Few-Task Imbalanced Multi-Task Learning via Alternating Optimization

Jiajun Xiao
Faculty of Science
University of Auckland
Auckland, New Zealand
jxia993@aucklanduni.ac.nz

Nanyuanyang Zhang
Faculty of Science
University of Auckland
Auckland, New Zealand
nzha229@aucklanduni.ac.nz

***Abstract*—**Recent works shows that alternating updates from task heads to a shared encoder achieve reliable feature sharing when tasks are many and balanced, offering geometric intuition for how subspaces emerge. In contrast, common practice often trains jointly with proportional sampling, and real deployments frequently involve few task counts and pronounced imbalance. We address this gap with an alternating schedule that couples two complementary levers. Temperature based task sampling improves minority task exposure and task wise uncertainty weighting stabilizes optimization across tasks. Our primary recipe, ALT-Temp+UW, combines both mechanisms, and a lightweight variant, ALT-Temp, removes uncertainty weighting for simpler deployment. We ground the approach with three hypotheses about coverage, stability, and early geometric signals, and we align method, evaluation, and diagnostics under a fairness first lens. The protocol emphasizes macro average and worst task metrics and uses subspace alignment and CKA as early indicators of transfer behavior. Positioned between the guarantees suggested by the anchor schedule and the demands of current practice, this design offers a concise path to more robust and fair multi task representations under few task imbalance and provides a clear blueprint for empirical validation.

Keywords—multi task learning, uncertainty weighting, temperature sampling, task imbalance, fairness metrics, subspace alignment

I. Introduction

A. Background and Motivation

Multi-task learning (MTL) has become a cornerstone paradigm in modern artificial intelligence, underpinning advances in computer vision, natural language processing, and reinforcement learning [1]. By training a shared encoder across tasks, MTL reduces annotation costs, improves transfer efficiency, and promotes generalization in downstream applications. This paradigm is particularly relevant in the era of large pretrained models, where transferability and data efficiency are decisive factors for both research and deployment.

Despite these advantages, fundamental questions remain about the conditions under which MTL produces robust shared representations. Linear learning theory provides formal guarantees that a common subspace can support task-specific predictors and reduce sample complexity [2]. At the same time, contrastive learning has connected representation geometry to separability in downstream tasks, while probing methods have clarified what encoders capture [3]. Recent studies also examine when frozen encoders with linear probes suffice and when fine-tuning becomes necessary [4]. These findings suggest promising mechanisms but also highlight the gap between theoretical predictions and empirical outcomes. We therefore evaluate shared encoders under a fairness-oriented lens using macro and worst-task metrics, and we monitor subspace alignment and CKA as early indicators of transfer trajectories.

A major source of this gap is that most guarantees are derived from linear models or single-task regimes, which do not fully explain the behavior of modern nonlinear encoders [5], [6]. In practice, encoders often succeed when pretrained but fail unpredictably in low-task or imbalanced-task regimes. Surveys show that large tasks tend to dominate optimization updates, overshadowing minority tasks, which results in biased or incomplete representations [1], [7], [8]. Such behavior undermines fairness, reliability, and trustworthiness in multi-task systems.

Understanding when and how MTL produces effective shared encoders is therefore of both theoretical and practical importance. Addressing this question requires connecting insights from learning theory with implementation strategies that account for limited task availability, strong imbalance, and the need for fairness-aware evaluation. Recent anchor results indicate that alternating updates from task heads to a shared encoder promote reliable feature sharing when tasks are many and roughly balanced, providing a geometric account of how subspaces emerge. In contrast, common practice often trains jointly with proportional sampling, which can amplify task-size imbalance and under-expose minority tasks.

B. Problem Statement and Research Questions

Although multi-task learning provides theoretical and practical benefits, two key challenges limit current understanding of how shared representations behave under realistic conditions. The first challenge is that most rigorous guarantees apply only to linear models or single-task regimes [2]. These results cannot fully account for the behavior of modern nonlinear encoders, which are widely used in practice [5], [6]. The second challenge is that many real-world scenarios involve only a handful of tasks or exhibit severe

task imbalance. In such cases, large datasets dominate optimization updates while small tasks contribute little, leading to biased or incomplete representations [1], [7], [8].

These limitations raise an important question: how can effective shared representations be learned when the number of tasks is limited and the distribution of data is imbalanced? This issue is not only theoretical but also practical, since projects often combine datasets of very different sizes or rely on a small in-house task suite [1], [7]. If the encoder is dominated by majority tasks, accuracy on minority tasks declines, undermining fairness, reliability, and trust in the system.

To address these concerns, this study is guided by three research questions:

- **RQ1:** Under limited tasks and imbalance, which sampling and weighting rules, when used with an alternating head→encoder schedule, help the encoder capture shared structure across tasks?
- **RQ2:** After pretraining, can a frozen encoder with a linear probe give strong transfer performance across all tasks, not only the majority ones?
- **RQ3:** Which training-time signals predict successful transfer? In particular, do subspace alignment and CKA track linear-probe accuracy or AUC on held-out tasks early in training?

These questions provide the foundation for the hypotheses and experimental plan developed in the following sections.

C. Scope, Contributions, and Structure

The scope of this study is deliberately focused on the quality of shared representations under conditions of limited tasks and task imbalance. Rather than pursuing state-of-the-art results on every benchmark, the emphasis is on understanding how training choices, update schedules, and evaluation protocols influence fairness and transferability across tasks. This focus ensures that the insights developed here remain broadly applicable, even when computational resources or task availability are restricted.

To address the research questions, we formulate three hypotheses:

- H1 (Coverage)**:** With few tasks, task coverage is the main bottleneck. Proper sampling and loss scaling can enlarge coverage and improve transfer without requiring additional data [1], [2].
- H2 (Stability)**:** Within an alternating schedule, task-wise uncertainty weighting stabilizes cross-task updates and regularizes shared geometry, improving minority-task transfer [5], [6].
- H3 (Early Signals)**:** A geometry-based diagnostic that measures cross-task alignment can predict probe outcomes and guide training schedules early in training [2], [5].

