Curvature-Weighted Capacity Allocation: A Minimum Description Length Framework for Layer-Adaptive Large Language Model Optimization

Theophilus Amaefuna[*1] **Hitesh Vaidya**[*1] **Anshuman Chhabra**[1] **Ankur Mali**[1]

[1]Bellini College of Artificial Intelligence, Cybersecurity and Computing
University of South Florida, Tampa, Florida, USA

Abstract

Layer-wise capacity in large language models is highly non-uniform: some layers contribute disproportionately to loss reduction, whereas others are nearly redundant. Existing layer-scoring methods provide sensitivity estimates but do not give a principled rule for converting those estimates into allocation or pruning decisions under a global hardware budget. We introduce a curvature-aware, MDL-inspired framework built around the layer gain $\zeta_k^2 = g_k^\top \widetilde{H}_{kk}^{-1} g_k$. This quantity equals twice the maximal decrease predicted by the regularized layer-restricted quadratic model and incorporates inverse local curvature; it is therefore a local surrogate for reducible risk, not a universal dominance claim over gradient-norm scores. After normalizing the gains into scores q_k, we formulate two convex programs: one allocates expert slots under diminishing returns, and the other assigns layer-wise pruning ratios while protecting high-score layers. Both continuous programs have unique globally optimal solutions characterized by one dual variable and computable in $O(K \log(1/\varepsilon))$ time by bisection. We also prove a quadratic transfer-regret bound: when source and target score vectors differ by at most δ, the target surrogate cost of the transferred decision is within $O(\delta^2)$ of the target optimum. Experiments on Mistral-7B and Gemma-7B show clear allocation gains in some settings and competitive, though mixed, pruning performance. The framework therefore replaces an empirical score-to-decision heuristic with a budget-feasible optimization procedure whose guarantees apply to the stated continuous surrogates.
Code is available on github repo - TKAI-LAB-Mali/Curvature-Weighted-Capacity-Allocation

1 INTRODUCTION

The representational capacity of a neural network is not uniformly distributed across its layers.Empirical studies of large language models (LLMs) consistently reveal that layers differ substantially in their contribution to the training objective: some layers hold the bulk of the model's expressive power, while others are near-redundant, contributing little to no loss reduction [Zhang et al., 2022, Vitel and Chhabra, 2026]. This non-uniformity has two practical consequences that current practice addresses only in isolation. On one hand, *capacity bottlenecks*: layers where representational power is insufficient, limit model performance even when global parameter counts are large. On the other hand, *capacity redundancy*: layers where parameters contribute negligibly to the objective, inflates model complexity without commensurate benefit.

As models scale into the hundreds of billions of parameters, both problems are exacerbated by the underlying physical reality: hardware constraints impose strict limits on memory, compute, and communication bandwidth. The central challenge, therefore, is not simply to make models larger or smaller, but to *allocate capacity where it matters and remove it where it does not*, within a global resource budget.

The missing ingredient: curvature. Existing approaches to layer importance estimation rely primarily on gradient magnitudes, activation statistics, or held-out accuracy drops [Askari et al., 2025, Song et al., 2024, Liu et al., 2021]. These signals share a common limitation: they do not account for the local curvature of the loss landscape. A layer may exhibit a large gradient norm yet reside in a region of high curvature, where the actual achievable loss reduction per unit of capacity is small. Conversely, a layer with a moderate gradient in a flat curvature region may offer substantial reducible risk. Without curvature information, capacity decisions are systematically misallocated.

This work. We develop a unified, curvature-aware framework for simultaneously allocating and pruning model capacity across layers under a global resource constraint. Our

[*]Equal contribution.

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:62–86.

central quantity is the *curvature-adjusted layer gain*:

$$\zeta_k^2 \;=\; g_k^\top \widetilde{H}_{kk}^{-1} g_k,$$

where g_k is the layer-k gradient and $\widetilde{H}_{kk}$ is a positive-definite surrogate for the layer-restricted Hessian block. We show that $\zeta_k^2/2$ is the maximal decrease of the regularized layer-restricted quadratic model. Appendix B.3 bounds its approximation error for the nonlinear empirical objective; for layers k and ℓ, the induced ordering is certified whenever $\frac{1}{2}|\zeta_k^2 - \zeta_\ell^2| > \varepsilon_k + \varepsilon_\ell$, where ε_k and ε_ℓ bound their respective approximation errors. With the normalized scores $q_k = \zeta_k^2 / \sum_j \zeta_j^2$, we then derive two complementary convex programs:

- **Capacity allocation** (Section 2.3): given a global hardware budget B, distribute additional capacity (e.g. mixture-of-experts slots) preferentially to high-q_k layers, with diminishing log-returns penalizing over-allocation. The program admits a closed-form *curvature-weighted water-filling* solution, computed in $O(K \log 1/\varepsilon)$ via bisection.
- **Capacity pruning** (Section 3.1): given a global sparsity target S, remove parameters aggressively from the low-q_k layers while protecting the high-gain layers from degradation. The program is strongly convex with a unique closed-form minimizer, again computed by bisection.

We further analyze **transfer stability** (Appendix C): when curvature scores drift between a source domain and a target domain by $\|q^{(A)} - q^{(B)}\|_2 \le \delta$, the excess cost of using source-derived allocations on the target task is bounded by $O(\delta^2)$, with explicit constants given by the condition number of the target program. This justifies warm-starting allocation and pruning decisions from source-domain curvature estimates, a practically important property for fine-tuning and domain adaptation.

Connections to information theory. Our objective functions are motivated by *Minimum Description Length* (MDL) [Rissanen, 1978, 1989]: model complexity is penalized by description length, while data fit is rewarded by a concave utility reflecting diminishing returns in code-length reduction [Lotfi et al., 2023]. This perspective connects our framework to compression-based generalization bounds, where shorter valid descriptions can yield tighter generalization bounds [Wilson, 2025, Schmidhuber, 1997]. The concrete objectives below are MDL-inspired surrogates rather than literal prefix-code lengths. Unless the empirical loss is a negative log-likelihood and the logarithm base and sample-size conversion are specified, $\zeta_k^2/2$ is measured in units of average loss, not bits.

Contributions. We summarize our contributions as follows.

1. **Curvature-adjusted layer gain.** We derive ζ_k^2 from first principles as twice the maximal second-order objective decrease attributable to layer k, and characterize the approximation error introduced by Tikhonov regularization of the Hessian block (Lemma 1, Appendix B.3).
2. **Curvature-weighted water-filling.** We formulate and solve in closed form a convex allocation program that distributes capacity according to q_k under diminishing returns and a global hardware budget (Theorem 2).
3. **Curvature-protected pruning.** We formulate and solve in closed form a strongly convex pruning program that concentrates sparsity on low-gain layers while meeting a global sparsity target (Theorem 3).
4. **Transfer stability.** We prove an $O(\delta^2)$ transfer regret bound under score drift, with explicit constants tied to the condition number and score-gradient Lipschitz constant of the target program (Theorem 4).
5. **Efficient algorithms.** We provide $O(K \log 1/\varepsilon)$ bisection algorithms for both programs, with explicit bisection brackets and compatibility with standard Hessian approximations (Algorithms 1 and 2).

2 BACKGROUND

Minimum Description Length. The Minimum Description Length (MDL) principle [Rissanen, 1978] formalizes the tradeoff between model complexity and data fit: the best model is the one that minimizes the total codelength required to describe both the model and the data under the model,

$$\mathrm{Cl}(D) \;=\; \underbrace{\mathrm{Cl}(\theta)}_{\text{model complexity}} \;+\; \underbrace{\mathrm{Cl}(D \mid \theta)}_{\text{data fit}}, \qquad (1)$$

where $\mathrm{Cl}(\theta)$ is the codelength (in bits) required to describe the parameters θ, and $\mathrm{Cl}(D \mid \theta)$ is the codelength required to describe the data given the model. Under this principle, a model with lower $\mathrm{Cl}(D)$ simultaneously achieves better compression and generalization, as well as lower unnecessary complexity [Lotfi et al., 2023, Prada et al., 2025].

MDL for capacity allocation. Adding capacity at layer k (e.g. adding mixture-of-experts slots by $e_k \ge 0$) increases model codelength by $\mathrm{Cl}(\theta_{e_k})$ while reducing data-fit codelength by $\Delta_{e_k}\mathrm{Cl}(D \mid \theta)$ [Blum and Langford, 2003]. Since the base terms $\mathrm{Cl}(\theta)$ and $\mathrm{Cl}(D \mid \theta)$ are constant with respect to the allocation decision, minimizing the total codelength Eq. (1) over $e = (e_1, \ldots, e_K)$ reduces to,

$$\min_{e_k \ge 0} \sum_{k=1}^{K} \Big[\mathrm{Cl}(\theta_{e_k}) - \Delta_{e_k}\mathrm{Cl}(D \mid \theta) \Big]. \qquad (2)$$

The first term penalizes complexity growth; the second rewards data-fit improvement. We instantiate Eq. (2) concretely in Section 2.3 by modeling $\mathrm{Cl}(\theta_{e_k}) \propto \alpha c_k e_k$ (linear in resource usage) and $\Delta_{e_k}\mathrm{Cl}(D \mid \theta) \propto \gamma q_k^\beta \log(1 + e_k)$ (concave, reflecting diminishing returns in code-length reduction).These proportional models specify a tractable

MDL-inspired objective; they are not derived as exact codelength identities. Consequently, the optimization results below establish optimality for the stated surrogate, not universal MDL optimality over all possible codes.

MDL for pruning. Pruning layer k by sparsity ratio $\rho_k \in [0,1]$ reduces the model codelength by $-\Delta_{\rho_k}\mathrm{Cl}(\theta)$ but increases data-fit codelength by $\Delta_{\rho_k}\mathrm{Cl}(D \mid \theta)$ [Frankle and Carbin, 2019]. Minimizing total codelength over $\rho_k = (\rho_1, \ldots, \rho_K)$ subject to a global sparsity target S

$$\min_{0\le\rho_k\le 1} \sum_{k=1}^{K} \Big[\Delta_{\rho_k}\mathrm{Cl}(\theta) + \Delta_{\rho_k}\mathrm{Cl}(D \mid \theta)\Big]. \qquad (3)$$

We instantiate Eq. (3) in Section 3.1 by modeling $\Delta_{\rho_k}\mathrm{Cl}(\theta) = -b\, n_k \rho_k$ (bits saved by removing $n_k\rho_k$ parameters) and $\Delta_{\rho_k}\mathrm{Cl}(D \mid \theta) = \eta\, q_k^{\kappa}\rho_k^2$ (convex degradation penalty, weighted by layer quality q_k). The quadratic degradation term is likewise a modeling surrogate. Its empirical adequacy is separate from the convexity and closed-form analysis of the resulting program.

Layer quality via second-order information. Both programs Eq. (2) and Eq. (3) depend on layer quality scores q_k that measure how much each layer contributes to reducible empirical risk. A natural candidate is the *Newton decrement* restricted to layer k: given gradient $g_k = E_k \nabla L(\theta)$ and positive-definite Hessian surrogate $\widetilde{H}_{kk} = E_k \nabla^2 L(\theta) E_k^\top + \tau I$, the quantity ζ_k^2 is twice the maximal second-order decrease in L achievable by updating layer k alone (derived in Section 2.2). We use normalized scores: $q_k = \zeta_k^2 \Big/ \sum_{j=1}^{K} \zeta_j^2$, throughout, ensuring scale invariance with respect to the global curvature magnitude. Layers with large q_k carry more reducible risk and should receive more capacity; layers with small q_k are candidates for pruning.

