COST-AWARE THRESHOLD ADAPTIVE-K LEARNING TO DEFER UNDER BUDGET CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning to defer routes each input to a prediction source, trading accuracy against consultation cost. In practice several predictors and human experts are available, their costs differ widely, and existing multi-expert and top-k methods regulate costs via a hyperparameter, yet they fail to guarantee strict adherence to budget constraints. We introduce CATABC, a budget-constrained adaptive top-k architecture that selects how many and which experts to consult per input, and weighs the consulted ones by masked attention over the image and expert features, under an average-cost constraint enforced by a dual price. We prove that the threshold selection optimally solves the Lagrangian of the average-budget proxy, that the price is the slope of the risk–budget frontier, and that the excess risk decomposes into pool, selection and fusion gaps. On CIFAR-10, CIFAR-100 and SVHN, CATABC stays within the prescribed budget and surpasses the baselines in most settings.

1 INTRODUCTION

A classification system rarely has a single source of truth: the predictors available at inference time include both pre-trained models and human experts (Hemmer et al., 2022; Madras et al., 2018; Mozannar & Sontag, 2020). The pool is complementary but heterogeneous: accuracies differ, and the price of consulting a member ranges from the inference cost of a model to the fee of a human expert, which is typically the most expensive resource to allocate (Alves et al., 2024; DeSalvo et al., 2025). Routing each input to a subset of the pool is therefore more attractive than invoking all members for every query, as in conditional computation and dynamic model routing (Shazeer et al., 2017; Fedus et al., 2022; Han et al., 2022). The key challenge is to decide which members each input needs under their heterogeneous costs.

Learning to Defer (L2D) is the standard framework for such instance-dependent routing (Strong et al., 2025): early work defers between one predictor and one expert (Madras et al., 2018; Mozannar & Sontag, 2020), and recent methods scale L2D to multiple experts (Verma et al., 2023; Mao et al., 2025), calibrate the deferral rule (Verma & Nalisnick, 2022; Cao et al., 2023), or add workload and budget constraints (Mozannar et al., 2023; Alves et al., 2024; DeSalvo et al., 2025). Top-k deferral selects a subset of experts per input (Mao et al., 2023; Montreuil et al., 2026), and cardinality-aware prediction lets that subset vary in size, allocating more predictors to difficult inputs (Cortes et al., 2024; Montreuil et al., 2026). These methods, however, control cost only indirectly through per-entity penalties whose coefficients are tuned before training, so a prescribed budget is not guaranteed at inference time; and since the selected subset determines both the inference cost and which predictions are available for aggregation, selection and aggregation should be learned jointly.

Motivated by these challenges, we propose *CATABC*, a framework that adaptively determines which predictors to consult for each input and outputs fusion weights for each selected predictor source. Crucially, these decisions are driven by the prescribed budget, so that the system attains its best performance under that budget. The model is trained end-to-end with an explicit expected-cost constraint, enabling adaptive-K routing that respects the budget without prescribing k.

Our main contributions are: (i) We propose CATABC, a budget-constrained adaptive-k framework that couples a feature extractor, a cost-aware selector, and a selection-aware attention fusion module. (ii) We induce the committee size per input by thresholding the price-adjusted utility, and drive one dual price by the realised average cost so that a prescribed budget is respected. (iii) We show that

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.

the threshold policy optimally solves the Lagrangian of the average-budget proxy, that the price is the slope of the risk–budget frontier, and that the excess risk decomposes into pool, selection, and fusion gaps. (iv) Experiments on three benchmarks show that our method stays within budget while surpassing the baselines in most settings, and ablations isolate each component.

2 Related Work

Learning to Defer. Learning to defer (L2D) extends selective prediction and learning with a reject option (Chow, 1970; Cortes et al., 2016; Geifman & El-Yaniv, 2017) by letting a model not only abstain on uncertain inputs but also hand them to an external expert; we refer to Strong et al. (2025) for a survey. Madras et al. (2018) introduced the fairness- and accuracy-aware formulation and Mozannar & Sontag (2020) the first Bayes-consistent surrogate for one-stage joint training, which later work improved in calibration (Verma & Nalisnick, 2022; Cao et al., 2023) and in realizability and $\mathcal{H}$-consistency guarantees (Mao et al., 2024b). A complementary *two-stage* regime keeps a pre-trained predictor and the experts fixed and learns only a deferral mechanism on top (Narasimhan et al., 2022; Mao et al., 2023); because it needs no access to predictor weights, it is the default when models are black boxes or API-only. The single-expert formulation has been generalized to *multiple* experts through consistent surrogates and conformal ensembles (Verma et al., 2023), complementary human-AI teams (Hemmer et al., 2022), and $\mathcal{H}$-consistency bounds and unified realizable losses (Mao et al., 2024a; 2025), and recent work adds workload and budget constraints (Mozannar et al., 2023; Alves et al., 2024; DeSalvo et al., 2025). Across this literature, however, the decision object remains a *single* expert per input: the system selects one entity, or abstains, rather than invoking and combining several at once.

Model Routing, Cascades, and Ensembling. A parallel line routes each input to a single model or sub-network to save computation, including sparsely-gated mixture-of-experts (Shazeer et al., 2017; Fedus et al., 2022), adaptive and early-exit networks (Teerapittayanon et al., 2016; Han et al., 2022), and cascades (Viola & Jones, 2001; Jitkrittum et al., 2023; Dohan et al., 2022). For LLMs, learned routers such as FrugalGPT (Chen et al., 2023), RouteLLM (Ong et al., 2024), and HybridLLM (Ding et al., 2024) dispatch each query to the cheapest sufficient model. These are almost uniformly winner-takes-all: one model answers each query, so complementary errors cannot be exploited. Multi-model aggregation instead belongs to ensemble literature, which combines all predictors (Wolpert, 1992; Lakshminarayanan et al., 2017) but ignores heterogeneous deployment cost, and to recent mixture-of-agents schemes fusing several LLMs' outputs (Wang et al., 2024) without budget-aware selection.

Top-k Classification and Deferral. Top-k classification predicts top-ranked label sets and has cardinality-aware losses (Lapin et al., 2017; Cortes et al., 2024). Extending deferral from a single expert to a committee remains largely open: to our knowledge, Montreuil et al. (2026) is the only top-k learning-to-defer framework, allocating each query to the k most cost-effective entities in one- and two-stage regimes and adding adaptive Top-$k(x)$ to select the committee size per input. Our method also allocates a variable number of predictors per query, but enforces an explicit additive *budget* via a dual-ascent Lagrangian instead of only per-entity penalties, and *fuses* selected predictors' probabilistic outputs with masked attention rather than treating each as an independent deferral decision.

3 APPROACH

3.1 Problem Formulation

We consider a pool of N frozen predictors P_i that return class distributions $p_i(\cdot \mid x) \in \Delta^{C-1}$. These include probabilistic models and experts whose categorical responses are represented by one-hot vectors. For each input, CATABC learns a binary selection vector $m(x)$ and combines the predictor outputs into a distribution $q(\cdot \mid x)$. The learning objective is

$$\min_{\theta,\psi} \mathbb{E}[\ell_\epsilon(q, Y)] \quad \text{s.t.} \quad \mathbb{E}_X[C_{\text{train}}(X)] \le B_{\text{train}}, \qquad \ell_\epsilon(q, y) = -\log(q_y + \epsilon_\ell). \tag{1}$$

Expectations include the data distribution and, where applicable, expert-response randomness. The parameters θ comprise the feature encoder and selector; ψ parameterizes fusion. Predictors remain

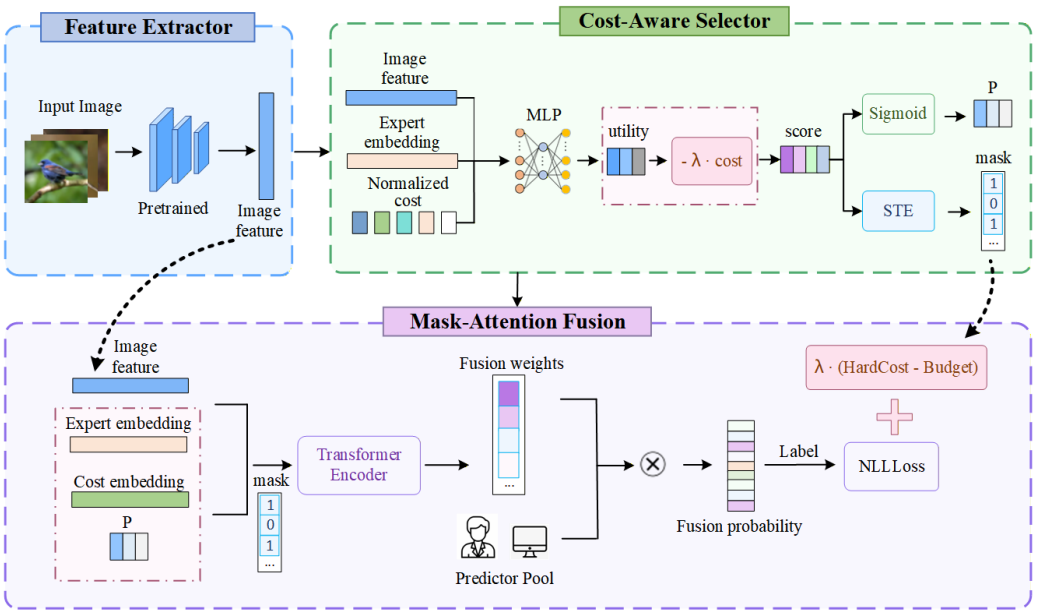


Figure 1: This figure illustrates our proposed framework, which comprises three core components: a pre-trained feature extractor, a cost-aware differentiable selector, and a selection-aware attention fusion module. We formulate the training objective as a constrained optimization problem and solve it via its Lagrangian dual. The final loss consists of the NLL loss and a penalty term for hard violations, scaled by the multiplier λ.

frozen throughout learning, following the two-stage deferral setting (Narasimhan et al., 2022; Mao et al., 2023). Figure 1 shows the framework of our method.

We distinguish the non-negative training costs $\tilde{c}_i$ from the reporting costs c_i, defining

$$C_{\text{train}}(x) = \sum_i m_i(x)\tilde{c}_i, \qquad C_{\text{report}}(x) = \sum_i m_i(x)c_i. \tag{2}$$

The budget B_{train} is in training units. The two cost vectors may coincide but need not be proportional; they are two accounting scales for the same selection. Both measure mask-based consultation cost. Evaluation obtains pool outputs before aggregation, so this accounting is distinct from measured execution time. The constraint in Eq. 1 allocates resources across inputs by an average, not a uniform per-input expenditure.

3.2 Cost-Aware Adaptive Selection

A shared encoder produces $z = f_\theta(x)$. A projection $v(z)$, a learnable predictor embedding e_i, and the normalized training cost form the input to the utility network:

$$\bar{c}_i = \frac{\tilde{c}_i}{N^{-1}\sum_j \tilde{c}_j + \epsilon_c}, \qquad U_i(x) = u_\theta([v(z); e_i; \bar{c}_i]), \qquad s_i(x) = U_i(x) - \lambda\tilde{c}_i. \tag{3}$$

The cost feature informs the learned utility, whereas the explicit price term adjusts the decision at a given utility. The shared scalar $\lambda \geq 0$ connects these input-dependent decisions to the average budget. Selection uses

$$\pi_i = \sigma\left(\frac{s_i - \tau}{T_s}\right), \qquad m_i = \mathbf{1}[s_i > \tau], \qquad m_i^{\text{ST}} = m_i + \pi_i - \text{sg}(\pi_i), \tag{4}$$

where $T_s > 0$ is the selector temperature and the default threshold is $\tau = 0$, as in the classical reject-option rule (Chow, 1970; Cortes et al., 2016; Geifman & El-Yaniv, 2017). Following the straight-through estimation approach (Bengio et al., 2013), the ST mask is binary in the forward pass and uses the sigmoid derivative in the backward pass. Below, m denotes this mask when it occurs in the training computation. No predictor is forced into the set $S_\lambda(x) = \{i : s_i(x) > \tau\}$; its size is determined by the scores.

Fixed-cardinality variant. CATABC-Topk replaces thresholding by the top-k entries of $\mathrm{softmax}(s/T_s)$ for a prescribed k, mirroring the top-k deferral framework of Montreuil et al. (2026). A straight-through mask uses these softmax probabilities for differentiation. It retains the feature encoder, fusion module, and cost-penalized training objective, while prescribing the number of selected predictors rather than inducing it through a threshold.