These hypotheses motivate a two-stage research plan. Stage one uses shallow encoders and controlled datasets to isolate the effects of task count, imbalance, sampling rules, and update schedules. Stage two examines deep pretrained backbones on a compact public multi-task suite, applying the same alternating schedule with temperature-based sampling and uncertainty weighting, and evaluating macro-averaged and worst-task metrics alongside subspace alignment and linear probes. Evaluation emphasizes fairness through macro-averaged and worst-task metrics, complemented by geometry-based diagnostics and probing protocols [4], [7], [8].

The contributions of this work can be summarized as follows. First, it provides a systematic analysis linking training choices to cross-task transfer performance under low task count and imbalance [1], [7]. Second, it introduces a practical recipe which alternating head→encoder updates with temperature-based task sampling and task-wise uncertainty weighting, which protects minority tasks without substantial overhead, and it quantifies the contribution of each component through ablations. Third, it proposes a geometry-based diagnostic that predicts probe outcomes and informs training schedules [5]. Finally, it outlines an evaluation protocol that emphasizes fairness through macro-averaged and worst-task metrics, ensuring reproducibility and comparability across future studies [7], [8].

The remainder of this paper is organized as follows. Section 2 reviews related work in multi-task representation learning. Section 3 presents the methodology, including the hypotheses, design axes, and evaluation strategy. Section 4 details the implementation setup, literature evidence, and integrated framework. Section 5 concludes the study, discusses its limitations, and outlines directions for future work.

II. Literature Review

A. Parameter sharing strategies

Parameter sharing lies at the heart of multi-task learning (MTL), and two main paradigms have emerged in literature: hard sharing and soft sharing. Hard sharing employs a single shared encoder with lightweight task-specific heads. This design significantly reduces the number of trainable parameters, lowers sample complexity, and naturally encourages a common latent representation across tasks. However, numerous surveys and empirical studies have documented its limitations in practice, most notably gradient conflicts between heterogeneous tasks and loss imbalance where large datasets dominate the optimization process [1], [7], [8].

In contrast, soft sharing retains per-task modules—such as adapters or low-rank layers—that are constrained to remain close in parameter space. This approach mitigates destructive interference by allowing a degree of independence across tasks, but it introduces extra parameters and tuning burdens [1], [8]. To further address imbalance, researchers have explored balancing mechanisms that operate at the loss or gradient level. Examples include uncertainty-based weighting, which scales losses according to predictive variance, multi-objective optimization frameworks, which treat each task as a separate objective, and gradient surgery techniques, such as PCGrad, which explicitly modify updates to reduce conflict [9], [10], [11].

Building on these insights, our methodology emphasizes combining sampling strategies with loss scaling and integrates an alternating update schedule. This hybrid approach aims to protect minority tasks while retaining the efficiency of a single shared encoder. Recent anchor results indicate that alternating head→encoder updates promote reliable feature sharing when tasks are many and roughly balanced, offering a geometric rationale for subspace formation. In practice, however, joint training with proportional sampling can amplify task-size imbalance and under-expose minority tasks, motivating hybrids that combine sampling, loss scaling, and an alternating schedule [1], [7], [8].

B. Representation learning paradigms

Representation learning for MTL can be broadly divided into supervised multi-task pretraining and self-supervised or contrastive pretraining. In the supervised paradigm, labeled data across tasks are used to shape the shared encoder, which is subsequently evaluated via a linear probe or light fine-tuning. This setup directly leverages task-specific supervision but suffers from label scarcity in some domains.

Self-supervised paradigms instead employ contrastive objectives with augmentations, learning invariances that generalize beyond task-specific supervision [3]. Such approaches often yield encoders with strong transferability, even to tasks with no labeled data. Recent studies have further examined the probe-then-fine-tune protocol, analyzing conditions under which a frozen encoder with linear probes suffices, versus when small amounts of task-specific fine-tuning unlock additional gains [4].

In this study, we adopt this widely used practice: encoders are pretrained (either supervised or contrastive), frozen, and first evaluated with lightweight probes. Fine-tuning is only introduced as a secondary check, ensuring that improvements are not merely due to heavy adaptation but reflect genuine representational quality.

C. Theoretical advances

Theoretical progress provides an essential foundation for understanding MTL. Classical linear multi-task theory shows that when tasks share a low-dimensional latent subspace, joint learning reduces the per-task sample complexity, effectively distributing statistical strength across tasks [2]. However, this benefit critically depends on **task diversity**: if tasks are too similar or unrelated, sharing may provide little gain or even harm performance (negative transfer).

More recently, nonlinear analyses have begun to shed light on modern neural encoders. Studies on shallow ReLU networks demonstrate that gradient descent can recover task-relevant features beyond what fixed kernel perspectives predict [5], [6]. These results suggest that even relatively simple nonlinear models can escape the limitations of linear theory. In parallel, contrastive learning theory has connected objective functions to representation geometry, demonstrating how properly designed augmentations lead to feature spaces that align with downstream separability [3].

Together, these strands motivate a focus on task coverage and geometry. These developments sharpen a central question: when tasks are few or imbalanced, can subspace alignment–style diagnostics serve as early predictors of downstream linear-probe performance, thereby guiding training schedules and balancing rules [2], [3], [5], [6]?

D. Applications and practice

Across domains, MTL has been applied with varied design choices and challenges. In natural language processing (NLP), large pretrained encoders are often extended with task-specific heads or lightweight adapters to balance efficiency with flexibility. In computer vision, dense prediction tasks such as segmentation, detection, and depth estimation frequently exhibit conflicts, making loss reweighting and adapter-style modules essential for stability [7]. In reinforcement learning (RL), shared encoders have been shown to reduce sample requirements in both offline and online regimes, especially under linear assumptions [12], [13].

A consistent finding across these domains is that imbalance is pervasive: proportional sampling tends to bias updates toward large tasks, causing minority tasks to collapse [1], [7], [8]. Practical systems therefore rely on combinations of balanced or temperature-scaled sampling, uncertainty weighting, or gradient surgery methods to ensure that smaller tasks remain viable contributors to the shared encoder [9], [11].

