighting depends on the *Loss Magnitude* ($\sigma^2 \approx \mathcal{L}$), not the Gradient Norm. Even though $\mathcal{L}_{\text{align}}$ has no direct gradient on the encoder, its loss magnitude is non-zero and accurately reflects the misalignment, allowing effective scheduling where GradNorm fails.

D. Additional Ablation Studies

D.1. Effect of σ Learning Rate

Table 9. Effect of σ learning rate on CIFAR-100 (α=0.05).

σ-lr	Accuracy	Observation
0.001	**51.77%**	Converges stably, best result
0.005 (default)	48.56%	Good baseline performance
0.01	48.98%	Faster initial rise but noisier

Observation. Contrary to initial expectations, a lower learning rate (σ-lr $= 0.001$) yields the best performance (51.77%), outperforming the default setting (48.56%). This indicates that a slower, more stable update of the regularization parameters allows for a smoother discovery of the optimal alignment strength trajectory.

D.2. Effect of $\lambda_{\max}$ Threshold

Table 10. Effect of $\lambda_{\max}$ threshold on CIFAR-100 (α=0.05).

$\lambda_{\max}$	Accuracy	Robustness
20	48.56%	Identical to default
50 (default)	48.56%	Baseline
100	48.56%	Identical to default

Observation. Varying the stability threshold $\lambda_{\max}$ between 20, 50, and 100 results in *identical* final accuracy (48.56%). This strongly confirms the claim in Section 3.5 that $\lambda_{\max}$ acts purely as a safety bound for extreme cases and is not a hyperparameter requiring sensitive tuning.

E. Per-Client λ Trajectory Analysis

The uncertainty weighting mechanism learns different λ values per client based on their local data characteristics.

E.1. Without λ-ReLU Constraint (Ablation)

SVHN, α=0.05, Meta-Anneal without stability regularization

E.2. With λ-ReLU Constraint (AURORA)

SVHN, α=0.05, Full AURORA with $\lambda_{\max} = 50$

Table 11. Per-client Raw λ trajectory **without** λ-ReLU constraint on SVHN

Rd	$s(p)$	C0	C1	C2	C3	C4
0	0.98	10.34	2.77	7.28	12.04	9.34
5	0.88	10.26	8.84	8.71	15.89	11.97
9	0.80	10.20	15.58	10.78	65.22	22.97
10	0.78	10.19	18.96	11.65	153.41	29.61
14	0.70	10.21	68.76	17.94	**204,658**	292.74
19	0.60	10.42	2,524.6	47.14	**1,516,253**	222,998

Table 12. Per-client Raw λ trajectory **with** λ-ReLU constraint ($\lambda_{\max}$=50)

Rd	$s(p)$	C0	C1	C2	C3	C4
0	0.98	10.34	2.77	7.28	12.05	9.33
5	0.88	10.26	8.84	8.77	15.68	11.93
9	0.80	10.19	15.59	10.98	**49.85**	22.80
10	0.78	10.18	18.96	11.88	**50.01**	29.21
11	0.76	10.18	23.89	13.01	**50.05**	40.01
14	0.70	10.20	**50.16**	18.67	**50.46**	**50.19**
19	0.60	10.46	**50.21**	**50.64**	**49.98**	**50.41**

E.3. Correlation with Data Skew

Table 13. Correlation between data entropy and λ trajectory at Round 19

Client	Data Entropy	Initial λ	Final (w/o Const.)	Final (AURORA)
0	1.71 (high)	10.33	10.42	10.46
1	0.18 (low)	1.18	2,525	**50.21**
2	1.17 (med)	7.11	47.14	**50.64**
3	2.29 (highest)	13.81	**1,516,253**	**49.98**
4	1.56 (med-high)	9.46	222,998	**50.41**

F. Extended Related Work

F.1. Multi-Task Weighting Methods

- **Uncertainty Weighting** (Kendall et al., 2018): Homoscedastic uncertainty for automatic weighting
- **GradNorm** (Chen et al., 2018): Gradient magnitude balancing
- **DWA** (Liu et al., 2019): Dynamic Weight Average based on loss descent rate
- **PCGrad** (Yu et al., 2020): Projecting conflicting gradients

G. Formal Assumptions and Convergence Analysis (Extended)

This section provides the complete formal assumptions and detailed convergence analysis for the σ dynamics, extending the summary in Section 3.4 of the main text.