3.3 Selection-Aware Prediction Fusion

The fusion network constructs an image token $h_0 = W_I z + b_I$ and one token per predictor. With predictor embeddings e_i^f, its auxiliary features are

$$t_i = e_i^f + m_i \,\mathrm{LN}(W_s \pi_i + b_s) + m_i \,\mathrm{LN}(W_c \bar{c}_i + b_c). \tag{5}$$

A Transformer (Vaswani et al., 2017) encodes $[h_0; t_1; \ldots; t_N]$. Its key-padding mask keeps the image token and the selected predictor tokens available as keys and values. The output head assigns a logit a_i to every predictor position. This construction learns fusion weights from image features, predictor identities, selection probabilities, and costs; predictor output distributions enter the final aggregation below. Direct access to image features allows fusion to distinguish inputs that share the same selection scores. Proposition 2 shows how this additional information can improve the achievable log-loss while keeping the selected predictors and consultation costs fixed.

Writing $b_i = a_i/T_f$, with fusion temperature $T_f > 0$, the implemented log-mask weighting is

$$Z = \sum_j e^{b_j}(m_j + \epsilon_m), \qquad \alpha_i = \frac{e^{b_i}(m_i + \epsilon_m)}{Z}, \qquad q(c \mid x) = \sum_i \alpha_i p_i(c \mid x). \tag{6}$$

Thus q combines probability forecasts and categorical expert responses in a common distribution (Lakshminarayanan et al., 2017; Wang et al., 2024); since the aggregation multiplies the predictors' probabilities, well-calibrated forecasts (Verma & Nalisnick, 2022; Cao et al., 2023) improve it. The positive constant ϵ_m makes the weighting well defined for every mask. It gives unselected positions residual weight; if the mask is empty, Eq. 6 reduces to a softmax over all fusion logits. For a nonempty mask, taking $\epsilon_m \to 0$ with fixed logits yields the softmax restricted to the selected set.

Numerical normalization. The implementation additionally divides α by $\sum_i \alpha_i + \epsilon_w$ before computing $-\log(\cdot + \epsilon_n)$. In exact arithmetic, $\sum_i \alpha_i = 1$, so its loss is $-\log(q_y + \epsilon_\ell) + \log(1 + \epsilon_w)$ with $\epsilon_\ell = (1 + \epsilon_w)\epsilon_n$. We use this effective ϵ_ℓ throughout, omitting the parameter-independent constant. This preserves the training derivatives while keeping q normalized in the notation.

3.4 Joint Training and Budget Feedback

Using a Lagrangian penalty for the average-cost constraint (Boyd & Vandenberghe, 2004), the objective for a mini-batch $\mathcal{B}$ is

$$\mathcal{L}_\mathcal{B} = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \ell_\epsilon(q(x), y) + \lambda \left(\frac{1}{|\mathcal{B}|} \sum_{x\in\mathcal{B}} C_{\mathrm{train}}(x) - B_{\mathrm{train}} \right). \tag{7}$$

The threshold rule specifies how utilities produce selections; the fused loss trains those utilities together with the fusion network. The same price enters the selection score and the cost penalty. After each epoch, the observed sample-weighted average training cost $\widehat{C}_{\mathrm{train},t}$ updates it, as in budget- and workload-constrained deferral (DeSalvo et al., 2025; Alves et al., 2024):

$$\lambda_{t+1} = \mathrm{clip}\Big(\lambda_t + \eta(\widehat{C}_{\mathrm{train},t} - B_{\mathrm{train}}), 0, \lambda_{\max}\Big). \tag{8}$$

An excess in average cost raises the price, increasing both the selection threshold in utility units and the cost penalty; spare budget lowers it.

4 Theoretical Analysis

We characterize the fused prediction and the information used to weight it, the average-budget utility proxy, the training gradients, and the response to the budget price, complementing the consistency

analyses of multi-expert deferral (Verma et al., 2023; Mao et al., 2024a; 2025). All proofs are reported in Appendix A, which follows the order of the statements. The selection results concern the default zero threshold; all costs and price coefficients are non-negative.

Definition 1 (Threshold selection policy) *Given utilities $U_i(x)$, training costs $\tilde{c}_i \geq 0$, and a budget price $\lambda \geq 0$, the* threshold policy *at price λ selects $m_i(x) = \mathbf{1}[U_i(x) > \lambda \tilde{c}_i]$, inducing the selected set $S_\lambda(x) = \{i : m_i(x) = 1\}$ with cardinality $k_\lambda(x) = |S_\lambda(x)| = \sum_i m_i(x)$.*

4.1 Fused Prediction

Proposition 1 (Probability forecasts and weighted votes) *Partition the pool into probability-output models $\mathcal{P}$ and one-hot experts $\mathcal{H}$ (Dietterich, 2000; Lakshminarayanan et al., 2017). For any non-negative weights summing to one,*

$$q(c \mid x) = \sum_{i \in \mathcal{P}} \alpha_i p_i(c \mid x) + \sum_{i \in \mathcal{H}} \alpha_i \mathbf{1}[\hat{y}_i(x) = c]. \tag{9}$$

When all outputs are one-hot, $\arg\max_c q(c \mid x)$ is weighted majority voting, with any fixed common tie rule. The stabilized NLL is the negative logarithm of the total probability assigned to the true class, plus its stabilizing offset.

Proof. We report the proof in Appendix A.1.

Why learn fusion separately from selection? Selection scores determine which predictors to consult, while fusion determines how their predictions are combined. The next result isolates the value of giving fusion direct access to the image representation, with the selector held fixed.

Proposition 2 (Selection scores need not suffice for optimal fusion) *Fix the predictors, representation $z(x)$, selector, and price. Suppose the selected set $S(x)$ is nonempty almost surely and determined by the score vector $s(x)$. Let $\mathcal{A}_s$ and $\mathcal{A}_{s,z}$ be all measurable weight rules in the closed simplex on $S(x)$ based on $s(x)$ and $(s(x), z(x))$, respectively. Costs and predictor identities are fixed side information. For standard NLL, define*

$$R(\alpha) = \mathbb{E}\left[-\log\left(\sum_{i \in S(X)} \alpha_i(X) p_i(Y \mid X) \right) \right]. \tag{10}$$

If $\mathcal{A}_s$ contains a finite-risk rule, then

$$\inf_{\alpha \in \mathcal{A}_{s,z}} R(\alpha) \leq \inf_{\alpha \in \mathcal{A}_s} R(\alpha). \tag{11}$$

The inequality can be strict even when every input selects the same two predictors, with identical training and reporting costs for both classes.

Proof. We report the proof in Appendix A.2.

The proposition compares the information available to unrestricted weighting rules. Its selected-set formulation matches the nonempty-mask limit described in Section 3.3; the strict example also admits finite ϵ_m, since both predictors are selected and their common mask factor cancels. The two feature-dependent weights are realized by softmax logit differences $\log 3$ and $-\log 3$. Thus, even at a fixed selected set, image features can support different mixtures when selection scores coincide. This motivates the direct image-feature input and separately learned fusion weights in our architecture. Equality is possible when the scores already retain the information needed for optimal mixing.

4.2 Average-Budget Selection and Learning

Theorem 1 (Threshold optimization of the average-budget proxy) *Let the fixed utilities $U_i(X)$ be measurable and integrable. Over all measurable binary selection policies, consider*

$$\sup_{m(\cdot)} \mathbb{E}_X\left[\sum_i m_i(X) U_i(X) \right] \quad \textit{s.t.} \quad \mathbb{E}_X\left[\sum_i m_i(X) \tilde{c}_i \right] \leq B_{\text{train}}. \tag{12}$$

For fixed $\lambda \geq 0$, its Lagrangian (Boyd & Vandenberghe, 2004) is maximized by the threshold rule (Chow, 1970; Cortes et al., 2016) $m_i^\star(x;\lambda) = \mathbf{1}[U_i(x) > \lambda \tilde{c}_i]$. Its maximized value is

$$d(\lambda) = \mathbb{E}_X \left[\sum_i \max\{0, U_i(X) - \lambda \tilde{c}_i\} \right] + \lambda B_{\text{train}}, \tag{13}$$

which upper-bounds the constrained proxy optimum.

Proof. We report the proof in Appendix A.3.

The average constraint couples inputs through one price, while fixed-price selection separates across inputs and predictors. For this utility proxy, $B_{\text{train}} - \mathbb{E}C_{\text{train}}(m^\star)$ is a subgradient of the convex function d: the maximizing policy supplies a supporting affine function. Descending along it gives the price-update direction in Eq. 8, as in primal–dual methods for long-term constraints (Mahdavi et al., 2012; Yu & Neely, 2020). The actual utilities are learned using the fused objective, whose local training signal is characterized next.

Theorem 2 (Selection-score gradient of the fused objective) *Fix input features, realized predictor outputs, λ, and positive temperatures. For one example, use the straight-through graph (Bengio et al., 2013) of Eq. 4, with Boolean attention masks held at their forward values, and set*

$$g_i = \frac{\pi_i(1-\pi_i)}{T_s}, \qquad D_j = \frac{\alpha_j[p_j(y \mid x) - q_y]}{q_y + \epsilon_\ell}. \tag{14}$$

At differentiability points of the remaining network operations, the single-example objective $\mathcal{L} = \ell_\epsilon(q, y) + \lambda(C_{\text{train}} - B_{\text{train}})$ has the local ST derivative

$$-\frac{\partial \mathcal{L}}{\partial s_i} = g_i \left[\frac{D_i}{m_i + \epsilon_m} - \lambda \tilde{c}_i \right] + \sum_j D_j \frac{\partial b_j}{\partial s_i}. \tag{15}$$

The last derivative includes the differentiable probability and mask paths through the fusion network.

Proof. We report the proof in Appendix A.4.

The direct prediction term is positive when predictor i assigns more probability to the true label than the current mixture. The fusion-weight adjustment term accounts for how changing the selection score changes all fusion logits. The cost term subtracts a price proportional to consultation cost. Together these terms train selection from the final prediction objective. For an unselected position the direct term remains $D_i/\epsilon_m = e^{b_i}(p_i(y \mid x) - q_y)/(Z(q_y + \epsilon_\ell))$, so it also supplies a local learning signal.

4.3 Adaptive Cardinality

Corollary 1 (Induced cardinality and expected usage) *The threshold set contains exactly the $k_\lambda(x) = \sum_i \mathbf{1}[U_i(x) > \lambda \tilde{c}_i]$ highest scores (Lapin et al., 2017; Cortes et al., 2024), where $k_\lambda(x)$ may be zero. Moreover,*

$$\begin{aligned} \mathbb{E}[k_\lambda(X)] &= \sum_i \Pr(U_i(X) > \lambda \tilde{c}_i), \\ \mathbb{E}[C_{\text{report}}(X)] &= \sum_i c_i \Pr(U_i(X) > \lambda \tilde{c}_i). \end{aligned} \tag{16}$$

Replacing c_i by $\tilde{c}_i$ gives the expected training cost.

Proof. We report the proof in Appendix A.5.

4.4 Budget-Price Response

Proposition 3 (Nested selections under a frozen utility function) *Freeze the encoder and utility network, with $\tilde{c}_i, c_i \geq 0$. For $0 \leq \lambda_1 \leq \lambda_2$ and every input,*

$$S_{\lambda_2}(x) \subseteq S_{\lambda_1}(x), \qquad k_{\lambda_2}(x) \leq k_{\lambda_1}(x). \tag{17}$$

Both training and reporting costs, and their expectations, are non-increasing in λ, the behaviour exploited by cost- and budget-aware deferral (Mozannar et al., 2023; DeSalvo et al., 2025).

Proof. We report the proof in Appendix A.5.

This property gives the price a predictable cost-adjustment direction for a fixed learned utility. During joint training, Eq. 8 integrates the observed budget deviations. In particular, with fixed step size η and no clipping of any preceding update,

$$\lambda_t = \lambda_0 + \eta \sum_{r=0}^{t-1} (\widehat{C}_{\mathrm{train},r} - B_{\mathrm{train}}). \tag{18}$$

This follows by telescoping the price increments. Persistent excess cost raises the price in both the forward selector and the training penalty; spare budget reduces it. The fused prediction objective, adaptive threshold rule, and average cost feedback consequently form a single learning mechanism.