E. Summary and placement.

In summary, prior work provides:

- A taxonomy of parameter sharing and its practical challenges [1], [7], [8].

- Baseline theory for task coverage and subspace recovery [2].
- A case for probe-based evaluation and contrastive links to representation geometry [3], [4].
- Practical mechanisms for mitigating imbalance, including loss reweighting and gradient surgery [9]–[11].

Our work extends this foundation by focusing explicitly on few-task regimes with pronounced imbalance. We integrate temperature-scaled task sampling and uncertainty-based loss weighting within an alternating head–encoder schedule, and we test whether subspace-alignment proxies reliably predict probe outcomes under these stress conditions.

III. BENCHMARK

A. Dataset

1) Source and construction

We build a compact three-task benchmark by deriving class pairs from a common image corpus. Concretely, each task is a binary classification problem formed by a semantically coherent pair of categories:

- T1: Cat vs. Dog
- T2: Bird vs. Frog
- T3: Airplane vs. Ship

All images are drawn from the same underlying distribution to ensure comparable visual statistics across tasks while preserving meaningful inter-task diversity (animals vs. vehicles) [14]. This design yields a few-task multi-task learning (MTL) setting with moderate cross-task transfer and minimal label leakage.

2) Imbalance protocol (task-level IR).

To study robustness under few-task imbalances [7], [8], we control the per-task training set size n_t and define a task-level instance ratio (IR) vector $(n_1 : n_2 : n_3)$. Unless otherwise specified, we use a default imbalanced regime $\mathrm{IR} = (8:1:1)$ implemented by random subsampling without replacement on the training split while keeping validation and test splits balanced for fair evaluation.

3) Data splits

For each task, images are partitioned into train/validation/test sets with non-overlapping identities:

- Train: imbalanced per task according to the IR protocol above; stratified by class.
- Validation: fixed size per task with equal positives/negatives (class-balanced) to decouple selection bias from performance estimation.
- Test: full, class-balanced split per task, shared across all methods and seeds.

All experiments use identical split indices across methods to guarantee comparability.

B. Experimental Context

To evaluate the robustness and fairness of our method under *few-task and imbalanced* conditions, we benchmark four representative training strategies that together cover both theoretical and practical perspectives [1], [7], [8].

1) JOINT + Proportional (Strong Baseline).

A In this baseline setting, all task heads and the shared encoder are trained jointly within a single optimization loop. Mini-batches are sampled from each task in proportion to its dataset size, so tasks with more data are updated more frequently. This mirrors common practice in industrial multi-task systems, where training frequency naturally follows data availability to ensure stable convergence and efficient use of computation.

The advantage of this configuration lies in its simplicity, scalability, and stability; the encoder quickly learns dominant patterns that generalize across tasks. However, because larger tasks contribute more gradients, the model tends to bias toward majority tasks, resulting in weaker generalization for minority ones. Therefore, JOINT + Proportional serves as a strong yet imbalance-sensitive baseline, against which more fairness-aware or adaptive strategies can be evaluated.

2) ALT + Balanced (Anchor Reference).

The ALT + Balanced strategy follows the alternating optimization framework introduced in the anchor paper [15], where the model alternates between updating task-specific heads and the shared encoder. Unlike the proportional joint training, this variant enforces balanced task sampling, ensuring that each task receives equal training exposure regardless of its dataset size.

This balanced scheduling emphasizes fairness over efficiency. It prevents overfitting to data-rich tasks and encourages the encoder to learn equally representative features across all tasks. Conceptually, ALT + Balanced represents an idealized upper bound for alternating methods—showing how well the alternating mechanism could perform when data imbalance is entirely removed. In our experiments, it serves as the reference configuration to which all adaptive sampling variants are compared.

3) ALT-Temp (Our Lightweight Variant).

Building upon the alternating optimization framework, ALT-Temp introduces a temperature-based task sampling mechanism that smoothly transitions between data-proportional and task-balanced regimes [8]. The sampling probability for each task is defined as

$$p_t \propto n_t^{1/T_t}$$

where n_t is the number of samples for task t and τ_t denotes a task-specific, learnable temperature parameter controlling the degree of balance. A larger τ_t makes the sampling distribution closer to data-proportional sampling (favoring majority tasks), while a smaller τ_t flattens the distribution and increases the exposure of minority tasks.

This mechanism provides a simple yet powerful way to mitigate task-level imbalance without explicitly reweighting losses or altering optimization dynamics. By allowing each task's τ_t to adapt during training, the model can flexibly trade off between efficiency and fairness, improving macro-level performance stability and tail-task robustness while maintaining computational simplicity.

4) ALT-Temp + UW (Our Full Model).

To further enhance robustness under severe imbalance, ALT-Temp + UW augments temperature-based sampling with uncertainty weighting (UW) [9]. The total loss is formulated as:

$$\mathcal{L}_{total} = \sum_t e^{-st}\mathcal{L}_t + 0.5 s_t$$

where $s_t = \log \sigma_t^2$ is a learnable log-variance parameter that reflects each task's predictive uncertainty.

This formulation ensures that tasks with higher predictive uncertainty contribute less to the overall gradient, while confident tasks are weighed more strongly. As a result, the model achieves balanced gradient magnitudes and improved optimization stability even under severe task-level imbalance. Combined with temperature-based sampling, ALT-Temp + UW forms our full model, offering a robust and fair solution for few-task, imbalanced multi-task learning.

C. Implementation Details

Following the alternating gradient descent configuration proposed in the anchor paper [15], we adopt the same two-stage training strategy that alternates between optimizing task-specific heads and the shared encoder representation. To ensure fairness, all methods are trained under an equivalent optimization budget measured by the total number of backward passes. Effective batch exposure per task is matched by gradient accumulation to maintain comparable optimization steps across strategies.

