Multi-Source Sparse Interpolated Experts: Task-Conditioned Composition of Multiple Pretrained Priors with Tight Statistical Guarantees

Liz Lemma Future Detective

January 20, 2026

Abstract

Modern 2026-era foundation model stacks rarely ship a single checkpoint: practitioners maintain multiple pretrained variants (domain-specialized, safety-tuned, supervised vs self-supervised, multilingual, etc.). The SMAT framework shows that sparse interpolated experts between a pretrained model and a meta-tuned delta can improve few-shot OOD generalization in vision. We generalize this idea from a single anchor θ_{pre} to a set of R pretrained sources $\{\theta^{(r)}\}$ by learning task-conditioned mixing weights over sources and sparse parameter corrections. We formalize a multi-source sparse interpolation model where each task's optimum lies in the convex hull of pretrained priors plus a sparse residual, and we study the corresponding few-shot estimator. In a convex surrogate regime (frozen features + regularized squared/logistic loss), we prove an excess-risk upper bound scaling as $O((R + k\log(d/k))/n)$ from n support examples and match it with an information-theoretic lower bound, yielding tight rates. We also show hardness of exact sparse source selection and justify relaxations/approximations. Finally, we propose Multi-Source SMAT: an amortized router producing per-task mixture weights from support-set prototypes, plus learned sparse masks for residual specialization under explicit constraints, and outline experiments on cross-domain few-shot classification and safety/robustness shifts to demonstrate that a single meta-tuned router composes multiple pretrained 'world models' and improves OOD performance without sacrificing ID.

Table of Contents

2. 2. Related Work: SMAT/meta-tuning; model soups/task arithmetic; adapter composition; mixture-of-experts routing; constrained sparsity; robust and distributionally robust generalization under shift.

3. 3. Problem Setup and Notation: task distribution, few-shot support/query; multi-source parameter space; constraints (simplex mixing, sparsity/latency budgets); evaluation metrics (ID/OOD, tail risk).

4. 4. Multi-Source Sparse Interpolation Model: formal generative assumption $\theta_T^\star = \sum_r w_r^\star \theta^{(r)} + \Delta_T^\star$; structured/block sparsity; when assumption is plausible (domain priors, safety priors).

5. 5. Oracle Per-Task Solver: constrained ERM over simplex + ℓ_0-sparse residual; relaxations (group lasso, hard-concrete) and Frank–Wolfe; guarantees and failure modes.

6. 6. Multi-Source SMAT (Amortized Router): architecture for predicting mixing weights from support-set prototypes; sparse masks and residual delta; optional distillation teacher; deployment modes (no-adapt / gradient-free / gradient-based).

7. 7. Theory I (Upper Bounds): excess-risk bounds for convex losses; decomposition into (a) simplex estimation and (b) sparse residual estimation; extension to block sparsity and approximate source model mismatch.

8. 8. Theory II (Lower Bounds and Hardness): minimax lower bounds for mixture+sparseness; NP-hardness of sparse source subset selection and ℓ_0 residual optimization; implications for approximation.

9. 9. Complexity and Systems Considerations: time/memory for oracle solver vs amortized router; impact of structured sparsity; latency proxies; discussion of checkpoint compatibility constraints.

10. 10. Experimental Plan (Flagged as Strengthening): cross-domain few-shot suites; additional OOD/safety shifts; ablations on number/diversity of sources; structured vs unstructured masks; comparison to soups/arithmetic baselines; reporting Pareto fronts under latency budgets.

11. 11. Discussion and Future Directions: multi-modal sources; privacy/edge routing; uncertainty-aware mixing; continual addition/removal of sources; open problems.

1 Introduction

By 2026, the operational reality of pretrained models is not the existence of a single canonical checkpoint, but rather a growing collection of partially overlapping ones: foundation models at multiple scales; instruction-tuned variants; domain-adapted checkpoints for code, biomedical text, or multilingual data; safety-aligned snapshots; and organization-specific finetunes produced by continuous training pipelines. When a new downstream task arrives with only a small labeled support set, the question "which initialization should we use?" is often ill-posed, because several candidate checkpoints are simultaneously plausible and none is uniformly dominant across tasks, domains, or evaluation criteria. Moreover, this selection problem is dynamic: tasks sampled from a mixture distribution (including out-of-distribution episodes) may demand different degrees of specialization, and the best choice among available checkpoints can change abruptly with the task.

Standard meta-tuning and few-shot adaptation methods typically impose a *single-anchor* view of transfer: one chooses a distinguished pretrained initialization and learns an adaptation rule (explicit optimization, an amortized update, or a low-rank/adapter parameterization) around that anchor. This paradigm is effective when there exists a universally good starting point and task variation is well-modeled as a small perturbation. However, the multi-checkpoint regime violates both assumptions. First, if tasks decompose into clusters aligned with different pretraining distributions, any single anchor may be systematically suboptimal for a nontrivial portion of tasks. Second, if different checkpoints encode complementary inductive biases, then restricting to a neighborhood of one anchor discards the possibility of composing these biases in a task-dependent way. The failure mode is not merely constant-factor inefficiency: in low-data settings, committing to the wrong anchor can induce an irreducible bias that cannot be corrected by a small update without overfitting.

We therefore adopt a compositional viewpoint in parameter space. The central modeling decision is to represent the task-specific parameter as a *mixture* of available sources together with a *sparse residual* that accounts for task idiosyncrasies not expressible by mixing alone. Concretely, we seek a representation of the form

$$\theta_T \approx \sum_{r=1}^{R} w_{T,r}\, \theta^{(r)} \;+\; \Delta_T,$$

where the mixing weights lie in the simplex (thereby restricting to the convex hull of pretrained sources) and the residual is constrained to be sparse (or block-sparse under a fixed modular partition). The simplex constraint is not merely aesthetic: it induces a low-dimensional search space for routing across sources, avoids uncontrolled extrapolation in parameter space, and

admits efficient approximate optimization via conditional-gradient methods. The residual constraint plays a complementary role: it provides a controlled mechanism for specialization when the convex hull is insufficient, while keeping the effective degrees of freedom commensurate with the limited support set size.

From this representation we obtain two algorithmic regimes. In the oracle regime, given the support set for a task, we solve a constrained empirical risk minimization problem over mixing weights and sparse residuals, producing a per-task parameter vector used for prediction on query inputs. This captures the idealized statistical behavior of multi-source composition under explicit constraints. In the amortized regime, we learn a meta-router that maps the support set to mixing weights (and optionally to a combination of sparse residual "experts"), enabling rapid deployment-time adaptation without solving a nonconvex sparse optimization problem from scratch. The amortized model is designed to treat pretrained checkpoints as read-only: we construct the task model by combining sources and adding masked deltas, with no need to backpropagate through source parameters.

Our contributions are as follows.

(1) A multi-source sparse interpolation framework for few-shot adaptation. We formalize the setting in which a task-specific optimum is approximated by (i) convex mixing over R pretrained checkpoints and (ii) an additional k-sparse (or block-k sparse) residual. This formulation unifies several empirical observations—that interpolating checkpoints can yield strong performance, and that task-specific deltas can be efficiently represented in structured sparse form—into a single constrained estimation problem. The formulation cleanly separates *routing* (estimating w_T in a low-dimensional simplex) from *specialization* (estimating Δ_T under a sparsity budget).

(2) Tight statistical rates with a matching minimax lower bound. In a convex surrogate regime (e.g., linear prediction on frozen features with ridge regularization), we analyze the constrained empirical risk minimizer and prove an excess-risk bound scaling as

$$\mathbb{E}\big[\ell_T(\widehat{\theta}_T) - \ell_T(\theta_T^\star)\big] \;=\; O\!\left(\frac{R + k\log(d/k)}{n}\right),$$

under standard smoothness and strong convexity assumptions and a realizability condition consistent with the proposed representation. The rate decomposes into an R term associated with estimating mixture weights and a $k\log(d/k)$ term associated with sparse specialization, reflecting the distinct effective dimensions of the two components. We further show that this scaling is minimax-optimal over the corresponding model class: no estimator can improve the dependence on R, k, d, and n in worst case beyond constant

factors. This establishes that multi-source composition with sparse residuals is not only algorithmically plausible but statistically well-calibrated to the few-shot regime.

(3) Practical optimization and amortization in the multi-checkpoint setting. Exact optimization with explicit ℓ_0 constraints is computationally intractable in general, and sparse source subset selection is similarly hard. We therefore emphasize relaxations that preserve the statistical structure while enabling computation. On the simplex side, smooth convex objectives admit Frank–Wolfe optimization, which yields sparse mixtures as a byproduct of its vertex-based updates and provides an anytime trade-off between the number of active sources and suboptimality. On the residual side, we employ standard sparse approximations (hard-thresholding, hard-concrete relaxations, or structured masks) to satisfy a budget on nonzeros. Finally, we introduce an amortized router that predicts the mixing weights (and optionally residual expert weights) directly from the support set. This router is trained on meta-training tasks to approximate the oracle mapping, yielding constant-time routing overhead at test time beyond encoding the support examples.

The resulting perspective shifts the role of meta-learning: rather than learning *one* initialization that must serve all tasks, we learn how to *navigate* a library of pretrained checkpoints and how to allocate a limited specialization budget. This is particularly natural under distribution shift, where different sources may be variably relevant and where conservative convex mixing provides a robust default. The next section situates this approach relative to existing work on meta-tuning and sparse adaptation, checkpoint interpolation and task arithmetic, adapter composition, mixture-of-experts routing, and constrained generalization under shift.

2 Related Work

Our setting intersects several lines of work that, when viewed together, motivate a compositional treatment of pretrained checkpoints together with a constrained mechanism for task-specific specialization. We summarize the closest connections and distinguish the aspects that are essential for our analysis and algorithmic design.