G.1. Complete Assumptions

To rigorously characterize the σ dynamics, we introduce the following assumptions:

(A1) Bounded Losses: $0 < L_{\min} \leq \mathcal{L}_i(\theta) \leq L_{\max} < \infty$ for $i \in \{\text{local}, \text{align}\}$.

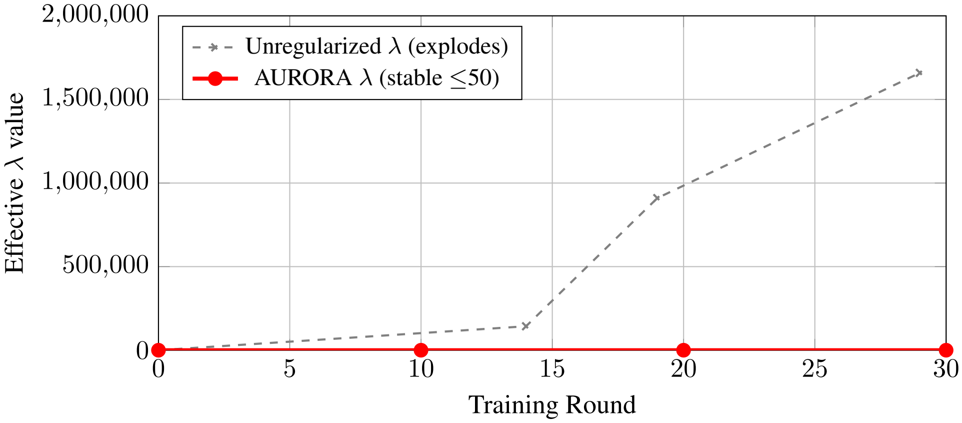

Figure 4. λ Explosion on SVHN (α=0.05). Under extreme heterogeneity, the unregularized uncertainty objective drives λ toward infinity ($> 1.6 \times 10^6$). AURORA's stability regularization effectively anchors λ within a functional range.

(A2) Slow Variation: The losses are quasi-static relative to σ dynamics: $|\mathcal{L}_i(\theta_{t+1}) - \mathcal{L}_i(\theta_t)| \leq \delta$ where $\delta/\eta_\sigma \to 0$ as $\eta_\sigma \to 0$.

(A3) Learning Rate Separation: $\eta_\sigma \ll \eta_\theta$, meaning σ parameters adapt faster than model parameters (timescale separation).

(A4) Schedule Regularity: $s(p) : [0, 1] \to (0, 1]$ is Lipschitz continuous with $|s'(p)| \leq S_{\max}$ and $s(p) \geq \epsilon > 0$.

G.2. Theorem Statement and Proof

Theorem G.1 (Stationary Points and Convergence)**.** *Under assumptions (A1)–(A4), the σ^2 dynamics induced by gradient descent on $\mathcal{L}_\sigma$ satisfy:*

1. ***Stationary Points:** The unique stationary point is:*

$$\sigma^{2*}_{local} = \mathcal{L}_{local}, \quad \sigma^{2*}_{align} = \frac{\mathcal{L}_{align}}{s(p)} \tag{35}$$

2. ***Local Stability:** The stationary point is locally asymptotically stable with convergence rate $O(\eta_\sigma)$.*

3. ***Tracking Error:** Under slow loss variation (A2), the tracking error satisfies:*

$$|\sigma^2(t) - \sigma^{2*}(t)| = O(\delta/\eta_\sigma + e^{-c\eta_\sigma t}) \tag{36}$$

for some constant $c > 0$ depending on $L_{\min}$.

Proof Sketch. The gradient of $\mathcal{L}_\sigma$ with respect to σ^2 yields: $\frac{\partial \mathcal{L}_\sigma}{\partial \sigma_i^2} = -\frac{\mathcal{L}_i}{2\sigma_i^4} + \frac{c_i}{2\sigma_i^2}$ where $c_{\text{local}} = 1$ and $c_{\text{align}} = s(p)$. Setting to zero gives $\sigma_i^{2*} = \mathcal{L}_i/c_i$. The Hessian at equilibrium is $\frac{\partial^2 \mathcal{L}_\sigma}{\partial(\sigma^2)^2} = \frac{c_i}{2\sigma_i^4} > 0$, confirming local convexity. Full proof in Appendix B.