All hyperparameters are kept consistent with the anchor setup unless otherwise noted: learning rate $\eta = 0.05$, weight decay $\lambda_w = 0.05$, head regularization $\lambda_a = 0.5$, and momentum = 0.9. Each task uses equal mini-batch sizes ($n_1 = n_2 = 64$), and alternating updates occur every 50 steps.

To ensure robustness, all experiments are conducted under identical data splits and repeated with five different random seeds (42–46). All reported results are averaged over these seeds with 95% confidence intervals; pairwise t-tests and Cliff's δ are computed to evaluate statistical significance.

For consistency with few-task and imbalanced conditions, we follow the anchor paper in using subspace-based evaluation metrics—namely subspace alignment error and condition number—to assess representation consistency [3], [5], [6], [13]. All experiments are automated through unified script controlling methods, imbalance levels (IR = 8:1:1), and random seeds to ensure full reproducibility.

.

D. Evaluation Metrics

a) Primary Metrics

- Macro Accuracy – average accuracy across all tasks, reflecting task-level fairness.
- Worst Accuracy – accuracy of the lowest-performing task, measuring tail-task robustness.

b) Secondary Metrics

- Subspace Alignment – cosine similarity between task representation subspaces, indicating geometric consistency.
- Linear CKA (Centered Kernel Alignment) – measures the representational similarity between task-specific embeddings in a linearized space, providing a complementary view of cross-task feature alignment. Unless otherwise noted, all features are extracted from the penultimate encoder layer, and the subspace dimension is fixed at k = 15 for consistency across methods.

IV. Methodology[1]

This section details the proposed methodology, which operationalizes the theoretical alternating training (ALT) framework under *few-task and imbalanced* multi-task learning conditions [15]. While the original ALT formulation assumes many balanced tasks, our setup considers scenarios where the number of tasks is small, and the task instance ratio (IR) is highly imbalanced. In such settings, conventional joint training tends to overfit dominant (head) tasks and underperform on minority (tail) tasks.

To address this challenge, we extend the ALT framework with two complementary mechanisms:

1) Temperature-based task sampling (ALT-Temp) to rebalance task exposure, and
2) Uncertainty-based loss weighting (ALT-Temp + UW) to stabilize gradients and improve fairness across tasks.

Together, these techniques aim to enhance *macro-average* and *worst-task* performance without sacrificing efficiency or balanced-task performance.

A. Alternating Training Framework (ALT)

Our approach builds upon the alternating training (ALT) paradigm introduced in the benchmark section.

Instead of updating all parameters jointly, ALT alternates between two phases in each iteration:

1) Head update: Task-specific heads are updated while the shared encoder is frozen.
2) Encoder update: The shared encoder is optimized while task heads are fixed.

This alternation produces a *pseudo-contrastive* effect that samples with the same semantic meaning across different tasks become geometrically aligned in the shared representation space [3], [5], [6].

In few-task regimes, such alternating updates help prevent feature collapse by maintaining diverse gradients from different tasks.

Formally, for task t with loss $\mathcal{L}_t$, the ALT process can be summarized as:

Head step:

$$\theta_t \leftarrow \theta_t - \eta \nabla_{\theta_t} \mathcal{L}_t(\theta_t; f_\phi)$$

Encoder step:

$$\phi \leftarrow \phi - \eta \sum_t \nabla_\phi \mathcal{L}_t(\theta_t; f_\phi)$$

where ϕ denotes encoder parameters, and θ_t are task-specific head parameters. To ensure fairness in comparison across methods, all variants are trained under an equivalent optimization budget measured by the total number of backward passes per epoch.

B. Temperature-Based Task Sampling (ALT-Temp)

Under few-task and imbalanced conditions, a key challenge is unequal task exposure: larger datasets dominate encoder updates, while smaller tasks contribute less frequently. To address this imbalance, we employ temperature-based task

[1] https://github.com/JamesYuanyang/764project

sampling [8], where the probability of selecting task t is proportional to its dataset size n_t raised to a learnable temperature exponent:

$$p_t \propto n_t^{1/\tau_t},$$

where $\tau > 0$regulates the balance between proportional and uniform sampling.

- When $\tau = 1$, the method reduces to proportional sampling (baseline).
- When $\tau \to 0$, it approaches uniform sampling, giving equal exposure to all tasks.

where $\tau_t > 0$ is a task-specific, learnable parameter that dynamically adjusts the balance between proportional and uniform sampling during training. Initially, all τ_t are set to 1 and are optimized jointly with other model parameters, constrained within [0.1, 5.0]. This adaptive mechanism continuously balances data efficiency and task fairness without manual tuning. It directly operationalizes Hypothesis H1 (Coverage) that improving the exposure of minority tasks enhances both Macro and Worst accuracy, especially under high imbalance ratios (IR > 1) [7].

C. Uncertainty-Based Loss Weighting (ALT-Temp + UW)

While temperature-based sampling improves the exposure frequency of minority tasks, it does not address the imbalance in *gradient magnitude* among tasks. In few-task settings, this imbalance becomes particularly severe: large tasks produce stronger and more stable gradients, while small tasks yield sparse and noisy signals. As a result, the shared encoder tends to over-fit the dominant tasks even when sampling is balanced.

To mitigate this issue, we introduce Uncertainty-Based Loss Weighting (UW) on top of ALT-Temp [9]. The key idea is to let the model learn how much to trust each task's loss by assigning every task t a learnable uncertainty parameter s_t.This parameter represents the intrinsic noise or volatility of that task's objective. During training, the total objective is formulated as:

$$\mathcal{L}_{total} = \sum_t e^{-st}\mathcal{L}_t + 0.5 s_t$$

where $\mathcal{L}_t$denotes the task-specific loss, and $s_t = \log(\sigma_t^2)$is optimized jointly with other model parameters.

The exponential factor e^{-s_t} automatically adjusts the contribution of each task:

- tasks with higher uncertainty (larger s_t) receive smaller weights [9], [10], [11].
- tasks with lower uncertainty are given stronger influence on the shared encoder.