Meta-tuning and few-shot adaptation around a single anchor. Classical meta-learning methods aim to produce a single initialization (or update rule) that can be rapidly adapted to new tasks, most prominently via gradient-based schemes such as MAML and its variants **??**, as well as implicit or amortized alternatives (e.g., learning update directions, learning per-parameter learning rates, or learning task-conditioned hypernetworks)

??. In these approaches the pretrained or meta-trained parameter θ_{pre} acts as a distinguished anchor, and per-task adaptation is modeled as a small update constrained by a limited number of steps, a norm bound, a low-rank parameterization, or an adapter bottleneck. Our multi-checkpoint regime departs from this single-anchor view: rather than committing to a particular checkpoint and adapting locally, we treat the available pretrained models as a structured set that can be navigated by task-dependent routing. The residual component we introduce is compatible with meta-tuning in the sense that it can be implemented through standard efficient finetuning parameterizations, but the conceptual separation between *routing* (selecting a point in a low-dimensional hull of sources) and *specialization* (a sparse modification) is not explicit in most single-anchor analyses.

Sparse masked adaptation and structured parameter-efficient finetuning. A large body of work seeks to reduce adaptation cost by restricting trainable degrees of freedom: adapters **?**, prompt/prefix tuning **??**, IA^3 **?**, and low-rank updates such as LoRA **?**. In parallel, sparse finetuning and masked adaptation methods train a subset of parameters (or sparse deltas) using pruning-style masks or learned gates, often aiming at favorable accuracy–compute trade-offs **??**. Our practical residual component is closest in spirit to this literature, with the additional constraint that the residual is *task-conditioned* and budgeted (via ℓ_0 or structured sparsity) to match few-shot sample sizes. The key distinction is that we couple sparse specialization to multi-source mixing; this changes both the statistical story (effective dimension decomposes into a simplex part and a sparse part) and the computational story (one may amortize routing while keeping sources frozen and residual sparse).

Checkpoint interpolation, model soups, and task arithmetic. A complementary line of work studies the empirical phenomenon that linear combinations of neural network parameters can preserve or improve performance, especially when combining finetuned models from a common base. This includes model soups and weight-space averaging **?**, linear mode connectivity and related connectivity results **??**, and "task arithmetic" / weight-space composition of deltas **?**. Related ideas appear in merging and averaging LoRA or adapter deltas, and in post-hoc combination of domain specialists. These works provide strong empirical evidence that parameter-space composition is viable, but typically focus on *global* merges (one merged model for all tasks) or on compositions that do not explicitly incorporate a few-shot support set at test time. Our formulation instead treats the mixing weights as *per-task* variables inferred from S_T, and it supplements mixing with a constrained residual when the convex hull is insufficient. The simplex constraint we impose can be interpreted as a conservative variant of

weight-space averaging that avoids extrapolation, which is particularly relevant under distribution shift.

Adapter composition and modular reuse across tasks. Modular transfer has a long history, including composing separately trained modules, routing among a library of adapters, and assembling models from reusable components. Empirically, one can attach multiple adapters and select or combine them based on task descriptors or learned gating **?**, and more broadly, one can view parameter-efficient finetuning as learning a task-specific element in a low-dimensional subspace that may be shared or composed across tasks. Our approach may be viewed as operating one level higher: the "modules" we combine are entire pretrained checkpoints (or, in the blockwise variant, checkpoint parameters at a module granularity), and the residual plays the role of a lightweight task-specific correction. The emphasis is not merely on modularity, but on a precise accounting of degrees of freedom: the routing component has complexity scaling with R, while the specialization component scales with sparsity k.

Mixture-of-experts routing and conditional computation. Mixture-of-experts (MoE) architectures **???** route tokens (or examples) to a subset of experts to achieve conditional computation and improved scaling. Routing is typically learned end-to-end with differentiable gates and load-balancing regularizers; the experts themselves are trained jointly, and the router operates at token-level or sequence-level granularity. Our routing problem differs in three ways. First, the experts (sources) are pretrained checkpoints treated as read-only, rather than jointly trained expert subnetworks. Second, routing is performed at the *task* level using a few-shot support set, producing a single set of weights w_T that defines the task model. Third, we focus on constrained estimation with explicit simplex and sparsity budgets, enabling statistical characterization in the convex surrogate regime and motivating optimization strategies such as Frank–Wolfe for simplex variables and thresholding/relaxations for sparse residuals. Nonetheless, the amortized meta-router we study is conceptually aligned with MoE gating in that it learns a fast mapping from data summaries to mixture weights.

Constrained sparsity, subset selection, and optimization under budgets. Imposing hard constraints such as $\|\Delta_T\|_0 \le k$ connects our residual estimation to sparse regression and best subset selection, which are computationally intractable in the worst case and thus commonly approached via relaxations (e.g., ℓ_1 penalties, group lasso, or stochastic gates such as hard-concrete) **??**. Similarly, sparsity over sources (selecting a small subset of checkpoints) is a form of combinatorial model selection; conditional-gradient methods offer a natural relaxation when the objective is convex in the mix-

ture and the feasible set is the simplex, producing sparse mixtures as a byproduct of vertex-based updates **?**. Our algorithmic choices mirror these classical themes: we retain explicit constraints at the modeling level, and we use computational surrogates that preserve the structure needed for analysis and for deployment-time budgeting.

Robustness, distribution shift, and multi-source generalization. Finally, our emphasis on task distributions that may mix in-distribution and out-of-distribution episodes is connected to robust optimization and distributionally robust generalization, including group DRO, worst-case risk, and tail-risk objectives such as CVaR **??**. Multi-source domain adaptation and hypothesis aggregation also motivate convex combinations of predictors or source hypotheses under shift **??**. While our mixing occurs in parameter space rather than prediction space, the simplex restriction serves a similar purpose: it constrains adaptation to a conservative region defined by available sources, which can mitigate overconfident extrapolation when support data are scarce or shifted. The sparse residual then provides a controlled escape hatch, whose complexity is calibrated to the amount of per-task data.

These connections clarify the design choices in our framework: we combine (i) parameter-space composition inspired by checkpoint interpolation and modular transfer, (ii) constrained sparse specialization aligned with parameter-efficient and sparse finetuning, and (iii) task-conditioned routing analogous to MoE gating but over pretrained sources. We now make these components precise by formalizing the task distribution, support/query protocol, and the constrained parameter class used throughout our analysis and algorithms.

3 Problem Setup and Notation

We consider a few-shot task distribution $\mathcal{P}$ over episodes T. Each task T induces an example distribution $\mathcal{D}_T$ on input–label pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$. An episode provides a labeled *support set* $S_T = \{(x_i, y_i)\}_{i=1}^{n}$ sampled i.i.d. from $\mathcal{D}_T$, and an independent *query set* $Q_T = \{(x'_j, y'_j)\}_{j=1}^{n_q}$ sampled i.i.d. from the same $\mathcal{D}_T$. The support set is the only information available to construct a task-adapted predictor; performance is evaluated on the query distribution (or the realized query set, when we report empirical metrics).

We work with a shared parameterization $\theta \in \mathbb{R}^d$ (e.g., a linear predictor on frozen features in the convex surrogate regime, or a full neural network parameter vector in practice). Each task T is associated with a population risk

$$\ell_T(\theta) \;:=\; \mathbb{E}_{(x,y)\sim\mathcal{D}_T}\big[\ell(\theta;(x,y))\big],$$

and its empirical counterpart on the support set,

$$\widehat{\ell}_T(\theta) \;:=\; \frac{1}{n} \sum_{(x_i,y_i)\in S_T} \ell(\theta;(x_i,y_i)).$$

In the theory sections we assume ℓ_T is convex, L-smooth, and μ-strongly convex in θ for each T (e.g., squared loss or logistic loss with ridge regularization on fixed features). This allows us to interpret excess population risk $\ell_T(\widehat{\theta}_T) - \ell_T(\theta_T^\star)$ as the primary measure of statistical error. In the practical regime, $\widehat{\ell}_T$ is the training objective used to infer task-specific parameters, while evaluation uses standard predictive metrics on Q_T (accuracy for classification, negative log-likelihood, etc.).

Our central resource is a library of R pretrained source checkpoints $\{\theta^{(r)}\}_{r=1}^R \subset \mathbb{R}^d$ that share architecture and parameter indexing. We treat these sources as read-only at test time: the per-task model is obtained by composing them in parameter space, optionally supplemented by a restricted task-specific residual. To formalize this composition, we use mixing weights $w_T = (w_{T,1}, \ldots, w_{T,R})$ constrained to the probability simplex

$$\Delta_R \;:=\; \big\{w \in \mathbb{R}^R :\; w_r \geq 0,\; \sum_{r=1}^R w_r = 1\big\}.$$

Given $w_T \in \Delta_R$, the associated convex-hull mixture is $\sum_{r=1}^R w_{T,r}\theta^{(r)}$. The simplex constraint enforces nonnegative interpolation among sources and rules out extrapolation; operationally, it yields a low-dimensional adaptation space of (effective) dimension at most $R-1$.

To permit limited specialization beyond the convex hull, we add a task-dependent residual vector $\Delta_T \in \mathbb{R}^d$ subject to a hard sparsity constraint. In the unstructured case we require $\|\Delta_T\|_0 \leq k$, where $\|\cdot\|_0$ denotes the number of nonzero entries. In structured variants we assume a fixed partition of parameters into L blocks (modules), and impose block sparsity: at most k blocks are allowed to be nonzero, with arbitrary values within each selected block. These constraints encode the few-shot regime: the residual carries additional degrees of freedom, but its complexity is explicitly budgeted.