Layer Influence scores and their limitations. The closest prior measure to ζ_k^2 is the *Layer Influence score* (LayerIF) of Askari et al. [2025], which localizes the classical influence function [Koh and Liang, 2017] to individual layers. Given training data $D^{\mathrm{train}} = \{x_i\}_{i=1}^m$ and validation data $D^{\mathrm{valid}} = \{x_j^\nu\}_{j=1}^n$, the global influence score of training sample x_i is

$$I(x_i) \;=\; -\sum_{j=1}^{n} \nabla_\theta \ell(x_j^\nu, \theta)^\top H(\theta)^{-1} \nabla_\theta \ell(x_i, \theta), \qquad (4)$$

measuring the sensitivity of validation loss to upweighting x_i. The LayerIF score restricts Eq. (4) to layer k by replacing full-model quantities with their layer-k counterparts:

$$I^{(k)}(x_i) \;= \\ -\sum_{j=1}^{n} \nabla_{\theta^{(k)}} \ell(x_j^\nu, \theta)^\top \big(H^{(k)}(\theta)\big)^{-1} \cdot \nabla_{\theta^{(k)}} \ell(x_i, \theta). \qquad (5)$$

A large $|I^{(k)}(x_i)|$ indicates that layer k is sensitive to the training sample x_i, and Askari et al. [2025] used the aggregated magnitude as a proxy for layer quality to guide expert allocation and pruning. LayerIF combines training and validation gradients through an inverse-Hessian operator. Its magnitude therefore reflects gradient alignment and inverse curvature and cannot, by itself, be identified with high curvature. The gain $\zeta_k^2/2$ instead has an exact interpretation as the decrease of the regularized quadratic surrogate. When L is an average negative log-likelihood measured in nats, the corresponding local total-codelength scale is $n\zeta_k^2/(2\log 2)$ bits, up to damping and Taylor-remainder errors.

While LayerIF captures data-dependent sensitivity, it has two structural limitations that motivate our approach. ***First***, $I^{(k)}$ depends on individual training samples and must be aggregated heuristically into a layer-level score; in contrast, ζ_k^2 is defined directly on the empirical objective and has a closed-form interpretation as reducible risk. ***Second***, and more importantly, LayerIF provides a *signal* but not an *objective*: translating $\{I^{(k)}\}$ into concrete expert counts or pruning ratios requires a separate heuristic (a knapsack assignment with first-come-first-served residual allocation [Askari et al., 2025]), with no budget constraint and no optimality guarantee. Our MDL-inspired programs Eq. (2) and Eq. (3) replace this two-stage heuristic with a single convex program that jointly determines the allocation of all layers under a global resource constraint, with closed-form solutions and provable optimality.

We study the problem of allocating limited model capacity across layers to maximize locally reducible empirical risk under a global resource constraint. The method consists of two components: (i) a curvature-aware measure of layer-wise reducible risk derived from a second-order expansion of the training objective, and (ii) a convex allocation program that distributes capacity according to these gains under diminishing returns. See Appendix A for related works.

2.1 OBJECTIVE AND NOTATION

Let: $L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \varphi(f(x_i;\theta),\, y_i)$, denote the empirical objective, where φ is a per-sample loss and $f(\cdot\,;\theta)$ is the model. We write the gradient and Hessian at $\theta \in \mathbb{R}^p$ as: $g := \nabla_\theta L(\theta)$, $H := \nabla_\theta^2 L(\theta)$. Partition θ into K disjoint layer blocks $\theta = (\theta_1, \ldots, \theta_K)$, where $\theta_k \in \mathbb{R}^{p_k}$ and $\sum_k p_k = p$. For each layer k, let $E_k \in \{0,1\}^{p_k \times p}$ be the coordinate-selection matrix and define: $g_k := E_k g \in \mathbb{R}^{p_k}$, $H_{kk} := E_k H E_k^\top \in \mathbb{R}^{p_k \times p_k}$.

2.2 SECOND-ORDER EXPANSION AND LAYER-RESTRICTED DECREASE

Second-order Taylor expansion. Suppose $\nabla^2 L$ is locally Lipschitz with constant $M > 0$ in a neighborhood of θ. Taylor's theorem with integral remainder gives,

$$L(\theta+\Delta)-L(\theta)=g^\top\Delta+\frac{1}{2}\,\Delta^\top H\Delta+R(\Delta),$$
$$|R(\Delta)|\;\le\;\frac{M}{6}\,\|\Delta\|^3. \quad (6)$$

Layer-restricted quadratic model. Restrict perturbations to layer k by setting $\Delta = E_k^\top d_k$ for $d_k \in \mathbb{R}^{p_k}$. Since $E_k E_k^\top = I_{p_k}$ and the blocks are disjoint, substitution into Eq. (6) yields the layer-restricted quadratic:

$$Q_k(d_k)\;=\;g_k^\top d_k+\frac{1}{2}\,d_k^\top H_{kk}\,d_k.$$

Because neural network Hessians are generally indefinite, minimizing Q_k directly is ill-posed. We introduce the Tikhonov-regularized surrogate:

$$\widetilde{H}_{kk}\;:=\;H_{kk}+\tau I,\qquad \tau>0, \quad (7)$$

and analyze $\widetilde{Q}_k(d_k)=g_k^\top d_k+\frac{1}{2}\,d_k^\top\widetilde{H}_{kk}\,d_k$. This regularization is standard in second-order deep learning [Martens and Grosse, 2015],Botev et al. [2017]. Quadratic damping is the Lagrangian form associated with a trust-region subproblem for a suitable multiplier; an arbitrary fixed τ

Lemma 1 (Layer-restricted optimum). If $\widetilde{H}_{kk}\succ 0$, the unique minimizer of $\widetilde{Q}_k(d_k)$ over $d_k\in\mathbb{R}^{p_k}$ is:

$$d_k^\star\;=\;-\widetilde{H}_{kk}^{-1}\,g_k,$$

and the corresponding decrease equals,

$$\min_{d_k}\widetilde{Q}_k(d_k)\;=\;-\frac{1}{2}\,g_k^\top\widetilde{H}_{kk}^{-1}\,g_k.$$

The proof for lemma 1 is shown in Appendix B.1.

Curvature-adjusted layer gain. Define:

$$\zeta_k^2\;:=\;g_k^\top\widetilde{H}_{kk}^{-1}\,g_k\;\ge\;0. \quad (8)$$

By Lemma 1, $\zeta_k^2/2$ is exactly the maximal decrease predicted by the regularized quadratic model $\widetilde{Q}_k$. For $d_k^\star = -\widetilde{H}_{kk}^{-1}g_k$ and $\Delta_k^\star = E_k^\top d_k^\star$, Taylor's theorem gives,

$$L(\theta+\Delta^\star)-L(\theta)=-\frac{1}{2}\zeta_k^2-\frac{\tau}{2}\|d_k^\star\|_2^2+R(\Delta^\star),$$
$$|R(\Delta^\star)|\le\frac{M}{6}\|d_k^\star\|_2^3.$$

Thus ζ_k^2 is a local surrogate for reducible empirical risk rather than a global characterization of the nonlinear loss landscape. Moreover,

$$\frac{\|g_k\|_2^2}{\lambda_{\max}(\widetilde{H}_{kk})}\le\zeta_k^2\le\frac{\|g_k\|_2^2}{\lambda_{\min}(\widetilde{H}_{kk})},$$

so the score incorporates curvature but need not preserve the ranking induced by $\|g_k\|_2^2$. Appendix B.3 gives the associated approximation and ranking conditions.

2.3 CAPACITY ALLOCATION UNDER DIMINISHING RETURNS

Setup. Let $e_k \ge 0$ denote continuous capacity allocated to layer k (e.g., number of active units or effective precision bits), with per-unit resource cost $c_k > 0$. We impose the global budget constraint

$$\sum_{k=1}^{K} c_k\,e_k\;\le\;B. \quad (9)$$

Define the normalized curvature weights:

$$q_k\;:=\;\frac{\zeta_k^2}{\sum_{j=1}^K\zeta_j^2},\qquad q_k\ge 0,\quad\sum_k q_k=1 \quad (10)$$

This definition assumes $\sum_j\zeta_j^2>0$. When zero or near-zero scores are possible, we use the smoothed simplex weights

$$q_k^{(\varepsilon)}=\frac{\zeta_k^2+\varepsilon_q}{\sum_{j=1}^K\zeta_j^2+K\varepsilon_q},\qquad\varepsilon_q>0,$$

which satisfy $q_k^{(\varepsilon)}>0$ and provide the positive lower bound needed for uniform strong-convexity and score-Lipschitz constants.

Optimization program. We seek an allocation that concentrates capacity where reducible risk is largest, while penalizing over-allocation via diminishing log-returns. Setting $\alpha_k=\alpha c_k$ (so that linear cost scales with resource usage and no layer-dependent free parameter is introduced), we solve:

$$\min_{e_k\ge0}\;\sum_{k=1}^{K}\Big[\alpha c_k\,e_k\;-\;\gamma\,q_k^\beta\phi(e_k)\Big]$$
$$\text{s.t.}\quad\sum_{k=1}^{K}c_k\,e_k\;\le\;B, \quad (11)$$

where $\alpha,\gamma>0$ are scalar hyperparameters, $\beta\ge0$ controls gain emphasis (see below) and $\phi(e_k)=\log(1+e_k)$. The objective is convex: $\alpha c_k e_k$ is linear, and $-\gamma q_k^\beta\log(1+e_k)$ is convex on $e_k\ge0$ since $-\log(1+e)$ is convex. The feasible set is a convex polytope. Slater's condition Boyd and Vandenberghe [2004] holds (any strictly feasible $e_k>0$ with $\sum_k c_k e_k < B$ is a Slater point), so strong duality applies.

Closed-form solution. Forming the Lagrangian:

$$\mathcal{L}(e,\lambda)=\sum_k\Big[\alpha c_k e_k-\gamma q_k^\beta\log(1+e_k)\Big]+\lambda\Big(\sum_k c_k e_k-B\Big),$$

and imposing stationarity: $\partial\mathcal{L}/\partial e_k=0$ for $e_k>0$ gives,

$$\alpha c_k+\lambda c_k-\frac{\gamma q_k^\beta}{1+e_k}=0.$$

Solving and incorporating the non-negativity constraint yields the *curvature-weighted water-filling* allocation,

$$\boxed{e_k(\lambda) \;=\; \max\left\{ \frac{\gamma\, q_k^{\beta}}{(\alpha+\lambda)\, c_k} - 1,\; 0 \right\}.} \qquad (12)$$

If the unconstrained optimum ($\lambda = 0$) violates the budget, the constraint is active and the dual variable $\lambda^\star > 0$ satisfies $\sum_k c_k\, e_k(\lambda^\star) = B$. The map $\lambda \mapsto \sum_k c_k e_k(\lambda)$ is continuous and nonincreasing and is strictly decreasing on every interval containing at least one active coordinate. Strict convexity gives a unique primal minimizer; when the active budget equation is nondegenerate, its dual multiplier is unique and can be computed in $O(K \log(1/\varepsilon))$ time by bisection. A zero-allocation plateau can admit several dual values representing the same primal solution.

Role of β. The exponent β interpolates between three regimes: (i) $\beta = 0$: $q_k^{\beta} = 1$ uniformly, so curvature information is discarded and the allocation depends only on costs c_k; (ii) $\beta = 1$: capacity is allocated proportionally to normalized gains q_k, recovering a natural baseline; (iii) As β increases, relative weight is increasingly placed on the largest scores. With fixed γ and normalized $q_k < 1$, however, all q_k^{β} vanish as $\beta \to \infty$, so the allocation may collapse to zero. Literal concentration on the maximizers requires rescaling γ with β or normalizing the powered weights as $q_k^{\beta} / \sum_j q_j^{\beta}$. In practice, $\beta = 1$ or $\beta = 2$ work well across architectures (Section 3.3).

3 METHOD

In this section, we present our MDL framework for expert allocation and pruning as displayed in Figure 1.