This adaptive re-weighting effectively reduces gradient domination from head tasks and suppresses instability caused by noisy or low-resource ones. Unlike fixed heuristic weights, UW continuously updates these coefficients throughout training, allowing the model to self-balance according to the current learning dynamics. This mechanism operationalizes Hypothesis H2 (Stability), which aims to stabilize the gradient magnitudes across tasks and improve convergence robustness.

In implementation, each $s_t$is learned jointly with the model parameters and constrained within a stable range to prevent numerical overflow. In our alternating setup, uncertainty weighting is applied conditionally to the sampled task in each iteration rather than aggregated across all tasks, following a stochastic approximation scheme. Although the current implementation does not explicitly apply L_2 regularization, such a term can be added to further stabilize optimization if needed. This design maintains robustness while allowing the model to remain sensitive to task difficulty.

Within the ALT-Temp + UW scheme, temperature sampling and uncertainty weighting play complementary roles:

- Temperature sampling controls how often each task is selected, shaping data-level exposure.
- Uncertainty weighting controls how strongly each selected task affects parameter updates, shaping gradient-level balance.

Together, these two mechanisms form a theoretically consistent, two-stage balancing framework that is particularly suited for few-task and highly imbalanced learning scenarios. The combined formulation is expected to improve convergence stability and fairness compared with either ALT-Temp or JOINT training, although this remains to be empirically verified in our subsequent analysis.

D. Model Architecture and Optimization

All experiments were conducted using lightweight multi-task architecture designed for clarity and reproducibility rather than scale. The shared encoder is a two-layer multilayer perceptron (MLP) with ReLU activation and a hidden width of 512, mapping each input image $x \in \mathbb{R}^{32\times32\times3}$into a 512-dimensional latent representation. Each task head is an independent linear classifier that maps the shared representation into a two-class output space. This shallow configuration follows the theoretical anchor model while remaining computationally efficient for systematic comparison [15].

For optimization, all models use stochastic gradient descent (SGD) with a learning rate of 0.05, momentum of 0.9, and weight decay of 5×10^{-4}. The batch size is 64, and each model is trained for 20 epochs without task-specific tuning. All methods share identical hyperparameters to ensure fairness. Temperature parameters τ_t and uncertainty parameters s_t are initialized uniformly (log-space zero initialization) and optimized jointly with the encoder weights. No explicit clipping or L_2 regularization is applied to $s_t$in the current implementation; however, both can be incorporated if stronger numerical stabilization is required. This setup provides a consistent and reproducible optimization environment across all baselines and proposed variants.

E. Training Procedure under Few-Task and Imbalanced Conditions

The complete training pipeline follows an alternating update schedule inspired by the ALT framework. In each iteration, a task t is sampled according to a temperature-adjusted probability

$$p_t = \frac{n_t^{1/\tau_t}}{\sum_j n_j^{1/r_j}}$$

where $n_t$denotes the number of training samples of task $t$and τcontrols the degree of re-balancing between head and tail tasks. Lower τ_t values flatten the sampling distribution,

increasing the exposure frequency of under-represented tasks and improving overall coverage.

Training alternates between two coordinated phases:

1) Head-update phase – The shared encoder is frozen, and each task-specific head is updated independently. This allows every task to refine its decision boundary and define its gradient direction without interference.
2) Encoder-update phase – All heads are frozen, and the shared encoder is optimized based on the aggregated gradients from the task heads. This step aligns representation of subspace across tasks, reinforcing the pseudo-contrastive effect discussed before.

When the uncertainty-weighting (UW) module is enabled, each task's loss is scaled adaptively using the formulation described in Section 5.4, ensuring that tasks with higher inherent noise or smaller sample size contribute proportionally without dominating the optimization. The UW parameters $s_t$are learned jointly with the encoder and clipped within a stable range to prevent numerical drift.

This two-phase schedule—temperature-based sampling for balanced exposure and uncertainty-weighted loss for adaptive scaling—forms the core of the proposed ALT-Temp (+ UW) training strategy. It ensures that minority tasks maintain sufficient representation during learning while the shared encoder remains geometrically consistent and stable across updates.

F. Evaluation Metrics and Hypothesis Validation

Model performance is assessed using both performance-level and geometry-level indicators, each aligned with the hypotheses introduced earlier.

- Macro Accuracy (Macro) – the mean of task-specific accuracies, measuring overall fairness across tasks.
- Worst-Task Accuracy (Worst) – the minimum accuracy among all tasks, reflecting robustness to imbalance.
- Subspace Alignment – the cosine alignment between singular subspaces of task embeddings, used as an early geometric signal of representation consistency [3], [5], [6], [13]. Linear CKA – measures representational similarity between task-specific embeddings in a linearized space. Unless otherwise noted, features are extracted from the penultimate encoder layer, and the subspace dimension is fixed at $k = 15$.

These metrics correspond directly to the theoretical assumptions:

- H1 (Coverage): Macro/Worst improvements indicate better coverage of minority tasks via temperature sampling.
- H2 (Stability): Reduced variance of alignment curves across epochs supports greater training stability.
- H3 (Early Signals): Correlation between alignment trends and final linear-probe accuracies confirms the predictive value of geometric diagnostics.

All experiments are repeated with five random seeds. We report 95 % confidence intervals and apply paired-sample t-tests to verify statistical significance ($p < 0.05$).

G. Summary

This methodology integrates theoretical insights from alternating optimization into a practical framework tailored to few-task, imbalanced multi-task learning. By jointly applying temperature-based task sampling and uncertainty-aware loss weighting, the proposed ALT-Temp (+UW) scheme aims to address two core challenges: insufficient exposure of minority tasks and unstable gradient magnitudes. The unified training schedule, shared optimization configuration, and consistent evaluation protocol are designed to isolate the effects of these mechanisms from other confounding factors.

The framework is theoretically expected to improve macro-level fairness and gradient stability under varying imbalance ratios; empirical verification of these hypotheses will be presented in subsequent sections.

V. EXPERIMENTS

A. Objectives & Success Criteria

We evaluate in a few-task, highly imbalanced regime with a fixed imbalance ratio of IR = 8, representing a realistic long-tail setting where head tasks dominate training data.