Corollary G.2 (Equilibrium λ_{eff} Dynamics)**.** *The equilibrium alignment weight satisfies:*

$$\lambda^*_{eff} = s(p) \cdot \frac{\mathcal{L}_{local}}{\mathcal{L}_{align}} \tag{37}$$

with the following properties:

1. ***Monotonic Decay:** Since $s(p) \downarrow$ monotonically, λ^*_{eff} exhibits a decreasing trend (curriculum behavior).*

2. ***Data-Adaptivity:** The ratio $\mathcal{L}_{local}/\mathcal{L}_{align}$ introduces client-specific variation based on local data characteristics.*

3. ***Bounded Range (without stability reg):*** *Under (A1),* $\lambda^*_{\mathit{eff}} \in [s(p) \cdot L_{\min}/L_{\max}, s(p) \cdot L_{\max}/L_{\min}]$.
4. ***Explosion Risk:*** *When* $\mathcal{L}_{\mathit{align}} \ll \mathcal{L}_{\mathit{local}}$ *(extreme non-IID), the ratio can exceed practical bounds, motivating stability regularization.*

Why This is Fundamentally Different from a Fixed Schedule. Unlike a fixed schedule $\lambda(t) = \lambda_0 \cdot s(t)$, AURORA's λ_{eff} emerges from the joint dynamics of loss magnitudes and the monotonic prior. The σ parameters capture *meta-level task uncertainty* through $\mathcal{L}_\sigma$ (with detached losses); this uncertainty does not rescale ∇_θ, but induces a ratio $\lambda_{\text{eff}} = \sigma^2_{\text{local}}/\sigma^2_{\text{align}}$ that modulates alignment in $\mathcal{L}_W$. The key distinction: $s(p)$ *only imposes a monotonic prior; magnitude and inter-client variation emerge from optimization*.

H. Implementation Details for Reproducibility

This section provides additional implementation details for reproducibility.

H.1. Prototype and Alignment Details

- **Prototype representation:** Learnable prototypes $\mathbf{p}_c \in \mathbb{R}^d$ are *not* L2-normalized during alignment computation. The ETF anchors are normalized to unit norm.
- **Alignment loss:** We use L2 (MSE) distance rather than cosine similarity, as MSE provides stronger gradients when prototypes are far from anchors.
- **Class mask per batch:** During training, alignment loss is computed only over classes appearing in the current batch.
- **Missing class initialization:** Prototypes are initialized to their corresponding ETF anchor positions (with small random perturbation). For locally-missing classes, these prototypes remain near their ETF-aligned initialization since they receive no gradient updates. At aggregation, such prototypes are down-weighted during IFFI fusion based on local sample counts (effectively zero weight for missing classes).

H.2. Full Experimental Setup

Training Configuration:

- Backbone: ResNet-18
- Total local epochs: 500 (CIFAR-10), 100 (CIFAR-100, SVHN)
- Optimizer: SGD with momentum 0.9, weight decay 5e-4
- Learning rate: 0.05 (cosine annealing over local training)
- AURORA-specific: σ learning rate = 0.005, $\lambda_{\max}$ = 50.0, γ = 0.001
- Default: $K = 5$ clients; scalability study with $K \in \{5, 10, 20\}$
- Evaluation checkpoints: Every 10 epochs (offline, no communication)

One-shot Protocol. We strictly follow the one-shot FL protocol: each client trains locally for multiple epochs, then uploads its model/prototypes to the server *exactly once*. The server performs a single aggregation.

Clarification on "epoch checkpoints": We record intermediate states every 10 local epochs *for offline analysis only*—no parameters are communicated. These checkpoints enable studying training dynamics without violating the one-shot constraint.

Loss Scaling. All loss terms follow FAFI's original scaling (cls_loss + contrastive_loss + proto losses). We keep these fixed across all baselines to ensure σ adapts to training dynamics rather than arbitrary rescaling.

Quantifying Reduced Hyperparameter Burden. Manual λ annealing requires tuning: (1) initial λ value, (2) decay shape (linear/exponential/cosine), and (3) decay rate—typically requiring a grid search over 20+ configurations per dataset/α combination. In contrast, AURORA uses *the same three hyperparameters* (σ-lr=0.005, $\lambda_{\max}$=50, γ=0.001) across all experiments without per-setting adjustment.