We thus define the per-task parameter as

$$\theta_T(w_T, \Delta_T) \;:=\; \sum_{r=1}^R w_{T,r}\theta^{(r)} + \Delta_T,$$

with feasibility constraints $w_T \in \Delta_R$ and Δ_T sparse (unstructured or block-sparse). The oracle per-task estimator is the constrained empirical risk minimizer

$$(\widehat{w}_T, \widehat{\Delta}_T) \;\in\; \arg\min_{w\in\Delta_R,\; \Delta:\; \|\Delta\|_0\leq k} \widehat{\ell}_T\Big(\sum_{r=1}^R w_r\theta^{(r)} + \Delta\Big), \qquad \widehat{\theta}_T := \theta_T(\widehat{w}_T, \widehat{\Delta}_T).$$

This optimization problem separates the adaptation degrees of freedom into (i) a simplex-constrained mixture over sources and (ii) a sparse correction. The two components will later admit distinct statistical accounting: the mixture scales with R, while the residual scales with k (and, in unstructured form, a $\log(d/k)$ term).

In practical implementations, sparsity is enforced via a family of masks $\mathcal{Z}$, where a mask $z \in \{0,1\}^d$ (or a structured mask over blocks) selects coordinates to update; the residual is realized as $z \odot \delta$ for some learned or optimized $\delta \in \mathbb{R}^d$. We also consider amortized settings in which multiple masks $\{z_m\}_{m=1}^M$ ("experts") are learned during meta-training and combined at test time via task-conditioned weights, but for the present section it suffices to view the constraint as limiting $\mathrm{nnz}(\Delta_T)$ and hence the incremental compute required to apply the residual.

Beyond statistical constraints, we optionally impose a deployment budget through a cost proxy $\widehat{\mathrm{cost}}(\theta_T)$ capturing latency, memory traffic, or energy. In the simplest case, this proxy depends primarily on the number of nonzeros in the residual (and, in variants that restrict source usage, on the number of active sources). Formally, one may constrain $\widehat{\mathrm{cost}}(\theta_T) \le C$ and solve either a projected problem or a Lagrangian relaxation; we treat such budgets as orthogonal to the statistical formulation, but they motivate the same sparsity and modular structure.

Finally, we specify how task distributions and evaluation under shift are represented. We allow $\mathcal{P}$ to mix in-distribution and out-of-distribution tasks, for instance

$$\mathcal{P} \;=\; (1-\gamma)\mathcal{P}_{\mathrm{ID}} + \gamma\,\mathcal{P}_{\mathrm{OOD}},$$

with $\gamma \in [0,1]$ unknown at test time. While our default objective is the mean query risk $\mathbb{E}_{T\sim\mathcal{P}}[\ell_T(\theta_T)]$, we also consider tail-focused criteria such as $\mathrm{CVaR}_\alpha(\ell_T(\theta_T))$ over $T \sim \mathcal{P}$, which emphasize worst-case or rare but important tasks. In experiments, we report both average performance and stratified performance on ID and OOD subsets, as well as tail-risk summaries when appropriate.

This notation isolates the elements that will recur throughout: sources $\{\theta^{(r)}\}_{r=1}^R$, task-conditioned simplex weights w_T, sparse specialization Δ_T, and the support/query protocol defining adaptation and evaluation. We next formalize the structural assumption relating tasks to the multi-source parameter class.

4 Multi-Source Sparse Interpolation Model

We now state the structural hypothesis that links the task distribution $\mathcal{P}$ to the library of pretrained sources $\{\theta^{(r)}\}_{r=1}^R$. The hypothesis specifies a restricted parameter class in which (i) coarse task variation is captured by

interpolation among sources, while (ii) residual task idiosyncrasies are localized to a small subset of coordinates or modules. This is a statement about the *population* minimizers $\theta_T^\star \in \arg\min_\theta \ell_T(\theta)$; the estimator of §3 is an empirical proxy constrained to the same class.

Generative form and hypothesis class. Our standing assumption is that for each task T there exist weights $w_T^\star \in \Delta_R$ and a sparse residual $\Delta_T^\star \in \mathbb{R}^d$ such that

$$\theta_T^\star \;=\; \sum_{r=1}^{R} w_{T,r}^\star\, \theta^{(r)} \;+\; \Delta_T^\star, \qquad \|\Delta_T^\star\|_0 \le k. \tag{1}$$

Equivalently, we posit that $\theta_T^\star$ lies in the Minkowski sum of the convex hull of sources and a k-sparse set. We write the corresponding hypothesis class as

$$\mathcal{H}_{R,k} \;:=\; \Big\{ \sum_{r=1}^{R} w_r \theta^{(r)} + \Delta :\; w \in \Delta_R,\; \|\Delta\|_0 \le k \Big\} \subset \mathbb{R}^d. \tag{2}$$

The simplex restriction is not purely technical: it encodes the inductive bias that tasks are generated by *interpolation* among known behaviors rather than extrapolation beyond them. In particular, if each $\theta^{(r)}$ corresponds to a model that has been pre-validated for a given domain, safety constraint, or calibration regime, then $w \in \Delta_R$ preserves nonnegativity and yields a form of convex averaging that can be easier to audit than arbitrary linear combinations.

We emphasize that representation (1) need not be unique: distinct pairs (w, Δ) may map to the same θ. Our analysis and algorithms depend only on membership in $\mathcal{H}_{R,k}$ and do not require identifiability of the decomposition.

Structured and block-sparse residuals. In large neural models, the unstructured ℓ_0 constraint can be too permissive (it may scatter updates across the network) or too restrictive (it may prevent coherent adaptation within a module). We therefore also consider a structured variant based on a fixed partition of coordinates into L disjoint blocks,

$$[d] \;=\; \bigsqcup_{\ell=1}^{L} \mathcal{I}_\ell, \qquad \Delta = (\Delta_{[\mathcal{I}_1]}, \dots, \Delta_{[\mathcal{I}_L]}),$$

where blocks may correspond to layers, attention heads, MLP submodules, or any architecturally meaningful grouping. Let the block support be $\mathrm{bsupp}(\Delta) := \{\ell : \Delta_{[\mathcal{I}_\ell]} \neq 0\}$ and define the block sparsity level $\|\Delta\|_{0,\mathrm{blk}} := |\mathrm{bsupp}(\Delta)|$. The block-sparse hypothesis is then

$$\theta_T^\star \;=\; \sum_{r=1}^{R} w_{T,r}^\star\, \theta^{(r)} \;+\; \Delta_T^\star, \qquad \|\Delta_T^\star\|_{0,\mathrm{blk}} \le k, \tag{3}$$

allowing dense changes within at most k selected blocks. This structure aligns with deployment constraints: updating a small number of modules can reduce memory traffic and can simplify interpretation (we can report which modules were altered). It also aligns with common fine-tuning phenomena in transformers, where adaptation often concentrates in specific layers or in low-rank directions within attention/MLP weights.

Interpretation: interpolation as task retrieval, residual as specialization. Model (1) admits a natural two-stage interpretation. The mixture $\sum_r w^\star_{T,r}\theta^{(r)}$ performs *task retrieval* from a finite library: the weights select a point in a low-dimensional simplex that approximates the task optimum up to a small error. The residual $\Delta^\star_T$ performs *specialization*: it corrects the retrieved point using a limited number of additional degrees of freedom. When $k = 0$ we recover pure convex-hull composition; when $R = 1$ we recover sparse adaptation from a single initialization (a convex surrogate for sparse fine-tuning, masking, or sparse adapters). Thus (1) interpolates between multi-checkpoint routing and sparse single-checkpoint adaptation.

When is the assumption plausible? We view (1) as plausible whenever the task distribution $\mathcal{P}$ can be described as a mixture of a small number of recurring "modes" plus bounded idiosyncratic variation.

First, in domain-mixture settings (e.g., a test-time stream mixing medical, legal, and general text), it is common to have sources pretrained or instruction-tuned on different domains. If task optima cluster near these sources, then a convex mixture can approximate the cluster centroid for a given task, with $\Delta^\star_T$ absorbing domain-specific label conventions or dataset artifacts that are not shared with any single source.

Second, in multi-objective or safety-constrained deployment, sources may encode different trade-offs (helpfulness, refusal, style, calibration, conservatism). The simplex constraint can be seen as a *safety prior*: rather than learning a fresh parameter vector from n samples, we restrict ourselves to averaging among pre-audited behaviors, and we budget any deviation via sparsity. In this interpretation, k plays the role of a deviation budget: it limits how much the task can "override" the library, and where.

Third, in out-of-distribution adaptation, we do not expect $\theta^\star_T$ to lie exactly in the convex hull; nevertheless, we may expect it to lie close to the hull in a restricted set of directions. For example, if the feature representation is largely reusable and only a few decision-boundary coordinates change, then $\Delta^\star_T$ can be sparse in the effective parameterization of the convex surrogate. In neural settings, the effective degrees of freedom relevant to a task may be concentrated in a small set of heads or layers; the block-sparse formulation (3) captures this.

Strength of the hypothesis and approximate variants. As stated, (1) is an exact realizability assumption. It is often sufficient (and more realistic) to assume an approximate form: there exists $\bar{\theta}_T \in \mathcal{H}_{R,k}$ such that $\ell_T(\bar{\theta}_T) - \inf_\theta \ell_T(\theta)$ is small. All subsequent procedures can then be interpreted as controlling an estimation term (from finite support samples) plus an approximation term (from model mismatch). In particular, the value of R and k should be chosen to reflect the intended trade-off: larger R and k enlarge $\mathcal{H}_{R,k}$ and reduce approximation error, but increase statistical and computational burden.

The role of this section is therefore to define a concrete, auditable parameter class connecting tasks to a source library, with explicit knobs (R, k, and the block partition) that quantify adaptation capacity. In the next section we study the per-task constrained ERM over this class and the algorithmic relaxations needed to compute it.