Our goal is to examine whether temperature-based sampling and uncertainty-aware weighting improve optimization stability and fairness under this single but challenging imbalance level.

1) Success criteria.

1. Macro/Worst performance:
   The proposed ALT-Temp + UW should match or exceed ALT-Temp and outperform both JOINT + Prop and ALT + Balanced on *macro* and *worst-task* accuracy.
2. Early-signal consistency:
   ALT-Temp + UW should exhibit earlier and more stable increases in geometric alignment and CKA, consistent in direction with final performance trends.
3. Statistical validation:
   For each method we use five random seeds and report *mean ± 95 % CI*. We perform paired tests between ALT-Temp + UW and each baseline, and report effect sizes (Cliff's δ or Hedges' g).

2) Protocol note.

All hyperparameters and implementation details are defined in preceding sections; this subsection specifies only evaluation and statistical criteria.

B. Data & Imbalance Protocol

Stage-1 (controlled mini-suite). We fix a compact mini-suite of 3–6 tasks as the evaluation basis. Task heterogeneity and difficulty are kept constant to ensure that any performance differences arise purely from optimization behavior rather than task selection or dataset variability. This stage serves as a controlled environment for all subsequent analyses.

Stage-2 (small real slices). We additionally verify whether the same training recipe transfers to one or two small, real multi-task subsets drawn from public benchmarks. Each slice

follows the identical configuration and imbalance construction described below. Results are summarized in a single compact table in the main text; any extended results are deferred to the supplementary material.

Constructing IR. We fix the imbalance ratio at IR = 8. Only the training split is modified, while validation and test splits remain unchanged. To achieve the desired imbalance, we apply stratified down-sampling of tail-task samples (and proportional re-sampling for head tasks when necessary) so that each task's internal label distribution is preserved. This procedure maintains within-task class balance while creating the intended cross-task long-tail structure.

Randomness & reproducibility. The task-subset list, down-sampling random seeds, and fold splits are all fixed before training. A full record of seed values, task indices, and sampling statistics is included in the released configuration artifact, ensuring that every reported run can be exactly reproduced.

Consistency check. After constructing the imbalanced training data, we compute the total variation (TV) and/or Kullback–Leibler (KL) divergence between the target 8 : 1 ratio and the realized task-level instance distribution. These values are reported as a single verification line in the supplementary configuration file, confirming that the intended imbalance was precisely achieved.

C. Methods & Budget Parity

1) Method order.

ALT-Temp + UW (primary) → ALT-Temp (lightweight) → JOINT + Prop (strong baseline) → ALT + Balanced (anchor). All methods use the same shared encoder and task-specific heads, identical data preprocessing and augmentations, the same optimizer type and learning-rate schedule, and identical early-stopping patience. This ensures that any observed difference in performance stems purely from the training strategy rather than architectural or regularization effects.

2) Budget accounting unit.

We equalize total backpropagation work. For ALT, one step is defined as a Head→Encoder pair of updates. All methods are matched in total backward passes (or equivalently, samples × forward-backward FLOPs).

3) Effective batch matching.

If memory footprints differ, we use gradient accumulation to match the effective batch size across methods.

4) Early stopping rule.

We only early-stop on val-Macro, using the same policy for all methods. Early signals (alignment/CKA) are not used for selection or tuning.

D. Hyperparameter Search & Model Selection

1) Shared search space & equal budget.

For *learning rate*, *weight decay*, *warm-up ratio*, *gradient clipping*, *data augmentations*, and all other shared hyperparameters, we use an identical candidate space across methods.

Each method is allocated the same number of hyperparameter trials under the fixed imbalance setting (IR = 8) to ensure comparable search effort and fair tuning opportunities.

2) Method-specific parity.

The temperature strength parameter (τ or γ) is searched with the same candidate grid and trial count for both ALT-Temp and ALT-Temp + UW. For methods including uncertainty weighting (UW), initialization and regularization parameters are searched exclusively within those methods, but with a trial budget equal to that of the temperature search, ensuring parity in total tuning resources.

3) Two-stage search.

We run a coarse random sweep followed by a Bayesian refinement. The number of trials in each stage is identical across methods and IRs. Trial-level random seeds are fixed and disclosed in the configuration artifact.

4) Model selection.

Model selection is based solely on validation-Macro performance. If multiple runs achieve similar Macro scores, validation-Worst is used as a tie-breaker. Test metrics and early-signal indicators (alignment/CKA) are recorded for analysis but never used for tuning or selection.

5) Overfitting control.

We use a single fixed validation split, pre-register the search space and trial quotas, and evaluate the final model once on test.

E. Metrics, Probing & Early Signals

1) Primary metrics:

- Classification: Macro-F1 or Macro-AUROC, averaged uniformly across tasks (not sample-weighted).
- Regression (if present): z-score standardize each task's outputs, then macro-average, avoiding unit-scale confounds.
- Worst-task: the minimum per-task score (optionally add the lower quartile in the configuration bundle).

2) Probing protocol.

We first freeze the encoder and train a linear probe. A small, uniform fine-tuning budget may be used only as a secondary check and never for model selection.

Early-signal logging:

- Extract features from the penultimate encoder layer.
- Subspace alignment: for each task, compute centered top-k (e.g., k=64) SVD subspaces and measure alignment.
- CKA: use linear CKA for cross-task representation similarity.
- Frequency: log once per epoch.
- Correlation: compute Pearson/Spearman correlations between early signals and final Macro/Worst; report in the configuration artifact.

VI. Results

A. Macro Accuracy Performance

As shown in Table I, ALT, which is our anchor baseline hat achieves the highest macro accuracy (72.687%), outperforming JOINT (72.37%) by approximately 0.32 percentage points. The temperature-augmented variants are numerically very close to ALT: ALT-Temp (72.603%) and ALT-Temp + UW (72.663%), with absolute gaps below 0.08 percentage points.