Theorem 3 (Excess-risk decomposition) *Let the system produce selected sets $S(x)$, nonempty almost surely, and fused predictions q, and let $R^{\star}_{\mathrm{pool}} = \mathbb{E}[-\log \max_i p_i(Y \mid X)]$ denote the pool-oracle risk. Then*

$$\mathbb{E}[\ell(q, Y)] = R^{\star}_{\mathrm{pool}} + \mathbb{E}\left[\log \frac{\max_i p_i(Y \mid X)}{\max_{i \in S(X)} p_i(Y \mid X)}\right] + \mathbb{E}\left[\log \frac{\max_{i \in S(X)} p_i(Y \mid X)}{q_Y}\right], \tag{19}$$

where both expectations are non-negative: the selection gap vanishes iff a pool-wise best expert is selected almost surely, and the fusion gap vanishes iff $q_Y = \max_{i \in S(X)} p_i(Y \mid X)$ almost surely.

Proof. We report the proof in Appendix A.5.

The decomposition attributes the realized risk to the pool, to selection, and to fusion: threshold learning (Theorem 1) reduces the selection gap subject to the budget, while the fused-loss gradient (Theorem 2) reduces the fusion gap. Appendix A.9 characterizes the pointwise-optimal fusion behind both gaps and the attainable risk–budget frontier.

5 Experiments

We evaluate the proposed framework on three image classification benchmarks—CIFAR-100, CIFAR-10 and SVHN—through two sets of experiments: (1) a comparison with the only existing framework for top-k learning-to-defer (Montreuil et al., 2026) and Standard Top-1 L2D (Hemmer et al., 2022; Narasimhan et al., 2022) in the two-stage setting under matched budgets, (2) an ablation study that isolates the contribution of each component of the architecture.

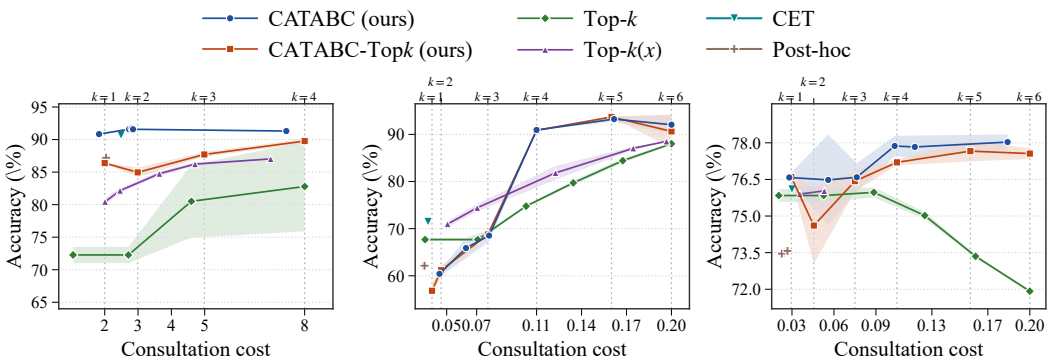
5.1 Experimental Setup

Additional details, including the dataset splits, the exact expert-construction protocols with the resulting per-expert accuracies, and the full hyperparameter settings, are provided in Appendix B.1.

Datasets. We use three standard benchmarks. CIFAR-100 and CIFAR-10 (Krizhevsky, 2009) contain $60{,}000$ colour images of resolution 32×32 distributed over 100 and 10 classes respectively, and SVHN (Goodfellow et al., 2013) is a large-scale street-view digit benchmark with more than $600{,}000$ images.

Models. All methods share the same feature extractor, and our fusion module is a two-layer Transformer encoder with masked self-attention. The baseline keeps the original implementation of the reference, in which the adaptive Top-$k(x)$ cardinality function is built on a CLIP-based image encoder (Radford et al., 2021). CIFAR-100 and SVHN adopt the model settings of Montreuil et al. (2026), whereas CIFAR-10 uses a new configuration. The exact backbone and trainable parameters of each dataset is specified in Appendix B.1 B.2.

Experts. The expert pools follow Montreuil et al. (2026) and differ per benchmark: CIFAR-100 uses a trained main predictor together with five synthesised experts of strongly overlapping competence, SVHN uses six experts trained on partially overlapping subsets of the training set, and CIFAR-10 uses four experts with heterogeneous backbones (He et al., 2016; Huang et al., 2017) specialised in overlapping class subsets. In all three cases the pool is frozen.



(a) CIFAR-10 (b) CIFAR-100 (c) SVHN

Figure 2: Test accuracy against the realised consultation cost on CIFAR-10, CIFAR-100 and SVHN. The bottom axis is labelled with the budgets used for training, while the top axis marks the expected number of queried experts k of CATABC-Top$k$, positioned at the costs that method actually realises. Results are averaged over four independent runs. Detailed data are reported in Appendix B.4.1.

Budgets. The budget of each dataset is derived from the reference top-k method (Montreuil et al., 2026): we take the cost it actually realises on that dataset, round it up to the next integer, and train our method under exactly that budget.

Metrics. We report two metrics: the classification accuracy and the realised cost, defined as the average consultation cost incurred per input. For the baseline of Montreuil et al. (2026) we report the *weighted-voting* accuracy produced by its own aggregation rule.

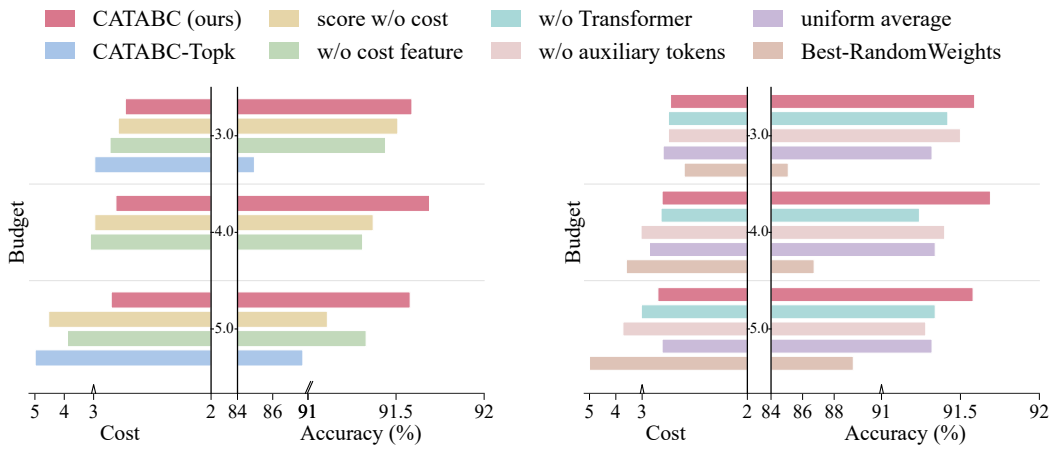
5.2 Main Experiments

Comparison with the baselines. Figure 2 compares our two methods with fixed-cardinality Top-k and adaptive Top-$k(x)$ (Montreuil et al., 2026), CET (Hemmer et al., 2022), and the two-stage PostHoc estimator (Narasimhan et al., 2022). On CIFAR-10 and SVHN, our method attains the highest accuracy at every budget and, at several budgets, lower cost than any baseline (e.g., at budget 5 on CIFAR-10, 91.58% at cost 2.85 vs. Top-$k(x)$: 86.23% at cost 4.71). On CIFAR-100, it is not competitive at low budgets but overtakes baselines once relaxed (93.25% at $B = 0.17$ vs. 87.05% for Top-$k(x)$). CET is strongest at the smallest budgets (71.5% at $B = 0.05$), consistent with its aggregation: CET commits to one entity and concentrates a small budget on the most reliable source, whereas our fusion combines machine probabilities with one-hot human encodings and needs a few entities to become informative.

Budget control and best performance. On all three datasets, both our methods never exceed the prescribed budget, whereas the baselines use a hyperparameter fixed before training that cannot enforce an inference-time budget. Our methods also achieve the highest reported accuracy on each dataset—91.69% on CIFAR-10, 93.71% on CIFAR-100, and 78.03% on SVHN. Figure 2 shows that budget usage differs across benchmarks: CIFAR-100 is exploited rather fully, while CIFAR-10 and SVHN leave intermediate budgets partly unused (at budget 5 on CIFAR-10, only 2.85 is spent), because the optimal cost lies below the budget there. Finally, because we fuse predicted probabilities rather than taking a majority vote, Top-k's SVHN accuracy first rises then falls (76.0% to 71.9%), while ours does not: adding an entity only reweights the existing evidence.

5.3 Ablation Study

We ablate the two learned modules of our framework, the selector and the fusion network, as well as the threshold within the selector and the Lagrangian multiplier that controls the budget, using CIFAR-10 under the budgets 3, 4 and 5. The detailed results of every configuration are reported in Appendix B.4.2.



(a) Effect of selector (b) Effect of fusion network

Figure 3: Subfigure (a) presents the experimental results of various variants related to the selector, while subfigure (b) shows those for the fusion network. w/o denotes without, specifically, w/o Transformer replaces the Transformer blocks with standard linear layers and w/o auxiliary tokens removes both the cost embeddings and selector outputs from the Transformer blocks. Best-RandomWeights refers to the optimal weights obtained from 50 random sampling trials.

Effect of the selector. Removing the cost-related components of the selector raises the realised cost without improving the accuracy, and the degradation grows with the budget. This is most visible at high budgets, where the cost exceeds that of the complete method by a wide margin while the drop in accuracy is also larger than at low budgets. CATABC-Top$k$, which forces a fixed number of experts, spends almost the whole budget yet attains the lowest accuracy of every variant, indicating that thresholding is preferable to a softmax top-k selection (Figure 3a).

Effect of the fusion network. The complete method achieves the best accuracy at every budget. Replacing the Transformer with a linear layer (*w/o Transformer*) degrades both metrics across all budgets, supporting self-attention for learning predictor-token interactions. Removing cost and selector-output embeddings (*w/o auxiliary tokens*) weakens cost awareness: at budget 5, cost rises to 3.73 and accuracy drops to 91.28%. Uniform-weight averaging (*uniform average*) also underperforms CATABC at every budget, highlighting the value of learned combination weights in ensemble methods (Dietterich, 2000).

6 CONCLUSION

We introduced CATABC, a budget-constrained adaptive-k architecture for learning to defer to a heterogeneous pool of predictors and human experts. It decides per input which experts to consult, fuses their predictions with masked attention, and keeps the average consultation cost within a prescribed budget through a dual price. We proved that the threshold selection optimally solves the Lagrangian of the average-budget proxy, that the price is the slope of the attainable risk–budget frontier, and that the excess risk decomposes into pool, selection and fusion gaps. Experiments on CIFAR-10, CIFAR-100 and SVHN show that the realised cost never exceeds the budget while the accuracy surpasses the baselines in most settings.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and for LaTeX typesetting of author-written material, and for drafting and restructuring the prose of the manuscript, specifically in the introduction, related work, method, theory and experiment sections. We have not used generative AI tools to generate research ideas, hypotheses or research directions, to design the architecture or the training algorithm, to implement, configure or run the experiments, or to produce experimental results. Every reported number, table entry and figure comes from our own runs of the code. The

literature search, the selection of the cited works and the bibliography were likewise carried out by us.

Reproducibility statement

We have organised the paper so that every reported result can be reproduced from the material provided. Appendix B documents the construction of the expert pools, the dataset splits and the hyperparameters of our method and of each baseline, together with the accuracy and cost of every expert; the code that trains and evaluates all methods and the scripts that produce our figures and tables accompany the submission. Each reported accuracy and realised cost is averaged over four runs with different random seeds and is accompanied by one standard deviation. For the theoretical part, Appendix A states the assumptions and proves each claim, while Appendix A.12 summarises the training procedure.

References

Jean Vieira Alves, Diogo Leitão, Sérgio Jesus, Marco O. P. Sampaio, Javier Liébana, Pedro Saleiro, Mario A. T. Figueiredo, and Pedro Bizarro. Cost-sensitive learning to defer to multiple experts with workload constraints. *Transactions on Machine Learning Research (TMLR)*, 2024.

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

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

Yuzhou Cao, Hussein Mozannar, Lei Feng, Hongxin Wei, and Bo An. In defense of softmax parametrization for calibrated and consistent learning to defer. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 38485–38503, 2023.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory (ALT)*, 2016.