5 Oracle Per-Task Solver: Constrained ERM over Simplex Mixing and Sparse Residual

Given a new task T with support set S_T of size n, our oracle estimator is the constrained empirical risk minimizer over the hypothesis class $\mathcal{H}_{R,k}$ introduced in §4. Concretely, we define

$$(\widehat{w}_T, \widehat{\Delta}_T) \in \arg \min_{w \in \Delta_R,\ \Delta : \|\Delta\|_0 \le k} \widehat{\ell}_T(\theta(w,\Delta)), \qquad \theta(w,\Delta) := \sum_{r=1}^{R} w_r \theta^{(r)} + \Delta, \tag{4}$$

and output $\widehat{\theta}_T := \theta(\widehat{w}_T, \widehat{\Delta}_T)$. In the convex surrogate regime (e.g., linear prediction on frozen features with ridge regularization), $\widehat{\ell}_T(\theta)$ is convex in θ and hence convex in w for fixed Δ, but the joint optimization in (4) remains nonconvex due to the ℓ_0 constraint on Δ (and, in the block-sparse variant, due to combinatorial block selection). For the statistical statements in §4, we treat (4) as defining an oracle solution; we then separate statistical error (finite n) from any additional optimization error incurred by practical solvers.

Reparameterization and gradients. Let $g(\theta) := \widehat{\ell}_T(\theta)$. Since $\theta(w,\Delta)$ is affine in (w,Δ), we have

$$\nabla_w g(\theta(w,\Delta)) = \Big(\langle \nabla_\theta g(\theta(w,\Delta)), \theta^{(1)} \rangle, \ldots, \langle \nabla_\theta g(\theta(w,\Delta)), \theta^{(R)} \rangle \Big) \in \mathbb{R}^R, \tag{5}$$

and $\nabla_\Delta g(\theta(w,\Delta)) = \nabla_\theta g(\theta(w,\Delta))$. Thus, any first-order method requires (i) computing $\nabla_\theta g(\cdot)$ on the support set, and (ii) aggregating inner products with the source parameters to obtain a gradient in the simplex coordinates.

Simplex optimization via Frank–Wolfe. When Δ is held fixed, minimizing $w \mapsto g(\theta(w, \Delta))$ over the simplex is a smooth convex problem under our assumptions. We therefore use Frank–Wolfe (conditional gradient), which avoids explicit projection and enjoys an intrinsic sparsity property of the iterates. Starting from some $w^{(0)} \in \Delta_R$, the t-th step computes

$$s^{(t)} \in \arg\min_{s \in \Delta_R} \langle s, \nabla_w g(\theta(w^{(t)}, \Delta)) \rangle, \qquad w^{(t+1)} = (1 - \gamma_t) w^{(t)} + \gamma_t s^{(t)}, \quad (6)$$

where $\gamma_t \in (0, 1]$ is a step size. Since the linear minimization oracle over Δ_R selects a vertex, $s^{(t)} = e_{r_t}$ for some $r_t \in [R]$, and consequently $w^{(t)}$ has at most $t + 1$ nonzeros. This provides an anytime trade-off between optimization accuracy and the number of active sources, without imposing an explicit $\|w\|_0$ constraint. In settings where source retrieval cost scales with the number of active sources, this property can be used to control deployment overhead.

Residual optimization: exact ℓ_0 is intractable, practical relaxations. For fixed w, (4) reduces to minimizing $g(\sum_r w_r \theta^{(r)} + \Delta)$ subject to $\|\Delta\|_0 \le k$. Even for quadratic losses this contains sparse regression as a special case and is NP-hard in general. We therefore distinguish three practical surrogates, each appropriate under different structure assumptions.

First, in the unstructured sparse case we can apply iterative hard thresholding (IHT) to a gradient step on Δ:

$$\Delta^{(t+1)} = \mathcal{H}_k\Big(\Delta^{(t)} - \eta_t \nabla_\Delta g(\theta(w, \Delta^{(t)}))\Big), \quad (7)$$

where $\mathcal{H}_k(\cdot)$ keeps the k largest-magnitude coordinates and zeros the rest. Under restricted strong convexity/smoothness conditions familiar from sparse estimation, IHT converges to a neighborhood whose size is controlled by noise and model mismatch; in our usage it serves as a fast heuristic that enforces the budget exactly.

Second, for block sparsity with a fixed partition $\{\mathcal{I}_\ell\}_{\ell=1}^L$, we may use a group-lasso relaxation:

$$\min_{w \in \Delta_R,\ \Delta \in \mathbb{R}^d} \ g(\theta(w, \Delta)) + \lambda \sum_{\ell=1}^{L} \|\Delta_{[\mathcal{I}_\ell]}\|_2, \quad (8)$$

followed by selecting the top-k blocks by $\|\Delta_{[\mathcal{I}_\ell]}\|_2$ and optionally refitting on the selected support. This replaces the combinatorial constraint by a convex penalty (for convex g) and yields stable solutions when blocks correspond to semantically meaningful modules. The trade-off is that the penalty introduces bias, and the conversion from penalized to hard-sparse solutions depends on separation among block magnitudes.

Third, in neural implementations where one wishes to learn masks by gradient descent, we can use a hard-concrete (or related) relaxation for binary gates. Writing $\Delta = z \odot \delta$ with trainable $\delta \in \mathbb{R}^d$ and stochastic/relaxed gates $z \in [0,1]^d$, we optimize a differentiable surrogate of (4) with an expected sparsity penalty $\lambda\, \mathbb{E}[\|z\|_0]$ and anneal the relaxation. This approach supports end-to-end optimization and amortization (used in §6), but the obtained sparsity is only approximate and must typically be enforced by post hoc thresholding.

Alternating minimization and stopping criteria. A practical per-task solver alternates between (approximately) minimizing over w using a small number of Frank–Wolfe steps and (approximately) minimizing over Δ using one of the sparse updates above. Since the objective is smooth in θ, we may monitor the decrease in $\widehat{\ell}_T(\theta(w, \Delta))$ and stop when improvements fall below a tolerance, or when a fixed budget of gradient evaluations is reached. While joint global optimality is not guaranteed, this block-coordinate strategy is consistent with the structure of (4): the simplex part is well-conditioned and low-dimensional, whereas the residual part is high-dimensional but constrained by a strict budget.

Guarantees and accounting for optimization error. The excess-risk rate in Theorem 1 is stated for the exact constrained ERM. If a solver returns $\widetilde{\theta}_T$ satisfying

$$\widehat{\ell}_T(\widetilde{\theta}_T) \le \min_{\theta \in \mathcal{H}_{R,k}} \widehat{\ell}_T(\theta) + \varepsilon_{\mathrm{opt}}, \tag{9}$$

then, by standard stability arguments for μ-strongly convex and L-smooth losses, the population excess risk increases by at most an additional term on the order of $\varepsilon_{\mathrm{opt}}$ (up to problem-dependent constants). Thus, in the convex surrogate regime, the statistical rate and the algorithmic approximation may be separated: we first control estimation over $\mathcal{H}_{R,k}$ and then add the optimization gap.

Failure modes. We record three limitations that recur in practice. (i) If the sources $\{\theta^{(r)}\}$ are nearly collinear in the geometry induced by the task loss, then w may be poorly identified, and Frank–Wolfe may oscillate among similar vertices; this is benign for prediction but degrades interpretability of the routing. (ii) If the true residual is not sparse in the chosen parameterization (or block partition), then enforcing $\|\Delta\|_0 \le k$ induces approximation error that cannot be removed by additional samples; this motivates the approximate-realizability viewpoint and the use of structured residuals. (iii) When n is extremely small, the empirical objective may be dominated by noise, and the ℓ_0 constraint may lead to overfitting on the support set;

in such regimes one should increase regularization (e.g., ridge in the surrogate) or reduce k, and one should evaluate with held-out queries as in the meta-learning protocol.

6 Multi-Source SMAT (Amortized Router): Predicting Simplex Weights and Sparse Residuals from Few-Shot Support Sets

The oracle program in (4) defines, for each task T, an implicit mapping from a support set S_T to mixing weights and a residual, $(S_T \mapsto \widehat{w}_T, \widehat{\Delta}_T)$. In this section we describe an amortized approximation to this mapping which we call *Multi-Source SMAT*. The goal is to replace per-task iterative optimization by a single forward pass on S_T, producing $w_T \in \Delta_R$ (and optionally a sparse residual Δ_T) at essentially constant overhead beyond encoding the support examples.

Support-set encoding and prototype statistics. We assume each task T comes with labeled support examples $S_T = \{(x_i, y_i)\}_{i=1}^n$, typically in a few-shot classification format with a small number of classes. We form permutation-invariant task features by computing class-wise prototypes in an embedding space. Concretely, let $f_\phi : \mathcal{X} \to \mathbb{R}^p$ be a (meta-trained) encoder applied to support inputs; for each class c present in S_T define the prototype

$$p_c \;:=\; \frac{1}{|S_{T,c}|} \sum_{(x_i,y_i)\in S_{T,c}} f_\phi(x_i), \qquad S_{T,c} := \{(x_i, y_i) \in S_T :\; y_i = c\}.$$

We then aggregate $\{p_c\}$ into a fixed-dimensional summary $u_T \in \mathbb{R}^q$ using a symmetric operator (e.g., concatenation of moments, attention pooling over prototypes, or DeepSets-style pooling). The only requirement for the sequel is that u_T is a deterministic function of S_T (or of a stochastic augmentation thereof) and that it can be computed in $O(n \cdot \mathrm{enc})$ time.

Routing to simplex weights over source checkpoints. Given u_T, the router produces logits $\rho_T \in \mathbb{R}^R$ via a small network $h_\zeta : \mathbb{R}^q \to \mathbb{R}^R$, and outputs mixing weights by

$$w_T \;=\; \mathrm{softmax}(\rho_T) \in \Delta_R, \qquad \rho_T := h_\zeta(u_T).$$

This guarantees the simplex constraint by construction. The resulting mixed initialization (without residual) is

$$\theta_T^{\mathrm{mix}} \;:=\; \sum_{r=1}^{R} w_{T,r}\, \theta^{(r)}.$$

When the architecture admits a meaningful partition into L blocks (layers, attention/MLP modules, heads), we may also predict blockwise mixing weights $\{w_{T,\ell} \in \Delta_R\}_{\ell=1}^{L}$ by outputting $\rho_{T,\ell} \in \mathbb{R}^R$ for each block and setting

$$\theta_{T,\ell}^{\text{mix}} \;:=\; \sum_{r=1}^{R} (w_{T,\ell})_r \, \theta_\ell^{(r)}, \qquad w_{T,\ell} = \text{softmax}(\rho_{T,\ell}),$$

which increases flexibility while keeping the constraint local to each block. Inference cost scales as $O(R)$ (or $O(LR)$ for blockwise mixing) beyond the support encoding.