In terms of variability, JOINT exhibits the **lowest cross-run** standard deviation (0.19), while ALT-Temp shows a favorable stability–accuracy trade-off (Std 0.215, slightly lower mean than ALT). Although ALT-Temp + UW maintains similar accuracy to ALT, its variance increases (0.364), indicating that the uncertainty weighting adds minor optimization noise without yielding clear accuracy gains.

A similar trend appears in the worst-task performance summarized in Table II. Here, JOINT (64.882%) attains the highest single-task macro accuracy, while ALT (64.167%), ALT-Temp (64.183%), and ALT-Temp + UW (64.143%) remain within a narrow band of ±0.04 percentage points.

Despite negligible differences in mean accuracy, ALT-Temp again records the lowest standard deviation (0.215), suggesting that temperature-based task sampling improves convergence consistency across runs even when it does not enhance absolute performance.

Taken together, these findings indicate that temperature sampling and uncertainty weighting primarily contribute to stability rather than mean accuracy. When the number of tasks is small and the imbalance ratio is high, macro-level accuracy quickly saturates, and improvements are better reflected in reduced variance and smoother optimization dynamics rather than higher scores.

Method	Best Macro	Macro Std
JOINT	72.37%	0.19
ALT	72.687%	0.347
ALT-Temp	72.603%	0.215
ALT-Temp + UW	72.663%	0.364

Table I — *Best Macro Accuracy Results*

Method	worst Macro	Macro Std
JOINT	64.882%	0.19
ALT	64.167%	0.347
ALT-Temp	64.183	0.215
ALT-Temp + UW	64.143%	0.364

Table Ⅱ — *Worst Macro Accuracy Results*

B. Representation Consistency and Significance

1) Alignment

All four methods yield nearly identical alignment accuracy, with JOINT (94.422%), ALT (94.433%), ALT-Temp (94.441%), and ALT-Temp + UW (94.467%) differing by less than 0.05 percentage points. This negligible gap confirms that the proposed temperature and uncertainty mechanisms do not materially alter the overall feature alignment performance, at least when measured by mean alignment scores. However, the variance across runs paints a more nuanced picture: ALT achieves the lowest alignment deviation (Align Std = 0.075), indicating stable and consistent convergence behavior, while both ALT-Temp (0.142) and ALT-Temp + UW (0.164) exhibit noticeably higher variability.

This increase in variance suggests that while the additional sampling and weighting introduce more flexibility in task selection and gradient contribution, they also inject extra stochasticity into the optimization process. In practice, temperature-based sampling adjusts the probability of task updates dynamically, which can amplify fluctuations in task exposure from one iteration to another—particularly under few-task conditions where sample diversity is already limited. Similarly, uncertainty weighting scales individual task losses adaptively based on estimated prediction variance, but such adjustments can cause momentary instability when uncertainty estimates fluctuate early in training. Consequently, the small numerical improvement in mean alignment seen in ALT-Temp + UW does not necessarily reflect better representation geometry; rather, it may stem from noisy gradient reweighting effects that trade off stability for flexibility.

Overall, the alignment results reinforce that ALT provides the most consistent and reliable task alignment, while the two extended variants introduce slight volatility without yielding meaningful structural gains.

Method	Best Align	Align Std
JOINT	94.422%	0.09
ALT	94.433%	0.075
ALT-Temp	94.441%	0.142
ALT-Temp + UW	94.467%	0.164

Table III — *Best Alignment Error Results*

2) CKA similarity

The CKA-based analysis provides further insight into how task representations overlap in the shared encoder.

Across all methods, the overall level of feature similarity remains high—around 82%—indicating that the encoder effectively maintains a common representational basis across tasks.

However, subtle quantitative differences emerge: JOINT achieves a mean CKA of 82.814%, ALT slightly decreases to 81.861%, while ALT-Temp (82.748%) and ALT-Temp + UW (82.347%) show a small recovery toward the joint level. This marginal increase suggests that temperature-based task alternation can occasionally encourage a *tighter coupling* among task embeddings, possibly by exposing minority tasks more frequently and improving feature alignment in under-represented regions of the shared space.

Nevertheless, this gain comes at the cost of higher instability. The CKA standard deviation rises from 0.543 (ALT) to 0.749 (ALT-Temp) and 0.725 (ALT-Temp + UW), reflecting that the representational geometry becomes more sensitive to initialization and sampling stochasticity. Temperature sampling adjusts task selection probabilities at each iteration, and under limited task diversity, this stochastic exposure amplifies gradient fluctuations that perturb the encoder's subspace alignment. Similarly, uncertainty weighting dynamically rescales task losses based on variance estimates, but these estimates are noisy during early optimization, leading to temporary over- or under-emphasis on tasks.

As a result, while the temperature-augmented models appear slightly closer to JOINT in mean CKA, their higher variance suggests less reproducible representational coherence across runs.

Taken together, the CKA results reinforce the alignment findings: ALT achieves the most stable inter-task representation structure, whereas the proposed extensions

introduce mild geometric variability without producing statistically meaningful gains.

Method	Best CKA	CKA Std
JOINT	82.814%	0.574
ALT	81.861%	0.543
ALT-Temp	82.748%	0.749
ALT-Temp + UW	82.347%	0.725

Table IV — *Best CKA Similarity Results*

3) Statistical significance

Pairwise tests against the anchor (ALT) show no statistical significance for the observed gaps (all $p > 0.05$). Effect-size estimates (δ^2, γ^2) fluctuate around zero, corroborating that the small differences in Align/CKA (and macro accuracy in §6.1) are best interpreted as *non-robust fluctuations* rather than reliable gains.

Method	p-value	δ	g	Sig.
JOINT	0.068	0.44	0.816	No
ALT	0.866	-0.04	-0.053	No
ALT-Temp	0.765	0.12	0.162	No
ALT-Temp + UW	-	-	-	-

Table Ⅵ — *Statistical Significance Analysis*

4) Summary

Taken together, the analyses across alignment, CKA similarity, and statistical significance consistently demonstrate that temperature sampling and uncertainty weighting do not fundamentally reshape the shared encoder's representational geometry in our few-task, imbalanced setting.