Corinna Cortes, Anqi Mao, Christopher Mohri, Mehryar Mohri, and Yutao Zhong. Cardinality-aware set prediction and top-k classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Giulia DeSalvo, Clara Mohri, Mehryar Mohri, and Yutao Zhong. Budgeted multiple-expert deferral. *arXiv preprint arXiv:2510.26706*, 2025.

Thomas G. Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems (MCS)*, pp. 1–15, 2000.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed H. Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*, 2024.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A. Saurous, Jascha Sohl-Dickstein, et al. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

Hugh Everett. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.

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.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7436–7456, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Patrick Hemmer, Sebastian Schellhammer, Michael Vössing, Johannes Jakubik, and Gerhard Satzger. Forming effective human-ai teams: Building machine learning models that complement the capabilities of multiple experts. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2478–2484, 2022.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.

Wittawat Jitkrittum, Neha Gupta, Aditya K. Menon, Harikrishna Narasimhan, Ankit Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 9891–9906, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 6402–6413, 2017.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1533–1554, 2017.

David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: Online convex optimization with long term constraints. *Journal of Machine Learning Research*, 13(1):2503–2528, 2012.

Anqi Mao, Christopher Mohri, Mehryar Mohri, and Yutao Zhong. Two-stage learning to defer with multiple experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 3578–3606, 2023.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Principled approaches for learning to defer with multiple experts. In *International Workshop on Combinatorial Image Analysis (IWCIA)*, pp. 107–135, 2024a.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Realizable h-consistent and bayes-consistent loss functions for learning to defer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Mastering multiple-expert routing: Realizable h-consistency and strong guarantees for learning to defer. In *International Conference on Machine Learning (ICML)*, 2025.

Yannis Montreuil, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. Why ask one when you can ask k? learning-to-defer to the top-k experts. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2504.12988.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning (ICML)*, pp. 7076–7087, 2020.

Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David Sontag. Who should predict? exact algorithms for learning to defer to humans. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 10520–10545, 2023.

Harikrishna Narasimhan, Wittawat Jitkrittum, Aditya K. Menon, Ankit Rawat, and Sanjiv Kumar. Post-hoc estimators for learning to defer to an expert. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 29292–29304, 2022.

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of Optimization Theory and Applications*, 142(1):205–228, 2009.

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 with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

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 (ICLR)*, 2017.

Joshua Strong, Emma Sun, Harry Rogers, Helen Higham, and J. Alison Noble. Learning to defer: A survey. Preprint, 2025. https://zenodo.org/records/17843044.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL `https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

Rajeev Verma and Eric Nalisnick. Calibrated learning to defer with one-vs-all classifiers. In *International Conference on Machine Learning (ICML)*, pp. 22184–22202, 2022.

Rajeev Verma, Daniel Barrejón, and Eric Nalisnick. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 11415–11434, 2023.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 511–518, 2001.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*, 2024.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

Hao Yu and Michael J. Neely. A low complexity algorithm with $o(\sqrt{T})$ regret and $o(1)$ constraint violations for online convex optimization with long term constraints. *Journal of Machine Learning Research*, 21(1):1–24, 2020.

A Detailed Proofs and Supplementary Analysis

This appendix expands the arguments in Section 4 in the same order as the main text. We use training costs $\tilde{c}_i$, reporting costs c_i, and the average training budget B_{train}. The default selection threshold is zero. In the gradient calculations, m_i denotes the forward value of the straight-through (ST) mask, with its backward derivative defined by Eq. 4. Throughout, $b_i = a_i/T_f$ denotes the temperature-scaled fusion logit.

A.1 Probability Fusion and Single-Predictor Strategies

Proof of Proposition 1. For each realized response, a categorical predictor has distribution $p_i(c \mid x) = \mathbf{1}[\hat{y}_i(x) = c]$. Partitioning the pool into probability models $\mathcal{P}$ and categorical predictors $\mathcal{H}$ therefore gives

$$\begin{aligned} q(c \mid x) &= \sum_{i=1}^{N} \alpha_i(x) p_i(c \mid x) \\ &= \sum_{i \in \mathcal{P}} \alpha_i(x) p_i(c \mid x) + \sum_{i \in \mathcal{H}} \alpha_i(x) \mathbf{1}[\hat{y}_i(x) = c]. \end{aligned}$$

If every response is categorical, this reduces to the total weight of the predictors voting for class c. In particular,

$$\begin{aligned} q(y \mid x) &= \sum_{i:\hat{y}_i(x)=y} \alpha_i(x), \\ \ell_\epsilon(q, y) &= -\log\left(\sum_{i:\hat{y}_i(x)=y} \alpha_i(x) + \epsilon_\ell \right), \\ \arg\max_c q(c \mid x) &= \arg\max_c \sum_{i:\hat{y}_i(x)=c} \alpha_i(x). \end{aligned}$$

Thus probability fusion includes weighted majority voting, with the same tie-breaking convention. The identities hold conditionally on any realized responses, so stochastic categorical predictors are covered as well. ∎

Corollary 2 (Single-predictor strategies are contained in fusion) *Fix a nonempty available set S. Over measurable weight functions taking arbitrary values in the closed simplex on S,*

$$\inf_{\alpha(\cdot)} \mathbb{E}[\ell_\epsilon(q_\alpha, Y)] \le \min_{i \in S} \mathbb{E}[\ell_\epsilon(p_i, Y)]. \tag{20}$$

The same statement holds for standard NLL whenever the expectations are defined.

Proof. Fix a nonempty available set S, and denote its closed simplex by $\Delta(S)$. For any $i \in S$, the constant measurable rule $\alpha(x) = e_i$ belongs to the allowed class. Consequently,

$$\begin{aligned} q_{e_i}(c \mid x) &= \sum_{j \in S} (e_i)_j p_j(c \mid x) = p_i(c \mid x), \\ \inf_{\alpha(\cdot) \in \Delta(S)} \mathbb{E}[\ell_\epsilon(q_\alpha, Y)] &\le \mathbb{E}[\ell_\epsilon(q_{e_i}, Y)] \\ &= \mathbb{E}[\ell_\epsilon(p_i, Y)]. \end{aligned}$$

Taking the minimum over $i \in S$ proves the result. The same substitution applies to standard NLL, with expectations interpreted in the extended nonnegative reals. This establishes inclusion of single-predictor strategies in the unrestricted fusion class. ∎

A.2 Information for Selecting and Combining Predictors

Proof of Proposition 2. Keep the predictors, representation, selector, and price fixed as in the proposition. Any measurable rule $f(s)$ can be written as a rule $\tilde{f}(s,z) = f(s)$ that ignores its second argument. Its weights and selected set are unchanged, so

$$\mathcal{A}_s \subseteq \mathcal{A}_{s,z},$$
$$\inf_{\alpha \in \mathcal{A}_{s,z}} R(\alpha) \le \inf_{\alpha \in \mathcal{A}_s} R(\alpha).$$

To establish strictness, consider a binary problem with two equally likely inputs. Let their conditional label probabilities and predictor outputs be

$$\begin{aligned}\Pr(X = x_a) = \Pr(X = x_b) &= \tfrac{1}{2},\\ \Pr(Y = 1 \mid x_a) = 0.7, &\qquad \Pr(Y = 1 \mid x_b) = 0.3,\\ p_1(\cdot \mid x_a) = p_1(\cdot \mid x_b) &= (0.9, 0.1),\\ p_2(\cdot \mid x_a) = p_2(\cdot \mid x_b) &= (0.1, 0.9).\end{aligned}$$

Set $\tilde{c}_1 = \tilde{c}_2 = c_1 = c_2 = 1$, $B_{\text{train}} = 2$, $U_1 = U_2 = 1$, and $\lambda = 0$. Both inputs then have $s = (1,1)$ and $S = \{1,2\}$. Every score-only rule uses the same first-predictor weight $w \in [0,1]$ on both inputs. Writing $r = 0.1 + 0.8w$, its risk is

$$\begin{aligned}R_s(w) &= -\tfrac{1}{2}\big[0.7 \log r + 0.3 \log(1-r)\big] - \tfrac{1}{2}\big[0.3 \log r + 0.7 \log(1-r)\big]\\ &= -\tfrac{1}{2} \log\big(r(1-r)\big)\\ &\ge \log 2,\end{aligned}$$

where equality holds at $r = 1/2$, equivalently $w = 1/2$.

Now choose $z(x_a) \neq z(x_b)$. A measurable rule of (s,z) can assign $w_a = 3/4$ and $w_b = 1/4$. These weights recover the two conditional label probabilities:

$$\begin{aligned}r_a &= 0.1 + 0.8(3/4) = 0.7,\\ r_b &= 0.1 + 0.8(1/4) = 0.3.\end{aligned}$$

For $\eta \in (0,1)$, conditional NLL is minimized at $r = \eta$, since

$$\begin{aligned}L_\eta(r) &= -\eta \log r - (1-\eta)\log(1-r),\\ \frac{dL_\eta}{dr} &= \frac{r-\eta}{r(1-r)}, \qquad 0 < r < 1.\end{aligned}$$

The constructed rule attains this minimum at both inputs. Hence, writing $H_{\text{bin}}(u) = -u \log u - (1-u)\log(1-u)$,

$$\begin{aligned}\inf_{\alpha \in \mathcal{A}_{s,z}} R(\alpha) &= \tfrac{1}{2} H_{\text{bin}}(0.7) + \tfrac{1}{2} H_{\text{bin}}(0.3)\\ &= H_{\text{bin}}(0.7) < \log 2 = \inf_{\alpha \in \mathcal{A}_s} R(\alpha).\end{aligned}$$

Both rules have $C_{\text{train}} = C_{\text{report}} = 2$ at every input. Moreover, because both predictors are selected, their common factor $1 + \epsilon_m$ cancels from Eq. 6. The weights $3/4$ and $1/4$ are obtained by scaled-logit differences $b_1 - b_2 = \log 3$ and $-\log 3$, respectively. The construction therefore exhibits the benefit of additional fusion information at exactly the same consultation cost. ∎

A.3 Threshold Selection under an Average Budget

Proof of Theorem 1. Let $V(B_{\text{train}})$ be the optimal value of Eq. 12, with fixed measurable integrable utilities. For a measurable binary policy and $\lambda \ge 0$, its Lagrangian can be rearranged as

$$\begin{aligned}\mathcal{J}(m,\lambda) &= \mathbb{E}\left[\sum_i m_i(X) U_i(X)\right] + \lambda\left(B_{\text{train}} - \mathbb{E}\left[\sum_i m_i(X)\tilde{c}_i\right]\right)\\ &= \mathbb{E}\left[\sum_i m_i(X)(U_i(X) - \lambda \tilde{c}_i)\right] + \lambda B_{\text{train}}.\end{aligned}$$

For every input and coordinate, maximizing the corresponding binary term gives

$$\max_{m_i \in \{0,1\}} m_i(U_i(x) - \lambda \tilde{c}_i) = \max\{0, U_i(x) - \lambda \tilde{c}_i\},$$
$$m_i^\star(x;\lambda) = \mathbf{1}[U_i(x) > \lambda \tilde{c}_i].$$

At equality, either binary choice is optimal; the strict threshold selects zero. The displayed policy is measurable and simultaneously attains the coordinatewise maxima. Thus

$$d(\lambda) := \sup_m \mathcal{J}(m,\lambda) = \mathbb{E}\left[\sum_i \max\{0, U_i(X) - \lambda \tilde{c}_i\}\right] + \lambda B_{\text{train}}.$$

For any feasible policy, nonnegativity of the multiplier implies

$$\mathbb{E}\left[\sum_i m_i(X) U_i(X)\right] \le \mathcal{J}(m,\lambda) \le d(\lambda),$$
$$V(B_{\text{train}}) \le \inf_{\lambda \ge 0} d(\lambda).$$

This is the weak-duality upper bound for the utility maximization problem. The shared multiplier coordinates the average budget, while its fixed value permits pointwise selection. ∎

Price direction for the fixed-utility proxy. The function d is convex as a supremum of affine functions. For any other price $\mu \ge 0$, evaluate its supremum at $m_\lambda^\star$ to obtain

$$\begin{aligned} d(\mu) &\ge \mathcal{J}(m_\lambda^\star, \mu) \\ &= d(\lambda) + (\mu - \lambda)\big(B_{\text{train}} - \mathbb{E}[C_{\text{train}}(m_\lambda^\star)]\big). \end{aligned}$$

Therefore $B_{\text{train}} - \mathbb{E}[C_{\text{train}}(m_\lambda^\star)]$ is a subgradient. A projected subgradient descent step on this proxy dual is

$$\lambda^+ = \Pi_{[0,\lambda_{\max}]}\big(\lambda + \eta\big(\mathbb{E}[C_{\text{train}}(m_\lambda^\star)] - B_{\text{train}}\big)\big).$$

This establishes the feedback direction used in Eq. 8. In joint training, the utilities evolve through the fused objective; the next derivation characterizes that learning signal directly.

