OPPORTUNITY, RECOVERABILITY, AND HARDWARE REALIZATION IN CONDITIONAL NEURAL COMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Conditional computation assigns different amounts of computation to different inputs. Poor performance can have several causes. The candidate models may not make complementary errors, the router may choose poorly, or dynamic execution may erase the expected computational savings. We study these effects under a common expected-compute budget. Our framework measures oracle opportunity relative to a matched static model and compares it with an input-independent mixture control. It also measures how much of that opportunity a learned router recovers and provides an exact finite-sample accuracy oracle. We test the framework in three settings: a nested CNN on CIFAR-10, independently trained ResNet-18 candidates on CIFAR-100, and an MLP bank on CoverType. Candidate and router training use disjoint data splits. All three candidate banks show substantial oracle opportunity. Recovering it is harder. The prespecified marginal-utility router falls below the static reference on both image tasks and gives only a small gain on CoverType. In a secondary exploratory analysis, a ResNet margin policy frozen on validation data recovers about 10% of the available opportunity. The hardware results reveal a separate limitation. On an RTX 3090, matched MACs do not imply matched latency or energy. The dynamic policy is slower and more energy-consuming than the static medium model, while its comparison with the static large model changes with batch size. Our results prove that we should therefore report candidate opportunity, routing recovery, and hardware realization as separate quantities.

1 INTRODUCTION

Conditional computation replaces a fixed inference graph with an input-dependent allocation of computation. Early-exit systems stop once an intermediate prediction satisfies a confidence rule (Teerapittayanon et al., 2016; Kaya et al., 2019); routing networks skip blocks or select branches (McGill & Perona, 2017; Wu et al., 2018; Wang et al., 2018; Mullapudi et al., 2018); dynamic-width models select channel configurations (Yu et al., 2019; Yu & Huang, 2019; Li et al., 2021). Such systems require more than a favorable average accuracy–FLOPs point. The candidate computations must make complementary errors, the useful computation for a particular input must be identifiable from information available before the routing decision, and the implementation must convert the arithmetic allocation into lower latency or energy.

These conditions are often conflated. A learned router can fail because the candidate set contains little per-example opportunity, because the available representation does not support the required counterfactual prediction, because the fitted router does not recover information that is present, or because dynamic dispatch removes any arithmetic advantage. Conversely, a large label-informed oracle gain does not imply that its allocation is predictable without labels. MAC counts also omit indexing, synchronization, route imbalance, memory traffic, and kernel irregularity.

We evaluate these three questions separately under a common expected-compute constraint: how much opportunity exists in the candidate bank, how much a learned router recovers, and whether the resulting allocation translates into hardware savings. Within learned-router regret, we theoretically distinguish restriction from observing only the routing representation from approximation, estimation, optimization, and calibration error; our experiments do not identify these components separately. Let U_s be the utility of a compute-matched static reference, U_o the utility of a metric-matched

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

label-informed oracle over a fixed candidate bank, and U_r the utility of a learned router. We define oracle opportunity $\mathcal{O} = U_o - U_s$, routing regret $\mathcal{G} = U_o - U_r$, and realized gain

$$\Delta = U_r - U_s = \mathcal{O} - \mathcal{G}. \tag{1}$$

This identity makes explicit that poor learned-router performance does not imply a lack of candidate complementarity. We also compare the oracle with the best input-independent randomized mixture of candidates, which separates per-example allocation value from a better average point on the candidate-bank frontier.

The framework distinguishes privileged label information from information available to a deployable router. For a routing representation Z, the optimal feature-restricted rule selects the candidate with the largest conditional expected utility net of compute cost. The difference between the label-informed and feature-restricted penalized values is a value-of-information gap. For empirical top-1 accuracy, the budgeted deterministic oracle has a simple exact solution: start from the cheapest candidate and purchase the least expensive correctness-producing upgrades until the budget is exhausted. This avoids using cross-entropy routing as a surrogate for an accuracy oracle.

We evaluate three capacity-routing settings. The first uses nested widths of a shared CNN on CIFAR-10 at a target budget of 9.73% of dense MACs. The second uses independently trained 25%, 50%, and 100%-width ResNet-18 bodies on CIFAR-100 behind a common frozen stem, with a target of 27.32%. The third uses independently trained small, medium, and large MLPs on CoverType at a target equal to the medium model, 19.12% of large-model MACs. The image routers observe a pooled stem representation; the CoverType router observes the original preprocessed 54-dimensional input. Candidate training and router-target construction use disjoint data splits.

The experiments exhibit distinct regimes. Exact accuracy opportunity is substantial in all three candidate banks, but the prespecified marginal router is below static execution on both image studies and reproducibly positive on CoverType. Small-model uncertainty is harmful on CIFAR-10 but positive on ResNet and CoverType. A strictly validation-frozen five-seed ResNet margin policy also improves accuracy at approximately the matched-static MAC budget. Under explicit dispatch on an RTX 3090, this policy has higher latency and GPU energy than static medium at all tested batch sizes. Relative to static large, it is dominated at batch size 1; at batch size 64 it trades lower GPU energy for higher latency; and at batch size 256 it forms an intermediate accuracy–latency–energy operating point. Specifically, we make the following contributions:

1. We define a metric-matched opportunity–regret decomposition, an input-independent mixture control, and a hardware realization criterion for conditional computation under an expected-compute constraint.
2. We characterize the optimal feature-restricted rule for the penalized objective and its value-of-information gap, and give an exact finite-sample solver for the deterministic budgeted accuracy oracle.
3. Paired multi-seed experiments across three routing settings, together with a five-seed hardware benchmark, show that substantial instance-dependent opportunity can coexist with weak recovery and hardware behavior that depends on the reference model.

2 Related Work

Conditional routing and adaptive depth. Conditional computation increases effective model capacity without evaluating every parameter on every input (Bengio et al., 2013; Shazeer et al., 2017). Image models have learned residual paths, branches, spatial support, and inference graphs (Figurnov et al., 2017; McGill & Perona, 2017; Wu et al., 2018; Wang et al., 2018; Mullapudi et al., 2018; Veit & Belongie, 2018). These approaches differ in granularity, but share the objective of adapting computational effort to the input rather than executing a fixed graph. Early-exit systems use intermediate confidence estimates to terminate inference before the deepest layers (Teerapittayanon et al., 2016; Bolukbasi et al., 2017; Kaya et al., 2019); related ideas appear in autoregressive models through CALM and LayerSkip (Schuster et al., 2022; Elhoushi et al., 2024). Mixture-of-Depths routes token positions through transformer computation while retaining fixed tensor sizes (Raposo et al., 2024). In these formulations, the main reported quantity is typically the cost–quality tradeoff produced by the learned or prescribed routing rule. This does not by itself distinguish whether a

weak tradeoff is caused by insufficient complementarity among candidate computations, limited information at the routing point, or errors of the fitted router. Our analysis isolates candidate-bank opportunity from the recovery achieved by an evaluated routing policy and from the systems cost of dynamic execution. The further decomposition of routing regret into feature-information restriction and fitting error is theoretical; the experiments do not identify those two terms separately.

Model selection, cascades, and dynamic width. Learning-to-defer assigns inputs between decision makers and therefore makes performance depend on the information available before deferral (Madras et al., 2018; Mozannar & Sontag, 2020). Model cascades replace the external decision maker with predictors of different cost, so the relevant routing quantity becomes the conditional advantage of one predictor over another rather than unconditional model quality. FrugalGPT and RouteLLM learn query-dependent selection between language models (Chen et al., 2024; Ong et al., 2025), illustrating the same statistical problem at the model level. Confidence-based cascades and learned selectors can improve average cost when model errors are sufficiently structured, but their observed gains combine candidate complementarity with the accuracy of the routing rule. Our oracle and mixture controls isolate these effects by comparing input-dependent selection with the best input-independent allocation under the same expected-compute constraint.

A related line of work exposes multiple capacities within a single parameterization. Slimmable and universally slimmable networks execute different channel widths from shared weights (Yu et al., 2019; Yu & Huang, 2019); once-for-all networks train a supernetwork from which subnetworks are selected for different deployment constraints (Cai et al., 2020). Dynamic Slimmable Networks add an input-dependent gate (Li et al., 2021), while SortedNet and MatFormer extend nested parameterization across architectural dimensions (Valipour et al., 2024; Devvrit et al., 2024). Shared-weight designs reduce storage and can make switching between capacities inexpensive, but they also couple the candidate functions through joint training. Our experiments include both regimes. The CIFAR-10 candidate bank uses nested widths from a shared convolutional model, whereas the ResNet-18 and CoverType banks contain independently trained candidates. The latter remove shared-body interference as a necessary explanation for routing behavior, at the cost of larger stored parameter counts.

Confidence, calibration, and counterfactual utility. Confidence and uncertainty are common routing signals because they are available without labels at inference time. Their interpretation, however, differs from the quantity required for optimal capacity allocation. A low-confidence prediction indicates uncertainty under one candidate but does not determine whether a more expensive candidate will correct the error. Conversely, a confident prediction may still be wrong and correctable by additional computation. Calibration methods address the relation between confidence and predictive correctness, but do not directly estimate the counterfactual gain of switching candidates. We therefore distinguish generic difficulty scores, such as entropy, margin, and maximum probability, from predictors trained explicitly on candidate-specific counterfactual losses or correctness. This distinction is central to the empirical analysis: uncertainty can be informative for routing even when it is only weakly aligned with the marginal utility of additional computation.

Efficiency measurement. Operation counts, parameter counts, latency, throughput, memory, and energy measure different properties and need not rank models identically (Dehghani et al., 2022). Dynamic execution introduces costs absent from nominal MAC accounting, including feature extraction for routing, indexing, route partitioning, synchronization, branch imbalance, and reduced kernel regularity. These effects can dominate when route groups are small or when most inputs follow the same path. MoE systems such as Tutel and MegaBlocks explicitly address communication, padding, and sparse-kernel overheads in routed workloads (Hwang et al., 2022; Gale et al., 2023). Direct energy measurement introduces additional dependence on hardware utilization, batch size, memory traffic, and measurement protocol (Henderson et al., 2020; Schwartz et al., 2020). Consequently, arithmetic savings are not sufficient evidence of deployment savings. We report active MACs and stored parameters separately from measured latency and energy, and compare dynamic execution against the same matched static reference used in the utility analysis. This separates statistical routing quality from the efficiency of a particular execution system.