3.1 CAPACITY ALLOCATION AND PRUNING UNDER CURVATURE-WEIGHTED TRADEOFFS

The curvature-adjusted layer gains ζ_k^2 defined in Eq. (8) and derived in Section 2.2, quantify how much empirical risk can be locally reduced by updating layer k alone. We now use these gains to drive two complementary capacity decisions: allocating additional capacity to under-resourced layers, and pruning parameters from layers that contribute little to risk reduction (see Figure 1). In both programs we work with the normalized quality scores introduced in Eq. (10). Normalization ensures that the allocation parameters (γ, β) are invariant to the global scale of the curvature signal, and that q_k can be interpreted directly as the relative share of reducible risk attributable to layer k. A layer with large q_k has high potential to reduce empirical risk; accordingly, it should receive more capacity and be protected from pruning.

Capacity allocation. Given the setup in Section 2.3 and Eq. (9) let e_k denote the effective capacity added at layer k (e.g., LoRA rank or number of mixture-of-experts slots) [Baldi and Vershynin, 2019], setting $\alpha_k = \alpha c_k$ so that linear penalty scales with hardware usage (and no layer-dependent free parameter is introduced), we model the tradeoff between model complexity and risk reduction via the separable convex program in Eq. (11), where $\gamma > 0$ scales benefit strength, $\beta \geq 0$ controls curvature emphasis, and ϕ is increasing and strictly concave with $\phi(0) = 0$. The concavity of ϕ models diminishing returns: each additional unit of capacity yields progressively smaller reductions in empirical risk.

Theorem 2 (Convexity and closed form for allocation). The objective in Eq. (11) is convex. If ϕ is strictly concave and $q_k > 0$ for all k, the objective is strictly convex and the minimizer is unique.

For $\phi(e) = \log(1+e)$, there exists a unique $\lambda^\star \geq 0$ such that the optimal allocation is as shown in Eq. (13).

$$\boxed{e_k^\star \;=\; \max\left\{ \frac{\gamma\, q_k^{\beta}}{(\alpha+\lambda^\star)\, c_k} - 1,\; 0 \right\}.} \qquad (13)$$

When $\lambda^\star > 0$ the budget constraint is active, $\sum_k c_k e_k^\star = B$; when $\lambda^\star = 0$ the unconstrained optimum is feasible. On the active set $\{k : e_k^\star > 0\}$, $e_k^\star$ is strictly increasing in q_k. A detailed proof for this theorem is discussed in Appendix B.2.1

Interpretation. At optimality, the marginal benefit equals the marginal cost at every active layer:

$$\underbrace{\gamma q_k^{\beta}\, \phi'(e_k^\star)}_{\text{marginal benefit}} \;=\; \underbrace{(\alpha+\lambda^\star)\, c_k}_{\text{marginal cost}}.$$

Layers with larger curvature signal q_k receive more capacity, while the global multiplier $\lambda^\star$ enforces the budget. The exponent β interpolates between three regimes: $\beta = 0$ produces a uniform allocation that ignores curvature; $\beta = 1$ allocates proportionally to normalized gains; and as $\beta \to \infty$, allocation concentrates on the layer(s) with the largest ζ_k^2. In practice, $\beta \in \{1, 2\}$ works well across architectures (Section 3.3).

Layer-wise pruning. We now consider the complementary problem of removing parameters from layers that carry little curvature signal. Let $\rho_k \in [0, 1]$ denote the fraction of parameters pruned at layer k, and let n_k denote the total number of parameters in layer k. The number of retained parameters is $n_k(1-\rho_k)$, contributing $b\, n_k(1-\rho_k)$ bits to model size, where $b > 0$ is bits per parameter.

Pruning a fraction ρ_k of layer k degrades data fit. Since ζ_k^2 measures the maximal second-order decrease achievable by updating layer k (Section 2.2), layers with larger $q_k \propto \zeta_k^2$ are more sensitive to parameter removal. We model the resulting degradation as $\eta q_k^{\kappa} \psi(\rho_k)$, where ψ is convex with $\psi(0) = 0$, $\kappa \geq 0$ controls curvature emphasis, and $\eta > 0$ scales the penalty. This is a design choice grounded in the

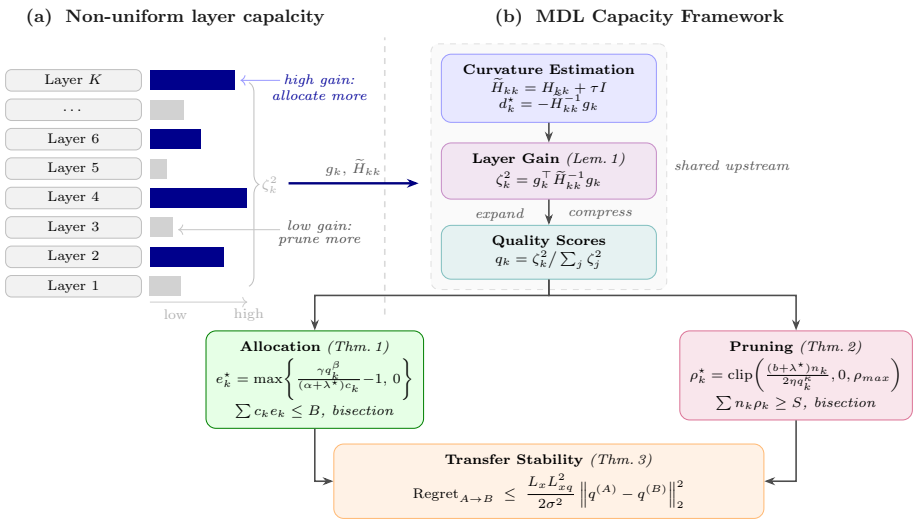


Figure 1: **(a)** Layer-wise curvature scores ζ_k^2 vary substantially across the transformer stack. High-gain layers (dark bars) hold disproportionate reducible risk and should receive additional capacity; low-gain layers (light bars) are candidates for aggressive pruning. **(b)** Our framework computes ζ_k^2 from per-layer gradients g_k and regularized Hessian blocks $\widetilde{H}_{kk}$, normalizes them to quality scores q_k, and solves two convex programs: a capacity allocation program (Theorem 2) that enriches high-gain layers, and a pruning program (Theorem 3) that concentrates sparsity on low-gain layers. Both programs admit closed-form solutions via $O(K \log 1/\varepsilon)$ bisection (Algorithms 1–2). Theorem 4 bounds the cost of transferring source-domain allocations to a target domain.

curvature interpretation of q_k; layers with larger curvature signal attract a higher degradation penalty, so the optimizer naturally protects them.

We enforce a minimum global sparsity target $S \le \sum_k n_k \rho_k$ and solve:

$$\min_{\rho_1,\dots,\rho_K} \sum_{k=1}^{K} \Big[b\, n_k (1-\rho_k) \;+\; \eta\, q_k^\kappa\, \psi(\rho_k) \Big]$$
$$\text{s.t.} \quad \sum_{k=1}^{K} n_k\, \rho_k \;\ge\; S, \;\; 0 \le \rho_k \le \rho_{\max}. \qquad (14)$$

The objective minimizes total model size subject to bounded degradation, with the constraint enforcing that at least S parameters are pruned globally.

Theorem 3 (Strong convexity and closed form for pruning). The objective in Eq. (14) is convex. If ψ is strongly convex and $q_k^\kappa > 0$ for all k, the objective is strongly convex and the primal minimizer is unique. A uniform modulus requires $q_k \ge q_{\min} > 0$, or the smoothed scores above, because q_k^κ vanishes at zero when $\kappa > 0$. Feasibility requires $0 \le S \le \rho_{\max} \sum_k n_k$.

For $\psi(\rho) = \rho^2$ and $q_k^\kappa > 0$ for all k, there exists $\lambda^\star \ge 0$ such that

$$\boxed{\rho_k^\star \;=\; \mathrm{clip}\left(\frac{(b+\lambda^\star)\, n_k}{2\,\eta\, q_k^\kappa},\; 0,\; \rho_{\max} \right).} \qquad (15)$$

When $\lambda^\star > 0$ the sparsity constraint is active, $\sum_k n_k \rho_k^\star = S$; when $\lambda^\star = 0$, the unconstrained solution already meets or exceeds the target. On interior coordinates ($0 < \rho_k^\star < \rho_{\max}$), $\rho_k^\star$ is strictly decreasing in q_k. A detailed proof for this theorem is discussed in Appendix B.2.2.

Interpretation. The pruning solution mirrors the allocation solution but in the opposite direction: layers with small q_k (low curvature signal, little reducible risk) are pruned aggressively, while high-gain layers are protected. The global multiplier $\lambda^\star$ calibrates the pruning depth to meet the sparsity target S, playing the same role as the budget multiplier in the allocation program. The quadratic choice $\psi(\rho) = \rho^2$ is a surrogate selected for analytic transparency. Convexity requires convex ψ; monotone degradation is modeled by nondecreasing ψ; uniqueness requires strict or strong convexity; and a closed-form inverse-gradient representation requires ψ' to be invertible on the relevant interval. Together, the two programs provide a unified curvature-aware framework: allocation enriches layers where gradient information is underexploited, and pruning removes redundancy where unnecessary.

Transfer Stability Under Score Drift The allocation and pruning programs depend on the normalized score vector q. A domain change can move this vector from $q^{(A)}$ to $q^{(B)}$. Appendix C bounds the target-objective excess cost incurred when the source score vector is used in place of the target score vector.

3.2 ALGORITHMS: CLOSED-FORM SOLUTIONS VIA ONE-DIMENSIONAL DUAL SEARCH

The convex programs of Theorems 2 and 3 share a common structure (see Figure 1): strict (or strong) convexity guarantees a unique primal minimizer, and the KKT conditions reduce each constrained program to a monotone scalar equation in the Lagrange multiplier λ. Both programs therefore admit $O(K \log(1/\varepsilon))$ algorithms via bisection, with primal variables recovered in closed form at each function evalua-

tion. Throughout, $q_k = \zeta_k^2 / \sum_j \zeta_j^2$ denotes the normalized curvature score from Eq. (10), $\alpha_k = \alpha c_k$ as established in Section 2.3, and we use the canonical utility choices $\phi(e) = \log(1+e)$ and $\psi(\rho) = \rho^2$. Since capacity is equivalent to an expert in mixture-of-expert, from here on out, we use expert allocation in place of capacity allocation.

Expert Allocation. From Theorem 2, the unique minimizer of Eq. (11) takes the closed form in Eq. (12) where $\lambda \geq 0$ is the dual variable for the budget constraint. The sum $\lambda \mapsto \sum_k c_k\, e_k(\lambda)$ is continuous and strictly decreasing: each active term $e_k(\lambda) > 0$ decreases strictly in λ, and once e_k hits zero it stays there. Therefore $\lambda^\star$ is unique when the constraint is active. To bracket the bisection, note that $e_k(0) = \max\{\gamma q_k^\beta / (\alpha c_k) - 1,\, 0\}$ gives the unconstrained solution. A valid upper bracket is

$$\lambda_{\max} = \max\left\{0,\ \max_k \left(\frac{\gamma q_k^\beta}{c_k} - \alpha\right)\right\},$$

because $e_k(\lambda) = 0$ whenever $\lambda \geq \gamma q_k^\beta / c_k - \alpha$.