H.3. Ablation Variant Definitions

- **AURORA (no stability):** Meta-annealing without stability regularization
- **AURORA (no decouple):** Standard Kendall formulation without gradient decoupling
- **Learnable-$\lambda(t)$:** $\lambda = \text{softplus}(a + b \cdot \phi(p))$ where $\phi(p) = \cos(\pi p)$, allowing nonlinear schedule learning
- **Cosine λ schedule:** Pure schedule $\lambda(t) = \lambda_0 \cdot s(p)$, no learning
- **GradNorm-style:** λ adjusted based on gradient magnitude ratio

H.4. Extended Experimental Setup

Datasets.

- **CIFAR-10:** 10-class natural image classification (50,000 training / 10,000 test)
- **CIFAR-100:** 100-class fine-grained classification (50,000 training / 10,000 test)
- **SVHN:** Street View House Numbers digit recognition (73,257 training / 26,032 test)

Baselines.

- **FedAvg (One-shot):** Simple averaging of locally trained models
- **FAFI:** Feature-Anchored Integration with contrastive learning (Zeng et al.)
- **FAFI+Annealing:** FAFI with manually-tuned linear λ annealing schedule

I. Analysis: AURORA Learns the Optimal Schedule (Extended)

This section provides extended analysis of how AURORA learns effective regularization schedules, complementing Section 4.4 of the main text.

Table 14. λ Evolution Comparison (CIFAR-100, α=0.05)

Checkpoint	$s(p)$	**AURORA λ_{eff}**	**Manual λ**
0 (start)	0.9	11.6	18.0
2	0.7	10.0	12.6
5	0.4	7.2	7.2
9 (end)	0.1	4.9	1.8

J. Robustness Study: The λ Explosion Problem (Extended)

This section provides extended analysis of the λ explosion problem and its mitigation, complementing Section 4.5 of the main text.

Table 15. SVHN Performance Under Extreme Heterogeneity (α=0.05)

Method	**Peak Acc**	**Final Acc**	**λ Behavior**
Meta-Anneal (no stab.)	49.5%	16.4%	Explodes $> 10^6$
+ Weak Reg (γ=1e-5)	50.0%	17.7%	Still explodes
AURORA (γ=1e-3)	**55.4%**	**52.9%**	Stable $\leq$50

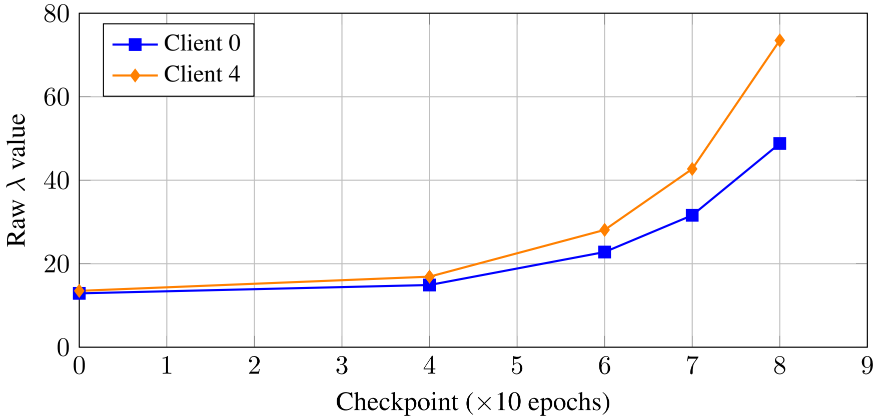

Figure 5. Per-Client λ Divergence. Despite sharing the same $s(p)$ prior, clients develop divergent λ trajectories based on their local data characteristics. By checkpoint 8, Client 4's λ is 51% higher than Client 0's—demonstrating AURORA is *data-dependent*, not merely *time-dependent*.

K. Hyperparameter Sensitivity Analysis (Extended)

This section provides extended hyperparameter sensitivity analysis, complementing Section 4.6 of the main text.

K.1. Qualitative Distinction: Safety Bounds vs. Performance-Critical Hyperparameters

Key Insight: The hyperparameters introduced by AURORA ($\lambda_{\max}$, γ, σ-lr) are *fundamentally different* from the manual annealing hyperparameters (initial λ, decay rate, decay shape) they replace. The former are *safety bounds*—they define when a fail-safe mechanism activates, not the core learning dynamics. The latter are *performance-critical*—small changes directly impact accuracy.