3 FEASIBILITY FRAMEWORK

Let $(X, Y) \sim P$ with $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. A candidate bank contains K predictors $f_k : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ indexed by $k \in \mathcal{K} = \{1, \ldots, K\}$. Candidate k has nonnegative active-compute cost c_k. A deterministic routing policy $\pi : \mathcal{Z} \to \mathcal{K}$ observes $Z = \phi(X)$ and selects $f_{\pi(Z)}$. The feature extractor ϕ may itself have cost $a_\phi(X)$. For a bounded utility $u(f_k(X), Y) \in [0, 1]$, we define

$$U(\pi) = \mathbb{E}[u_{\pi(Z)}], \qquad C(\pi) = \mathbb{E}[a_\phi(X) + c_{\pi(Z)}].$$

Classification accuracy corresponds to $u(f_k(X), Y) = \mathbf{1}\{\arg\max_j f_k(X)_j = Y\}$. A policy is compute-feasible at budget b when $C(\pi) \le b$. A static reference s_b is trained independently subject to approximately the same active-compute budget; its utility is $U_s = U(s_b)$.

The diagnostic label-informed oracle may depend on (X, Y) and solves

$$\pi_o \in \arg \max_{\pi:\, C(\pi) \le b} U(\pi), \tag{2}$$

over the fixed candidate bank. Because Y is observed, π_o is not deployable. It quantifies error complementarity under the budget rather than the performance of a realizable inference procedure.

Definition 1 (Opportunity, regret, and recovery)**.** *For a learned policy π_r with utility U_r, define*

$$\mathcal{O} = U_o - U_s \qquad \textit{(oracle opportunity)}, \tag{3}$$

$$\mathcal{G} = U_o - U_r \qquad \textit{(routing regret)}, \tag{4}$$

$$\Delta = U_r - U_s \qquad \textit{(realized gain)}. \tag{5}$$

When $\mathcal{O} > 0$, the recovery fraction is $q = \Delta/\mathcal{O}$.

Direct substitution gives $\Delta = \mathcal{O} - \mathcal{G}$; hence $\Delta > 0$ iff $\mathcal{G} < \mathcal{O}$. The oracle must use the evaluation metric itself; cross-entropy assignments are retained only as feasible witnesses.

Input-independent control. Let $U_k = \mathbb{E}[u_k]$ and let Δ_K be the probability simplex. To separate instance-dependent selection from the aggregate quality of the candidate bank, we define the best input-independent randomized policy as

$$U^\star_{\text{mix}} = \max_{p \in \Delta_K} \sum_k p_k U_k \quad \text{s.t.} \quad \sum_k p_k c_k \le b. \tag{6}$$

For a metric-matched oracle, the opportunity decomposes as

$$\mathcal{O} = (U_o - U^\star_{\text{mix}}) + (U^\star_{\text{mix}} - U_s), \tag{7}$$

where the first term measures instance-dependent selection within the bank and the second measures the bank–static frontier gap. Because the independently trained static reference need not belong to the candidate bank, $U^\star_{\text{mix}} - U_s$ may be negative, allowing $U_o - U^\star_{\text{mix}}$ to exceed O without contradiction.

Exact deterministic accuracy oracle. For an evaluation sample $\{(x_i, y_i)\}_{i=1}^n$, let $a_{ik} \in \{0, 1\}$ indicate whether candidate k is correct on example i.

Theorem 1 (Exact empirical accuracy oracle)**.** *Assume a unique minimum-cost candidate k_0 and $b \ge c_{k_0}$. For every example on which k_0 ($a_{ik_0} = 0$) is wrong but at least one candidate is correct ($\max_k a_{ik} = 1$), let*

$$d_i = \min_{k: a_{ik} = 1} (c_k - c_{k_0}), \qquad k^\star_i \in \arg \min_{k: a_{ik} = 1} c_k. \tag{8}$$

Order the finite values as $d_{(1)} \le \cdots \le d_{(M)}$ and let

$$m^\star = \max \left\{ m : \; n c_{k_0} + \sum_{j=1}^{m} d_{(j)} \le nb \right\}. \tag{9}$$

An exact deterministic accuracy oracle assigns the $m^\star$ examples with smallest incremental costs to their $k^\star_i$ actions and assigns k_0 otherwise. The resulting deterministic assignment maximizes empirical accuracy; among accuracy-maximizing assignments it has minimum total cost.

The proof is in Appendix A.4.1. In all reported seeds the budget admits every correctness-producing upgrade, so the exact oracle equals union accuracy.

Let $\Pi_Z(b)$ denote the set of budget-feasible policies measurable with respect to $Z = \phi(X)$, and define the best feature-restricted utility

$$U_Z^\star = \sup_{\pi \in \Pi_Z(b)} U(\pi). \tag{10}$$

Theorem 2 (Optimal feature-restricted rule for the penalized objective)**.** *Let $u_k = u(f_k(X), Y)$ be integrable and define $m_k(z) = \mathbb{E}[u_k \mid Z = z]$. For any multiplier $\lambda \geq 0$, an optimal Z-measurable selector for the penalized objective $\mathbb{E}[u_{\pi(Z)} - \lambda c_{\pi(Z)}]$ is*

$$\pi_Z^\lambda(z) \in \arg\max_{k \in \mathcal{K}} \{m_k(z) - \lambda c_k\}. \tag{11}$$

Its value is

$$V_Z(\lambda) = \mathbb{E}\left[\max_k \{m_k(Z) - \lambda c_k\}\right]. \tag{12}$$

If $W = (X, Y)$ denotes the privileged information available to a label-informed selector and $Q_k = u_k - \lambda c_k$, then the value lost by observing only Z is

$$\mathcal{J}_Z(\lambda) = V_W(\lambda) - V_Z(\lambda) = \mathbb{E}[\max_k Q_k] - \mathbb{E}\left[\max_k \mathbb{E}[Q_k \mid Z]\right] \geq 0. \tag{13}$$

The gap is zero if and only if a Z-measurable selector is almost surely optimal under the privileged information, up to ties.

A complete proof is given in Appendix A.5.1.

For any learned Z-measurable policy $\pi_r \in \Pi_Z(b)$, adding and subtracting $U_Z^\star$ yields

$$\mathcal{G} = \underbrace{U_o - U_Z^\star}_{\mathcal{G}_{\text{info}}} + \underbrace{U_Z^\star - U_r}_{\mathcal{G}_{\text{est}}}. \tag{14}$$

Here G_{info} is the population-level restriction induced by observing only Z, while G_{est} contains approximation, estimation, optimization, and calibration error. We estimate neither term separately. Consequently, the empirical routing-regret results reported below estimate only their sum G; they must not be interpreted as separate measurements of information limitation and fitting error.

If Z_1 is measurable with respect to Z_2 and both policy classes use identical total-cost accounting, then $\Pi_{Z_1}(b) \subseteq \Pi_{Z_2}(b)$ and consequently

$$U_{Z_1}^\star \leq U_{Z_2}^\star. \tag{15}$$

The requirement of identical cost accounting is material. A deeper feature can enlarge the policy class while consuming most of the target budget.

Hardware realization. If a reported route cost omits a nonnegative feature or probe cost $a_\phi(X)$, then

$$C_{\text{true}}(\pi) = C_{\text{reported}}(\pi) + \mathbb{E}[a_\phi(X)] \geq C_{\text{reported}}(\pi). \tag{16}$$

Thus a policy feasible under omitted-probe accounting need not be feasible under total-cost accounting. For a hardware quantity H such as latency or energy, define

$$\rho_H = \frac{H(\pi_r)}{H(s_b)}. \tag{17}$$

Values below one indicate a measured improvement. Relative to the chosen static reference, simultaneous utility and hardware dominance requires $\Delta > 0$ and $\rho_H < 1$. For a metric-matched oracle, the first condition is equivalent to $\mathcal{G} < \mathcal{O}$.

Equal MAC counts need not imply equal hardware cost: dispatch, kernel shape, memory traffic, occupancy, and route imbalance can change latency and energy even at matched active compute.

4 EXPERIMENTAL PROTOCOL

Data and candidate banks. CIFAR-10 and CIFAR-100 use a fixed seed-1729 split of the $50{,}000$ training images into $40{,}000$ candidate-training, $5{,}000$ router-training, and $5{,}000$ validation examples; the standard $10{,}000$ images are held out for test. Candidate parameters are frozen before router targets are constructed on the disjoint router split. The nested-CNN study uses three contiguous-width candidates at a target of 0.0973 dense MACs and seeds 0–4. The ResNet study uses independently trained 25%, 50%, and 100%-width bodies behind a frozen stem at target 0.2732; seeds 0–2 are confirmatory and seeds 3–4 are locked-protocol replications. Image routers observe the 64-dimensional pooled stem representation. CoverType uses a fixed stratified $60/15/10/15\%$ candidate/router/validation/test split of all $581{,}012$ examples. Its independently trained small, medium, and large MLPs use $8{,}000$, $40{,}576$, and $212{,}224$ MACs; the medium candidate is the static reference and target budget (0.1912 of large-model MACs). The router observes the preprocessed 54-dimensional input. CoverType uses seeds 0–4. Neural routers are trained for 200 epochs with AdamW; full architectures and optimization details are in Appendix B.