Sparse residuals via shared deltas and masks. To capture task-specific deviations not expressible by convex combinations of sources, we add a residual term whose nonzeros are controlled by a mask family $\mathcal{Z}$. We parameterize a small set of residual "experts" by a shared delta tensor $\theta_\delta \in \mathbb{R}^d$ and M masks $\{z_m\}_{m=1}^{M} \subset \{0,1\}^d$ (or structured binary tensors aligned with blocks). Each expert corresponds to the masked delta $z_m \odot \theta_\delta$, and we let the router output combination weights $\alpha_T \in \Delta_M$ (or allow a sparse α_T) to form

$$\Delta_T \;:=\; \sum_{m=1}^{M} \alpha_{T,m} \, (z_m \odot \theta_\delta), \qquad \theta_T \;:=\; \theta_T^{\text{mix}} + \Delta_T.$$

This construction separates *where* adaptation is allowed (the support of z_m) from *how much* to adapt (the values in θ_δ and the mixture α_T). The budget can be enforced by requiring each z_m to have at most k nonzeros (or at most k active blocks), yielding $\text{nnz}(\Delta_T) \le k$ when α_T selects a single expert, and a controlled multiple of k more generally.

Learning masks: relaxations and structured sparsity. In practice we learn masks using continuous relaxations. A common choice is to treat each coordinate of z_m as a hard-concrete gate, optimize the expected sparsity $\mathbb{E}[\|z_m\|_0]$ via a Lagrangian penalty, and then discretize at deployment by thresholding so that $\text{nnz}(z_m) \le k$. When blocks $\{\mathcal{I}_\ell\}_{\ell=1}^{L}$ are prescribed, we instead gate at the block level, so that z_m is constant on each block and the residual cost is proportional to the number of activated blocks. This aligns the amortized model with the block-k hypothesis class used in the theory and yields predictable compute (masked-add at module granularity).

Meta-training objective and optional teacher distillation. We meta-train (ϕ, ζ) and the residual parameters by episodic training. For each sampled task T, we compute (w_T, α_T) from S_T, form θ_T, and minimize the query loss on Q_T:

$$\min_{\phi,\zeta,\theta_\delta,\{z_m\}} \; \mathbb{E}_{T\sim\mathcal{P}}\Big[\, \widehat{\ell}_{Q_T}(\theta_T) \,\Big] \;+\; \lambda_w \, \Omega_w(w_T) \;+\; \lambda_z \, \Omega_z(\{z_m\}),$$

where Ω_w may encourage low-entropy routing (for interpretability or reduced source access) and Ω_z enforces sparsity/structure. To better approximate the oracle mapping (4), we may distill from a teacher that solves (approximately) for $(\widehat{w}_T, \widehat{\Delta}_T)$ on the support set. A simple distillation loss is

$$\mathcal{L}_{\text{distill}}(T) \;=\; \tau^2\, \mathrm{KL}\big(\mathrm{softmax}(\widehat{\rho}_T/\tau) \,\big\|\, \mathrm{softmax}(\rho_T/\tau)\big)\,,$$

where $\widehat{\rho}_T$ are teacher logits over sources and $\tau > 0$ is a temperature. One may analogously distill α_T or the induced predictions on Q_T. This hybrid objective reduces the amortization gap when the router is small relative to the complexity of the task family.

Deployment modes and budgeted adaptation. We distinguish three deployment regimes. (i) *No-adapt:* run the router once, form θ_T from w_T (and α_T if used), and evaluate on queries; this yields fixed per-task overhead and never backpropagates through $\theta^{(r)}$. (ii) *Gradient-free refinement:* initialize w at the router output and take a small number of Frank–Wolfe steps on the support loss (optionally keeping the residual fixed); because the iterate remains sparse in the number of visited vertices, this can be stopped early to meet a source-access budget. (iii) *Gradient-based refinement:* optionally fine-tune only low-dimensional variables (e.g., α_T or a small continuous relaxation of the mask) on S_T while keeping sources frozen; this interpolates between pure amortization and the oracle solver, with compute governed by the number of refinement steps and the residual sparsity pattern.

The resulting procedure preserves the core structure of $\mathcal{H}_{R,k}$: prediction is obtained by convex mixing of pretrained checkpoints, augmented by a sparse residual whose support is controlled by masks. The next section analyzes the statistical consequences of this structure in the convex surrogate regime by decomposing estimation into the simplex and sparse components.

7 Theory I (Upper Bounds): Excess-Risk Rates from Simplex Mixing and Sparse Residuals

We analyze the constrained estimator induced by the oracle program (4) in the convex surrogate regime. Fix a task T and write the population and empirical losses as $\ell_T(\theta)$ and $\widehat{\ell}_T(\theta)$, where $\widehat{\ell}_T$ is formed from n i.i.d. support examples. Let the structured hypothesis class be

$$\mathcal{H}_{R,k} \;:=\; \Big\{\theta \in \mathbb{R}^d :\; \theta = \textstyle\sum_{r=1}^{R} w_r \theta^{(r)} + \Delta,\; w \in \Delta_R,\; \|\Delta\|_0 \le k\Big\}.$$

The estimator $\widehat{\theta}_T$ is the empirical risk minimizer over $\mathcal{H}_{R,k}$, and our goal is to bound its expected excess risk.

Assumptions and a reduction via strong convexity. Throughout this section we assume that for every task T, the function ℓ_T is μ-strongly convex and L-smooth in θ. Strong convexity converts excess risk to parameter estimation error:

$$\ell_T(\widehat{\theta}_T) - \ell_T(\theta_T^\star) \;\geq\; \frac{\mu}{2}\,\|\widehat{\theta}_T - \theta_T^\star\|_2^2, \qquad \ell_T(\widehat{\theta}_T) - \ell_T(\theta_T^\star) \;\leq\; \frac{L}{2}\,\|\widehat{\theta}_T - \theta_T^\star\|_2^2, \tag{10}$$

where $\theta_T^\star$ denotes the (assumed unique) minimizer of ℓ_T over $\mathcal{H}_{R,k}$ in the realizable setting. Thus it suffices to control $\|\widehat{\theta}_T - \theta_T^\star\|_2$ in expectation (or with high probability). Technically, our bounds may be obtained by localized Rademacher complexity arguments or by stability of ERM for strongly convex objectives; in either case the statistical complexity will be governed by the metric entropy of $\mathcal{H}_{R,k}$ around $\theta_T^\star$.

Decomposition into simplex and sparse components. A key point is that $\mathcal{H}_{R,k}$ decomposes as a Minkowski sum of two sets:

$$\mathcal{H}_{R,k} \;=\; \underbrace{\Big\{ \sum_{r=1}^{R} w_r \theta^{(r)} :\; w \in \Delta_R \Big\}}_{\mathcal{C}_R \text{ (convex hull of sources)}} \;+\; \underbrace{\Big\{ \Delta :\; \|\Delta\|_0 \leq k \Big\}}_{\mathcal{S}_k \ (k\text{-sparse residuals})}.$$

Consequently, the estimation error may be controlled by separately accounting for (i) uncertainty in the $(R-1)$-dimensional simplex weights, and (ii) uncertainty in the combinatorial support of a k-sparse residual. At a high level, the simplex term contributes a parametric rate proportional to R/n, whereas the sparse term contributes the familiar $k\log(d/k)/n$ rate.

Simplex estimation term. Consider first the residual-free class $\mathcal{C}_R$. Writing $\Theta := [\theta^{(1)} \cdots \theta^{(R)}] \in \mathbb{R}^{d\times R}$, the mixed parameters are $\theta = \Theta w$ with $w \in \Delta_R$. Since Δ_R is a compact convex set of intrinsic dimension $R-1$, covering number bounds yield metric entropy scaling like $(R-1)\log(1/\varepsilon)$ in the natural geometry for w. Under standard boundedness/sub-Gaussian assumptions ensuring Lipschitzness of $\ell_T(\Theta w)$ as a function of w, uniform convergence over Δ_R gives an estimation contribution of order R/n to the excess risk. Intuitively, despite θ living in $\mathbb{R}^d$, the mixing weights are low-dimensional; the statistical price depends on R rather than d.

Sparse residual estimation term. Now consider the residual-only class $\mathcal{S}_k$. Even when the loss is strongly convex, estimating an unrestricted vector in $\mathbb{R}^d$ from n samples would scale with d/n; sparsity replaces d by k up to a combinatorial $\log(d/k)$ factor. Formally, the number of possible supports is $\binom{d}{k}$, and $\log\binom{d}{k} \asymp k\log(d/k)$. Under restricted strong convexity (or an

appropriate compatibility/restricted eigenvalue condition in the linear prediction surrogate), ERM over $\mathcal{S}_k$ achieves excess risk of order $k\log(d/k)/n$. This is the same term that appears in classical sparse regression, and it is unavoidable even when the sources are absent.

Combined upper bound (realizable case). Putting the two components together and applying (10), we obtain the rate stated in Theorem 1: if the task satisfies the representation hypothesis (H1), namely $\theta_T^\star = \sum_r w_{T,r}^\star \theta^{(r)} + \Delta_T^\star$ with $w_T^\star \in \Delta_R$ and $\|\Delta_T^\star\|_0 \le k$, then

$$\mathbb{E}\big[\ell_T(\widehat{\theta}_T) - \ell_T(\theta_T^\star)\big] \;\le\; C\,\frac{R + k\log(d/k)}{n}, \tag{11}$$

for a constant C depending on (L,μ) and on distributional quantities (noise level, feature norms) required to ensure concentration and restricted curvature. The salient feature of (11) is additivity: the convex-hull part behaves as an R-parameter model, while the sparse residual behaves as a k-sparse model.