Their primary impact lies in slightly altering optimization dynamics, introducing modest stochasticity that may smooth convergence but fail to produce reproducible or statistically significant improvements. Overall, ALT remains the most stable and reliable configuration across both feature-level and statistical perspectives, while the extended variants contribute only marginal and non-significant adjustments to training behavior.

These findings collectively highlight that the potential of alternating optimization is already largely realized by the ALT baseline—offering a useful reference point for future efforts aimed at enhancing stability or fairness without sacrificing efficiency.

VII. Conclusion and Future Work

A. Conclusion

This work examined the challenges of multi-task representation learning (MTL) under two realistic constraints: a limited number of tasks and severe task imbalance. Building on both theoretical insights and controlled experiments with a fixed imbalance ratio **(**IR = 8**)**, we proposed an implementation framework that integrates architectural design choices, task sampling and weighting strategies, and fairness-oriented evaluation protocols.

Section 4 described the technical setup and mechanisms such as proportional versus balanced sampling, adaptive loss weighting, and alternating update schedules. Section 5 consolidated empirical evidence that the proposed ALT-Temp (+ UW) variants offer marginal yet more stable performance relative to the ALT anchor under few-task, imbalanced conditions. Section 6 expanded the evaluation to representation-level diagnostics, showing that temperature sampling and uncertainty weighting mainly influence optimization stability rather than mean accuracy or alignment. Together, these findings suggest that the potential of alternating optimization is largely realized by the ALT baseline, while additional sampling or weighting improves consistency but not absolute performance.

The main contribution of this study lies in presenting a structured blueprint for analyzing representation quality under constrained MTL regimes. By unifying theoretical guarantees, practical training strategies, and fairness-oriented evaluation standards, this framework provides a conceptual foundation and experimental template for future empirical exploration.

B. Limitations

Several limitations of this work should be acknowledged. First, the study did not implement large-scale experiments, so the proposed framework remains conceptual and illustrative rather than empirically validated at scale. Although the methodology integrates strategies such as uncertainty weighting, gradient surgery, and subspace-alignment diagnostics, their effectiveness under different architectures, task distributions, and data regimes has not been systematically tested. Second, while the analysis connects theoretical results with practical considerations, it is largely informed by findings in vision, language, and reinforcement learning domains, leaving open the question of generalization to areas such as healthcare or robotics. Finally, the evaluation relies heavily on fairness-oriented metrics and geometry-based diagnostics, which may not capture all dimensions of practical utility (e.g., interpretability or efficiency). A broader framework incorporating these aspects would be needed for deployment in real-world systems.

C. Future Work

Future research can extend this study in several directions:

- Empirical validation: Implement the proposed framework on standard MTL benchmarks (e.g., GLUE for NLP, multi-task dense prediction for vision) to quantify the impact of task number, diversity, and imbalance mitigation strategies.
- Adaptive scheduling: Investigate dynamic mechanisms that adjust sampling, weighting, and update frequency during training, guided by online geometry metrics.
- Cross-domain generalization: Explore whether the principles identified here transfer to domains

beyond vision and language, such as control systems, healthcare, and multi-modal settings.

- Scalable diagnostics: Develop efficient proxies for subspace alignment and fairness metrics that can be deployed in large-scale pretraining, reducing computational overhead.
- Integration with foundation models: Examine how MTL strategies interact with large pretrained models, where task imbalance and transferability issues may manifest differently from shallow or mid-sized setups.

In summary, this work consolidates theoretical and empirical knowledge into a conceptual framework for studying MTL under practical constraints. While primarily a blueprint, it establishes the groundwork for future empirical investigations aimed at building robust, fair, and generalizable shared representations.

REFERENCES

[1] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," IEEE Trans. Knowl. Data Eng., vol. 34, no. 12, pp. 5586–5609, 2022.

[2] A. Maurer, M. Pontil, and B. Romera-Paredes, "The Benefit of Multitask Representation Learning," J. Mach. Learn. Res., vol. 17, no. 81, pp. 1–32, 2016.

[3] J. Z. HaoChen, C. Wei, A. Gaidon, and T. Ma, "Provable Guarantees for Self-Supervised Deep Learning with Spectral Contrastive Loss," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 34, pp. 5000–5011, 2021.

[4] A. Tomihari and I. Sato, "Understanding Linear Probing then Fine-tuning Language Models from NTK Perspective," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 37, 2024.

[5] S. Karp, E. Winston, Y. Li, and A. Singh, "Local Signal Adaptivity: Provable Feature Learning in Neural Networks Beyond Kernels," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 34, 2021.

[6] Z. Shi, J. Wei, and Y. Liang, "Provable Guarantees for Neural Networks via Gradient Feature Learning," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 36, 2023.

[7] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-Task Learning for Dense Prediction: A Survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 7, pp. 3614–3633, 2022.

[8] J. Yu, Y. Dai, X. Liu, J. Huang, Y. Shen, K. Zhang, et al., "Unleashing the Power of Multi-Task Learning," arXiv:2404.18961, 2024.

[9] A. Kendall, Y. Gal, and R. Cipolla, "Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 7482–7491, 2018.

[10] O. Sener and V. Koltun, "Multi-Task Learning as Multi-Objective Optimization," in Proc. Int. Conf. Mach. Learn. (ICML), pp. 4344–4352, 2018.

[11] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient Surgery for Multi-Task Learning," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, pp. 5824–5836, 2020.

[12] H. Ishfaq, T. Nguyen-Tang, S. Feng, R. Arora, M. Wang, M. Yin, and D. Precup, "Offline Multitask Represe

[13] L. Lin and S. Mei, "A Statistical Theory of Contrastive Learning via Approximate Sufficient Statistics," arXiv preprint arXiv:2503.17538, 2025.

[14] T. D. Standley, A. S. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which Tasks Should Be Learned Together in Multi-task Learning?" in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 9129–9140, 2020.

[15] Chen, Z., Mehta, R., & Zhang, T. (2023). Theoretical Insights into Alternating Gradient Descent for Multi-Task Representation Learning. arXiv preprint arXiv:2307.06887.