Marginal-utility routing. The evaluated routers are diagnostic policies rather than proposed routing architectures. Let the three capacities be indexed by s, m, l, with $c_s < c_m < c_l$. For each candidate, let $\ell_k(x, y)$ denote its per-example cross-entropy. Relative to the medium candidate, define the downsize penalty and upgrade benefit per normalized MAC as

$$g_{\text{down}}(x, y) = \frac{\ell_s(x, y) - \ell_m(x, y)}{c_m - c_s}, \quad g_{\text{up}}(x, y) = \frac{\ell_m(x, y) - \ell_l(x, y)}{c_l - c_m}.$$

A marginal-utility predictor $r_\theta(Z) = (\hat{g}_{\text{down}}, \hat{g}_{\text{up}})$ maps the routing representation $Z = \phi(X)$ to these quantities. The corresponding candidate-loss vector, expressed relative to the medium candidate, is

$$\widehat{\boldsymbol{\ell}}(x) = \begin{bmatrix} \hat{g}_{\text{down}}(c_m - c_s) \\ 0 \\ -\hat{g}_{\text{up}}(c_l - c_m) \end{bmatrix}. \tag{18}$$

Proposition 13 shows that exact conditional marginal utilities recover the conditional candidate losses up to an additive term common to all actions; hence they induce the same cost-sensitive route ordering. The primary marginal predictor is a one-hidden-layer MLP of width 16 with SiLU activation. Features and targets are standardized using the router-training split, and each marginal target is clipped at its 1st and 99th training percentiles. The training objective combines Huber regression with pairwise ranking:

$$\mathcal{L}_{\text{router}} = \frac{1}{2} \sum_{q \in \{\text{down}, \text{up}\}} \text{Huber}_1(\hat{g}_q, g_q) + \beta \mathcal{L}_{\text{rank}}, \tag{19}$$

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j,q) \in \mathcal{P}} \log(1 + \exp[-s_{ijq}(\hat{g}_{iq} - \hat{g}_{jq})]), \tag{20}$$

$$s_{ijq} = \text{sign}(g_{iq} - g_{jq}), \tag{21}$$

where $\mathcal{P}$ contains training-example pairs whose target difference has magnitude greater than 10^{-3}, and $\beta = 0.25$.

Routing policies and budget calibration. The fixed suite contains linear and MLP marginal predictors, candidate-loss, correctness, and oracle-route predictors, histogram gradient boosting, and small-model entropy, margin, and maximum-probability controls. Supervised targets use only the router-training split. Each policy produces action scores $\widehat{v}_k(x)$ and selects

$$\hat{k}(x; \lambda) = \arg \min_{k \in \{s,m,l\}} \{\widehat{v}_k(x) + \lambda c_k\}. \tag{22}$$

For loss-prediction policies, $\widehat{v}_k$ is the predicted candidate loss; for marginal routing, it is obtained from equation 18. The multiplier λ therefore implements a common cost-sensitive decision rule across predictor types.

For a controlled matched-budget diagnostic, we choose λ from predicted scores and known costs on the unlabeled evaluation batch. This uses no test labels but is transductive, and therefore does

Table 1: Exact opportunity and input-independent control. Accuracies are percentages. Opportunity intervals are paired seed-level 95% Student t intervals. "Inst." denotes $U_o - U^\star_{\text{mix}}$. Because the separately trained static reference need not belong to the candidate bank, Inst. can exceed O when $U^\star_{\text{mix}} < U_s$.

Study	Static	$U^\star_{\text{mix}}$	Oracle	$\mathcal{O}$	Inst.
CNN / CIFAR-10	80.56	79.82	89.39	8.83 [8.50, 9.17]	9.57
ResNet / CIFAR-100	73.12	73.12	84.93	11.81 [10.98, 12.64]	11.81
MLP / CoverType	89.04	89.04	95.84	6.80 [6.60, 6.99]	6.80

not represent a fully frozen deployment policy. We separately report a non-transductive evaluation in which λ is selected on validation data and applied unchanged at test time; all deployment claims rely on this validation-frozen protocol. For the secondary five-seed ResNet margin analysis, both uncertainty-score normalization and the budget multiplier are estimated exclusively from validation data and frozen before test evaluation. The margin policy was selected for this five-seed follow-up after inspection of the initial three-seed uncertainty-control suite. Accordingly, its five-seed accuracy interval is an exploratory, descriptive interval and is not adjusted for policy selection. Validation freezing prevents test-time normalization or calibration from using test inputs or labels, but does not convert this post-specified policy comparison into a confirmatory result. Neural-router MACs are included in route costs. Uncertainty controls first execute the small candidate, whose probe cost is charged when execution continues. Gradient-boosted traversal cost is omitted, so that method is treated only as an information probe. Detailed objectives and router hyperparameters are given in Appendix B.

Statistics and hardware. We report paired seed-level 95% Student t intervals. The clipped marginal router is prespecified; broader router-suite and uncertainty comparisons are exploratory. ResNet seeds 0–2 form the confirmatory study, while seeds 3–4 are locked-protocol replications. The five-seed margin result is post-specified and its interval is descriptive rather than confirmatory. The hardware benchmark evaluates the strictly validation-frozen ResNet margin policy for all five seeds against the corresponding static candidates at batch sizes 1, 64, and 256. Measurements use one RTX 3090 in FP32 with TF32 disabled, deterministic GPU-resident inputs, 20 warm-up and 200 timed iterations over five repetitions, and randomized interleaved method order. Primary GPU energy uses the NVML cumulative-energy counter; CodeCarbon provides a software cross-check. Full systems metadata are in Appendix D.4.

5 Results

Opportunity exists in all three candidate banks. Table 1 separates exact opportunity from the input-independent control. The exact oracle improves over the matched static reference by 8.83 points on CIFAR-10, 11.81 points on the three-seed CIFAR-100 study, and 6.80 points on CoverType. On CIFAR-10, the decomposition is $8.83 = 9.57 - 0.74$ points, up to rounding: the best input-independent candidate-bank mixture is 0.74 points below the separately trained static CNN. The oracle-over-mixture gap can therefore exceed the static-referenced opportunity without contradiction. On CIFAR-100 and CoverType, the input-independent optimum selects the medium candidate and coincides with the static reference; their entire 11.81- and 6.80-point opportunities are instance-dependent. The ResNet opportunity remains 11.674 points with interval $[11.284, 12.064]$ after adding seeds 3 and 4. The exact solver equals candidate-union accuracy in every reported seed. Its minimum-cost assignments use only 5.99%, 14.98%, and 9.48% of dense/large-model MACs for CIFAR-10, CIFAR-100, and CoverType, respectively, all below the nominal budgets. Pairwise correctness statistics are reported in Appendix D.

Recoverability differs across routing signals. Figure 1 compares the prespecified clipped marginal router across studies. Its realized gains are -1.29, -0.83, and $+0.225$ points on CIFAR-10, CIFAR-100, and CoverType. The CoverType gain is positive in all five seeds, with interval $[0.138, 0.312]$, but recovers only 3.3% of the exact opportunity. Across five ResNet seeds, the same router gains -0.834 points with interval $[-0.970, -0.698]$; validation-frozen calibration gives essentially the same result.

Uncertainty provides a contrasting exploratory signal. All three controls are harmful on CIFAR-10. On the three initial ResNet seeds they have positive means but wide pointwise intervals. In a post-

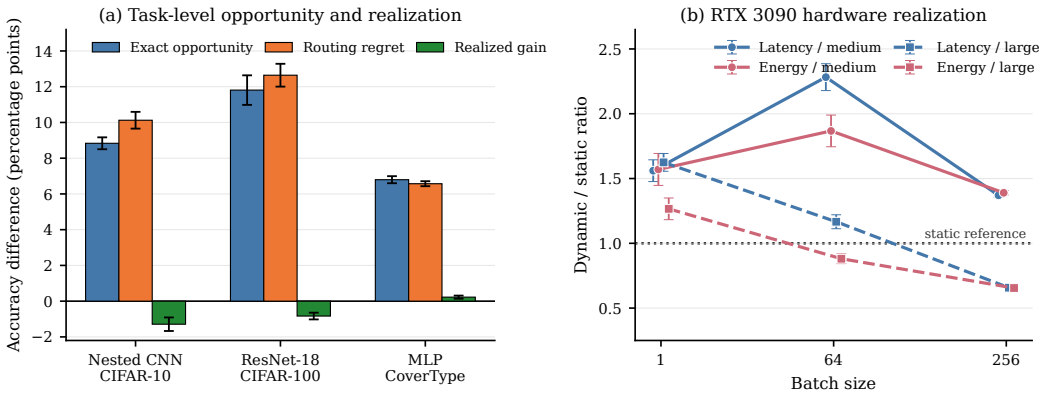


Figure 1: **Opportunity, routing, and hardware realization. Left:** Exact opportunity, routing regret, and realized gain for the clipped MLP-16 marginal-utility router; bars show paired-seed means with 95% Student t intervals. **Right:** Five-seed latency and primary NVML GPU-energy ratios of the strictly validation-frozen ResNet margin policy on an RTX 3090, relative to static medium and static large. Error bars show paired seed-level 95% Student t intervals, and the horizontal line marks ratio 1. Dynamic execution has higher latency and GPU energy than static medium at every batch size. Relative to static large, it is dominated at batch 1, is slower but less energy-consuming at batch 64, and is both faster and less energy-consuming at batch 256.

Table 2: Selected routing results. Δ is gain over the matched static reference; recovery is relative to exact opportunity. "Val." freezes budget calibration on validation data; "strict Val." also freezes uncertainty-score normalization. Other rows use unlabeled batch projection. Secondary-policy intervals are unadjusted pointwise intervals.

Study	Policy	Seeds	Δ (pp)	Recovery
CIFAR-10	Marginal, clipped	5	$-1.29\,[-1.67, -0.91]$	< 0
CIFAR-100	Marginal, clipped	3	$-0.83\,[-1.02, -0.65]$	< 0
CIFAR-100	Marginal, clipped	5	$-0.834\,[-0.970, -0.698]$	< 0
CIFAR-100	Margin, strict Val.	5	$+1.192\,[0.498, 1.886]$	10.13%
CoverType	Marginal, clipped	5	$+0.225\,[0.138, 0.312]$	3.3%
CoverType	Entropy	5	$+0.546\,[0.480, 0.612]$	8.0%

specified five-seed follow-up, the strictly validation-frozen margin policy gains 1.192 points, recovers 10.13% of the exact opportunity, and uses a mean MAC ratio of 0.2722 versus the 0.2732 target. On CoverType, entropy has the largest observed uncertainty- control gain, 0.546 points, corresponding to 8.0% recovery. These secondary policy comparisons are descriptive and are not adjusted for selection among the evaluated controls. The broader router suite sharpens the interpretation rather than changing it. Larger candidate-loss, correctness, and route-imitation networks remain below static accuracy in both image studies; several are also negative on CoverType despite the positive marginal and uncertainty policies. The full eleven-policy comparison is reported in Appendix D. Higher predictor capacity is therefore not sufficient; the policy must estimate a cost-sensitive ordering of counterfactual outcomes at the routing boundary. A later ResNet probe provides only modest utility association and no stable gain. Its mean accuracy difference is about -0.82 points with a wide interval, and obtaining its penultimate representation already requires the medium computation. It is therefore a representation diagnostic, not a deployable policy at the displayed selected-route cost.