Layer-wise Pruning. From Theorem 3, the unique minimizer of Eq. (14) is obtained as follows. The Lagrangian subtracts the dual term for the lower-bound constraint in Eq. (17),

The full KKT stationarity condition is

$$-bn_k + \eta q_k^\kappa \psi'(\rho_k) - \lambda n_k - \mu_k + \nu_k = 0, \qquad \lambda, \mu_k, \nu_k \geq 0,$$

where μ_k and ν_k correspond to $-\rho_k \leq 0$ and $\rho_k - \rho_{\max} \leq 0$. On an interior coordinate, $0 < \rho_k < \rho_{\max}$, complementary slackness gives $\mu_k = \nu_k = 0$. For $\psi(\rho) = \rho^2$, this yields,

$$\rho_k^{\text{int}}(\lambda) \;=\; \frac{(b+\lambda)\, n_k}{2\,\eta\, q_k^\kappa}.$$

Strong convexity guarantees a unique primal minimizer, although a plateau of the projected scalar map can make the dual multiplier non-unique. The positive numerator $b + \lambda > 0$ (since $b, \lambda \geq 0$) ensures $\rho_k^{\text{int}} \geq 0$, consistent with the lower-bound structure of the sparsity constraint. Projection onto $[0, \rho_{\max}]$ enforces the box constraints Eq. (15).The sum $\lambda \mapsto \sum_k n_k \rho_k(\lambda)$ is continuous and non-decreasing: the interior solution increases linearly in λ, and clipping to $[0, \rho_{\max}]$ preserves monotonicity since each ρ_k is non-decreasing before and after projection. Strict increase holds whenever at least one coordinate remains in the interior $(0, \rho_{\max})$, guaranteeing a unique $\lambda^\star$ when binding.

The unconstrained solution at $\lambda = 0$ gives $\rho_k(0) = \text{clip}(b\, n_k / (2\eta q_k^\kappa),\, 0,\, \rho_{\max})$. Feasibility under the common layer cap requires $S \leq \rho_{\max} \sum_k n_k$. A valid upper bisection bracket is

$$\lambda_{\max} = \max\left\{0,\ \max_k \left(\frac{2\eta q_k^\kappa \rho_{\max}}{n_k} - b\right)\right\},$$

which places every coordinate at its upper cap.

Remark (Budget flexibility). The budget B in Algorithm 1 need not equal the base-model FLOPs used in our experiments. Any computationally feasible value of B yields a valid allocation that is globally optimal for the continuous surrogate program; the bisection in Algorithm 1 adapts automatically to the chosen constraint.

Complexity and practical notes. Each bisection iteration evaluates the primal sum in $O(K)$ time. Both algorithms therefore run in $O(K \log(1/\varepsilon))$ time to achieve dual gap ε. This is substantially cheaper than general-purpose interior-point methods, which require $O(K^3)$ per iteration. The curvature scores $\{q_k\}$ enter only through the closed-form expressions and need not be recomputed during the dual search, so the algorithms are fully compatible with any approximation of $H_{kk}^{-1} g_k$ (e.g., diagonal Fisher, K-FAC, or randomized Nyström sketches). Together, Algorithms 1 and 2 provide practical implementations of the curvature-weighted water-filling framework developed throughout this section.

3.3 EXPERIMENTAL SETUP

Models. All experiments are conducted on two publicly available 7B-parameter large language models: Mistral-7B-v0.1 Jiang et al. [2023] and Gemma-7B Team et al. [2024]. Parameter-efficient fine-tuning is performed using LoRA-MoE Gao et al. [2024], which augments each layer with a mixture of low-rank adapters and serves as the capacity expansion mechanism targeted by Algorithm 1.

Score estimation. The theoretical score is the normalized Newton-decrement gain q_k. For scalability, the experiments instead use the proxy

$$\widehat{q}_k = \frac{\widehat{s}_k}{\sum_j \widehat{s}_j},$$

where $\widehat{s}_k$ is the chosen nonnegative aggregation of LayerIF scores. Algorithms 1 and 2 are instantiated with $\widehat{q}$, not with the theoretical Newton-decrement weights q. The convex optimization results apply to any fixed positive weight vector; the experiments therefore evaluate the allocation and pruning decision rules under this proxy.

Expert allocation. *Datasets.* We evaluate on five classification and question-answering benchmarks: CoLA Warstadt et al. [2019], MRPC Dolan and Brockett [2005], CommonsenseQA Talmor et al. [2019], ScienceQA Lu et al. [2022], and OpenBookQA Mihaylov et al. [2018].

Curvature scores. Layer-wise influence scores are computed for each (model, dataset) pair following the procedures of Askari et al. [2025] and Kwon et al. [2023]. We consider two variants of the influence score pool: **All**, in which every computed influence score is used, and **+ve**, in which only positively influential samples (those with negative influence score values, indicating a beneficial effect on validation loss) are retained. Both variants are evaluated on Mistral-7B; only the **+ve** variant is used for Gemma-7B.

Algorithm 1 MDL-Inspired Expert Allocation

Require: Scores $\{q_k\}$, costs $\{c_k\}$, budget B, hyperparameters $\alpha, \gamma, \beta > 0$

Define closed-form primal: Eq. (12)
Set $\lambda_{\min} \leftarrow 0$ and $\lambda_{\max} \leftarrow \max\{0, \max_k(\gamma q_k^{\beta}/c_k - \alpha)\}$.
if $\sum_k c_k\, e_k(0) \le B$ **then**
  $\lambda^\star \leftarrow 0$ {budget constraint inactive}
else
  Find $\lambda^\star \in (\lambda_{\min}, \lambda_{\max})$ via bisection on $\sum_k c_k\, e_k(\lambda) = B$
end if
return $e_k^\star \leftarrow e_k(\lambda^\star)$ for all k

Algorithm 2 MDL-Inspired Layer-wise Pruning

Require: Scores $\{q_k\}$, sizes $\{n_k\}$, sparsity target S, hyperparameters $b, \eta, \kappa > 0$

Define projected primal: Eq. (15)
Set $\lambda_{\min} \leftarrow 0$ and $\lambda_{\max} \leftarrow \max\{0, \max_k(2\eta q_k^{\kappa} \rho_{\max}/n_k - b)\}$.
if $\sum_k n_k\, \rho_k(0) \ge S$ **then**
  $\lambda^\star \leftarrow 0$ {sparsity constraint inactive}
else
  Find $\lambda^\star \in (\lambda_{\min}, \lambda_{\max})$ via bisection on $\sum_k n_k\, \rho_k(\lambda) = S$
end if
return $\rho_k^\star \leftarrow \rho_k(\lambda^\star)$ for all k

Budget and allocation. Following He et al. [2023], the FLOPs of a multi-expert LoRA model are empirically upper-bounded by the FLOPs of the base network. We therefore set the budget B in Algorithm 1 equal to the FLOPs of a standard single-expert layer in the respective base model, with scaling factors $\sigma = 0.0276$ for Mistral-7B and $\sigma = 0.02$ for Gemma-7B (i.e., $B = \sigma \times$ base model FLOPs). Per-layer expert counts are computed using Algorithm 1, and fine-tuning follows the protocols of Qing et al. [2024] and Gao et al. [2024] for 5 epochs per (model, dataset) pair. The convex program produces continuous allocations e_k^*. We convert them to integer expert counts by setting $m_k = \lfloor e_k^* \rfloor$. This rule preserves budget feasibility, $\sum_k c_k m_k \le \sum_k c_k e_k^* \le B$, but it can leave unused budget and is not, in general, the exact solution of the corresponding integer program.

Layer-wise Pruning. *Datasets.* Calibration uses the C4 dataset Raffel et al. [2023]. Zero-shot post-pruning evaluation is performed on seven benchmarks: RTE Wang et al. [2019], OpenBookQA Mihaylov et al. [2018], ARC-Easy and ARC-Challenge Clark et al. [2018], HellaSwag Zellers et al. [2019], BoolQ Clark et al. [2019], and WinoGrande Sakaguchi et al. [2021].

Pruning configurations. Layer-wise pruning ratios $\rho_k^\star$ are computed using Algorithm 2 with influence scores carried over from the expert allocation experiments. The computed ratios are applied under three structural pruning configurations — Magnitude Han et al. [2015], SparseGPT Frantar and Alistarh [2023], and Wanda Sun et al. [2023] — using the framework of Lu et al. [2024]. A global sparsity target of $S = 0.5 \times$ (total parameters) is enforced for both models, corresponding to 50% sparsity.

Evaluation. We report the mean zero-shot accuracy for expert allocation after finetuning and across all seven evaluation benchmarks for each pruning configuration and compare against the baseline Askari et al. [2025].

Refer to Appendix D for hardware and parameter settings used in the experiments.

3.4 RESULTS AND DISCUSSION

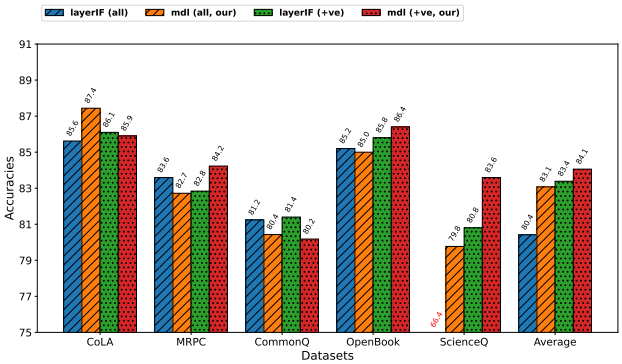


Figure 2: Expert allocation accuracy (%) on Mistral-7B-v0.1 (5 epochs)

Expert allocation. Figures 2 and 3 report zero-shot accuracy after five epochs of fine-tuning on Mistral-7B and Gemma-7B respectively, with per-layer expert counts determined by Algorithm 1 (MDL) and the LayerIF heuristic baseline. We note that LayerIF has previously been shown to outperform AlphaLoRA Qing et al. [2024] and MoLA Gao et al. [2024] on these benchmarks; we omit those comparisons here for brevity and focus on the MDL-vs-LayerIF contrast. On Mistral-7B, the MDL allocation outperforms LayerIF on average under both influence score variants: **83.07%** vs. 80.41% (All) and **84.06%** vs. 83.39% (+ve), corresponding to absolute improvements of 2.66 and 0.67 percentage points respectively. The gains are most pronounced on ScienceQA, where MDL improves over LayerIF by 13.4 points (All) and 2.8 points (+ve), suggesting that curvature-weighted allocation is particularly beneficial for knowledge-intensive reasoning tasks where representational capacity is unevenly demanded across layers. On Gemma-7B, Algorithm 1 produces identical expert counts under the **All** and **+ve** variant (hence only +ve results are reported). The MDL allocation yields a marginal improvement to the LayerIF allocation (**87.52%** vs. 87.46%), confirming that the two methods agree in structure when curvature scores

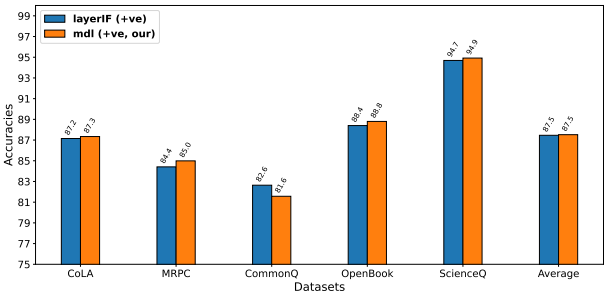


Figure 3: Expert allocation accuracy (%) on Gemma-7B, +ve variant only (5 epochs)

are relatively uniform, MDL provides a cleaner theoretical justification for the same decision. These results demonstrate that replacing the knapsack heuristic of LayerIF with the convex MDL program of Theorem 2 yields consistent improvements in some cases without additional compute: Both methods share the same influence-score inputs, and Algorithm 1 adds only an $O(K \log 1/\varepsilon)$ bisection step. See more in Tables 5, 7 and 6 for results on Mistral-7B and Gemma-7B respectively.