A.4 Full Selection-Score Gradient

Proof of Theorem 2. Fix the input features, realized responses, price, and Boolean attention masks as in the theorem. Write $p_i = p_i(y \mid x)$ and $q_y = \sum_i \alpha_i p_i$. First, the additional normalization in Section 3.3 preserves the gradient after adjusting the NLL offset. Indeed, since $\sum_i \alpha_i = 1$,

$$\begin{aligned} \bar{\alpha}_i = \frac{\alpha_i}{1+\epsilon_w}, \qquad \bar{q}_y &= \frac{q_y}{1+\epsilon_w}, \\ -\log(\bar{q}_y + \epsilon_n) &= -\log\big(q_y + (1+\epsilon_w)\epsilon_n\big) + \log(1+\epsilon_w) \\ &= \ell_\epsilon(q,y) + \log(1+\epsilon_w), \end{aligned}$$

where $\epsilon_\ell = (1+\epsilon_w)\epsilon_n > 0$. All offsets are fixed, so the last term is constant with respect to the model parameters.

Next, introduce $h_j = b_j + \log(m_j + \epsilon_m)$. Differentiating the softmax and its probability mixture yields

$$\begin{aligned} \alpha_k &= \frac{e^{h_k}}{\sum_j e^{h_j}}, \\ \frac{\partial \alpha_k}{\partial h_j} &= \alpha_k(\mathbf{1}[k=j] - \alpha_j), \\ \frac{\partial q_y}{\partial h_j} &= \sum_k p_k \alpha_k(\mathbf{1}[k=j] - \alpha_j) \\ &= \alpha_j(p_j - q_y), \\ \frac{\partial \ell_\epsilon}{\partial h_j} &= -\frac{\alpha_j(p_j - q_y)}{q_y + \epsilon_\ell} = -D_j. \end{aligned}$$

The ST graph supplies the local mask derivative, and hence

$$g_i = \frac{\pi_i(1-\pi_i)}{T_s}, \qquad \frac{\partial m_j}{\partial s_i} = \mathbf{1}[i=j]g_i,$$
$$\frac{\partial h_j}{\partial s_i} = \frac{\partial b_j}{\partial s_i} + \frac{\mathbf{1}[i=j]g_i}{m_j+\epsilon_m},$$
$$\frac{\partial C_{\text{train}}}{\partial s_i} = \sum_j \tilde{c}_j \frac{\partial m_j}{\partial s_i} = \tilde{c}_i g_i.$$

Combining the prediction and cost derivatives now gives the full expression:

$$\begin{aligned} -\frac{\partial \mathcal{L}}{\partial s_i} &= -\sum_j \frac{\partial \ell_\epsilon}{\partial h_j}\frac{\partial h_j}{\partial s_i} - \lambda \frac{\partial C_{\text{train}}}{\partial s_i} \\ &= \sum_j D_j \left(\frac{\partial b_j}{\partial s_i} + \frac{\mathbf{1}[i=j]g_i}{m_j+\epsilon_m} \right) - \lambda \tilde{c}_i g_i \\ &= g_i \left[\frac{D_i}{m_i+\epsilon_m} - \lambda \tilde{c}_i \right] + \sum_j D_j \frac{\partial b_j}{\partial s_i}. \end{aligned} \tag{21}$$

The derivative of b_j includes all differentiable paths through the fusion network. In particular, changing s_i changes both π_i and the ST mask used in the token features. Treating these two token inputs as separate arguments of b_j, the chain rule reads

$$\frac{\partial b_j}{\partial s_i} = \frac{\partial b_j}{\partial \pi_i} g_i + \frac{\partial b_j}{\partial m_i} g_i,$$

with Boolean attention masks held fixed. Finally, substitution of the actual weight formula makes the direct prediction term explicit:

$$\frac{D_i}{m_i+\epsilon_m} = \frac{e^{b_i}}{Z}\frac{p_i - q_y}{q_y + \epsilon_\ell}.$$

This identity holds for both selected and unselected positions, including an empty forward mask. Its sign is determined by whether predictor i assigns more probability to the true class than the current mixture. Equation 21 combines this prediction signal with the fusion-weight adjustment and consultation-cost penalty to explain how the fused objective trains the selection scores. ∎

A.5 Adaptive Cardinality and Price Response

Proof of Corollary 1. For any included index i and excluded index j,

$$i \in S_\lambda(x),\ j \notin S_\lambda(x) \implies s_i(x) > 0 \geq s_j(x).$$

Thus the selected set consists of exactly the $k_\lambda(x)$ highest scores, including the empty set when $k_\lambda(x) = 0$. Let $A_i = \{U_i(X) > \lambda \tilde{c}_i\}$. Linearity of expectation gives

$$\mathbb{E}[k_\lambda(X)] = \mathbb{E}\left[\sum_i \mathbf{1}[A_i]\right] = \sum_i \Pr(A_i),$$
$$\mathbb{E}[C_{\text{train}}(X)] = \sum_i \tilde{c}_i \mathbb{E}[\mathbf{1}[A_i]] = \sum_i \tilde{c}_i \Pr(A_i),$$
$$\mathbb{E}[C_{\text{report}}(X)] = \sum_i c_i \mathbb{E}[\mathbf{1}[A_i]] = \sum_i c_i \Pr(A_i).$$

Linearity makes these identities valid for statistically dependent gates as well, expressing average usage directly through each gate's selection probability. ∎

Proof of Proposition 3. Fix the encoder and utility network, and let $0 \le \lambda_1 \le \lambda_2$. Nonnegative training costs imply, for each input,

$$\begin{aligned} i \in S_{\lambda_2}(x) &\implies U_i(x) > \lambda_2 \tilde{c}_i \\ &\implies U_i(x) > \lambda_1 \tilde{c}_i \\ &\implies i \in S_{\lambda_1}(x). \end{aligned}$$

Hence $S_{\lambda_2}(x) \subseteq S_{\lambda_1}(x)$. For any nonnegative weights w_i, summing over the removed indices yields

$$\sum_i w_i m_i(x;\lambda_1) - \sum_i w_i m_i(x;\lambda_2) = \sum_{i \in S_{\lambda_1}(x) \setminus S_{\lambda_2}(x)} w_i \ge 0.$$

Taking $w_i = 1$, $\tilde{c}_i$, or c_i proves monotonicity of cardinality, training cost, and reporting cost, respectively. Taking expectations preserves the inequalities. For $\tilde{c}_i = 0$, the gate is independent of the price; at threshold equality the strict inequality sets the gate to zero. ∎

Proof of Theorem 3. Fix a realization (x, y) and abbreviate $p_i = p_i(y \mid x)$ and $S = S(x)$. Since $S \subseteq [N]$ and the fusion weights lie in the simplex on S,

$$\max_i p_i \ge \max_{i \in S} p_i = \Big(\max_{i \in S} p_i\Big)\Big(\sum_{j \in S} \alpha_j\Big) \ge \sum_{j \in S} \alpha_j p_j = q_y,$$

where the last inequality bounds each p_j by the selected-set maximum. All three quantities are positive, so the log-ratio of the fused probability splits through the selected-set maximum:

$$-\log q_y = -\log \max_i p_i + \log \frac{\max_i p_i}{\max_{i \in S} p_i} + \log \frac{\max_{i \in S} p_i}{q_y},$$

and both log-ratios are non-negative by the chain above. Taking expectations over (X, Y) gives Eq. 19. The first gap vanishes iff $\max_{i \in S(X)} p_i = \max_i p_i$ almost surely, i.e., a pool-wise best predictor is selected; the second vanishes iff $q_Y = \max_{i \in S(X)} p_i$ almost surely, which requires the mixture to concentrate on the best selected predictors. ∎

Accumulated budget deviations and clipping. Write $\delta_t = \widehat{C}_{\mathrm{train},t} - B_{\mathrm{train}}$. If no previous update is clipped and the step size η is fixed, Eq. 8 implies

$$\begin{aligned} \lambda_{r+1} - \lambda_r &= \eta \delta_r, \\ \lambda_t - \lambda_0 &= \sum_{r=0}^{t-1} (\lambda_{r+1} - \lambda_r) \\ &= \eta \sum_{r=0}^{t-1} (\widehat{C}_{\mathrm{train},r} - B_{\mathrm{train}}). \end{aligned}$$

For completeness, define the correction introduced by clipping as $e_r = \Pi_{[0,\lambda_{\max}]}(\lambda_r + \eta\delta_r) - (\lambda_r + \eta\delta_r)$. The exact identity for all updates is then

$$\lambda_t = \lambda_0 + \eta \sum_{r=0}^{t-1} \delta_r + \sum_{r=0}^{t-1} e_r.$$

This expresses the price as accumulated budget feedback adjusted at the projection boundaries. For $\eta > 0$ and $\lambda_{\max} > 0$, an unchanged price in a single update satisfies

$$\lambda_{t+1} = \lambda_t \quad \implies \quad \begin{cases} \delta_t = 0, & 0 < \lambda_t < \lambda_{\max}, \\ \delta_t \le 0, & \lambda_t = 0, \\ \delta_t \ge 0, & \lambda_t = \lambda_{\max}. \end{cases}$$

The interior equality identifies a balanced epoch, while the boundary conditions describe the deviations absorbed by clipping. Together with the fixed-utility nesting result, these identities explain how the shared price responds to resource usage.

A.6 FINITE-MASK RESIDUAL WEIGHTS

The following supplementary result quantifies the finite-ϵ_m weighting used in Section 3.3.

Lemma 1 (Residual weight outside a nonempty mask) *Fix a nonempty selected set S and scaled logits satisfying $|b_i| \le A$. Let $R_S = \sum_{j \notin S} \alpha_j$ and $\kappa_S = (N - |S|)e^{2A}/|S|$. Then*

$$R_S \le \frac{\epsilon_m \kappa_S}{1 + \epsilon_m + \epsilon_m \kappa_S} \le \epsilon_m \kappa_S. \tag{22}$$

For fixed logits, as $\epsilon_m \to 0$, the weights converge to the softmax on S, with zero weight outside S.

Proof. Set $A_S = \sum_{i \in S} e^{b_i} > 0$ and $A_{\bar{S}} = \sum_{j \notin S} e^{b_j}$. The implemented normalization gives

$$\begin{aligned} Z &= (1 + \epsilon_m)A_S + \epsilon_m A_{\bar{S}}, \\ R_S &= \frac{\epsilon_m A_{\bar{S}}}{(1 + \epsilon_m)A_S + \epsilon_m A_{\bar{S}}} \\ &= \frac{\epsilon_m r_S}{1 + \epsilon_m + \epsilon_m r_S}, \qquad r_S = \frac{A_{\bar{S}}}{A_S}. \end{aligned}$$

Bounded logits imply

$$0 \le r_S \le \frac{(N - |S|)e^{A}}{|S|e^{-A}} = \kappa_S.$$

The expression for R_S is increasing in r_S, proving Eq. 22. For fixed logits, direct substitution gives

$$\lim_{\epsilon_m \to 0} \alpha_i = \begin{cases} e^{b_i}/A_S, & i \in S, \\ 0, & i \notin S. \end{cases}$$

This proves the selected-set limit. ∎

The residual also controls the change in the prediction distribution. Define $q_S = \sum_{i \in S} e^{b_i} p_i / A_S$. When $S \neq [N]$, define $q_{\bar{S}}$ analogously. Then

$$q = (1 - R_S)q_S + R_S q_{\bar{S}}, \tag{23}$$

$$\|q - q_S\|_1 = R_S \|q_{\bar{S}} - q_S\|_1 \le 2R_S, \tag{24}$$

$$|q_y - (q_S)_y| \le R_S, \tag{25}$$

$$|\ell_\epsilon(q, y) - \ell_\epsilon(q_S, y)| \le \frac{R_S}{\epsilon_\ell}. \tag{26}$$

The last inequality follows from the mean value theorem, because the absolute derivative of $-\log(u + \epsilon_\ell)$ on $u \in [0, 1]$ is at most $1/\epsilon_\ell$. If $S = [N]$, then $R_S = 0$ and $q = q_S$ exactly.