Hardware realization is reference- and batch-dependent. The five-seed hardware benchmark evaluates the same strictly validation-frozen margin policy used above. Its mean accuracy is 74.498%, compared with 73.306% for static medium and 77.588% for static large. Its mean MAC ratio is 0.2722, compared with 0.2732 and 1.00, respectively. Relative to static medium, the latency ratios are 1.561, 2.283, and 1.370 at batch sizes 1, 64, and 256. The corresponding primary NVML GPU-energy ratios are 1.570, 1.867, and 1.390. Thus, the 1.192-point accuracy gain over static medium incurs higher realized latency and GPU energy at every evaluated batch size despite approximately matched MACs. CodeCarbon task-level GPU energy gives corresponding ratios of 1.517, 1.920, and 1.393, preserving the conclusion.

Relative to static large, the latency ratios are 1.625, 1.166, and 0.658, while the primary GPU-energy ratios are 1.266, 0.881, and 0.655. At batch 1, static large is more accurate, faster, and less energy-consuming. At batch 64, static large remains more accurate and faster, whereas the dynamic policy uses 11.9% less GPU energy. At batch 256, the dynamic policy reduces latency by 34.2% and GPU energy by 34.5% relative to static large. Therefore, at batch 64 the ordering depends on whether latency or GPU energy is prioritized, while at batch 256, the dynamic policy lies between static medium and static large in the measured accuracy–latency–energy comparison.

6 Discussion and Limitations

Opportunity and recoverability are distinct. The exact oracle and input-independent control show substantial per-example allocation value in all three banks, yet the same clipped marginal router is negative in the two image studies and positive on CoverType. The result rules out both simple interpretations: large oracle opportunity neither guarantees routing success nor implies that the evaluated routing procedure must fail. The CoverType result does not establish that tabular data are intrinsically easier to route; dataset, architecture, dimensionality, sample size, error structure, and routing representation change together. It establishes only that recoverability varies across the evaluated settings. Likewise, a negative result for the evaluated router does not establish an information-theoretic impossibility. The image experiments show that the evaluated policies, when given the pooled stem representation, recover little useful task gain. They do not establish whether the remaining routing gap is caused primarily by limited information in that representation or by approximation, estimation, optimization, or calibration error. Several CoverType policies recover a small but reproducible amount from the original input. Even the strongest positive exploratory policies recover only about 8–10% of the exact opportunity, leaving most label-informed allocation value unrealized.

Difficulty scores can help without estimating marginal utility. Small-model uncertainty measures generic difficulty rather than the counterfactual benefit of increasing capacity. This distinction is visible empirically: uncertainty is harmful on CIFAR-10, positive on ResNet, and consistently positive on CoverType. The strictly validation-frozen ResNet margin result shows that a heterogeneous positive-utility policy can be obtained without test-batch normalization or calibration, but it remains a secondary analysis and should not be conflated with the prespecified supervised marginal router.

Scope and systems limits. The study covers three controlled classification settings rather than large early-exit, transformer, or MoE deployments. Some diagnostic costs are omitted as stated in the appendix. Hardware is measured on one RTX 3090 and one explicit-dispatch implementation; the five seeds quantify policy variation, not hardware diversity. Host-to-device transfer is excluded, clocks are unfixed, and the implementation uses no compilation, CUDA graphs, fused routing, or asynchronous execution. NVML GPU energy is cross-checked with CodeCarbon, but both depend on NVIDIA telemetry. The hardware results therefore characterize this platform and implementation rather than an accelerator-independent ordering.

7 Conclusion

Our experiments separate three questions that are often bundled together in evaluations of conditional computation: whether useful input-dependent allocation exists, whether a deployable router can identify it, and whether the resulting computation pattern improves measured hardware cost. The candidate banks considered here offer substantial accuracy opportunity 8.83, 11.81, and 6.80 percentage points on CIFAR-10, CIFAR-100, and CoverType but the prespecified marginal router recovers little of it and is worse than the matched static reference on both image tasks. A post-specified, validation-frozen ResNet margin policy performs better, gaining 1.192 points and recovering 10.13% of the available opportunity. This statistical gain does not translate directly into hardware savings: relative to static medium, the dynamic implementation remains slower and more energy-consuming at every tested batch size, whereas the comparison with static large changes with batch size. These results show why MAC-matched accuracy alone is insufficient to characterize conditional computation. Candidate complementarity, routing performance, and realized hardware cost should be measured separately.

Reproducibility Statement

The candidate/router data separation, seed sets, budget rules, model and router hyperparameters, exact-oracle construction, per-seed decompositions, calibration sensitivity, and hardware metadata are specified in the main text and appendices. Complete proofs are included in the theoretical appendix. An anonymized repository containing the training, aggregation, routing-diagnostic, hardware-benchmarking, and figure-generation code is available at https://anonymous.4open.science/r/conditional_routing_submission_artifact-0B2E/.

AI Use Statement

Generative AI tools were used to assist with implementation and debugging of analysis scripts and language editing of the manuscript. The authors reviewed the experimental procedures, code, mathematical statements, and manuscript text and take responsibility for the final content of the submission.

References

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. volume abs/1308.3432, 2013. URL http://arxiv.org/abs/1308.3432.

Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 527–536. JMLR.org, 2017.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once for all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/pdf/1908.09791.pdf.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=cSimKw5p6R. Featured Certification.

Mostafa Dehghani, Yi Tay, Anurag Arnab, Lucas Beyer, and Ashish Vaswani. The efficiency misnomer. In *ICLR*, 2022. URL https://openreview.net/forum?id=iulEMLYh1uR.

Devvrit, Sneha Kudugunta, Aditya Kusupati, Tim Dettmers, Kaifeng Chen, Inderjit Dhillon, Yulia Tsvetkov, Hannaneh Hajishirzi, Sham Kakade, Ali Farhadi, and Prateek Jain. Matformer: Nested transformer for elastic inference. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 140535–140564. Curran Associates, Inc., 2024. doi: 10.52202/079017-4461. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/fe066022bab2a6c6a3c57032a1623c70-Paper-Conference.pdf.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A Aly, Beidi Chen, and Carole-Jean Wu. Layerskip: Enabling early exit inference and self-speculative decoding. In *Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://api.semanticscholar.org/CorpusID:269362647.

Michael Figurnov, Maxwell D. Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry P. Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. pp. 1790–1799, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/papers/Figurnov_Spatially_Adaptive_Computation_CVPR_2017_paper.pdf.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. Megablocks: Efficient sparse training with mixture-of-experts. In D. Song, M. Carbin, and T. Chen

(eds.), *Proceedings of Machine Learning and Systems*, volume 5, pp. 288–304. Curran, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/file/5a54f79333768effe7e8927bcccffe40-Paper-mlsys2023.pdf.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, Joe Chau, Peng Cheng, Fan Yang, Mao Yang, and Yongqiang Xiong. Tutel: Adaptive mixture-of-experts at scale. July 2022. URL https://www.microsoft.com/en-us/research/publication/tutel-adaptive-mixture-of-experts-at-scale/.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3301–3310. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kaya19a.html.

Changlin Li, Guangrun Wang, Bing Wang, Xiaodan Liang, Zhihui Li, and Xiaojun Chang. Dynamic slimmable network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8607–8617, June 2021.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 6150–6160, Red Hook, NY, USA, 2018. Curran Associates Inc.

Mason McGill and Pietro Perona. Deciding how to decide: Dynamic routing in artificial neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2363–2372. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/mcgill17a.html.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Ravi Teja Mullapudi, William R. Mark, Noam Shazeer, and Kayvon Fatahalian. Hydranets: Specialized dynamic architectures for efficient inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

David Raposo, Samuel Ritter, Blake A. Richards, Timothy P. Lillicrap, Peter C. Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *ArXiv*, abs/2404.02258, 2024. URL https://api.semanticscholar.org/CorpusID:268876220.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Commun. ACM*, 63(12): 54–63, November 2020. ISSN 0001-0782. doi: 10.1145/3381831. URL https://doi.org/10.1145/3381831.

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.

Surat Teerapittayanon, Bradley McDanel, and H.T. Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2464–2469, 2016. doi: 10.1109/ICPR.2016.7900006.

Mojtaba Valipour, Mehdi Rezagholizadeh, Hossein Rajabzadeh, Parsa Kavehzadeh, Marzieh Tahaei, Boxing Chen, and Ali Ghodsi. Sortednet: A scalable and generalized framework for training modular deep neural networks. 2024. URL https://arxiv.org/abs/2309.00255.

Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, pp. 3–18, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01245-8. doi: 10.1007/978-3-030-01246-5_1. URL https://doi.org/10.1007/978-3-030-01246-5_1.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, pp. 420–436, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01260-1. doi: 10.1007/978-3-030-01261-8_25. URL https://doi.org/10.1007/978-3-030-01261-8_25.

Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S. Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8817–8826, 2018. doi: 10.1109/CVPR.2018.00919.

Jiahui Yu and Thomas Huang. Universally slimmable networks and improved training techniques. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1803–1811, 2019. doi: 10.1109/ICCV.2019.00189.

Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gMCsAqY7.

A Proofs and Additional Theory

This appendix gives complete proofs of the results used in the main text. The candidate set is finite, with indices $k \in [K]$, utilities $u_k(x, y)$, and costs $c_k \geq 0$. A routing policy π is feasible at budget b when its expected total cost is at most b. All expectations below are assumed to exist. Ties are resolved by a fixed ordering of the candidates; because the candidate set is finite, this convention gives measurable selectors whenever the candidate scores are measurable.

A.1 Opportunity, regret, and surrogate witnesses

Proposition 3 (Opportunity–regret identity)**.** *Let U_s, U_r, and U_o denote the utilities of a static reference, a feasible learned router, and an optimal label-informed policy, respectively. Define*

$$\mathcal{O} = U_o - U_s, \qquad \mathcal{G} = U_o - U_r, \qquad \Delta = U_r - U_s.$$

Then

$$\Delta = \mathcal{O} - \mathcal{G}.$$

If $\mathcal{O} > 0$, the recovered fraction $q = \Delta/\mathcal{O}$ satisfies

$$q = 1 - \frac{\mathcal{G}}{\mathcal{O}}.$$

If the oracle optimizes over a policy class containing the learned router, then $\mathcal{G} \geq 0$ and hence $q \leq 1$.