Layer-wise Pruning. Tables 1 and 2 report mean zero-shot accuracy across seven evaluation benchmarks at 50% global sparsity, under Magnitude, Wanda, and SparseGPT pruning configurations. We impose the box constraint before optimization, using $\rho_{\max} = 0.51$ for Mistral-7B and $\rho_{\max} = 0.55$ for Gemma-7B. The global 50% target remains feasible without fully pruning any targeted parameter block. For Mistral-7B, the combination of a 50% global target and $\rho_{\max} = 0.51$ leaves little room for heterogeneous layer ratios, which partly explains the near parity between the two decision rules. Fully pruning a targeted transformation can cause severe degradation, although residual connections may still preserve a network-level information path. On Mistral-7B (Table 1), MDL pruning ratios match LayerIF closely across all three configurations: average accuracies are within 0.05 points for Wanda and identical for SparseGPT (60.18%), while Magnitude shows a marginal difference of 0.18 points (56.23% MDL vs. 56.41% LayerIF). On Gemma-7B (Table 2), MDL outperforms LayerIF under Magnitude (**33.34%** vs. 32.91%) while LayerIF leads under Wanda (52.3% vs. 49.47%) and SparseGPT (50.79% vs. 49.22%). The pruning parity between MDL and LayerIF is itself a meaningful result: it shows that the principled convex program of Theorem 3 recovers the empirically tuned LayerIF ratios without manual calibration, while providing the theoretical guarantees — strong convexity, unique minimizer, and budget feasibility, which are lacking in LayerIF. The Wanda and SparseGPT gaps on Gemma-7B suggest that the quadratic degradation model $\psi(\rho) = \rho^2$ may underestimate pruning sensitivity in certain architectural regimes; exploring richer ψ (e.g., $\psi(\rho) = -\log(1-\rho)$) is a natural direction for future work. Overall, the allocation results are strongest on Mistral-7B, while Gemma-7B allocation is essentially tied with LayerIF and the pruning results are mixed. The smaller pruning differences are partly explained by the 50% global sparsity target and the tight per-layer caps, which leave limited freedom for heterogeneous pruning ratios. Our optimality guarantees apply to the continuous surrogate programs; downstream accuracy can additionally be affected by integer rounding and the specific structural pruning method.

Table 1: Mean zero-shot accuracy (%) across 7 benchmarks for pruning on Mistral-7B-v0.1 at 50% sparsity (layer cap 0.51). Rows show the calibration dataset used to compute influence scores.

Calibration dataset	Magnitude	Wanda	SparseGPT
CoLA	56.47	58.59	59.66
MRPC	56.23	58.48	60.12
CommonsenseQA	56.32	58.65	60.77
OpenBookQA	56.22	58.46	60.05
ScienceQA	55.89	58.99	60.30
Average (MDL)	56.23	58.63	**60.18**
LayerIF	56.41	58.67	**60.18**

Table 2: Mean zero-shot accuracy (%) across 7 benchmarks for pruning on Gemma-7B at 50% sparsity (layer cap 0.55). Rows show the calibration dataset used to compute influence scores.

Calibration dataset	Magnitude	Wanda	SparseGPT
CoLA	33.03	48.36	49.28
MRPC	33.49	52.06	50.77
CommonsenseQA	33.51	50.26	49.28
OpenBookQA	33.52	48.09	48.11
ScienceQA	33.15	48.58	48.66
Average (MDL)	**33.34**	49.47	49.22
LayerIF	32.91	**52.30**	**50.79**

4 CONCLUSION

We presented a curvature-aware framework for layer-wise capacity allocation and pruning in large language models, grounded in the Minimum Description Length principle. The central quantity, $\zeta_k^2 = g_k^\top \widetilde{H}_{kk}^{-1} g_k$, measures reducible empirical risk at each layer and drives two convex programs with unique closed-form solutions, each computable in $O(K \log 1/\varepsilon)$ via bisection. Experiments on Mistral-7B and Gemma-7B show clear allocation gains in one model, an allocation tie in the other, and competitive but mixed pruning performance. Because both methods use the same influence-derived proxy scores, the comparison isolates the decision rule. The formal guarantees are global optimality for the continuous surrogate programs and quadratic stability with respect to score perturbations under the stated positivity and strong-convexity assumptions. See Appendix G and Appendix H for limitations and directions for future work, respectively.

References

Hadi Askari, Shivanshu Gupta, Fei Wang, Anshuman Chhabra, and Muhao Chen. LayerIF: Estimating Layer Quality for Large Language Models using Influence Functions. In *Advances in Neural Information Processing Systems*, 2025.

Pierre Baldi and Roman Vershynin. The capacity of feedforward neural networks. *Neural Netw.*, 116(C):288–311, August 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.04.009. URL https://doi.org/10.1016/j.neunet.2019.04.009.

Avrim Blum and John Langford. Pac-mdl bounds. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 344–357, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45167-9.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical gauss-newton optimisation for deep learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 557–565, Sydney, Australia, 2017. PMLR. URL https://proceedings.mlr.press/v70/botev17a.html.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. Available at https://web.stanford.edu/~boyd/cvxbook/.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, January 2005.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot, 2023. URL https://arxiv.org/abs/2301.00774.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. Higher layers need more lora experts, 2024. URL https://arxiv.org/abs/2402.08562.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1135–1143, Cambridge, MA, USA, 2015. MIT Press.

B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1, 1993. doi: 10.1109/ICNN.1993.298572.

Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. Merging experts into one: Improving computational efficiency of mixture of experts. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14685–14691, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.907. URL https://aclanthology.org/2023.emnlp-main.907/.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv preprint arXiv:2310.00902*, 2023.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Hongyang Liu, Sara Elkerdawy, Nilanjan Ray, and Mostafa Elhoushi. Layer importance estimation with imprinting for neural network quantization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2408–2417, 2021. doi: 10.1109/CVPRW53098.2021.00273.

Sanae Lotfi, Marc Finzi, Yilun Kuang, Tim G. J. Rudner, Micah Goldblum, and Andrew Gordon Wilson. Non-vacuous generalization bounds for large language models. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:266573256.

Haiquan Lu, Yefan Zhou, Shiwei Liu, Zhangyang Wang, Michael W Mahoney, and Yaoqing Yang. Alphapruning: Using heavy-tailed self regularization theory for improved layer-wise pruning of large language models. In *Thirty-eighth Conference on Neural Information Processing Systems*, 2024.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR, 2015.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Benjamin Prada, Shion Matsumoto, Abdul Malik Zekri, and Ankur Mali. Bridging predictive coding and mdl: A two-part code framework for deep learning. *ArXiv*, abs/2505.14635, 2025. URL https://api.semanticscholar.org/CorpusID:278768469.

Peijun Qing, Chongyang Gao, Yefan Zhou, Xingjian Diao, Yaoqing Yang, and Soroush Vosoughi. Alphalora: Assigning lora experts based on layer training quality, 2024. URL https://arxiv.org/abs/2410.10054.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Jorma Rissanen. Modeling by shortest data description*. *Autom.*, 14:465–471, 1978. URL https://api.semanticscholar.org/CorpusID:30140639.

Jorma Rissanen. Stochastic complexity in statistical inquiry. In *World Scientific Series in Computer Science*, 1989. URL https://api.semanticscholar.org/CorpusID:9365056.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Jürgen Schmidhuber. Discovering neural nets with low kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(96)00127-X. URL https://www.sciencedirect.com/science/article/pii/S089360809600127X.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning : from theory to algorithms*. Cambridge University Press, Cambridge, 2014. ISBN 1-107-29801-6.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Zichen Song, Sitan Huang, Yuxin Wu, and Zhongfeng Kang. Layer importance and hallucination analysis in large language models via enhanced activation variance-sparsity, 2024. URL https://arxiv.org/abs/2411.10069.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27:1134–1142, 1984. URL https://api.semanticscholar.org/CorpusID:59712.

Dmytro Vitel and Anshuman Chhabra. First is Not Really Better Than Last: Evaluating Layer Choice and Aggregation Strategies in Language Model Data Influence Estimation. In *International Conference on Learning Representations*, 2026.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL https://arxiv.org/abs/1804.07461.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 09 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00290. URL https://doi.org/10.1162/tacl_a_00290.

Andrew Gordon Wilson. Position: Deep learning is not so mysterious or different. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025. URL https://openreview.net/forum?id=42Au7FoD8F.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Curvature-Weighted Capacity Allocation: A Minimum Description Length Framework for Layer-Adaptive Large Language Model Optimization

Theophilus Amaefuna[*1] **Hitesh Vaidya**[*1] **Anshuman Chhabra**[1] **Ankur Mali**[1]

[1]Bellini College of Artificial Intelligence, Cybersecurity and Computing

University of South Florida, Tampa, Florida, USA

SUPPLEMENTARY OVERVIEW

This supplement follows the same conceptual progression as the main paper. Section A places the proposed curvature-weighted framework in the context of layer-quality estimation, second-order compression, MDL, and adaptive capacity. Section B then supplies the detailed proofs and the surrogate-regularization analysis underlying the main theoretical claims. Section C develops the transfer-stability result and its cross-task diagnostic. The remaining sections provide implementation details, complete experimental results, ablations, proxy validation, statistical analysis, limitations, future work and the LLM-use disclosure.

[*]Equal contribution.

[*]Equal contribution.

CONTENTS

A RELATED WORK

The main paper combines a layer-wise curvature score with globally constrained allocation and pruning programs. We first position these two ingredients relative to prior work, beginning with layer-quality estimation and then moving to second-order compression, MDL, and adaptive-capacity models.

A.1 LAYER QUALITY ESTIMATION

Quantifying the per-layer contribution to model performance is central to pruning, expert allocation in mixture-of-experts (MoE) models, and network compression. Askari et al. [2025] proposed LayerIF, which uses influence functions to estimate layer quality and applies it to predict expert counts in Mistral-7B and to determine layer-wise pruning ratios. Song et al. [2024] introduced the Activation Variance-Sparsity Score (AVSS), a combined measure of normalized activation variance and sparsity that quantifies each layer's contribution to model performance. Liu et al. [2021] developed an accuracy-aware criterion for ranking layer importance, used to guide quantization decisions. A common limitation shared by these approaches is the absence of curvature information: gradient magnitudes and activation statistics do not account for the local geometry of the loss landscape. Our gain ζ_k^2 addresses this gap directly by incorporating the inverse Hessian block, yielding a measure of *reducible risk* rather than raw gradient magnitude.

A.2 SECOND-ORDER METHODS FOR MODEL COMPRESSION

Second-order information has been used for pruning since the seminal Optimal Brain Damage LeCun et al. [1989] and Optimal Brain Surgeon frameworks Hassibi et al. [1993], which identify parameters to remove by computing the Hessian of the training loss. More recent work scales these ideas to modern architectures using diagonal Fisher approximations Martens and Grosse [2015], Kronecker-factored curvature Botev et al. [2017], and randomized sketches. Our work extends this tradition from individual weight pruning to *layer-level* capacity allocation, and introduces a convex program that jointly optimizes the allocation across all layers under a global budget.

A.3 GENERALIZATION BOUNDS AND MINIMUM DESCRIPTION LENGTH

Generalization theory asks how well a model trained on finite data performs on unseen examples. Valiant [1984] formalized PAC learning, establishing that generalization depends on hypothesis space complexity and sample size. Shalev-Shwartz and Ben-David [2014] extend this to infinite hypothesis spaces via VC dimension. For LLMs, classical bounds are vacuous due to the enormous number of parameters; Lotfi et al. [2023] address this by introducing compression-based bounds via SubLoRA, a low-dimensional nonlinear parametrization that yields non-vacuous guarantees. Our MDL objective is directly motivated by this line of work: minimizing description length simultaneously controls generalization and penalizes unnecessary model complexity, grounding our convex programs in information-theoretic principles.

A.4 MIXTURE-OF-EXPERTS AND ADAPTIVE CAPACITY

Sparse MoE models Shazeer et al. [2017], Fedus et al. [2022] increase model capacity without proportional compute cost by routing tokens to a subset of expert layers. Existing routing mechanisms are learned end-to-end and do not explicitly account for the curvature-adjusted utility of adding capacity at a given layer. Our allocation program provides a principled, optimization-based alternative that is complementary to learned routing: given a fixed routing mechanism, it determines *how much* capacity to assign to each layer based on reducible risk.