For the empty mask, all positions share the same positive factor ϵ_m, so its exact numerical behavior is instead

$$S = \emptyset \quad \Longrightarrow \quad \alpha_i = \frac{e^{b_i}\epsilon_m}{\epsilon_m \sum_j e^{b_j}} = \frac{e^{b_i}}{\sum_j e^{b_j}}, \tag{27}$$

$$C_{\text{train}} = C_{\text{report}} = 0. \tag{28}$$

Thus the implementation produces a normalized mixture for every mask, while consultation costs retain the mask-based definition in Eq. 2.

A.7 FUSION QUALITY: WEIGHTED-AVERAGE BOUND AND EXPERT COMPLEMENTARITY

The following lemma collects two quantitative properties of the fused loss: a weighted-average bound with an exact gap, and the effect of error sharing between two categorical experts.

Lemma 2 (Weighted-average bound and correctness complementarity) *(a) Fix an input, a realized label y, a nonempty index set S, and weights α in the simplex on S. For $\phi(u) = -\log(u + \epsilon_\ell)$,*

$$\ell_\epsilon(q, y) = \phi\Big(\sum_{i \in S} \alpha_i p_i(y \mid x)\Big) \le \sum_{i \in S} \alpha_i \, \ell_\epsilon(p_i, y), \tag{29}$$

and the gap equals the log ratio of the arithmetic and geometric means of the shifted probabilities $\{p_i(y \mid x) + \epsilon_\ell\}_{i \in S}$*:*

$$\sum_{i \in S} \alpha_i \ell_\epsilon(p_i, y) - \ell_\epsilon(q, y) = \log \frac{\sum_{i \in S} \alpha_i \big(p_i(y \mid x) + \epsilon_\ell\big)}{\prod_{i \in S} \big(p_i(y \mid x) + \epsilon_\ell\big)^{\alpha_i}} \geq 0, \tag{30}$$

with equality iff $p_i(y \mid x)$ *is constant on the support of* α*. (b) Consider two categorical experts with fixed fusion weights* w *and* $1 - w$*,* $0 < w < 1$*. Let* $Z_i = \mathbf{1}[\hat{Y}_i = Y]$*,* $a_i = \Pr(Z_i = 1)$*, and* $t = \Pr(Z_1 = Z_2 = 1)$*. The expected stabilized NLL of the fused prediction is*

$$\begin{aligned} R(t) = & -t \log(1 + \epsilon_\ell) - (a_1 - t) \log(w + \epsilon_\ell) \\ & - (a_2 - t) \log(1 - w + \epsilon_\ell) - (1 - a_1 - a_2 + t) \log \epsilon_\ell, \end{aligned} \tag{31}$$

which is affine in t *with*

$$\frac{\partial R}{\partial t} = \log\left(1 + \frac{w(1-w)}{\epsilon_\ell(1 + \epsilon_\ell)}\right) > 0. \tag{32}$$

Given the marginals, t *ranges over the Fréchet interval* $\max\{0, a_1 + a_2 - 1\} \leq t \leq \min\{a_1, a_2\}$*; since the shared-error probability is* $\Pr(Z_1 = Z_2 = 0) = 1 - a_1 - a_2 + t$*, the risk is strictly decreasing as the two experts share fewer errors, at fixed individual accuracies and fixed weights.*

Proof. (a) The function $\phi(u) = -\log(u + \epsilon_\ell)$ satisfies $\phi''(u) = (u + \epsilon_\ell)^{-2} > 0$ on $u \geq 0$, so it is convex. Since the weights lie in the simplex, Jensen's inequality gives

$$\begin{aligned} \ell_\epsilon(q, y) &= \phi\Big(\sum_{i \in S} \alpha_i p_i(y \mid x)\Big) \\ &\leq \sum_{i \in S} \alpha_i \, \phi\big(p_i(y \mid x)\big) = \sum_{i \in S} \alpha_i \, \ell_\epsilon(p_i, y), \end{aligned}$$

which is Eq. 29. Expanding the gap between the two sides,

$$\begin{aligned} \sum_{i \in S} \alpha_i \ell_\epsilon(p_i, y) - \ell_\epsilon(q, y) &= -\sum_{i \in S} \alpha_i \log\big(p_i(y \mid x) + \epsilon_\ell\big) + \log\Big(\sum_{i \in S} \alpha_i \big(p_i(y \mid x) + \epsilon_\ell\big)\Big) \\ &= \log \frac{\sum_{i \in S} \alpha_i \big(p_i(y \mid x) + \epsilon_\ell\big)}{\prod_{i \in S} \big(p_i(y \mid x) + \epsilon_\ell\big)^{\alpha_i}} \geq 0, \end{aligned}$$

where the last line is the arithmetic-geometric mean inequality applied to the shifted probabilities $\{p_i(y \mid x) + \epsilon_\ell\}_{i \in S}$ with weights α; its equality condition is exactly that $p_i(y \mid x)$ is constant on the support of α.

(b) Condition on the four correctness patterns of (Z_1, Z_2). The fused probability assigned to the true class in each case is

$$q_Y = \begin{cases} w + (1 - w) = 1, & Z_1 = 1,\ Z_2 = 1, \\ w, & Z_1 = 1,\ Z_2 = 0, \\ 1 - w, & Z_1 = 0,\ Z_2 = 1, \\ 0, & Z_1 = 0,\ Z_2 = 0, \end{cases}$$

with respective probabilities t, $a_1 - t$, $a_2 - t$, and $1 - a_1 - a_2 + t$. Taking expectations of $-\log(q_Y + \epsilon_\ell)$ over the four cases,

$$\begin{aligned} R(t) = & -t \log(1 + \epsilon_\ell) - (a_1 - t) \log(w + \epsilon_\ell) \\ & - (a_2 - t) \log(1 - w + \epsilon_\ell) - (1 - a_1 - a_2 + t) \log \epsilon_\ell, \end{aligned}$$

which is Eq. 31. Differentiating this affine expression term by term,

$$\begin{aligned} \frac{\partial R}{\partial t} &= -\log(1 + \epsilon_\ell) + \log(w + \epsilon_\ell) + \log(1 - w + \epsilon_\ell) - \log \epsilon_\ell \\ &= \log \frac{(w + \epsilon_\ell)(1 - w + \epsilon_\ell)}{\epsilon_\ell(1 + \epsilon_\ell)} \\ &= \log\left(1 + \frac{w(1-w)}{\epsilon_\ell(1 + \epsilon_\ell)}\right) > 0, \end{aligned}$$

where the last equality expands the numerator as

$$(w+\epsilon_\ell)(1-w+\epsilon_\ell) = w(1-w) + \epsilon_\ell\big(w+(1-w)\big) + \epsilon_\ell^2 \\ = w(1-w) + \epsilon_\ell(1+\epsilon_\ell).$$

Finally, for two events with fixed marginals a_1, a_2, the joint probability $t = \Pr(Z_1 = Z_2 = 1)$ ranges over the Fréchet interval

$$\max\{0, a_1 + a_2 - 1\} \le t \le \min\{a_1, a_2\},$$

and the shared-error probability rewrites as

$$\Pr(Z_1 = Z_2 = 0) = 1 - \Pr(Z_1 = 1) - \Pr(Z_2 = 1) + \Pr(Z_1 = Z_2 = 1) = 1 - a_1 - a_2 + t.$$

Hence sharing fewer errors at fixed marginals means decreasing t, and the positive slope above makes R strictly decrease. ∎

For categorical experts, a wrong vote has $p_i(y \mid x) = 0$ and contributes $-\log \epsilon_\ell$ to the right-hand side of Eq. 29: the stabilized offset $\epsilon_\ell > 0$ is exactly what makes the bound non-vacuous for one-hot pool members. Without it, a single wrong selected expert would make the weighted average infinite while the fused loss remains finite.

Two quantitative readings follow directly from Eq. 32. First, the slope is maximal at $w = 1/2$: complementarity is most valuable when neither expert dominates the mixture. Second, the slope diverges as $\epsilon_\ell \to 0$ and vanishes as $\epsilon_\ell \to \infty$: the stabilizing offset controls how strongly the fused loss responds to complementarity, because the shared-error case carries loss $-\log \epsilon_\ell$. The analysis is at fixed weights; with learned weights, complementarity additionally shifts the optimal w.

A.8 Exact Optimality of the Threshold Rule

Theorem 1 establishes that the threshold policy maximizes the Lagrangian of the average-budget proxy. The following lemma sharpens this into per-price exact optimality and a computable optimality certificate. Utilities are fixed, measurable, and integrable throughout this subsection; write $V(m) = \mathbb{E}_X[\sum_i m_i(X)U_i(X)]$ and $\bar{C}(m) = \mathbb{E}_X[C_{\text{train}}(m(X))]$, and let $m_\lambda(x) = \mathbf{1}[U_i(x) > \lambda\tilde{c}_i]$ with $V_\lambda = V(m_\lambda)$, $\bar{C}_\lambda = \bar{C}(m_\lambda)$.

Lemma 3 (Everett exactness and optimality certificate (Everett, 1963)) *(a) For every $\lambda \ge 0$, the threshold policy m_λ maximizes the proxy utility among all measurable binary policies whose average training cost does not exceed its own:*

$$\bar{C}(m) \le \bar{C}_\lambda \quad \Longrightarrow \quad V(m) \le V_\lambda. \tag{33}$$

(b) Let $V^\star(B_{\text{train}})$ denote the optimum of the constrained proxy problem in Eq. 12. If m_λ is budget-feasible, $\bar{C}_\lambda \le B_{\text{train}}$, then

$$0 \le V^\star(B_{\text{train}}) - V_\lambda \le \lambda\big(B_{\text{train}} - \bar{C}_\lambda\big). \tag{34}$$

The certificate vanishes iff complementary slackness holds, $\lambda(B_{\text{train}} - \bar{C}_\lambda) = 0$, in which case m_λ attains the global proxy optimum at B_{train}.

Proof. We begin by recalling from the proof of Theorem 1 that m_λ attains the coordinatewise maximum of the Lagrangian at every input: for any measurable binary policy m,

$$\sum_i m_i(x)\big(U_i(x) - \lambda\tilde{c}_i\big) \le \sum_i \max\big\{0, U_i(x) - \lambda\tilde{c}_i\big\} \\ = \sum_i m_{\lambda,i}(x)\big(U_i(x) - \lambda\tilde{c}_i\big).$$

Taking expectations and adding λB_{train} to both sides,

$$\mathcal{J}(m,\lambda) = V(m) + \lambda\big(B_{\text{train}} - \bar{C}(m)\big) \\ \le V_\lambda + \lambda\big(B_{\text{train}} - \bar{C}_\lambda\big) = \mathcal{J}(m_\lambda, \lambda). \tag{35}$$

(a) Rearranging Eq. 35,

$$V(m) \le V_\lambda + \lambda\big(\bar{C}(m) - \bar{C}_\lambda\big) \le V_\lambda \qquad \text{whenever } \bar{C}(m) \le \bar{C}_\lambda,$$

since $\lambda \ge 0$. This is Eq. 33.

(b) Suppose $\bar{C}_\lambda \le B_{\text{train}}$. Then m_λ is feasible for Eq. 12, so

$$V_\lambda \le V^\star(B_{\text{train}}).$$

On the other hand, evaluating the dual function at its attaining policy,

$$d(\lambda) = \mathcal{J}(m_\lambda, \lambda) = V_\lambda + \lambda\big(B_{\text{train}} - \bar{C}_\lambda\big),$$

and weak duality (Theorem 1) gives $V^\star(B_{\text{train}}) \le d(\lambda)$. Subtracting V_λ from both sides,

$$0 \le V^\star(B_{\text{train}}) - V_\lambda \le d(\lambda) - V_\lambda = \lambda\big(B_{\text{train}} - \bar{C}_\lambda\big),$$

which is Eq. 34. Under feasibility both factors of $\lambda(B_{\text{train}} - \bar{C}_\lambda)$ are non-negative, so the certificate vanishes iff $\lambda(B_{\text{train}} - \bar{C}_\lambda) = 0$, and in that case $V_\lambda = V^\star(B_{\text{train}})$. ∎

The budget B_{train} cancels in part (a): each price λ solves the constrained proxy problem *exactly* for its own realized budget $\bar{C}_\lambda$. As λ varies, the threshold rule traces the Pareto frontier of achievable (utility, average cost) pairs with no duality gap at the traced points; the monotonicity of $\bar{C}_\lambda$ in λ is Proposition 3. Part (b) converts the budget slack into an explicit upper bound on the proxy suboptimality, computable from observed quantities. Combined with the stationary-point conditions of Eq. 18's clipping analysis, an interior stationary epoch ($\lambda > 0$, zero average violation) is exactly the case in which this certificate is zero for the frozen-utility proxy.