Proof. Adding and subtracting U_o gives

$$U_r - U_s = (U_o - U_s) - (U_o - U_r).$$

This is the first identity. Division by $\mathcal{O} > 0$ gives the expression for q. If the learned router belongs to the oracle's feasible policy class, optimality of the oracle gives $U_o \geq U_r$, so $\mathcal{G} \geq 0$. □

For a loss R to be minimized, define

$$\mathcal{O}_R = R_s - R_o, \qquad \mathcal{G}_R = R_r - R_o, \qquad \Delta_R = R_s - R_r.$$

The same calculation yields

$$\Delta_R = \mathcal{O}_R - \mathcal{G}_R.$$

Proposition 4 (Feasible surrogate assignments are accuracy witnesses)**.** *Let U_o^{acc} be the maximum accuracy among all policies feasible at budget b. Let $\widehat{\pi}$ be any feasible policy constructed using a surrogate loss, including a cross-entropy Lagrange sweep, and let $\widetilde{U} = U_{\mathrm{acc}}(\widehat{\pi})$. Then*

$$\widetilde{U} \leq U_o^{\mathrm{acc}}.$$

Consequently, for any static reference of accuracy U_s,

$$\widetilde{\mathcal{O}} := \widetilde{U} - U_s \leq U_o^{\mathrm{acc}} - U_s =: \mathcal{O}_{\mathrm{acc}}.$$

No claim that $\widehat{\pi}$ minimizes the surrogate risk is required.

Proof. The policy $\widehat{\pi}$ is one feasible point in the optimization defining U_o^{acc}. Its accuracy therefore cannot exceed the maximum over that feasible set. Subtracting U_s proves the second inequality. □

A.2 Input-independent mixtures

Let $U_k = \mathbb{E}[u_k(X, Y)]$ be the marginal utility of candidate k. An input-independent randomized policy draws $K' \sim p$, independently of the input, where $p \in \Delta_K$. Its utility and cost are

$$U(p) = \sum_{k=1}^{K} p_k U_k, \qquad C(p) = \sum_{k=1}^{K} p_k c_k.$$

Define

$$U^{\star}_{\mathrm{mix}} = \max_{p \in \Delta_K} \left\{ \sum_{k=1}^{K} p_k U_k : \sum_{k=1}^{K} p_k c_k \leq b \right\}. \tag{23}$$

Proposition 5 (Bank quality and instance-specific selection)**.** *For any label-informed value U_o and static-reference value U_s,*

$$U_o - U_s = \underbrace{U_o - U^{\star}_{\mathrm{mix}}}_{\mathcal{O}_{\mathrm{inst}}} + \underbrace{U^{\star}_{\mathrm{mix}} - U_s}_{\mathcal{O}_{\mathrm{bank}}}.$$

If the oracle policy class contains all feasible input-independent randomized mixtures, then $\mathcal{O}_{\mathrm{inst}} \geq 0$. Thus $\mathcal{O}_{\mathrm{inst}}$ is the advantage of input-dependent selection over the best input-independent use of the same candidate bank, whereas $\mathcal{O}_{\mathrm{bank}}$ measures the difference between that bank and the chosen static reference.

Proof. The equality follows by adding and subtracting $U^{\star}_{\mathrm{mix}}$. Under the stated containment assumption, every feasible optimizer of equation 23 is admissible in the oracle optimization. Hence $U_o \geq U^{\star}_{\mathrm{mix}}$. □

Remark 6 (Deterministic empirical oracles)**.** *If U_o is defined using only deterministic assignments on a finite sample, the algebraic decomposition remains valid, but $U_o \geq U^{\star}_{\mathrm{mix}}$ need not hold without an additional randomization convention. A finite-sample deterministic budget can have an integrality effect. The nonnegative interpretation of $\mathcal{O}_{\mathrm{inst}}$ therefore uses either randomized admissible policies or a deterministic mixture implemented with counts compatible with the empirical budget.*

Proposition 7 (Two-candidate support suffices)**.** *If equation 23 is feasible, it has an optimal solution supported on at most two candidates. If the cost constraint is inactive at an optimal extreme point, an optimal solution supported on one candidate exists.*

Proof. The feasible region in equation 23 is a compact polytope, so the linear objective attains its maximum at an extreme point. At an extreme point with an active cost constraint, the positive coordinates are determined by at most two linearly independent equalities: $\sum_k p_k = 1$ and $\sum_k p_k c_k = b$. More than two positive coordinates would admit a nonzero perturbation supported on those coordinates that preserves both equalities; sufficiently small perturbations in either direction would remain nonnegative, contradicting extremality. Thus the support size is at most two. If the cost constraint is inactive, only the simplex equality is active, and an extreme point has one positive coordinate. □

A.3 THE RELAXED EMPIRICAL ORACLE

For a sample $\{(x_i, y_i)\}_{i=1}^n$, write $u_{ik} = u_k(x_i, y_i)$. The randomized empirical oracle is the linear program

$$\begin{aligned} \max_{p_{ik} \geq 0} \quad & \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} p_{ik} u_{ik} && (24) \\ \text{subject to} \quad & \sum_{k=1}^{K} p_{ik} = 1, && i \in [n], \\ & \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} p_{ik} c_k \leq b. \end{aligned}$$

A.3.1 PROOF OF THE LAGRANGIAN CHARACTERIZATION

Proposition 8 (Lagrangian characterization)**.** *Assume that the relaxed oracle in 24 is feasible. There exists $\lambda^\star \geq 0$ such that every optimal assignment $p^\star$ has support only on*

$$\arg\max_{k \in \mathcal{K}} \{u_{ik} - \lambda^\star c_k\} \tag{25}$$

for each i. Moreover,

$$\lambda^\star \left(\frac{1}{n} \sum_{i,k} p^\star_{ik} c_k - b \right) = 0. \tag{26}$$

If the maximizer in equation 25 is unique for every example, the resulting assignment is deterministic, feasible, and optimal for both the relaxed and deterministic problems.

Proof of proposition 8. Multiply the objective and the budget constraint by n, which does not alter the optimizer. Introduce a multiplier $\lambda \geq 0$ for $\sum_{i,k} p_{ik} c_k \leq nb$ and unrestricted multipliers α_i for the simplex equalities. The dual program is

$$\begin{aligned} \min_{\lambda \geq 0,\, \alpha \in \mathbb{R}^n} \quad & nb\lambda + \sum_{i=1}^{n} \alpha_i && (27) \\ \text{subject to} \quad & \alpha_i \geq u_{ik} - \lambda c_k, && i \in [n],\ k \in [K]. \end{aligned}$$

The primal feasible set is compact and nonempty. Hence an optimum exists, and finite-dimensional linear-programming duality gives primal and dual optima with equal values. At a dual optimum,

$$\alpha_i^\star = \max_k \{u_{ik} - \lambda^\star c_k\};$$

otherwise $\alpha_i^\star$ could be decreased while retaining dual feasibility. Complementary slackness for $p_{ik} \geq 0$ gives

$$p^\star_{ik} \big(\alpha_i^\star - u_{ik} + \lambda^\star c_k \big) = 0.$$

Thus a positive $p^\star_{ik}$ is possible only when candidate k maximizes $u_{ik} - \lambda^\star c_k$. Complementary slackness for the global budget constraint gives the second displayed equality. If every per-example maximizer is unique, the simplex constraint forces its probability to be one, so the solution is deterministic. □

Proposition 9 (Extreme-point structure and deterministic rounding)**.** *The relaxed program equation 24 has an optimal extreme point at which all but at most one example are assigned deterministically. The exceptional example, if present, randomizes between at most two candidates. If* $u_{ik} \in [u_{\min}, u_{\max}]$*, replacing that fractional assignment by its lower-cost supported candidate preserves feasibility and decreases mean utility by at most*

$$\frac{u_{\max} - u_{\min}}{n}.$$

Proof. An optimum exists at an extreme point. Consider a feasible point with two distinct fractional example blocks. Within each fractional simplex block there is a nonzero direction whose coordinates sum to zero and whose support is restricted to that block. A linear combination of the two directions can be chosen to have zero total cost change. For sufficiently small positive and negative step sizes, both perturbed points remain nonnegative and satisfy all constraints. The original point is then their midpoint and is not extreme. Hence at most one example can be fractional.

If the remaining fractional block has at least three positive coordinates, there is a nonzero direction on that block orthogonal both to the all-ones vector and to its candidate-cost vector. Small perturbations in either direction again preserve feasibility, contradicting extremality. The block therefore has at most two positive coordinates.

Suppose the fractional example mixes two supported candidates. Select the one with lower cost. Its cost does not exceed the convex combination of the two costs, so feasibility is preserved. Only one example's utility changes, and its loss is at most $u_{\max} - u_{\min}$. Division by n gives the stated bound. □

A.4 Exact finite-sample accuracy oracle

Let $a_{ik} = \mathbf{1}\{f_k(x_i) = y_i\}$. Assume that k_0 is the unique cheapest candidate and that $b \geq c_{k_0}$. Define

$$\mathcal{I} = \{i : a_{ik_0} = 0,\ \max_k a_{ik} = 1\}.$$

For every $i \in \mathcal{I}$, let

$$k_i^\star \in \arg\min_{k: a_{ik}=1} c_k, \qquad d_i = c_{k_i^\star} - c_{k_0} > 0.$$

Order these incremental costs as $d_{(1)} \leq \cdots \leq d_{(|\mathcal{I}|)}$ and define

$$m^\star = \max\left\{ m \in \{0, \ldots, |\mathcal{I}|\} : nc_{k_0} + \sum_{j=1}^{m} d_{(j)} \leq nb \right\}. \tag{28}$$

A.4.1 Proof of the exact deterministic accuracy oracle

Proof of Theorem 1. Consider an arbitrary feasible deterministic assignment. If $a_{ik_0} = 1$, replacing its selected candidate by k_0 preserves correctness and weakly decreases cost. If $a_{ik} = 0$ for every k, assigning k_0 also preserves accuracy and weakly decreases cost. Finally, if $i \in \mathcal{I}$ and the assignment is correct, replacing the selected candidate by $k_i^\star$ preserves correctness and weakly decreases cost. Therefore some optimizer has the following canonical form: it assigns $k_i^\star$ to a subset $S \subseteq \mathcal{I}$ and assigns k_0 otherwise.