Together, these lines of work motivate the two-part construction analyzed next: a layer-wise second-order score that estimates locally reducible risk, and a convex resource-allocation rule that converts those scores into globally feasible decisions.

B MATHEMATICAL FOUNDATIONS AND PROOFS

With the surrounding literature established, we now give the detailed mathematical arguments omitted from the main paper. The presentation follows the logical dependency of the framework: first the layer-restricted quadratic optimum, then the allocation and pruning programs, and finally the effect of surrogate regularization.

B.1 LAYER-RESTRICTED QUADRATIC OPTIMUM

We begin with Lemma 1, because its regularized Newton step defines the layer gain used by every subsequent optimization program. Recall that $\widetilde{Q}_k(d_k)$ is the layer-restricted quadratic surrogate for the change in empirical loss produced by a perturbation d_k to layer k.

Proof. The gradient of $\widetilde{Q}_k$ with respect to d_k is $\nabla_{d_k}\widetilde{Q}_k = g_k + \widetilde{H}_{kk}\, d_k$. Setting this to zero and invoking $\widetilde{H}_{kk} \succ 0$ yields the unique stationary point $d_k^\star = -\widetilde{H}_{kk}^{-1} g_k$. Substituting back,

$$\begin{aligned}\widetilde{Q}_k(d_k^\star) &= g_k^\top(-\widetilde{H}_{kk}^{-1} g_k) + \frac{1}{2}(-\widetilde{H}_{kk}^{-1} g_k)^\top \widetilde{H}_{kk}(-\widetilde{H}_{kk}^{-1} g_k) \\ &= -\, g_k^\top \widetilde{H}_{kk}^{-1} g_k + \frac{1}{2}\, g_k^\top \widetilde{H}_{kk}^{-1} g_k \;=\; -\frac{1}{2}\, g_k^\top \widetilde{H}_{kk}^{-1} g_k.\end{aligned}$$

□

B.2 CLOSED-FORM ALLOCATION AND PRUNING SOLUTIONS

Having established the layer-wise gain, we next show how the two global resource-allocation programs inherit convexity and reduce to one-dimensional dual searches.

B.2.1 Allocation: Convexity and Closed Form

We first prove Theorem 2, which characterizes the optimal continuous capacity allocation $e_k^\star$ under the global budget.

Proof. Each term $\alpha c_k e_k$ is linear. Since ϕ is concave, $-\phi$ is convex, and since $\gamma q_k^\beta \geq 0$, the term $-\gamma q_k^\beta \phi(e_k)$ is convex on $e_k \geq 0$. The sum of convex functions over a convex feasible set is convex. When ϕ is strictly concave and $q_k > 0$, each term $-\gamma q_k^\beta \phi(e_k)$ is strictly convex, giving strict convexity of the full objective.

Slater's condition Boyd and Vandenberghe [2004] holds (any strictly feasible point satisfies the constraint with strict inequality), so strong duality applies. The Lagrangian is

$$\mathcal{L}(e, \lambda) = \sum_k \Big[\alpha c_k e_k - \gamma q_k^\beta \phi(e_k)\Big] + \lambda\left(\sum_k c_k e_k - B\right), \qquad \lambda \geq 0. \tag{16}$$

Stationarity at an interior point $e_k > 0$ gives

$$(\alpha + \lambda)\, c_k \;-\; \gamma q_k^\beta\, \phi'(e_k^\star) \;=\; 0.$$

For $\phi(e) = \log(1+e)$, $\phi'(e) = 1/(1+e)$, so

$$(\alpha + \lambda^\star)\, c_k \;=\; \frac{\gamma q_k^\beta}{1 + e_k^\star},$$

which yields Eq. (13) after incorporating non-negativity. Since $e_k(\lambda)$ is strictly decreasing and continuous in λ, the sum $\sum_k c_k e_k(\lambda)$ is strictly decreasing, so $\lambda^\star$ is unique and can be computed in $O(K \log(1/\varepsilon))$ time by bisection. Strict monotonicity of $e_k^\star$ in q_k follows directly from Eq. (13). □

B.2.2 Pruning: Strong Convexity and Closed Form

The pruning program is complementary to allocation: instead of distributing additional capacity, it assigns layer-wise sparsity while enforcing a global target. We now prove the closed-form characterization in Eq. (15).

Proof. The term $-b n_k \rho_k$ is linear. If ψ is strongly convex, $\eta q_k^\kappa \psi(\rho_k)$ is strongly convex for $q_k > 0$, making the full objective strongly convex. The feasible set $[0,1]^K \cap \{\sum_k n_k \rho_k \geq S\}$ is convex and compact, so a unique minimizer exists.

The constraint $\sum_k n_k \rho_k \geq S$ is a lower-bound constraint, so the Lagrangian is formed by subtracting the dual term:

$$\begin{aligned}\mathcal{L}(\rho, \lambda) = \;& \textstyle\sum_k \Big[b\, n_k(1-\rho_k) + \eta q_k^\kappa \psi(\rho_k)\Big] - \lambda\Big(\sum_k n_k \rho_k - S\Big) - \sum_{k=1}^K \mu_k \rho_k + \sum_{k=1}^K \nu_k(\rho_k - \rho_{\max}), \\ & \lambda \geq 0,\;\; \mu_k \geq 0,\;\; \nu_k \geq 0\end{aligned} \tag{17}$$

Stationarity at an interior point $\rho_k \in (0, \rho_{\max})$ gives

$$-b\, n_k + \eta\, q_k^{\kappa}\, \psi'(\rho_k^\star) - \lambda^\star\, n_k - \mu_k + \nu_k \;=\; 0.$$

For $\psi(\rho) = \rho^2$, $\psi'(\rho) = 2\rho$, so

$$\rho_k^\star \;=\; \frac{(b + \lambda^\star)\, n_k}{2\,\eta\, q_k^{\kappa}},$$

and box constraints $[0, \rho_{\max}]$ induce clipping, giving Eq. (15). Since $\lambda^\star \geq 0$, the numerator $b + \lambda^\star > 0$, so $\rho_k^\star > 0$ on the interior.

The mapping $\lambda \mapsto \sum_k n_k \rho_k(\lambda)$ is strictly increasing (larger λ increases every ρ_k through the numerator $b + \lambda$), so $\lambda^\star$ is unique when binding and can be found by bisection. Strict decrease of $\rho_k^\star$ in q_k follows directly from Eq. (15). □

B.3 VALIDITY UNDER SURROGATE REGULARIZATION

The preceding proofs establish optimality for the regularized quadratic surrogate. We now connect that surrogate back to the actual empirical objective and state the additional separation condition needed for layer rankings to be certified.

The step $\Delta^\star = E_k^\top d_k^\star$ is derived from the surrogate $\widetilde{H}_{kk}$, not from H_{kk} directly. We now show that the resulting ranking of layers by ζ_k^2 is consistent with the true objective decrease, and characterize the approximation gap explicitly.

Applying Eq. (6) to $\Delta^\star$ and using $\Delta^{\star\top} H_{kk}\, \Delta^\star = \Delta^{\star\top} \widetilde{H}_{kk} \Delta^\star - \tau \|\Delta^\star\|^2$ gives

$$L(\theta + \Delta^\star) - L(\theta) \;=\; -\frac{1}{2}\,\zeta_k^2 \;-\; \frac{\tau}{2}\,\|\Delta^\star\|^2 \;+\; R(\Delta^\star). \tag{18}$$

We bound the bias term as follows. Since $d_k^\star = -\widetilde{H}_{kk}^{-1} g_k$ and $\|E_k^\top d\| = \|d\|$ for any d,

$$\|\Delta^\star\|^2 \;=\; \|d_k^\star\|^2 \;=\; g_k^\top \widetilde{H}_{kk}^{-2}\, g_k.$$

The spectral inequality $\widetilde{H}_{kk}^{-2} \preceq \lambda_{\min}(\widetilde{H}_{kk})^{-1} \widetilde{H}_{kk}^{-1}$ (which follows because all eigenvalues of $\widetilde{H}_{kk}^{-1}$ lie in $(0, \lambda_{\min}(\widetilde{H}_{kk})^{-1}]$) then yields

$$\frac{\tau}{2}\,\|\Delta^\star\|^2 \;\leq\; \frac{\tau}{2\,\lambda_{\min}(\widetilde{H}_{kk})}\,\zeta_k^2. \tag{19}$$

Combining Eq. (18) and Eq. (19),

$$L(\theta + \Delta^\star) - L(\theta) \;\geq\; -\frac{1}{2}\left(1 + \frac{\tau}{\lambda_{\min}(\widetilde{H}_{kk})}\right)\zeta_k^2 + R(\Delta^\star).$$

The factor $\tau / \lambda_{\min}(\widetilde{H}_{kk})$ measures the relative damping term in this bound. When $H_{kk} \succeq 0$, one has $\lambda_{\min}(\widetilde{H}_{kk}) \geq \tau$ and the factor is at most one. For an indefinite block, $\lambda_{\min}(\widetilde{H}_{kk}) = \lambda_{\min}(H_{kk}) + \tau$ may be smaller than τ; positive definiteness requires $\tau > -\lambda_{\min}(H_{kk})$, and the ratio need not be at most one.

Small damping error alone does not preserve rankings when scores are nearly tied. Define

$$\varepsilon_k := \frac{\tau}{2}\|d_k^\star\|_2^2 + \frac{M}{6}\|d_k^\star\|_2^3.$$

The ordering of layers k and ℓ is certified whenever

$$\frac{1}{2}|\zeta_k^2 - \zeta_\ell^2| > \varepsilon_k + \varepsilon_\ell.$$

Neither batch normalization nor weight decay alone guarantees that an empirical Hessian block is positive semidefinite.

This completes the within-task justification of the curvature score. We next study a different source of error: replacing the target-task score vector by a score vector estimated on a related source task.

C TRANSFER STABILITY UNDER SCORE DRIFT

The allocation and pruning programs depend on a normalized score vector. We quantify the target-objective excess cost incurred when a source score vector $q^{(A)}$ is substituted for the target score vector $q^{(B)}$, while keeping the optimization objective, hyperparameters, and feasible set fixed.

C.1 PROBLEM SETUP

Let $\mathcal{J}_B(x; q)$ denote the MDL program objective on task B, as a function of the decision variable x and quality inputs q. Here $x = e \in \mathbb{R}^K_{\geq 0}$ for the allocation program (Section 2.3) and $x = \rho \in [0, \rho_{max}]^K$ for the pruning program (Section 3.1), with

$$\Delta_K := \{q \in \mathbb{R}^K_+ : \mathbf{1}^\top q = 1\}, \qquad q \in \Delta_K.$$

The score vector has K coordinates and affine dimension $K - 1$; it is distinct from the decision variable x.

Let $\mathcal{X}$ denote the corresponding feasible set (budget constraint or sparsity constraint). Since Theorems 2 and 3 guarantee a unique minimizer for each admissible q, define

$$\widehat{x}_B(q) \;:=\; \arg\min_{x \in \mathcal{X}} \mathcal{J}_B(x;\, q).$$

The *transfer regret* of deploying the source-derived decision $\widehat{x}_B(q^{(A)})$ on the target task is

$$\mathrm{Regret}_{A \to B} \;:=\; \mathcal{J}_B\Big(\widehat{x}_B(q^{(A)});\, q^{(B)}\Big) - \mathcal{J}_B\Big(\widehat{x}_B(q^{(B)});\, q^{(B)}\Big). \tag{20}$$

Both terms are evaluated under the target objective and target scores $q^{(B)}$, so Eq. (20) is the excess cost of a misspecified score input. The notation $\widehat{x}_B(q^{(A)})$ keeps the target objective and feasible set fixed. Using $\widehat{x}_A(q^{(A)})$ would additionally introduce objective, hyperparameter, or feasible-set drift and would require separate error terms.