A.9 Attainable Risk and the Budget Frontier

This subsection expands the reference quantities used by Theorem 3 for the attainable performance of the selection-plus-fusion architecture. Predictor outputs are fixed; expectations are over data and, where applicable, expert-response randomness.

Lemma 4 (Optimal fusion weights) *Fix a nonempty selected set S. (a) For fixed x and y,*

$$\min_{\alpha \in \Delta(S)} \Big[-\log\Big(\sum_{i \in S} \alpha_i p_i(y \mid x) \Big) \Big] = -\log \max_{i \in S} p_i(y \mid x), \tag{36}$$

attained at any vertex $e_{i^\star}$ with $i^\star \in \arg\max_{i \in S} p_i(y \mid x)$. The same holds for the stabilized loss, with value $-\log\big(\max_{i \in S} p_i(y \mid x) + \epsilon_\ell\big)$. Consequently, given a selection policy with $S(X)$ nonempty almost surely, the attainable fused risk is

$$R_S^\star = \mathbb{E}\left[-\log \max_{i \in S(X)} p_i(Y \mid X) \right]. \tag{37}$$

(b) Fix the information variable $z = z(x)$. Any minimizer $\alpha^\star(z) \in \Delta(S)$ of the conditional risk $\mathbb{E}[-\log \sum_{i \in S} \alpha_i p_i(Y \mid X) \mid z]$ satisfies the KKT condition: for some $\mu(z)$,

$$\mathbb{E}\left[\frac{p_i(Y \mid X)}{\sum_{j \in S} \alpha_j^\star p_j(Y \mid X)} \,\middle|\, z \right] \begin{cases} = 1, & \alpha_i^\star > 0, \\ \le 1, & \alpha_i^\star = 0. \end{cases} \tag{38}$$

In particular, a vertex e_k is conditionally optimal iff $\mathbb{E}[p_i(Y \mid X)/p_k(Y \mid X) \mid z] \le 1$ for all $i \in S$.

Proof. (a) Fix x and y and abbreviate $p_i = p_i(y \mid x)$. For any $\alpha \in \Delta(S)$,

$$\sum_{i \in S} \alpha_i p_i \le \sum_{i \in S} \alpha_i \max_{j \in S} p_j = \max_{j \in S} p_j,$$

with equality at any vertex $e_{i^\star}$ with $i^\star \in \arg\max_{i \in S} p_i$: the mixture is linear in α, so its maximum over the simplex is attained at a vertex. Since $-\log$ is strictly decreasing,

$$\begin{aligned} \min_{\alpha \in \Delta(S)} \Big[-\log\Big(\sum_{i \in S} \alpha_i p_i \Big) \Big] &= -\log \max_{\alpha \in \Delta(S)} \sum_{i \in S} \alpha_i p_i \\ &= -\log \max_{i \in S} p_i, \end{aligned}$$

which is Eq. 36. The stabilized loss follows the same chain with an added offset,

$$\min_{\alpha\in\Delta(S)}\Big[-\log\Big(\sum_{i\in S}\alpha_i p_i+\epsilon_\ell\Big)\Big]=-\log\Big(\max_{i\in S}p_i+\epsilon_\ell\Big).$$

Taking expectations over (X,Y), with $S(X)$ nonempty almost surely, gives Eq. 37.

(b) Write $F(\alpha)=\mathbb{E}[-\log\sum_{i\in S}\alpha_i p_i(Y\mid X)\mid z]$. Interchanging the derivative with the conditional expectation,

$$\frac{\partial F}{\partial\alpha_i}=-\mathbb{E}\left[\frac{p_i(Y\mid X)}{\sum_{j\in S}\alpha_j p_j(Y\mid X)}\,\middle|\,z\right]=:-G_i(\alpha).$$

F is convex in α, as a conditional expectation of the convex functions $\alpha\mapsto-\log(\alpha\cdot p)$, and the simplex constraints are linear; hence $\alpha^\star$ is a minimizer iff the KKT conditions hold: for some multiplier $\mu(z)$,

$$G_i(\alpha^\star)\begin{cases}=-\mu(z), & \alpha_i^\star>0,\\ \le-\mu(z), & \alpha_i^\star=0.\end{cases}\tag{39}$$

Multiplying by $\alpha_i^\star$ and summing over $i\in S$ identifies the multiplier:

$$-\mu(z)=\sum_{i\in S}\alpha_i^\star G_i(\alpha^\star)=\mathbb{E}\left[\frac{\sum_{i\in S}\alpha_i^\star p_i(Y\mid X)}{\sum_{j\in S}\alpha_j^\star p_j(Y\mid X)}\,\middle|\,z\right]=1.$$

Substituting $-\mu(z)=1$ into Eq. 39 gives Eq. 38. For a vertex $\alpha^\star=e_k$, the on-support condition holds trivially since the ratio at $i=k$ equals 1, and the off-support condition reads

$$\mathbb{E}\left[\frac{p_i(Y\mid X)}{p_k(Y\mid X)}\,\middle|\,z\right]\le 1\qquad\text{for all } i\in S,$$

which is the stated vertex criterion. ∎

Equation 37 evaluates both components of the architecture with one quantity, the largest true-class probability available in the selected set: selection decides whether a good expert is present, and fusion decides whether the mixture places its weight there. Theorem 3 in the main text splits the realized system risk accordingly: into the irreducible pool limit, the loss attributable to not selecting a best expert, and the loss attributable to not concentrating the mixture on the best selected expert. Under an average budget the selection gap generally cannot vanish; the frontier below parametrizes this trade-off.

The fusion gap is minimized by the conditional-optimal weight rule of part (b), which the fusion network approximates from its inputs. In the architecture of Section 3.3, the fusion weights are functions of the image representation, predictor identities, selection probabilities, and costs; the predictor outputs $p_i(\cdot\mid x)$ enter only the final aggregation, and part (b) makes the fusion target explicit. The gradient signal of Theorem 2 drives the system toward this condition: its direct term compares $p_i(y\mid x)$ with the current mixture q_y, and Eq. 38 equalizes exactly the posterior-weighted ratios p_i/q on the support of the optimal mixture. Proposition 2 shows that reaching this target can strictly require the image information z beyond the scores.

Lemma 5 (Budget frontier and the price as its shadow slope) *For randomized selection policies, let*

$$R^\star(B)=\inf_{m:\,\bar C(m)\le B}R_S^\star(m),\qquad B\ge 0.\tag{40}$$

Then $R^\star$ is non-increasing and convex in B. If strong duality holds at B with multiplier $\lambda^\star\ge 0$, i.e., $R^\star(B)=\inf_m\{R_S^\star(m)+\lambda^\star(\bar C(m)-B)\}$, then $-\lambda^\star$ is a subgradient of $R^\star$ at B:

$$R^\star(B')\ge R^\star(B)-\lambda^\star(B'-B)\qquad\textit{for all } B'\ge 0.\tag{41}$$

Proof. For monotonicity, observe that a larger budget enlarges the feasible set:

$$B\le B'\quad\Longrightarrow\quad\{m:\bar C(m)\le B\}\subseteq\{m:\bar C(m)\le B'\},$$

so the infimum in Eq. 40 satisfies $R^\star(B)\ge R^\star(B')$.

For convexity, fix $B_1, B_2 \geq 0$ and $\rho \in [0,1]$, and take policies m_1, m_2 feasible at B_1, B_2 with attainable risks R_1, R_2. Let m_ρ randomize between m_1 and m_2 with weights ρ and $1-\rho$, independently of the data. Its average cost and risk are

$$\begin{aligned}\bar{C}(m_\rho) &= \rho\,\bar{C}(m_1) + (1-\rho)\,\bar{C}(m_2) \leq \rho B_1 + (1-\rho)B_2,\\ R_S^\star(m_\rho) &= \rho R_1 + (1-\rho)R_2,\end{aligned}$$

so m_ρ is feasible at $\rho B_1 + (1-\rho)B_2$ and certifies

$$R^\star\big(\rho B_1 + (1-\rho)B_2\big) \leq \rho R_1 + (1-\rho)R_2.$$

Taking the infimum over m_1 and m_2 gives convexity.

For the subgradient, fix any policy m feasible at B', so that $\bar{C}(m) \leq B'$. Adding and subtracting the same multiplier term,

$$\begin{aligned}R_S^\star(m) &= R_S^\star(m) + \lambda^\star\big(\bar{C}(m) - B\big) - \lambda^\star\big(\bar{C}(m) - B\big)\\ &\geq R_S^\star(m) + \lambda^\star\big(\bar{C}(m) - B\big) - \lambda^\star(B' - B)\\ &\geq \inf_{\tilde{m}} \Big\{ R_S^\star(\tilde{m}) + \lambda^\star\big(\bar{C}(\tilde{m}) - B\big)\Big\} - \lambda^\star(B' - B)\\ &= R^\star(B) - \lambda^\star(B' - B),\end{aligned}$$

where the first inequality uses $\lambda^\star \geq 0$ together with $\bar{C}(m) \leq B'$, the second relaxes m to the unconstrained infimum, and the last equality is strong duality at B. Taking the infimum over all policies feasible at B' yields Eq. 41. ∎

Under strong duality, the optimal multiplier is thus the local slope of the attainable risk-cost frontier: the marginal risk reduction per unit of additional budget. The epoch-level price feedback of Eq. 8 plays the same structural role inside the utility proxy: Lemma 3 shows that each realized price marks an exactly optimal point of the proxy frontier and bounds the remaining proxy gap by the price times the budget slack.

A.10 Stability of Selection under Utility Perturbations

The nesting result (Proposition 3) varies the price at frozen utilities. The following lemma varies the utilities at a frozen price, covering epoch-to-epoch utility updates, numerical error, or input perturbations. Fix λ, costs, and the input distribution; let $U_i, \widehat{U}_i$ be two deterministic utility functions with $\sup_x |\widehat{U}_i(x) - U_i(x)| \leq \delta_i$, and write $s_i = U_i - \lambda\tilde{c}_i$, $m_i = \mathbf{1}[s_i > 0]$, $\widehat{m}_i = \mathbf{1}[\widehat{U}_i - \lambda\tilde{c}_i > 0]$.