Every selected example in S contributes exactly one additional correct prediction relative to the all-k_0 assignment and incurs incremental cost d_i. The remaining optimization is consequently

$$\max_{S \subseteq \mathcal{I}} |S| \quad \text{subject to} \quad \sum_{i \in S} d_i \leq n(b - c_{k_0}).$$

This is a unit-value knapsack problem. For any fixed cardinality m, the minimum-cost subset consists of the m smallest increments: if a subset contains d_j but omits $d_i < d_j$, exchanging j for i weakly decreases cost without changing cardinality. Thus cardinality m is feasible if and only if the sum of the m smallest increments is feasible. Definition equation 28 therefore gives the largest attainable number of corrections, proving accuracy optimality. The same exchange argument shows that the selected prefix has minimum incremental cost among all subsets with cardinality $m^\star$. The preceding

canonicalization weakly lowers the cost of every assignment without lowering its accuracy, which proves minimum cost among all accuracy maximizers. The displayed accuracy follows because each selected element of $\mathcal{I}$ adds one correct prediction. □

Corollary 10 (Union accuracy)**.** *If*

$$nc_{k_0} + \sum_{i \in \mathcal{I}} d_i \le nb,$$

then $m^\star = |\mathcal{I}|$ *and the exact oracle attains*

$$\frac{1}{n} \sum_{i=1}^{n} \max_{k \in [K]} a_{ik}.$$

The budget may be slack at this optimum.

Proof. The condition makes every correctable error of k_0 affordable, so $m^\star = |\mathcal{I}|$. The resulting assignment is correct exactly on the examples for which at least one candidate is correct. Additional cost cannot improve accuracy on the remaining examples, so the budget need not be tight. □

A.5 Feature-restricted routing and value of information

Let $Z = \phi(X)$ be the information available to the router, and define

$$m_k(z) = \mathbb{E}[u_k(X, Y) \mid Z = z].$$

For a fixed multiplier $\lambda \ge 0$, let

$$Q_k = u_k(X, Y) - \lambda c_k.$$

A.5.1 Proof of feature-restricted optimality and the information gap

Proof of Theorem 2. For any Z-measurable selector π,

$$\begin{aligned} &\mathbb{E}\big[u_{\pi(Z)}(X, Y) - \lambda c_{\pi(Z)}\big] \\ &\quad = \mathbb{E}\left[\mathbb{E}[Q_{\pi(Z)} \mid Z]\right] = \mathbb{E}\left[m_{\pi(Z)}(Z) - \lambda c_{\pi(Z)}\right]. \end{aligned}$$

The integrand is maximized pointwise by π_Z^λ, which proves the first two claims.

Conditional Jensen's inequality for the convex function $(q_1, \dots, q_K) \mapsto \max_k q_k$ gives

$$\max_k \mathbb{E}[Q_k \mid Z] \le \mathbb{E}[\max_k Q_k \mid Z].$$

Taking expectations proves $\mathcal{J}_Z(\lambda) \ge 0$.

For the equality characterization, fix a measurable tie-breaking rule and let π_Z^λ be the resulting conditional maximizer. The preceding identities imply

$$\mathcal{J}_Z(\lambda) = \mathbb{E}\left[\max_k Q_k - Q_{\pi_Z^\lambda(Z)}\right].$$

The random variable inside the expectation is nonnegative. Its expectation is zero if and only if it is zero almost surely, which is equivalent to $\pi_Z^\lambda(Z) \in \arg\max_k Q_k$ almost surely. This is precisely the existence of a Z-measurable privileged-optimal selector, with arbitrary choice among ties. □

The result identifies the quantity that a routing probe must estimate. The relevant score is the conditional expected utility of each action, not its realized utility on an individual labeled example. A low empirical correlation for one learned probe does not by itself estimate $\mathcal{J}_Z(\lambda)$ or prove that this gap is large.

Proposition 11 (Information and estimation components)**.** *Let U_o be the optimal utility in a privileged policy class, and let*

$$U_Z^\star = \sup_{\pi \in \Pi_Z(b)} U(\pi)$$

be the optimal utility among budget-feasible Z-measurable policies. If the privileged class contains $\Pi_Z(b)$ and the learned router $\pi_r \in \Pi_Z(b)$, then

$$\mathcal{G} := U_o - U_r = \underbrace{U_o - U_Z^\star}_{\mathcal{G}_{\text{info}}} + \underbrace{U_Z^\star - U_r}_{\mathcal{G}_{\text{est}}}, \tag{29}$$

and both terms are nonnegative.

Proof. Add and subtract $U_Z^\star$. Policy-class containment gives $U_o \geq U_Z^\star$, while the feasibility of π_r gives $U_Z^\star \geq U_r$. □

Proposition 12 (Nested representations)**.** *Let Z_1 and Z_2 be router representations such that $Z_1 = h(Z_2)$ almost surely for a measurable function h. Suppose the two policy classes use the same total-cost accounting and the same budget. Then*

$$U_{Z_1}^\star \leq U_{Z_2}^\star.$$

Equivalently, relative to a common privileged oracle, $\mathcal{G}_{\text{info}}(Z_2) \leq \mathcal{G}_{\text{info}}(Z_1)$.

Proof. Every Z_1-measurable router $\pi_1(Z_1)$ can be written as the Z_2-measurable router $\pi_1(h(Z_2))$. Under identical cost accounting, this inclusion also preserves feasibility. Thus $\Pi_{Z_1}(b) \subseteq \Pi_{Z_2}(b)$, and taking suprema proves the result. □

A.6 Marginal-utility parameterization

Consider three candidates with costs $c_s < c_m < c_l$ and losses ℓ_s, ℓ_m, ℓ_l. Define

$$g_{\text{down}} = \frac{\ell_s - \ell_m}{c_m - c_s}, \qquad g_{\text{up}} = \frac{\ell_m - \ell_l}{c_l - c_m}.$$

The relative-loss vector reconstructed from these two marginal utilities is

$$r(g) = \begin{bmatrix} g_{\text{down}}(c_m - c_s) \\ 0 \\ -g_{\text{up}}(c_l - c_m) \end{bmatrix}.$$

Proposition 13 (Exact reconstruction and conditional optimality)**.** *For exact marginal utilities,*

$$r(g) = \begin{bmatrix} \ell_s - \ell_m \\ 0 \\ \ell_l - \ell_m \end{bmatrix}.$$

Consequently, for every $\lambda \geq 0$,

$$\arg \min_{k \in \{s,m,l\}} \{r_k(g) + \lambda c_k\} = \arg \min_{k \in \{s,m,l\}} \{\ell_k + \lambda c_k\}.$$

Moreover, if a router observes Z and predicts the exact conditional means $\mathbb{E}[g_{\text{down}} \mid Z]$ and $\mathbb{E}[g_{\text{up}} \mid Z]$, then the reconstructed scores select

$$\arg\min_k \{\mathbb{E}[\ell_k \mid Z] + \lambda c_k\},$$

which is an optimal Z-measurable penalized loss router.

Proof. Substitution of the definitions of g_{down} and g_{up} gives the first equality. Adding the common quantity ℓ_m to every reconstructed score does not change the argmin and yields the original loss vector, proving the second equality.

Because candidate costs are constant, conditional expectation commutes with the two affine transformations:

$$\mathbb{E}[g_{\text{down}} \mid Z](c_m - c_s) = \mathbb{E}[\ell_s - \ell_m \mid Z]$$

and

$$-\mathbb{E}[g_{\text{up}} \mid Z](c_l - c_m) = \mathbb{E}[\ell_l - \ell_m \mid Z].$$

The reconstructed conditional scores therefore equal $\mathbb{E}[\ell_k \mid Z] - \mathbb{E}[\ell_m \mid Z]$. The subtracted term is common to all candidates and does not affect the argmin. Pointwise conditional minimization then proves optimality among Z-measurable selectors, by the same tower-property argument used in Theorem 2. □

A.7 COST ACCOUNTING AND HARDWARE REALIZATION

Lemma 14 (Omitted probe cost)**.** *Suppose a reported routing cost includes only the selected candidate,*

$$C_{\mathrm{reported}}(\pi) = \mathbb{E}[c_{\pi(Z)}],$$

while computing the routing representation incurs a nonnegative cost $a_\phi(X)$. Then the true expected cost is

$$C_{\mathrm{true}}(\pi) = C_{\mathrm{reported}}(\pi) + \mathbb{E}[a_\phi(X)] \geq C_{\mathrm{reported}}(\pi).$$

The same conclusion holds for a nonnegative route-dependent probe cost $a_\phi(X, \pi(Z))$ after replacing the final term by its expectation.

Proof. The equality follows from additivity of executed computation and linearity of expectation. The inequality follows from nonnegativity of the omitted cost. □

Let $H(\pi)$ denote a measured hardware quantity for policy π, such as latency or energy, and let s_b be the static reference. Define

$$\rho_H = \frac{H(\pi_r)}{H(s_b)}.$$

Proposition 15 (Utility and hardware dominance)**.** *The learned dynamic policy strictly dominates the selected static reference in both utility and the hardware metric if and only if*

$$\mathcal{G} < \mathcal{O} \qquad \textit{and} \qquad \rho_H < 1.$$

Under the illustrative additive model

$$H(\pi) = h(C(\pi)) + \delta(\pi),$$

if $C(\pi_r) = C(s_b)$, $\delta(s_b) = 0$, and $\delta(\pi_r) > 0$, then $\rho_H > 1$.

Proof. By Proposition 3, $U_r - U_s = \mathcal{O} - \mathcal{G}$. Hence $U_r > U_s$ if and only if $\mathcal{G} < \mathcal{O}$. By definition, $H(\pi_r) < H(s_b)$ if and only if $\rho_H < 1$, assuming $H(s_b) > 0$. This proves the first claim. Under the additive model and matched nominal cost,

$$\begin{aligned} H(\pi_r) - H(s_b) &= h(C(\pi_r)) - h(C(s_b)) \\ &\quad + \delta(\pi_r) - \delta(s_b) \\ &= \delta(\pi_r) > 0, \end{aligned}$$

which implies $\rho_H > 1$. □

The additive model is a sufficient diagnostic abstraction, not an assumption required by the empirical hardware measurements. In particular, direct measurement remains necessary because batching, dispatch, memory traffic, kernel selection, and device utilization need not be functions of nominal operation count alone.