Extension: block sparsity and structured partitions. Suppose $\{1,\dots,d\}$ is partitioned into L blocks $\mathcal{I}_1,\dots,\mathcal{I}_L$ (e.g., layers or modules), and we replace $\|\Delta\|_0 \le k$ by a block constraint $\|\Delta\|_{\text{block-0}} \le k$, meaning Δ is supported on at most k blocks. The combinatorial term becomes $\log\binom{L}{k} \asymp k\log(L/k)$, while estimation within the active blocks scales with their total dimension. A representative bound is therefore

$$\mathbb{E}\big[\ell_T(\widehat{\theta}_T) - \ell_T(\theta_T^\star)\big] \;\lesssim\; \frac{R}{n} \;+\; \frac{k\log(L/k)}{n} \;+\; \frac{1}{n}\sum_{\ell\in \mathrm{supp}_{\text{block}}(\Delta_T^\star)} |\mathcal{I}_\ell|, \tag{12}$$

with the last term simplifying to $k\,d_{\text{blk}}/n$ when blocks have comparable size $|\mathcal{I}_\ell| \approx d_{\text{blk}}$. In parallel, if one allows *blockwise mixing* with weights $w_{T,\ell} \in \Delta_R$ per block, the simplex term scales with the number of independent simplex parameters, yielding LR/n in the worst case (and smaller when blockwise weights are tied or low-rank across ℓ). This provides a direct statistical justification for using block structure only when it is needed: increased flexibility has a quantifiable sample cost.

Agnostic tasks and approximate source-model mismatch. The realizable assumption (H1) may fail when the source checkpoints do not span the relevant task optima, or when the residual budget k is too small. In this case we define the best approximation within $\mathcal{H}_{R,k}$,

$$\theta_T^\dagger \;:=\; \arg\min_{\theta\in\mathcal{H}_{R,k}} \ell_T(\theta), \qquad \mathrm{App}(T) \;:=\; \ell_T(\theta_T^\dagger) - \inf_{\theta\in\mathbb{R}^d} \ell_T(\theta),$$

and we bound the excess risk relative to the unconstrained optimum by the sum of estimation and approximation:

$$\mathbb{E}\big[\ell_T(\widehat{\theta}_T)\big] - \inf_{\theta} \ell_T(\theta) \;\le\; \mathrm{App}(T) \;+\; C\,\frac{R + k\log(d/k)}{n}. \tag{13}$$

Equivalently, if one has a quantitative mismatch in parameter space, e.g. there exist (w, Δ) with $\|\Delta\|_0 \le k$ such that $\|\theta_T^{\mathrm{opt}} - (\sum_r w_r \theta^{(r)} + \Delta)\|_2 \le \delta$, then smoothness implies $\mathrm{App}(T) \le \frac{L}{2}\delta^2$, so the penalty for imperfect source coverage appears as an additive $O(L\delta^2)$ term on top of the statistical rate.

The bounds above isolate the statistical benefits of multi-source transfer (a low-dimensional simplex) and the cost of specialization (a sparse residual). In the next section we show that these rates are minimax-tight and that exact optimization over more stringent sparsity constraints is computationally intractable in general, motivating the relaxations and amortization mechanisms used in practice.

8 Theory II (Lower Bounds and Hardness): Minimax Optimality and Computational Intractability

We now justify that the upper rate obtained for the oracle program is not an artifact of our analysis but rather reflects the intrinsic statistical difficulty of simultaneously (i) identifying a mixture over R pretrained sources and (ii) fitting a k-sparse specialization. We then complement this statistical perspective with computational hardness results showing that more stringent combinatorial routing objectives are intractable in general, thereby motivating the relaxations and amortization mechanisms used in practice.

Minimax lower bound for simplex mixing plus sparse residuals. Fix a task T and consider the family of realizable parameters

$$\mathcal{F}_{R,k} \;:=\; \Big\{\theta^\star :\; \theta^\star = \textstyle\sum_{r=1}^{R} w_r \theta^{(r)} + \Delta,\; w \in \Delta_R,\; \|\Delta\|_0 \le k\Big\}.$$

Under the same bounded-noise assumptions used to establish concentration for the upper bound (e.g., sub-Gaussian features and noise for a generalized linear model or ridge-regularized squared loss), we claim that any estimator $\widetilde{\theta}(S_T)$ suffers worst-case expected excess risk at least on the order of

$$\inf_{\widetilde{\theta}} \sup_{\theta^\star \in \mathcal{F}_{R,k}} \mathbb{E}\big[\ell_T(\widetilde{\theta}) - \ell_T(\theta^\star)\big] \;\gtrsim\; \frac{R + k\log(d/k)}{n}, \tag{14}$$

which matches Theorem 2. The key point is that the class $\mathcal{F}_{R,k}$ contains two statistically independent degrees of freedom: an $(R-1)$-dimensional simplex

component and a combinatorial sparse-support component of size $\binom{d}{k}$. Since the lower bound is additive in these contributions, there is no estimator that can uniformly beat the sum rate.

Proof strategy (packing and information bounds). We outline a standard route via Fano's inequality (or, equivalently, a multi-hypothesis Le Cam argument). We construct a finite subset $\{\theta^{(j)}\}_{j=1}^{N} \subset \mathcal{F}_{R,k}$ such that (i) parameters are well-separated in ℓ_2 (hence in excess risk by strong convexity), and (ii) the induced distributions over support sets have small pairwise Kullback–Leibler divergence. The packing is built as a Cartesian product of two packings:

$$\theta^{(j)} \;=\; \Theta w^{(j)} \;+\; \Delta^{(j)}, \qquad \Theta := [\theta^{(1)} \;\cdots\; \theta^{(R)}],$$

where $\{w^{(j)}\}$ is an ε-packing of Δ_R in ℓ_2 (or ℓ_1), and $\{\Delta^{(j)}\}$ is a k-sparse packing obtained by choosing supports among $\binom{d}{k}$ possibilities and assigning $\pm a$ signs on the active coordinates. The packing size satisfies

$$\log N \;\gtrsim\; (R-1)\log(1/\varepsilon) \;+\; k\log(d/k),$$

while strong convexity yields, for a suitable choice of amplitude a and packing resolution, that $\|\theta^{(i)} - \theta^{(j)}\|_2^2$ is proportional to the desired separation in excess risk. On the other hand, for common convex surrogate models, the KL divergence between the distributions induced by two parameters scales as n times a quadratic form in $\theta^{(i)} - \theta^{(j)}$ (e.g., $n\|\theta^{(i)} - \theta^{(j)}\|_2^2$ up to feature-covariance factors in linear regression). Choosing the separation small enough ensures $\mathrm{KL}(P_{\theta^{(i)}} \| P_{\theta^{(j)}}) \lesssim \log N$, which is the regime where Fano implies a constant probability of identification error. Translating this identification error back into excess risk via μ-strong convexity yields the rate (14).

Interpretation: why the rate decomposes as $R + k\log(d/k)$. The lower bound formalizes the intuition that the learner must pay at least (i) a parametric price for discovering an $(R-1)$-dimensional mixture, and (ii) a combinatorial price for discovering a sparse support. The term R/n persists even when $k = 0$, corresponding to estimation over the simplex (or, equivalently, a low-dimensional convex set in parameter space). The term $k\log(d/k)/n$ persists even when $R = 1$ (i.e., when there is only a single pretrained initialization), corresponding to classical sparse regression difficulty. Thus, absent further structure on the sources (e.g., orthogonality, known supports, or additional supervision on w_T), the additive form of the upper bound cannot be improved in general.

Hardness of sparse source subset selection. Statistical optimality does not imply computational tractability. A natural alternative to dense

simplex mixing is to restrict w to be k_s-sparse, i.e. to select at most k_s sources. Even in the simplest quadratic setting with $\Delta \equiv 0$ and

$$\widehat{\ell}_T(\theta) \;=\; \|A\theta - b\|_2^2,$$

the optimization

$$\min_{w \in \Delta_R,\ \|w\|_0 \le k_s} \ \widehat{\ell}_T\Big(\sum_{r=1}^{R} w_r \theta^{(r)}\Big) \tag{15}$$

is NP-hard (Theorem 5). The reduction is from subset selection (sparse regression): one encodes candidate columns as source parameters so that choosing a subset of sources corresponds to choosing a subset of regressors. The simplex constraint may be enforced by augmenting the construction (e.g., adding a bias coordinate and scaling) so that feasible w correspond to convex combinations without changing the combinatorial core of the problem. Consequently, exact best-k_s routing over sources is intractable in worst case, and one should not expect polynomial-time algorithms that solve (15) globally for arbitrary instances.

Hardness of exact ℓ_0 residual optimization. A second intractability arises from the residual itself. Setting $R = 1$ and $\theta^{(1)} = 0$ reduces the oracle program to

$$\min_{\|\Delta\|_0 \le k} \ \widehat{\ell}_T(\Delta),$$

which includes ℓ_0-constrained least squares as a special case and is NP-hard by classical reductions. Therefore, even if the mixture weights were known, globally optimizing the sparse specialization is computationally intractable in general. This motivates the use of relaxations (e.g., hard-concrete gates, ℓ_1 or group-lasso surrogates) and iterative heuristics (e.g., hard-thresholding schemes) in practical implementations.