C.2 REGULARITY CONDITIONS

Let $\Delta_K(q_{\min}) = \{q \in \Delta_K : q_k \geq q_{\min} \text{ for all } k\}$ for some $q_{\min} > 0$. We assume:

(i) $\mathcal{J}_B(\cdot; q)$ is σ-strongly convex on the common closed convex feasible set $\mathcal{X}$ for every $q \in \Delta_K(q_{\min})$;

(ii) the decision gradient is Lipschitz in the score vector:

$$\|\nabla_x \mathcal{J}_B(x; q) - \nabla_x \mathcal{J}_B(x; q')\|_2 \leq L_{xq} \|q - q'\|_2$$

for all $x \in \mathcal{X}$ and $q, q' \in \Delta_K(q_{\min})$.

For allocation with $\phi(e) = \log(1 + e)$, one may take

$$L^{\mathrm{alloc}}_{xq} \leq \gamma\beta \max_{t \in [q_{\min}, 1]} t^{\beta - 1}.$$

For pruning with $\psi(\rho) = \rho^2$,

$$L^{\mathrm{prune}}_{xq} \leq 2\eta\kappa\rho_{\max} \max_{t \in [q_{\min}, 1]} t^{\kappa - 1}.$$

The positive lower bound on q_k is necessary for uniform constants when $0 < \beta < 1$ or $0 < \kappa < 1$, and it also prevents the strong-convexity modulus from vanishing in the pruning program.

C.3 TRANSFER-REGRET BOUND

With the score domain and regularity conditions fixed, we can state the target-objective excess-cost guarantee.

Theorem 4 (Transfer regret under score drift). Under assumptions (i)–(ii),

$$0 \leq \mathrm{Regret}_{A \to B} \leq \frac{L^2_{xq}}{2\sigma} \|q^{(A)} - q^{(B)}\|_2^2. \tag{21}$$

If $|q^{(A)}_k - q^{(B)}_k| \leq \delta_k$ for every k, then

$$\mathrm{Regret}_{A \to B} \leq \frac{L^2_{xq}}{2\sigma} \sum_{k=1}^{K} \delta_k^2.$$

Proof. Write $x_A = \widehat{x}_B(q^{(A)})$ and $x_B = \widehat{x}_B(q^{(B)})$. Strong convexity of $\mathcal{J}_B(\cdot; q^{(B)})$ gives

$$\mathcal{J}_B(x_A; q^{(B)}) - \mathcal{J}_B(x_B; q^{(B)}) \leq \Big\langle \nabla_x \mathcal{J}_B(x_A; q^{(B)}), x_A - x_B \Big\rangle - \frac{\sigma}{2} \|x_A - x_B\|_2^2.$$

Because x_A minimizes $\mathcal{J}_B(\cdot; q^{(A)})$ over the closed convex set $\mathcal{X}$, its variational inequality is

$$\left\langle \nabla_x \mathcal{J}_B(x_A; q^{(A)}), x_B - x_A \right\rangle \geq 0.$$

Equivalently, $\langle \nabla_x \mathcal{J}_B(x_A; q^{(A)}), x_A - x_B \rangle \leq 0$. Subtracting this nonpositive term and applying assumption (ii) yields

$$\text{Regret}_{A \to B} \leq L_{xq} \|q^{(A)} - q^{(B)}\|_2 \|x_A - x_B\|_2 - \frac{\sigma}{2} \|x_A - x_B\|_2^2.$$

The right-hand side is maximized over $r = \|x_A - x_B\|_2 \geq 0$ at $r = L_{xq}\|q^{(A)} - q^{(B)}\|_2/\sigma$, which gives Eq. (21). Nonnegativity follows from the optimality of x_B under the target score vector. The coordinate-wise statement follows from $\|q^{(A)} - q^{(B)}\|_2^2 \leq \sum_k \delta_k^2$. □

C.4 INTERPRETATION AND BOUNDARY VALIDITY

The bound is valid for interior and boundary solutions, including active budget, sparsity, and box constraints; no multiplier-sensitivity argument or additional smoothness constant is required. The factor $L_{xq}^2/(2\sigma)$ separates sensitivity to score perturbations from the strong-convexity conditioning of the target surrogate program.

C.5 CROSS-TASK TRANSFER DIAGNOSTIC

We conduct a cross-task diagnostic on Gemma-7B by deriving the expert allocation from CommonsenseQA and deploying that allocation on the remaining tasks. Because the experimental weights are LayerIF-derived proxies, we report the squared proxy-score drift

$$\|\widehat{q}^{(\text{source})} - \widehat{q}^{(\text{target})}\|_2^2,$$

with each proxy vector normalized to sum to one. The accompanying performance quantity is a downstream accuracy difference, not the nonnegative optimization regret in Eq. (20).

Table 3: Cross-task proxy-score drift and downstream accuracy difference for a CommonsenseQA-derived allocation.

Dataset	**Proxy-score drift**	**Accuracy difference**
CommonQA (Source)	0.0000	0.00
MRPC	0.2447	-0.34
CoLA	0.6273	1.82
OpenBookQA	0.3127	1.20
ScienceQA	0.6698	1.44

The negative MRPC entry confirms that the third column is not the theorem's optimization regret, which is nonnegative by definition. The table therefore provides a qualitative cross-task diagnostic. The direct numerical quantity associated with Theorem 4 is

$$R_J = \mathcal{J}_B(\widehat{x}_B(q^{(A)}); q^{(B)}) - \mathcal{J}_B(\widehat{x}_B(q^{(B)}); q^{(B)}) \geq 0,$$

with upper bound $L_{xq}^2 \|q^{(A)} - q^{(B)}\|_2^2/(2\sigma)$.

The theorem controls the surrogate optimization objective, whereas the cross-task experiment additionally reflects training noise, integer expert counts, and downstream evaluation. We therefore keep the two quantities separate and now turn to the implementation details and complete empirical results.

D EXPERIMENTAL DETAILS AND FULL RESULTS

D.1 HARDWARE

Mistral-7B experiments are run on 4× NVIDIA L40S GPUs; Gemma-7B experiments are run on 4× NVIDIA A6000 GPUs.

D.2 HYPERPARAMETERS

Table 4 summarizes all hyperparameter settings. For the pruning program (Algorithm 2): $b = 16$ bits per retained parameter, $\eta = 2$, $\kappa = 1$. For the allocation program (Algorithm 1): $\alpha = 0.5$, $\gamma = 0.9$, $\beta = 1$. Layer sizes n_k are set to the number of parameters in layer k, and per-unit costs c_k are set to the FLOPs of a single LoRA expert at layer k.

Table 4: Hyperparameter configurations for allocation and pruning.

Program	Parameter	Mistral-7B	Gemma-7B
Allocation	α	0.5	0.5
	γ	0.9	0.9
	budget scaling, σ	0.0276	0.02
Pruning	b (bits)	16	16
	η	2	2
	κ	1	1
	Sparsity target, S	50%	50%

D.3 COMPLETE EXPERT-ALLOCATION RESULTS

Tables 5 and 6 provide the dataset-level results summarized in Section 3.4 of the main paper. These complete tables separate the effect of the convex decision rule from the influence-style score estimator shared with LayerIF.

Table 5: Expert allocation accuracy (%) on Mistral-7B-v0.1 (5 epochs). Best average per category in **bold**.

	All		+ve	
Dataset	**LayerIF**	**MDL**	**LayerIF**	**MDL**
CoLA	85.62	87.44	86.10	85.90
MRPC	83.59	82.72	82.84	84.23
CommonsenseQA	81.24	80.43	81.40	80.18
OpenBookQA	85.20	85.00	85.80	86.40
ScienceQA	66.41	79.77	80.80	83.59
Average	80.41	**83.07**	83.39	**84.06**

Table 6: Expert allocation accuracy (%) on Gemma-7B, +ve variant only (5 epochs). Best average in **bold**.

Dataset	LayerIF	MDL (+ve)
CoLA	87.15	87.34
MRPC	84.41	84.99
CommonsenseQA	82.64	81.57
OpenBookQA	88.40	88.80
ScienceQA	94.69	94.92
Average	87.46	**87.52**

The full results establish the primary comparison. We next examine which parts of the framework drive those outcomes by varying the allocation and pruning hyperparameters and by comparing alternative score-to-decision pipelines.

E ABLATIONS AND PROXY VALIDATION

This section complements the main experiments in three steps. We first compare the proposed allocation rule with uniform, hand-designed, and heuristic alternatives. We then study sensitivity to the exponents β and κ. Finally, we compare direct evaluations of ζ_k^2 with scalable proxy scores on a smaller network where the regularized curvature quantity can be computed explicitly.

E.1 MISTRAL-7B EXPERT-ALLOCATION COMPARISON

We compare uniform and non-uniform MoLA allocations [Gao et al., 2024], AlphaLoRA [Qing et al., 2024], a gradient-norm/Fisher score combined with the LayerIF heuristic, the original LayerIF decision rule [Askari et al., 2025], and our convex allocation program using the same scalable influence-style score inputs. All methods are evaluated on Mistral-7B-v0.1 under the common five-epoch protocol.

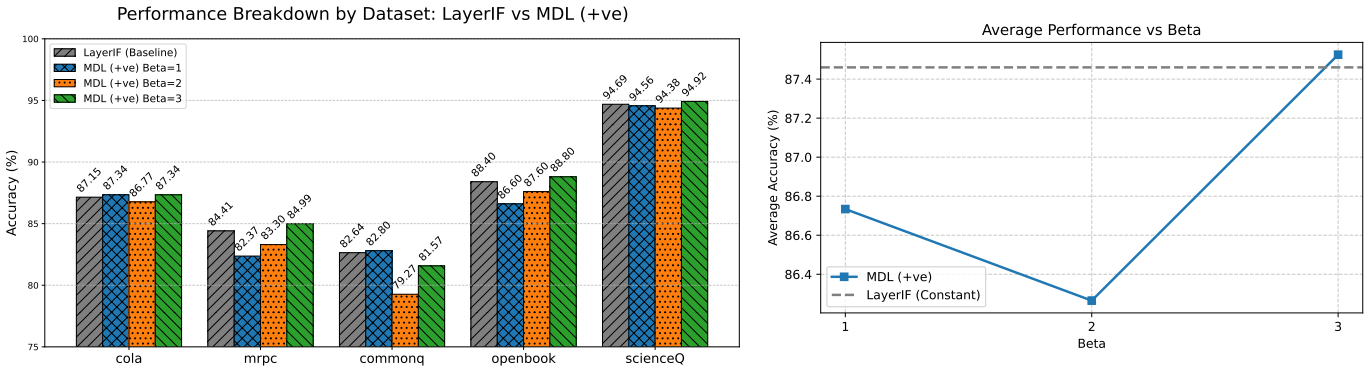
Table 7: Accuracy (%) across expert-allocation strategies (Mistral-7B-v0.1, 5 epochs, All).

Dataset	Uniform (MoLA 5555)	Non-uniform MoLA 2468	Non-uniform MoLA 8642	AlphaLora	Fisher (grad. norm)	LayerIF	MDL
Cola	85.52	86.29	85.43	87.44	85.33	85.62	87.44
MRPC	83.94	82.61	84.23	83.54	80.87	83.59	82.72
CommonsenseQA	81.24	81.08	63.47	81.90	79.52	81.24	80.43
OpenbookQA	85.40	86.60	82.20	82.00	86.20	85.20	85.00
ScienceQA	81.21	76.39	82.55	72.26	78.33	66.41	79.77

Table 7 shows that the proposed method remains competitive with the strongest alternatives while providing an explicit continuous objective and a globally enforced resource budget. The comparison also clarifies that the theoretical contribution concerns the score-to-decision rule rather than a uniformly superior score estimator.