B FROZEN HYPERPARAMETERS

B.1 DATA PARTITIONS

Table 3: Data-partition and evaluation protocol. CoverType counts correspond to the fixed $60/15/10/15\%$ stratified split of all $581{,}012$ examples.

Quantity	CIFAR studies	CoverType
Candidate-training examples	40,000	348,606
Router-training examples	5,000	87,152
Validation examples	5,000	58,102
Test examples	10,000	87,152
Partition seed	1729	1729
Candidate preprocessing	Crop with padding 4; horizontal flip	Standardize first 10 features
Preprocessing statistics	Dataset constants	Candidate-training split only
Binary features	—	44 unchanged
Primary budget calibration	Unlabeled test-batch projection	Unlabeled test-batch projection
Sensitivity calibration	Validation-selected multiplier	Validation-selected multiplier
Primary task metric	Top-1 accuracy	Top-1 accuracy
Bootstrap resamples	2,000	2,000

For CIFAR-10 and CIFAR-100, candidate models are trained only on the $40{,}000$-example candidate-training split. Counterfactual router targets are then computed on the disjoint $5{,}000$-example router-training split using the deterministic evaluation transform.

CoverType uses all $581{,}012$ examples and 54 input features. A fixed stratified split assigns 60% of the data to candidate training, 15% to router training, 10% to validation, and 15% to held-out testing. The first 10 continuous features are standardized using means and standard deviations computed only on the candidate-training split; the remaining 44 binary features are left unchanged.

In all three studies, candidate parameters are frozen before counterfactual router targets are generated. No test label is used to train router parameters or to select the multiplier of a learned policy under the primary budget-projected evaluation.

B.2 CANDIDATE MODELS

Table 4: Nested CNN/CIFAR-10 hyperparameters.

Quantity	Value
Seeds	$0, 1, 2, 3, 4$
Stem width	64
Dense body widths	$(128, 128, 256, 256)$
Candidate multipliers	$(0.125, 0.25, 0.5)$
Target dense-MAC ratio	0.0973
Saliency calibration limit	10,000
Dense optimizer	AdamW
Dense epochs / learning rate	50 / 10^{-3}
Candidate optimizer	AdamW
Candidate epochs / learning rate	40 / 3×10^{-4}
Global model batch size	128
Weight decay	10^{-4}
Distillation weight	1.0
Distillation temperature	2.0
Learning-rate schedule	Cosine

Table 5: ResNet-18/CIFAR-100 hyperparameters.

Quantity	Value
Confirmatory seeds	$0, 1, 2$
Supplementary seeds	$3, 4$
Stem width	64
Dense stage widths	$(64, 128, 256, 512)$
Candidate multipliers	$(0.25, 0.50, 1.00)$
Target dense-MAC ratio	0.2732
Dense optimizer	SGD, Nesterov
Dense epochs / learning rate	200 / 0.1
Candidate optimizer	SGD, Nesterov
Candidate epochs / learning rate	160 / 0.05
Global model batch size	128
Momentum	0.9
Weight decay	5×10^{-4}
Label smoothing	0.1
Distillation weight	1.0
Distillation temperature	2.0
Learning-rate schedule	Cosine

Seeds 3 and 4 use the unchanged candidate-training protocol and are used for the locked-protocol clipped-router replication, the strictly validation-frozen margin analysis, and the five-seed hardware benchmark.

Table 6: MLP/CoverType hyperparameters.

Quantity	Value
Seeds	$0, 1, 2, 3, 4$
Input features	54
Classes	7
Small hidden widths	$(64, 64)$
Medium hidden widths	$(128, 128, 128)$
Large hidden widths	$(256, 256, 256, 256)$
Small MACs	8,000
Medium MACs	40,576
Large MACs	212,224
Relative candidate costs	$(0.03770, 0.19119, 1.00000)$
Target large-MAC ratio	0.19119
Static reference	Medium MLP
Optimizer	AdamW
Epochs	40
Learning rate	10^{-3}
Batch size	1024
Evaluation batch size	4096
Weight decay	10^{-4}
Dropout	0.10
Label smoothing	0
Gradient-norm limit	5.0
Learning-rate schedule	Cosine
Checkpoint selection	Minimum validation cross-entropy

The three CoverType candidates are trained independently. Each hidden layer uses a ReLU activation followed by dropout with probability 0.1. The medium MLP is the matched static reference. Candidate checkpoints are selected solely by minimum validation cross-entropy before router, validation, and test counterfactual caches are constructed.

B.3 Router suite

Table 7: Hyperparameters shared by the neural router baselines.

Quantity	Value
CIFAR routing representation	Globally pooled 64-dimensional stem feature
CoverType routing representation	Preprocessed 54-dimensional input
Optimizer	AdamW
Epochs	200
Learning rate	3×10^{-3}
Weight decay	10^{-4}
Batch size	512
Learning-rate schedule	Cosine
Gradient-norm limit	5.0
Activation	SiLU
Marginal Huber threshold	1.0
Marginal ranking weight	0.25
Ranking exclusion threshold	10^{-3}
Clipped-target quantiles	0.01 and 0.99
Budget overshoot tolerance	10^{-4}

The fixed neural suite contains a linear marginal-utility predictor trained by mean-squared error; width-16 clipped and unclipped marginal predictors trained by Huber and pairwise ranking losses;

width-128 and width-256 candidate-loss predictors trained by mean-squared error; a width-256 candidate-correctness predictor trained by binary cross-entropy; and a width-256 exact-oracle-route classifier trained by class-weighted cross-entropy. Route-class weights are proportional to the square root of inverse class frequency and normalized to unit mean.

The histogram gradient-boosted baseline fits one candidate-loss regressor per route with learning rate 0.05, 200 boosting iterations, at most 31 leaves, and ℓ_2 regularization 10^{-3}. Its decision cost is not included in the reported MAC count and it is therefore treated as an information probe.

Entropy, margin, and maximum-probability policies first evaluate the small candidate. Its cost is consequently added whenever execution continues to the medium or large candidate. For neural routers, decision-network MACs are added to all routes before budget projection.

C Per-Seed Exact Decomposition

Table 8 uses the exact finite-sample accuracy oracle and the clipped width-16 marginal-utility router evaluated with unlabeled empirical budget projection. Opportunity, regret, and realized gain use top-1 accuracy throughout.

Table 8: Per-seed exact accuracy decomposition in percentage points. The identity residual is $\Delta - (\mathcal{O} - \mathcal{G})$.

Study	Seed	Static accuracy	Exact-oracle accuracy	Opportunity $\mathcal{O}$	Regret $\mathcal{G}$	Realized gain Δ	Residual
Nested CNN / CIFAR-10	0	80.330	88.830	8.500	10.140	−1.640	0.000
	1	80.360	89.100	8.740	9.540	−0.800	0.000
	2	80.410	89.530	9.120	10.420	−1.300	0.000
	3	80.990	89.700	8.710	10.030	−1.320	0.000
	4	80.710	89.810	9.100	10.490	−1.390	0.000
ResNet-18 / CIFAR-100	0	72.930	84.980	12.050	12.830	−0.780	0.000
	1	73.220	84.650	11.430	12.350	−0.920	0.000
	2	73.200	85.150	11.950	12.750	−0.800	0.000
MLP / CoverType	0	89.196	95.926	6.730	6.556	+0.173	0.000
	1	89.057	95.734	6.677	6.427	+0.250	0.000
	2	89.096	95.766	6.670	6.516	+0.154	0.000
	3	88.854	95.897	7.043	6.712	+0.330	0.000
	4	89.016	95.883	6.867	6.650	+0.217	0.000

The exact opportunity is positive in every seed of all three studies. For the clipped marginal-utility router, realized gain is negative in every nested-CNN and ResNet seed but positive in every CoverType seed. On CoverType, the mean gain is 0.225 points with paired 95% interval $[0.138, 0.312]$. The decomposition identity holds up to the displayed numerical precision.

D Additional Candidate and Router Diagnostics

D.1 ResNet candidate results

Table 9: Per-seed ResNet candidate, union, and exact-oracle results. Accuracy and MAC values are percentages. Oracle routes are reported in the order (s, m, l).

Seed	Small	Medium	Large	Union	Exact oracle	Oracle MACs	Oracle routes
0	69.11	72.93	77.51	84.98	84.98	14.893	$(84.13, 10.62, 5.25)$
1	68.33	73.22	77.41	84.65	84.65	15.038	$(83.68, 10.99, 5.33)$
2	68.67	73.20	77.88	85.15	85.15	15.011	$(83.52, 11.23, 5.25)$

All correctness-producing upgrades fit within the target budget in each ResNet seed. The exact oracle therefore equals union accuracy in all three runs and uses approximately 15% of dense-model MACs, below the nominal matched budget of 27.32%. The similar route distributions show that the exact allocation structure is stable across seeds.

D.2 Early and later ResNet features

Table 12: Per-seed ResNet robustness results for the clipped marginal-utility router. Gains, regrets, and opportunities are in percentage points.

Seed	$\mathcal{O}$	Δ	$\mathcal{G}$	Recovery	MAC	Δ_{val}	MAC_{val}
0	12.05	−0.78	12.83	−0.065	0.27315	−0.77	0.27015
1	11.43	−0.92	12.35	−0.080	0.27318	−0.94	0.27060
2	11.95	−0.80	12.75	−0.067	0.27314	−0.80	0.27444
3	11.60	−0.97	12.57	−0.084	0.27317	−0.97	0.27315
4	11.34	−0.70	12.04	−0.062	0.27321	−0.71	0.27110

Table 10: ResNet marginal-utility diagnostics. Correlations and gains are mean ± sample standard deviation across three seeds. The later probe's displayed route cost excludes computation of the medium representation.

Feature and predictor	Downsize ρ	Upgrade ρ	Accuracy gain (pp)
Pooled stem, clipped MLP-16	0.007 ± 0.006	0.004 ± 0.007	-0.833 ± 0.076
Medium penultimate, diagnostic probe	0.061 ± 0.032	0.079 ± 0.099	-0.830 ± 1.102

The later-feature downsize correlations are $(0.024, 0.082, 0.076)$ across the three seeds, while the upgrade correlations are $(-0.035, 0.133, 0.139)$. The later representation increases the mean association with both marginal-utility targets, but the correlations remain modest and do not produce a consistent accuracy improvement.