Implications for approximation and our algorithmic choices. The hardness results delineate the role of the convex-hull relaxation $w \in \Delta_R$: by allowing dense mixtures, we obtain a tractable convex domain for the mixing part (and, in smooth settings, Frank–Wolfe provides an efficient solver with sparse iterates as in Theorem 3). The residual component remains nonconvex under ℓ_0 constraints, but approximate solvers can be deployed with explicit budget control and predictable anytime behavior. From a systems standpoint, these considerations suggest that (i) dense simplex mixing can be used as the primary routing primitive, with optional sparsification via the inherent sparsity of Frank–Wolfe iterates, and (ii) specialization should be enforced through structured sparsity mechanisms amenable to masked computation rather than through exact combinatorial search. In the next section we quantify the resulting time and memory costs for oracle solvers

versus amortized routers and discuss how structured partitions affect deployment latency.

9 Complexity and Systems Considerations

Two cost models: oracle per-task optimization versus amortized routing. We distinguish (i) an *oracle* regime in which, given a support set S_T, we solve (approximately) the constrained empirical risk minimization over (w, Δ), and (ii) an *amortized* regime in which a meta-trained router outputs w_T (and optionally residual expert weights) in one forward pass over S_T. The salient systems question is how the marginal cost *per new task* scales with (n, R, d, k, L), and how this interacts with deployment-time latency budgets.

Oracle complexity: dependence on n and on the representation of sources. In the convex surrogate regime where $\widehat{\ell}_T$ is evaluated on frozen features, a direct implementation of the oracle program

$$\min_{w \in \Delta_R,\ \|\Delta\|_0 \le k} \widehat{\ell}_T(\Theta w + \Delta), \qquad \Theta = [\theta^{(1)} \ \cdots \ \theta^{(R)}] \in \mathbb{R}^{d \times R},$$

admits two natural representations. In a *parameter-centric* view, each iteration requires forming Θw (cost $O(Rd)$) plus evaluating gradients of $\widehat{\ell}_T$ on n samples (cost $O(nd)$), leading to a nominal per-iteration cost $O(nd + Rd)$. In a *sufficient-statistics* view (e.g., squared loss or generalized linear models), one can precompute task-specific statistics from S_T (cost $O(nd)$) and then evaluate $\widehat{\ell}_T(\Theta w + \Delta)$ and its gradients with respect to (w, Δ) without scanning all n samples, reducing subsequent iterations to a dependence on (R, d) alone. This distinction matters primarily when n is not tiny relative to (R, d), and it clarifies that the few-shot setting (n small) is generally dominated by feature extraction rather than by optimization over w.

When we solve for w over the simplex using Frank–Wolfe, each iteration requires a linear minimization oracle over Δ_R, which reduces to selecting the best vertex e_r under the current gradient; in the parameter-centric view this selection can cost $O(Rd)$ if one evaluates $\langle \nabla \widehat{\ell}_T(\Theta w + \Delta), \theta^{(r)} \rangle$ naively for all r. However, if $\theta^{(r)}$ are stored as deltas from a common base (or are blockwise addressable), the relevant inner products can be computed at lower overhead, and the number of nonzeros in the Frank–Wolfe iterate remains at most the iteration count (Theorem 3), yielding an explicit latency–accuracy trade-off.

Residual sparsity: masked updates and structured partitions. The residual variable Δ is the primary handle for controlling specialization cost. In unstructured sparsity, we store Δ as (index, value) pairs with $\mathrm{nnz}(\Delta) \le k$, and apply it as a sparse add; the incremental memory is $O(k)$ and the

incremental compute is proportional to $\mathrm{nnz}(\Delta)$ for those operations that can exploit sparsity (e.g., embedding updates, MLP weight updates under sparse kernels). In structured sparsity, we fix a partition of parameters into L blocks (layers, attention heads, MLP submodules, or tensor shards) and constrain Δ to be block-k sparse. This makes deployment costs substantially more predictable: if a block is inactive, we can skip loading and applying its delta entirely, while if it is active, the block is typically dense and amenable to optimized dense kernels. Thus structured sparsity replaces a theoretically sharper $k\log(d/k)$ combinatorial term with an engineering-friendly *module budget* that aligns with hardware execution.

For blockwise mixing over sources, we similarly allow weights $w_{T,\ell} \in \Delta_R$ per block $\ell \in [L]$, yielding

$$\theta_T \;=\; \big(\theta_{T,1},\ldots,\theta_{T,L}\big), \qquad \theta_{T,\ell} \;=\; \sum_{r=1}^{R} w_{T,\ell,r}\,\theta_\ell^{(r)} \;+\; \Delta_{T,\ell}.$$

This increases routing dimension from $R-1$ to $L(R-1)$, but it has two systems benefits: (i) we can form parameters *on demand* per module (streaming the required blocks), and (ii) we can define latency budgets per module in a way that matches transformer execution (attention and MLP dominate, while normalization and residual adds are typically negligible).

Latency and energy proxies as explicit constraints. To connect sparsity and routing to deployment budgets, we introduce a proxy $\widehat{\mathrm{cost}}(\theta_T)$ intended to correlate with wall-clock latency, memory bandwidth, or energy. A simple proxy in the block-structured setting is

$$\widehat{\mathrm{cost}}(\theta_T) \;:=\; \sum_{\ell=1}^{L} \Big(c_\ell^{\mathrm{mix}} \cdot \|w_{T,\ell}\|_0 \;+\; c_\ell^{\mathrm{res}} \cdot \mathbf{1}\{\Delta_{T,\ell} \neq 0\} \Big),$$

where c_ℓ^{mix} accounts for reading and combining multiple source blocks and c_ℓ^{res} accounts for loading and applying a residual block. In the unstructured case, one may replace $\mathbf{1}\{\Delta_{T,\ell} \neq 0\}$ by $\mathrm{nnz}(\Delta_{T,\ell})$ when sparse kernels are available. Such proxies can be enforced either by projection (discarding updates that exceed a budget C) or by a Lagrangian penalty $\widehat{\ell}_T(\theta_T) + \lambda\,\widehat{\mathrm{cost}}(\theta_T)$. While the proxy is not a perfect predictor of hardware time, it enables reporting *Pareto* behavior (risk versus cost) and guides the design of routers that explicitly trade accuracy for compute.

Amortized routing: constant overhead beyond feature extraction. In the amortized Multi-Source SMAT regime, the per-task overhead beyond encoding the support set is dominated by: (i) producing $w_T = \mathrm{softmax}(h_\zeta(S_T))$, which is $O(R)$ (or $O(LR)$ in the blockwise case), and (ii) applying a small

number of residual experts via masks, which is $O(\mathrm{nnz}(\Delta_T))$ for unstructured masks or $O(\#\text{active blocks})$ for structured masks. Importantly, amortization avoids iterative evaluation of $\widehat{\ell}_T$ and therefore avoids repeated forward/backward passes on S_T; this is the principal savings when tasks arrive online and must be served under tight latency constraints. The amortized model also admits caching: if multiple queries share a task identity, the computed w_T and active masks can be reused.

Memory footprint and checkpoint storage formats. Storing R full checkpoints of dimension d requires $O(Rd)$ parameters; when R is large, we may store each source as a delta to a base checkpoint θ_{pre} (or to a low-rank code), which reduces disk usage when sources are similar and also simplifies streaming. At inference, one can further reduce memory bandwidth by limiting $\|w_{T,\ell}\|_0$ (implicitly via Frank–Wolfe iterates or explicitly via top-k truncation), so that only a few source blocks must be fetched per module. Residual masks similarly permit loading only the active parameters. These considerations are often more consequential than raw floating-point operation counts, since routing and specialization mainly change *which* weights are read, not the structure of the dominant dense matrix multiplications.

Checkpoint compatibility constraints and alignment. All parameter-space mixing mechanisms require that sources share a common parameterization: identical architecture, tensor shapes, and ordering of parameters. Practical incompatibilities arise from changes in tokenizers/vocabularies, positional encodings, normalization conventions, and merged/unmerged projection matrices. Even with identical shapes, mixing can be degraded by permutation symmetries (e.g., permuting attention heads or MLP neurons across independently trained checkpoints). When such symmetries are present, a pre-alignment step (e.g., matching heads by maximizing correlation of activations on a calibration set, or applying weight-matching heuristics) may be necessary to ensure that convex combinations remain semantically meaningful. Finally, mixed-precision and quantization introduce additional constraints: if sources are stored in quantized form, the system must either dequantize selected blocks to a common compute type or implement mixing directly in the quantized domain, which alters both cost and numerical behavior.

10 Experimental Plan (Strengthening)

We structure the empirical evaluation to test three claims suggested by the model and theory: (i) multi-source simplex mixing improves adaptation relative to any single source when tasks draw from a heterogeneous mixture, (ii) sparse residual specialization is necessary when tasks lie outside the convex

hull of sources (or when source alignment is imperfect), and (iii) the amortized router can approximate the per-task oracle while achieving a favorable risk–cost trade-off under explicit deployment budgets.

Cross-domain few-shot suites. We evaluate on few-shot benchmarks designed to emphasize domain heterogeneity rather than within-domain class variation. Concretely, we assemble suites where each episode samples a domain and then a classification problem within that domain (e.g., natural images, sketches, rendering, satellite, medical; or, for language, topic and genre shifts). We report standard n-shot/q-query episodic metrics (mean accuracy and 95% confidence intervals over episodes) and additionally report per-domain accuracies to diagnose when mixing collapses to a single source. To connect to the theoretical task distribution $\mathcal{P}$, we explicitly define mixtures of in-distribution (ID) domains used for meta-training and held-out out-of-distribution (OOD) domains used only at test time.

Additional OOD and safety shifts. Beyond domain shifts, we introduce shifts intended to stress reliability: label shift (class priors changed at test time), corruption shift (e.g., blur/noise/compression), spurious-correlation shift (synthetic confounders), and prompt/instruction shift for language episodes. For safety-oriented evaluation, we include (when applicable) harmlessness/refusal and toxicity tasks with few-shot supervision, and we measure not only task accuracy but also constraint-violation rates under fixed prompting/decoding protocols. Our goal is not to claim safety from routing alone, but to test whether routing concentrates on safer sources under such shifts and whether residual specialization increases risk of constraint violations. Accordingly, we report (a) performance on the main task metric and (b) auxiliary safety metrics, both as a function of the residual budget k (or block budget) and the mixing sparsity induced by routing.