Table 16. Comparison of hyperparameter types

	Manual λ Schedule	**AURORA Stability Params**
Type	Performance-critical	**Safety bounds**
Sensitivity	Change shape → 2–5% acc drop	**5× range (20–100) → <1% variance**
Cross-setting	Requires re-tuning per dataset/α	Same defaults work across all
Trigger rate	Always active (shapes entire trajectory)	Rarely triggered in stable settings
Analogy	Curriculum design	Gradient clipping threshold

Why $\lambda_{\max}$ is Not λ in Disguise. The manual annealing schedule uses $\lambda(t) = \lambda_0 \cdot (1 - t/T)$, where λ_0 determines the *entire trajectory* and optimal values vary by 10× across datasets. In contrast, $\lambda_{\max}$ is a *ceiling*: it only activates when the learned λ_{eff} exceeds it, which rarely occurs under normal training conditions. Varying $\lambda_{\max}$ from 20 to 100 changes final accuracy by $<1\%$, demonstrating its role as a safety mechanism rather than a performance lever.

Table 17. Sensitivity Analysis on SVHN (α=0.05)

Parameter	**Values Tested**	**Accuracy (%)**	**Behavior**
$\lambda_{\max}$	20, **50**, 100	52.3, 52.9, 52.5	Stable within range
γ (reg strength)	0, 1e-5, **1e-3**	16.4, 17.7, 52.9	Collapse → Stable
σ-learning rate	1e-4, **5e-3**, 1e-2	51.2, 52.9, 51.8	Default: 5e-3

K.2. λ Sensitivity Analysis

To understand how different fixed λ values affect performance, we conducted experiments with $\lambda \in \{1.0, 2.5, 5.0, 10.0, 20.0, 50.0\}$ on CIFAR-10 (α=0.05):

Table 18. Effect of Fixed λ with Linear Annealing on CIFAR-10 (α=0.05)

λ_{initial}	**Accuracy (%)**	**g_protos_std**	**Observation**
1.0	58.89	0.987	Weak alignment
2.5	57.44	0.959	Destructive interference zone
5.0	58.77	0.914	Transition region
10.0	59.38	0.874	Strong alignment begins
20.0	**59.68**	0.597	Near-optimal manual tuning
50.0	59.39	0.503	Plateau—robust to over-tuning

Performance exhibits a U-shape: λ=2.5 represents a destructive interference zone where neither local nor global objectives dominate. This motivates the need for autonomous λ selection.

L. Scalability Study (Extended)

This section provides extended scalability analysis, complementing Section 4.7 of the main text.

Table 19. Performance with Varying Number of Clients on CIFAR-10 (α=0.1)

K **(Clients)**	**FAFI**	**FAFI+Ann.**	**AURORA**
10	54.16	55.70	**58.26**
20	46.39	48.22	**48.40**
30	40.23	39.70	**41.17**

Experimental Setup. To efficiently evaluate scalability across different federation sizes, we conduct a focused study with the following configuration:

- **Dataset:** CIFAR-10 with Dirichlet distribution (α=0.1)
- **Training:** 50 local epochs
- **Model:** ResNet-18 backbone
- **Optimization:** SGD with learning rate 0.05, momentum 0.9, weight decay 1e-4
- **AURORA-specific:** λ_{initial}=18.0, σ-lr=0.005, $\lambda_{\max}$=20.0

Note that this configuration differs from the main experiments (which use 300 rounds) to enable rapid evaluation across multiple client scales.

Purpose: Verify that AURORA's autonomous mechanism generalizes across different federation scales without re-tuning.

M. Scalability Analysis: Impact of Projector under Dimensionality Bottleneck

To address the dimensionality constraint $d \geq C - 1$ inherent to Simplex ETF, we evaluated the efficacy of adding a Projector layer when the feature dimension is insufficient. We conducted a stress test on CIFAR-100 ($C = 100$) by artificially constraining the ResNet-18 backbone output to $d = 32$, which violates the ETF condition ($32 < 99$).

As shown in Table 20, directly enforcing ETF alignment on the deficient dimension ($d = 32$) leads to poor performance. Introducing a lightweight MLP projector ($32 \rightarrow 128$) to map features to a sufficient logical dimension yields an absolute improvement of +1.85% (11.8% relative improvement), confirming that AURORA scales to many-class settings (e.g., ImageNet) via projection.

Table 20. Impact of Projector under Dimensionality Bottleneck (CIFAR-100, $d = 32$). Adding a projector significantly recovers performance by relieving the geometric bottleneck.

Configuration	**Feature Dim**	**Projector Dim**	**Test Accuracy (50 ep)**
Direct ETF (Collapse)	32	N/A	15.63%
AURORA + Projector	32	128	**17.48%**