E.2 SENSITIVITY TO β

We next isolate the gain-emphasis exponent β in Algorithm 1. Using Gemma-7B, we vary β while holding the remaining allocation hyperparameters and training protocol fixed. This experiment measures how strongly concentrating capacity on the largest scores affects dataset-level and average accuracy.



(a) (b)

Figure 4: (a) reveals the detailed accuracy per dataset as Beta goes from $1-3$. (b) As shown for Gemma-7B, $\beta = 3$ gives, on average, a better accuracy across the datasets.

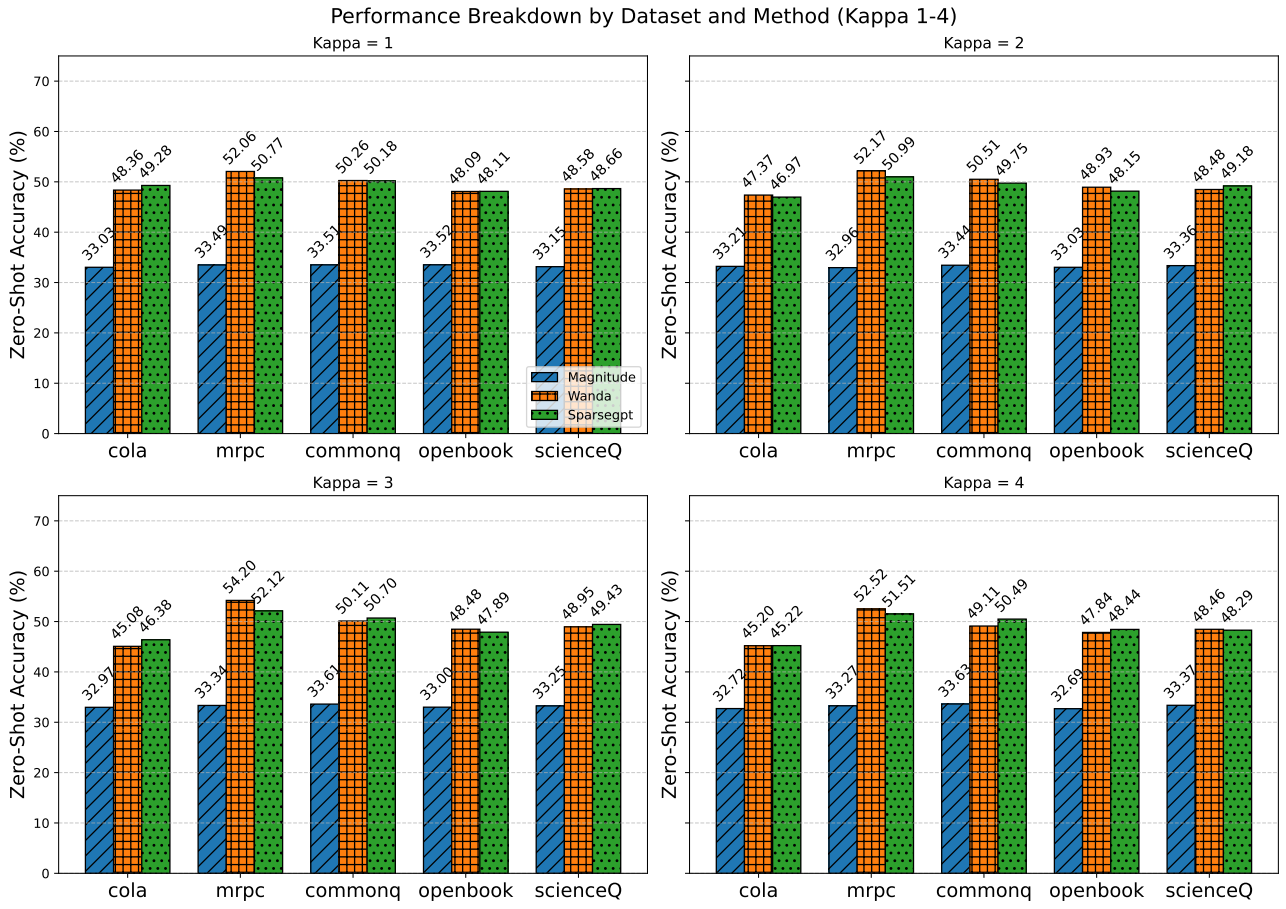
E.3 SENSITIVITY TO κ

The pruning exponent κ controls how sharply the degradation penalty distinguishes high- and low-score layers. We vary κ on Gemma-7B while holding the remaining pruning hyperparameters fixed, and report zero-shot accuracy for Magnitude, Wanda, and SparseGPT pruning over the tasks listed in Section 3.3 of the main paper.

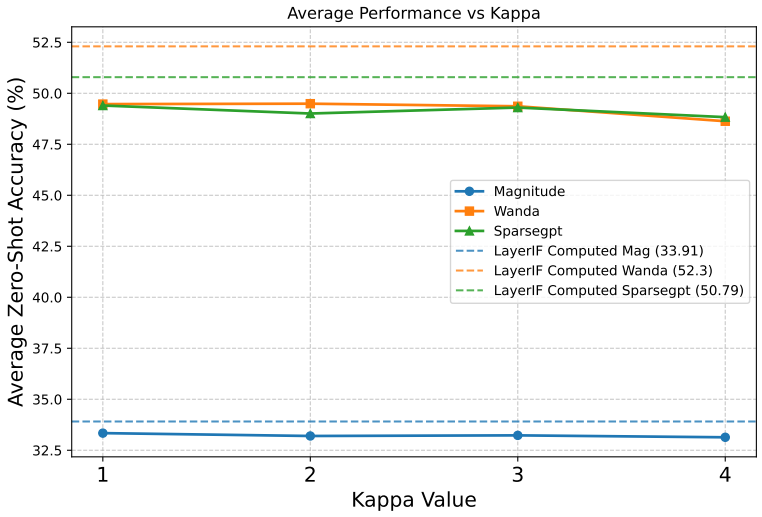
E.4 DIRECT AND PROXY ESTIMATES OF ζ^2 ON AN 8-LAYER MLP

The preceding ablations vary the optimization programs while keeping the score pipeline fixed. We now examine that pipeline directly on an 8-layer multilayer perceptron, where $\zeta_k^2(\tau) = g_k^\top \widetilde{H}_{kk}^{-1} g_k$ can be computed and compared with scalable alternatives. The damping parameter τ is central to this comparison: as τI increasingly dominates H_{kk} in Eq. (7),

$$\widetilde{H}_{kk}^{-1} \approx \tau^{-1} I, \qquad \zeta_k^2(\tau) \approx \tau^{-1} \|g_k\|_2^2,$$



(a)



(b)

Figure 5: (a) Performance breakdown for the three pruning types across varying values of κ (b) Across the pruning types, an increase in κ doesn't seem to be helpful when compared to the LayerIF baselines.

so the regularized second-order score approaches a rescaled gradient-norm ranking. Tables 8–12 show how the proxy correlations evolve across this regime.

The results reveal a damping-dependent transition. At small-to-moderate damping, the LayerIF-style proxy is most strongly

Table 8: Tau / damping = 0.001

Proxy/Baseline	Spearman	Kendall
LayerIF_style	0.659 ± 0.224	0.571 ± 0.210
diag_fisher	0.484 ± 0.204	0.381 ± 0.187
grad_norm	0.611 ± 0.157	0.452 ± 0.135

Table 9: Tau / damping = 0.01

Proxy/Baseline	Spearman	Kendall
LayerIF_style	0.754 ± 0.185	0.643 ± 0.210
diag_fisher	0.349 ± 0.238	0.310 ± 0.187
grad_norm	0.706 ± 0.137	0.571 ± 0.117

Table 10: Tau / damping = 0.1

Proxy/Baseline	Spearman	Kendall
LayerIF_style	0.881 ± 0.168	0.833 ± 0.236
diag_fisher	0.214 ± 0.418	0.167 ± 0.379
grad_norm	0.690 ± 0.085	0.571 ± 0.058

Table 11: Tau / damping = 1

Proxy/Baseline	Spearman	Kendall
LayerIF_style	0.738 ± 0.178	0.619 ± 0.221
diag_fisher	0.381 ± 0.152	0.333 ± 0.121
grad_norm	0.825 ± 0.119	0.738 ± 0.121

Table 12: Tau / damping = 2

Proxy/Baseline	Spearman	Kendall
LayerIF_style	0.595 ± 0.185	0.476 ± 0.168
diag_fisher	0.444 ± 0.011	0.381 ± 0.034
grad_norm	0.929 ± 0.034	0.833 ± 0.067

aligned with the direct regularized score, whereas at larger damping the gradient norm becomes the closest proxy, as predicted by $\widetilde{H}_{kk}^{-1} \approx \tau^{-1} I$. This diagnostic does not establish proxy equivalence for LLMs, but it clarifies when an influence-style proxy can retain information beyond raw gradient magnitude.

Having examined both program hyperparameters and score proxies, we finally assess whether the observed allocation differences are statistically distinguishable under the current evaluation budget.

F STATISTICAL SIGNIFICANCE AND INTERPRETATION

To assess the reliability of the allocation comparison, we performed paired t-tests between the proposed MDL-inspired decision rule and LayerIF on the matched model–dataset evaluations reported below. Because the two methods use the same influence-style layer-score inputs, this analysis isolates the difference between their allocation rules rather than the quality of the underlying sensitivity estimator.

Table 13: Paired comparisons between LayerIF and the proposed allocation rule. None of the reported differences is significant at the 0.05 level.

Algorithm (Mistral)	Average (%)	p-value
LayerIF (+ve)	83.39	0.1906
MDL (+ve)	84.06	–
Mean diff	0.67	–

Algorithm (Mistral)	Average (%)	p-value
LayerIF (All)	80.41	0.1917
MDL (All)	83.07	–
Mean diff	2.66	–

Algorithm (Gemma)	Average (%)	p-value
LayerIF (All)	87.46	0.4162
MDL (All)	87.52	–
Mean diff	0.07	–

The differences are not statistically significant at the 0.05 level, so the experiments support comparability rather than a universal accuracy improvement. This outcome should be interpreted in the context of the design: our method and LayerIF intentionally share the same influence-style layer scores, and the comparison therefore isolates the decision rule. Under the current evaluation budget, replacing the LayerIF knapsack heuristic with the convex MDL-inspired program preserves competitive empirical performance while adding an explicit risk–complexity objective, global budget feasibility, and the

transfer-stability guarantee proved above.

G LIMITATIONS

The quadratic degradation model $\psi(\rho) = \rho^2$ may underestimate pruning sensitivity in architectures with highly heterogeneous layer widths, as seen in the Gemma-7B Wanda results. The Hessian surrogate $\widetilde{H}_{kk}$ requires a curvature approximation (e.g., diagonal Fisher or K-FAC) whose quality affects the accuracy of ζ_k^2. The experiments instantiate the programs with LayerIF-derived proxy weights rather than directly computed Newton-decrement gains. They therefore evaluate the convex decision rule, but do not directly establish the empirical quality of the proposed curvature estimator.

H FUTURE WORK

Natural extensions include: (i) richer degradation penalties ψ calibrated to specific pruning methods; (ii) joint optimization of allocation and pruning in a single program; and (iii) online updating of ζ_k^2 during fine-tuning to track curvature drift adaptively.

This concludes the theoretical and empirical supplement. We finish with the required disclosure concerning the use of language-model tools during preparation.

I LLM USE DISCLOSURE

We employed Large Language Models (LLMs) to refine the text for grammar and clarity. Additionally, LLMs were used for debugging the published codebases of Askari et al. [2025], Kwon et al. [2023] and in writing visualization code. We confirm that LLMs were not used to implement any of the core algorithms or methodologies proposed in this work.