Source scaling and diversity ablations. We perform controlled ablations over the number of sources R and their diversity. Holding total pretraining compute approximately fixed, we compare: (i) many moderately specialized sources versus (ii) a few highly specialized sources, and we measure how performance and routing entropy change with R. To isolate the value of diversity, we construct groups of sources that are (a) near-duplicates (different seeds), (b) same domain but different objectives, and (c) different domains/objectives. We then quantify gains from mixing by comparing against the best single source chosen with access to S_T (an oracle selection baseline) and by analyzing the learned weights w_T (entropy, effective support size, and stability across resampled support sets). Where feasible, we also evaluate sensitivity to source misalignment by applying controlled permutations within symmetric substructures (e.g., attention head permutation) and

measuring degradation with and without any alignment heuristic.

Structured versus unstructured masks. We compare three specialization mechanisms at matched parameter budgets: (i) unstructured k-sparse residuals, (ii) block-k residuals under a fixed partition into L modules, and (iii) no residual ($k = 0$). We evaluate not only accuracy but also realized efficiency measured by (a) proxy cost $\widehat{\mathrm{cost}}(\theta_T)$ and (b) wall-clock latency on representative hardware, since unstructured sparsity may not translate into speedups without specialized kernels. We additionally report how specialization localizes across modules (which blocks activate) and whether structured sparsity produces more predictable costs across tasks, as suggested by the systems discussion.

Baselines: soups, arithmetic, and routing-free compositions. We include parameter-space baselines that are natural competitors to our mixing formulation. First, we compare to *model soups* (uniform or validation-weighted averages of checkpoints) and to fixed arithmetic compositions (e.g., $\theta_{\mathrm{pre}} + \sum_r a_r(\theta^{(r)} - \theta_{\mathrm{pre}})$ with fixed a_r), which do not condition on S_T. Second, we compare to per-task *single-source* adaptation baselines (fine-tuning from the best source chosen by support loss) under identical residual budgets. Third, we include a *mixture without residual* baseline (simplex mixing only), which isolates the contribution of Δ_T. For completeness, we compare amortized routing to the per-task oracle solver (approximate Frank–Wolfe plus a sparsification routine) to estimate the amortization gap in practice, aligning with the role of Theorem 4.

Pareto fronts under latency budgets. We treat deployment constraints as first-class and report Pareto curves of query risk (or accuracy) versus cost. Operationally, we sweep (i) the residual budget k (or the number of active blocks), (ii) any imposed sparsity on w_T (e.g., top-k_s truncation or limiting Frank–Wolfe iterations), and (iii) the router architecture capacity. For each setting we report $\big(\mathrm{metric}, \widehat{\mathrm{cost}}, \mathrm{latency}\big)$ triplets and plot Pareto fronts per benchmark and per OOD shift. We also evaluate explicit budget enforcement by solving $\min \widehat{\ell}_T(\theta_T)$ subject to $\widehat{\mathrm{cost}}(\theta_T) \le C$ (or using a Lagrangian) and measuring constraint satisfaction rates. This protocol clarifies whether improvements come from better use of a fixed budget or simply from spending more compute.

Implementation details and robustness checks. To ensure fair comparisons, we standardize: episodic sampling, support/query sizes, optimizer settings for any per-task procedures, and the total number of meta-training episodes. We report variability across random seeds and across independently sampled task sets. We further include robustness checks: (i) varying

n to test the few-shot dependence, (ii) degrading support labels with controlled noise to test sensitivity of routing, and (iii) stress-testing with mixed ID/OOD task mixtures to evaluate whether routing overfits to meta-training domains. Finally, we provide qualitative diagnostics (e.g., nearest-source behavior, weight trajectories, and module activation patterns) to make failure modes explicit, particularly in cases where mixing produces unstable or high-variance solutions.

11 Discussion and Future Directions

Our formulation isolates two mechanisms for per-task adaptation: (i) selecting a point in the convex hull of pretrained sources via $w_T \in \Delta_R$, and (ii) applying a sparse specialization Δ_T outside that hull. This separation is conceptually useful, but it also exposes several directions where the model class, the routing mechanism, and the associated guarantees can be strengthened.

Multi-modal and cross-modal source collections. A natural extension is to treat the sources $\{\theta^{(r)}\}_{r=1}^{R}$ as a heterogeneous library spanning modalities and interfaces (e.g., vision encoders, text encoders, multimodal fusion modules, and decoders). In such settings, one may not have a single shared parameterization across sources, so the literal mixing map $\theta \mapsto \sum_r w_r \theta^{(r)}$ requires modification. One approach is to define mixing on a shared subspace: for instance, restrict mixing to a subset of aligned modules (a common text backbone) while treating modality-specific components as fixed and selected (rather than mixed) at the level of discrete routing. Another approach is to define a learned alignment operator $A^{(r)}$ so that mixing occurs in an aligned coordinate system, e.g., $\theta_T = \sum_r w_{T,r} A^{(r)} \theta^{(r)} + \Delta_T$, where $A^{(r)}$ is block-diagonal and acts within permutation-symmetric structures (attention heads, MLP channels). This suggests a combined estimation problem over (w_T, Δ_T) and alignment parameters, where identifiability becomes nontrivial: multiple decompositions may yield indistinguishable predictors. Developing conditions under which w_T is interpretable (or at least stable) is an open question, particularly when the sources share function but differ by internal symmetries.

Privacy-preserving and edge-constrained routing. A second direction concerns deployment regimes where the support set S_T is sensitive and must remain on-device, or where compute/energy constraints require routing decisions with minimal overhead. Our amortized router naturally supports this setting because it can be implemented as a small encoder h_ζ applied to S_T, producing w_T (and possibly α_T) without backpropagating through large source models. One can further reduce leakage by transmitting only low-dimensional routing outputs, i.e., communicate w_T (or a small subset of

active sources) rather than any examples or gradients. This raises questions about privacy guarantees: even a weight vector w_T can leak task/domain information. It is therefore of interest to study differentially private routing, where $h_\zeta(S_T)$ is perturbed to satisfy (ε, δ)-DP, and to quantify the induced increase in risk relative to the oracle. In addition, edge constraints motivate explicit cost-aware objectives, e.g., minimizing $\widehat{\ell}_T(\theta_T) + \lambda\, \widehat{\mathrm{cost}}(\theta_T)$, and studying the resulting trade-offs between concentration of w_T (few active sources), structured sparsity of Δ_T, and achieved accuracy.

Uncertainty-aware mixing and abstention. Our current estimator outputs a single w_T and Δ_T, implicitly committing to a point estimate. In regimes with small n or ambiguous tasks, it is often preferable to represent uncertainty over routing. A simple mechanism is to interpret w_T as parameters of a Dirichlet distribution and perform either (i) posterior sampling of w (and possibly sparse supports of Δ) or (ii) risk-averse selection using a robust objective over a plausible set of weights. This connects to distributionally robust optimization: one may select θ_T minimizing the worst-case empirical risk over an ℓ_1-ball around w_T, thereby hedging against routing error (cf. the amortization gap perspective). Similarly, uncertainty can be used for selective prediction: if the inferred routing distribution is diffuse (high entropy) or the support loss landscape is flat, the system may abstain, request more labels, or fall back to a safer source. Providing formal guarantees of coverage or constraint satisfaction (e.g., via conformal prediction on top of routed models) is largely open, especially when Δ_T is enabled and can increase variance or enable spurious shortcuts.

Continual addition and removal of sources. In many applications the set of available checkpoints is not fixed: new sources are added over time (new domains, improved models), and old sources may be removed for licensing, privacy, or deprecation. This suggests studying dynamic R and maintaining a router that generalizes to unseen sources. A practical desideratum is modularity: adding a new $\theta^{(R+1)}$ should not require full retraining. One can view this as a meta-learning problem with a changing candidate set, where the router must embed sources (e.g., via source descriptors or probing features) and then compute w_T by matching S_T to sources in an embedding space. On the theoretical side, the statistical complexity in Theorem-type bounds depends on R, but in a continually growing library one would like rates in terms of an *effective* number of distinguishable sources (e.g., covering numbers of the set $\{\theta^{(r)}\}$) rather than raw R. Another open point is stability under removal: if a source is excised, we would like guarantees that performance degrades gracefully, perhaps by re-optimizing w_T within the reduced simplex while keeping Δ_T within the same budget.

Open problems: computation, structure, and nonconvexity. Several unresolved issues lie at the interface of optimization and modeling. First, our hardness results motivate relaxations, but we do not yet have sharp characterizations of when practical heuristics (Frank–Wolfe on w and thresholding on Δ) recover near-oracle solutions for realistic feature distributions. Second, the choice of sparsity structure remains delicate: unstructured ℓ_0 sparsity is statistically attractive but systemically expensive, whereas block sparsity is implementable but may be too coarse. Understanding the optimal partition into L modules—and whether it can be learned jointly with routing—is an important direction. Third, the convex surrogate theory provides clean rates, yet the transformer regime is nonconvex and exhibits additional symmetries; extending excess-risk guarantees to parameter-space mixing in deep networks likely requires new stability arguments, perhaps based on local linearization around pretrained checkpoints or on function-space metrics that avoid parameter non-identifiability. Finally, we note a modeling question: the decomposition $\theta_T^\star = \sum_r w_{T,r}^\star \theta^{(r)} + \Delta_T^\star$ implicitly assumes that "source knowledge" is additively composable in parameter space. Determining when this approximation is valid, and when one should instead mix in representation space or through conditional computation, remains central to understanding the limits of multi-source routing.