D.3 Uncertainty controls

Table 11: Per-seed accuracy gains of the ResNet uncertainty controls under unlabeled empirical budget projection. Values are percentage points relative to the matched static medium candidate.

Seed	Entropy	Margin	Maximum probability
0	+1.36	+2.05	+1.77
1	+0.54	+0.74	+0.57
2	+0.95	+1.42	+1.14

All three uncertainty controls improve accuracy in every ResNet seed. Their mean gains are 0.95, 1.40, and 1.16 points for entropy, margin, and maximum probability, respectively. The corresponding paired 95% intervals are $[-0.07, 1.97]$, $[-0.22, 3.03]$, and $[-0.33, 2.65]$. Thus, the observed gains are consistent in sign but remain statistically imprecise with three seeds.

D.4 Five-seed ResNet hardware realization

Table 13: Five-seed hardware realization of the strictly validation-frozen ResNet margin policy on an RTX 3090. Entries are mean dynamic/static ratios with paired seed-level 95% Student t intervals. NVML is the primary cumulative-counter GPU-energy measurement; CC denotes the CodeCarbon GPU-energy cross-check. Values below one favor dynamic execution.

	Dynamic / static medium			Dynamic / static large		
Batch	Latency	NVML GPU energy	CC GPU energy	Latency	NVML GPU energy	CC GPU energy
1	1.561 [1.477, 1.644]	1.570 [1.447, 1.694]	1.517 [1.433, 1.600]	1.625 [1.556, 1.693]	1.266 [1.182, 1.350]	1.278 [1.237, 1.319]
64	2.283 [2.179, 2.386]	1.867 [1.745, 1.990]	1.920 [1.789, 2.051]	1.166 [1.113, 1.220]	0.881 [0.843, 0.918]	0.881 [0.837, 0.924]
256	1.370 [1.356, 1.385]	1.390 [1.373, 1.407]	1.393 [1.376, 1.409]	0.658 [0.651, 0.664]	0.655 [0.647, 0.663]	0.656 [0.650, 0.662]

Hardware and measurement environment. Measurements use physical GPU 1 on a host containing two NVIDIA GeForce RTX 3090 devices. The selected device has 24 GiB of memory, NVIDIA driver 580.178.04, and a 420 W power limit. The benchmark uses single-GPU FP32 inference under `torch.inference_mode`; TF32 is disabled for both cuBLAS and cuDNN to reproduce the frozen full-precision prediction caches. cuDNN benchmarking is disabled and deterministic cuDNN execution is enabled. GPU clocks are not fixed. Before measurement, the launcher verifies that no competing compute process is active and that the NVML cumulative-energy counter is available.

For each seed and batch size, the dynamic policy and corresponding static small, medium, and large candidates receive the same deterministic set of 20 input batches already resident on the GPU; host-to-device transfer is therefore excluded. Each method uses 20 warm-up iterations followed by 200 timed iterations per repetition, with five repetitions. Method order is randomized and interleaved across repetitions to reduce systematic ordering and thermal effects. GPU synchronization is performed before and after every timed block. The benchmark does not use `torch.compile`, CUDA graphs, fused routing kernels, asynchronous route execution, or specialized sparse-control kernels.

Primary GPU energy is computed from the difference in *nvmlDeviceGetTotalEnergyConsumption* before and after each timed block and divided by the number of iterations, yielding energy per complete batch inference. CodeCarbon 3.3.1 task-level GPU energy is recorded over the same blocks with a 0.1-second power-measurement interval. Across the 60 seed–batch–method records, the mean CodeCarbon/direct-NVML ratio is 0.996, the median is 1.000, and the mean absolute relative difference is 1.85%. Both estimators give the same qualitative conclusions. Because CodeCarbon also uses NVIDIA telemetry, this comparison validates implementation consistency rather than providing an independent physical power measurement.

All five frozen deployments reproduce their cached decisions and candidate outputs on the complete 10,000-example test set before measurement. The five trained policies are the statistical units for the reported hardware intervals. We report paired seed-level 95% Student t intervals; the five timed repetitions within each seed quantify technical measurement variation rather than independent training variation. These measurements characterize this single-GPU explicit-dispatch implementation and should not be interpreted as accelerator-independent performance.

D.5 CoverType candidate results

Table 14: Per-seed CoverType candidate, union, and exact-oracle results. Accuracy and MAC values are percentages. Oracle routes are reported in the order (s, m, l).

Seed	Small	Medium	Large	Union	Exact oracle	Oracle MACs	Oracle routes
0	82.955	89.196	93.469	95.926	95.926	9.538	(87.03, 8.30, 4.67)
1	83.019	89.057	93.266	95.734	95.734	9.322	(87.29, 8.26, 4.45)
2	83.085	89.096	93.269	95.766	95.766	9.351	(87.32, 8.19, 4.49)
3	83.160	88.854	93.492	95.897	95.897	9.732	(87.26, 7.78, 4.95)
4	82.956	89.016	93.344	95.883	95.883	9.442	(87.07, 8.37, 4.56)

All correctness-producing upgrades fit within the CoverType target budget. The exact oracle therefore equals union accuracy in all five seeds. Its mean route distribution is approximately $(87.19, 8.18, 4.63)\%$, and its mean active cost is approximately 9.48% of the large-MLP MAC count, well below the nominal matched budget of 19.12%. The optimal input-independent mixture selects the medium candidate with probability one in every seed.

D.6 CoverType router diagnostics

Table 15: CoverType fixed-router suite under unlabeled empirical budget projection. Δ is accuracy gain relative to the matched static medium MLP. Intervals are paired seed-level 95% Student t intervals. ρ_d and ρ_u are Spearman correlations with the downsize and upgrade utilities. The gradient-boosted decision cost is omitted.

Table 16: Validation-frozen budget sensitivity. Each entry reports mean accuracy gain in percentage points followed by mean realized MAC percentage. The multiplier is selected on validation data and applied unchanged to the test set. The boosted-tree decision cost is omitted.

Method	Nested CNN / CIFAR-10	ResNet-18 / CIFAR-100	MLP / CoverType
Linear marginal MSE	$-0.932/9.721$	$-0.970/26.993$	$+0.015/19.087$
MLP-16 marginal, clipped	$-1.306/9.682$	$-0.837/27.173$	$+0.213/19.025$
MLP-16 marginal, unclipped	$-1.370/9.676$	$-0.693/27.591$	$+0.234/18.964$
MLP-128 candidate losses	$-1.728/9.667$	$-0.993/27.155$	$-0.157/19.091$
MLP-256 candidate losses	$-1.754/9.694$	$-1.047/27.346$	$-0.856/19.114$
MLP-256 candidate correctness	$-1.952/9.659$	$-0.793/27.559$	$-0.530/19.080$
MLP-256 oracle-route imitation	$-5.950/9.630$	$-2.103/27.616$	$-1.970/19.070$
Histogram gradient boosting	$-2.928/9.685$	$-1.097/27.573$	$+0.425/19.036$
Small-model entropy	$-2.990/9.780$	$+0.970/27.427$	$+0.547/19.126$
Small-model margin	$-2.524/9.738$	$+1.443/27.503$	$+0.337/19.040$
Small-model maximum probability	$-2.618/9.781$	$+1.207/27.644$	$+0.398/19.057$

Method	Δ static	ρ_d	ρ_u	MAC ratio
Linear marginal MSE	$+0.017; [-0.010, 0.044]$	.232	.282	.191195
MLP-16 marginal, clipped	$+0.225; [0.138, 0.312]$	.415	.452	.191194
MLP-16 marginal, unclipped	$+0.251; [0.184, 0.318]$	.417	.448	.191195
MLP-128 candidate losses	$-0.151; [-0.187, -0.115]$	.348	.437	.191195
MLP-256 candidate losses	$-0.855; [-0.919, -0.790]$	.349	.436	.191194
MLP-256 candidate correctness	$-0.524; [-0.605, -0.444]$	.289	.398	.191195
MLP-256 oracle-route imitation	$-1.965; [-2.070, -1.859]$	.338	$-.105$	.191194
Histogram gradient boosting	$+0.438; [0.320, 0.556]$	.344	.377	.191192
Small-model entropy	$+0.546; [0.480, 0.612]$	.481	.514	.191194
Small-model margin	$+0.350; [0.229, 0.472]$	.484	.510	.191194
Small-model maximum probability	$+0.405; [0.346, 0.463]$	.486	.515	.191195

The CoverType router suite spans both successful and unsuccessful policies. The prespecified clipped marginal router improves over static accuracy in all five seeds and recovers 3.3% of the exact opportunity. Small-model entropy has the largest mean gain, 0.546 points, corresponding to approximately 8.0% recovery. In contrast, increasing neural predictor capacity does not guarantee better routing: the width-128 and width-256 candidate-loss predictors, the correctness predictor, and oracle-route imitation all remain below static accuracy.

E Validation-Frozen Budget Sensitivity

The primary router comparison projects the multiplier onto the target budget using predicted test values and route costs. Table 16 reports a non-transductive sensitivity analysis in which the multiplier is selected on validation data and applied unchanged to the test set. The supervised counterfactual-target predictors remain below static accuracy in the two image studies. On CoverType, the clipped and unclipped marginal predictors, gradient boosting, and all three uncertainty policies retain positive gains. The differences from test-batch projection are small relative to the observed method differences.

For the prespecified CoverType clipped marginal router, the validation-frozen multiplier produces gains of 0.171, 0.242, 0.145, 0.308, and 0.199 points across seeds 0–4. Its mean gain is 0.213 points with paired interval $[0.133, 0.292]$, and its mean MAC ratio is 0.19025, approximately 0.50% below the target. Small-model entropy is particularly insensitive to the calibration protocol: its validation-frozen gain is 0.547 points with interval $[0.487, 0.607]$, compared with 0.546 points under test-batch projection.

The validation-frozen multiplier produces modest budget deviations because the validation and test score distributions differ. In the two image studies, these deviations do not change the conclusion for the supervised counterfactual-target predictors: their mean gains remain negative. On CoverType, the clipped and unclipped marginal predictors retain positive mean gains, while the higher-capacity candidate-loss, correctness, and route-imitation predictors remain below static accuracy. The uncertainty controls likewise retain the same qualitative behavior as under unlabeled test-batch projection. Thus the principal cross-study conclusions are not driven by transductive test-batch budget calibration.