Lemma 6 (Margin identity and perturbation certificate) *(a) A gate changes only near the threshold:*

$$m_i(x) \neq \widehat{m}_i(x) \quad \Longrightarrow \quad |s_i(x)| \leq \delta_i. \tag{42}$$

Moreover, the fixed-price Lagrangian proxy gap has the exact form

$$\mathcal{J}(m,\lambda) - \mathcal{J}(\widehat{m},\lambda) = \mathbb{E}\left[\sum_i |s_i(X)|\,\mathbf{1}[m_i \neq \widehat{m}_i]\right] \leq \sum_i \delta_i \Pr\big(|s_i(X)| \leq \delta_i\big), \tag{43}$$

and the average costs satisfy

$$\begin{aligned}\big|\bar{C}(m) - \bar{C}(\widehat{m})\big| &\leq \sum_i \tilde{c}_i\,\Pr\big(|s_i(X)| \leq \delta_i\big),\\ \big|\mathbb{E}C_{\mathrm{report}}(m) - \mathbb{E}C_{\mathrm{report}}(\widehat{m})\big| &\leq \sum_i c_i\,\Pr\big(|s_i(X)| \leq \delta_i\big).\end{aligned} \tag{44}$$

If threshold neighbourhoods have bounded mass, $\Pr(|s_i(X)| \leq r) \leq A_i r^\kappa$ *for some* $A_i, \kappa > 0$*, then*

$$\mathcal{J}(m,\lambda) - \mathcal{J}(\widehat{m},\lambda) \leq \sum_i A_i\,\delta_i^{\kappa+1}, \qquad \big|\bar{C}(m) - \bar{C}(\widehat{m})\big| \leq \sum_i \tilde{c}_i A_i\,\delta_i^\kappa. \tag{45}$$

(b) If the perturbed policy is budget-feasible, $\bar{C}(\widehat{m}) \leq B_{\mathrm{train}}$*, then relative to the reference utilities,*

$$V^\star(B_{\mathrm{train}}) - V(\widehat{m}) \leq \sum_i \delta_i \Pr\big(|s_i(X)| \leq \delta_i\big) + \lambda\big(B_{\mathrm{train}} - \bar{C}(\widehat{m})\big). \tag{46}$$

Proof. (a) Write $\widehat{s}_i(x) = s_i(x) + \Delta_i(x)$ with $\Delta_i(x) = \widehat{U}_i(x) - U_i(x)$, so that $|\Delta_i(x)| \le \delta_i$. If the gate changes, the two scores lie on opposite sides of zero, and the score must sit within one perturbation radius of the threshold:

$$\begin{aligned} m_i(x) \neq \widehat{m}_i(x) &\implies \big(s_i(x) > 0 \ge \widehat{s}_i(x)\big) \text{ or } \big(s_i(x) \le 0 < \widehat{s}_i(x)\big) \\ &\implies |s_i(x)| \le |s_i(x) - \widehat{s}_i(x)| = |\Delta_i(x)| \le \delta_i, \end{aligned}$$

which is Eq. 42. Next, expanding the Lagrangian difference,

$$\begin{aligned} \mathcal{J}(m,\lambda) - \mathcal{J}(\widehat{m},\lambda) &= \mathbb{E}\left[\sum_i \big(m_i(X) - \widehat{m}_i(X)\big)\big(U_i(X) - \lambda\tilde{c}_i\big)\right] \\ &= \mathbb{E}\left[\sum_i \big(m_i(X) - \widehat{m}_i(X)\big)s_i(X)\right]. \end{aligned}$$

Coordinatewise, the product vanishes off the disagreement event, and on that event the sign of s_i agrees with the gate difference:

$$\begin{aligned} \big(m_i - \widehat{m}_i\big)s_i &= \begin{cases} s_i, & m_i = 1,\ \widehat{m}_i = 0 \ \ (\text{so } s_i > 0), \\ -s_i, & m_i = 0,\ \widehat{m}_i = 1 \ \ (\text{so } s_i \le 0), \end{cases} \\ &= |s_i|\,\mathbf{1}[m_i \neq \widehat{m}_i]. \end{aligned}$$

Substituting this identity and bounding $|s_i|$ by δ_i on the disagreement event via Eq. 42,

$$\begin{aligned} \mathcal{J}(m,\lambda) - \mathcal{J}(\widehat{m},\lambda) &= \mathbb{E}\left[\sum_i |s_i(X)|\,\mathbf{1}[m_i \neq \widehat{m}_i]\right] \\ &\le \sum_i \delta_i \,\Pr\big(|s_i(X)| \le \delta_i\big), \end{aligned}$$

which is Eq. 43. The same event decomposition controls the average training cost:

$$\begin{aligned} \big|\bar{C}(m) - \bar{C}(\widehat{m})\big| &= \left|\mathbb{E}\left[\sum_i \tilde{c}_i\big(m_i(X) - \widehat{m}_i(X)\big)\right]\right| \\ &\le \sum_i \tilde{c}_i \,\Pr\big(m_i \neq \widehat{m}_i\big) \\ &\le \sum_i \tilde{c}_i \,\Pr\big(|s_i(X)| \le \delta_i\big), \end{aligned}$$

and replacing $\tilde{c}_i$ by c_i line by line gives the reporting-cost bound, proving Eq. 44. Under the bounded-mass assumption $\Pr(|s_i(X)| \le r) \le A_i r^{\kappa}$, substituting $r = \delta_i$,

$$\begin{aligned} \mathcal{J}(m,\lambda) - \mathcal{J}(\widehat{m},\lambda) &\le \sum_i \delta_i \left(A_i \delta_i^{\kappa}\right) = \sum_i A_i\, \delta_i^{\kappa+1}, \\ \big|\bar{C}(m) - \bar{C}(\widehat{m})\big| &\le \sum_i \tilde{c}_i \left(A_i \delta_i^{\kappa}\right) = \sum_i \tilde{c}_i A_i\, \delta_i^{\kappa}, \end{aligned}$$

which is Eq. 45.

(b) Since m is the threshold policy at price λ under the reference utilities, weak duality (Theorem 1) gives

$$V^{\star}(B_{\text{train}}) \le d(\lambda) = \mathcal{J}(m,\lambda).$$

Expanding the perturbed utility through the Lagrangian,

$$V(\widehat{m}) = \mathcal{J}(\widehat{m},\lambda) - \lambda\big(B_{\text{train}} - \bar{C}(\widehat{m})\big).$$

Subtracting the two displays and applying Eq. 43,

$$\begin{aligned} V^{\star}(B_{\text{train}}) - V(\widehat{m}) &\le \mathcal{J}(m,\lambda) - \mathcal{J}(\widehat{m},\lambda) + \lambda\big(B_{\text{train}} - \bar{C}(\widehat{m})\big) \\ &\le \sum_i \delta_i \Pr\big(|s_i(X)| \le \delta_i\big) + \lambda\big(B_{\text{train}} - \bar{C}(\widehat{m})\big), \end{aligned}$$

which is Eq. 46. ∎

Two readings of part (a): gates with scores far from the threshold never move, and gates that do move contribute at most their small score margin to the proxy gap. No independence assumptions on the gates are used, and the perturbation radii δ_i may be heterogeneous. The reference utilities need not be a verified "true" utility: any two deterministic utility maps may be compared, e.g., consecutive training epochs. Part (b) splits the proxy suboptimality of a perturbed feasible policy into a score-perturbation term, controlled by the threshold-neighbourhood mass, and a budget-slack term, exactly the certificate of Lemma 3 evaluated at the perturbed policy.

A.11 Sample-Average Price Updates and Soft Usage

The telescoping identity of Eq. 18 is algebraic; the following statement lifts it to population quantities when each epoch's cost is a sample average.

Lemma 7 (Concentration of the integrated price deviation) *Suppose all training costs lie in $[0, \tilde{c}_{\max}]$, epoch r averages n_r i.i.d. samples, and no price update is clipped. Write $\mu_r = \mathbb{E}[\widehat{C}_{\mathrm{train},r} \mid \mathcal{F}_r]$ for the epoch's population average cost given the history $\mathcal{F}_r$ (which includes λ_r and the current parameters), and $v_r = \widehat{C}_{\mathrm{train},r} - \mu_r$. Then v_r is a martingale difference, and with probability at least $1-\delta$,*

$$\left| \lambda_t - \lambda_0 - \eta \sum_{r=0}^{t-1} (\mu_r - B_{\mathrm{train}}) \right| \le \eta\, \tilde{c}_{\max} \sqrt{2t \log(2/\delta)}. \tag{47}$$

Proof. Substituting $\widehat{C}_{\mathrm{train},r} = \mu_r + v_r$ into the telescoping identity of Eq. 18,

$$\begin{aligned} \lambda_t &= \lambda_0 + \eta \sum_{r=0}^{t-1} \big(\widehat{C}_{\mathrm{train},r} - B_{\mathrm{train}}\big) \\ &= \lambda_0 + \eta \sum_{r=0}^{t-1} (\mu_r - B_{\mathrm{train}}) + \eta \sum_{r=0}^{t-1} v_r. \end{aligned}$$

Given the history $\mathcal{F}_r$, the epoch cost is an average of n_r independent variables taking values in $[0, \tilde{c}_{\max}]$, so its deviation satisfies

$$\mathbb{E}[v_r \mid \mathcal{F}_r] = \mathbb{E}[\widehat{C}_{\mathrm{train},r} \mid \mathcal{F}_r] - \mu_r = 0, \qquad |v_r| \le \tilde{c}_{\max},$$

i.e., (v_r) is a bounded martingale difference sequence. Azuma's inequality applied to the martingale $M_t = \sum_{r=0}^{t-1} v_r$ gives, for any $u > 0$,

$$\Pr\left(\left| \sum_{r=0}^{t-1} v_r \right| \ge u \right) \le 2 \exp\left(-\frac{u^2}{2t\, \tilde{c}_{\max}^2} \right).$$

Setting the right-hand side equal to δ and solving for u,

$$2 \exp\left(-\frac{u^2}{2t\, \tilde{c}_{\max}^2} \right) = \delta \quad \Longrightarrow \quad u = \tilde{c}_{\max} \sqrt{2t \log(2/\delta)},$$

and multiplying the deviation by η yields Eq. 47. ∎

Thus, away from the clipping boundaries, the realized price equals the integral of population budget deviations plus noise of order $O(\eta\sqrt{t})$. Treating epoch means through Hoeffding's inequality (Hoeffding, 1963) per epoch refines the bound by replacing $t\, \tilde{c}_{\max}^2$ with $\tilde{c}_{\max}^2 \sum_r 1/n_r$.

Soft usage is likewise monotone. Proposition 3 covers the forward (hard) quantities. The soft selection probabilities that carry the differentiable cost path are monotone as well: with the encoder and utility network frozen, $\pi_i(\lambda) = \sigma\big((U_i(x) - \lambda \tilde{c}_i - \tau)/T_s\big)$ is non-increasing in λ for every input and predictor, hence both

$$\mathbb{E}\left[\sum_i \pi_i(X) \tilde{c}_i \right] \quad \text{and} \quad \mathbb{E}\left[\sum_i \pi_i(X)\, m_i(X)\, \tilde{c}_i \right] \tag{48}$$

are non-increasing in λ, since each summand is a product of non-negative non-increasing factors. The same holds with reporting costs c_i in place of $\tilde{c}_i$.

Algorithm 1 Joint selection, fusion, and budget-price feedback

Require: Frozen predictors, training costs $\tilde{c}$, budget B_{train}, parameters (θ, ψ), initial price λ_0

for each epoch t **do**
  Fix $\lambda \leftarrow \lambda_t$; initialize cost sum and sample count
  for each mini-batch $\mathcal{B}$ **do**
    Encode inputs; compute utilities, scores, and ST masks
    Obtain predictor outputs; compute fusion weights and q
    Update (θ, ψ) using Eq. 7 with λ fixed
    Accumulate forward mask costs and sample count
  end for
  Compute $\widehat{C}_{\text{train},t}$; update λ by Eq. 8
end for

A.12 Training Procedure

Algorithm 1 summarises the joint training of the selector and the fusion network together with the budget-price feedback of Section 3.4. The procedure has two nested levels. In the inner loop the price is held fixed: each mini-batch performs one primal step on (θ, ψ) with the Lagrangian of Eq. 7, where the straight-through mask of Eq. 4 supplies the derivative of the discrete selection, and the realised forward-mask costs are accumulated over the epoch. In the outer step the sample-weighted average training cost $\widehat{C}_{\text{train},t}$ is formed from these accumulations and the price is updated by Eq. 8, which raises λ when the average cost exceeds the budget and lowers it otherwise. The two levels therefore implement the primal–dual scheme analysed in Appendix A.3 and Appendix A.5: the primal step reduces the fused loss at the current price, and the dual step moves the price along the direction that activates or relaxes the average-cost constraint. Because the price enters the selection score of Eq. 3, a single scalar couples all input-dependent decisions to the budget.

B Experimental Details

B.1 Additional Details

We detail the configuration of our empirical evaluation, dataset by dataset. Across the three benchmarks the selector and the fusion network share the same architecture. The selector projects the shared feature into a query embedding, concatenates it with a learnable expert embedding and the normalised expert cost, and scores the resulting pairs with a two-layer MLP; the fusion network builds one token per expert from a learnable embedding plus masked projections of the selection probability and the normalised cost, prepends an image token, and encodes the sequence with a two-layer Transformer encoder (model width 128, 4 attention heads, dropout 0.1). Only the feature extractor and the expert pool vary across datasets, as described below.

B.1.1 Baselines

We compare against three baselines. The first is Montreuil et al. (2026), which is to our knowledge the first framework to introduce adaptive top-k deferral in learning-to-defer and which we reproduce in the two-stage setting. The second is CET (Hemmer et al., 2022), a complementary expert-team method that likewise combines the experts' predictions with the machine probabilities in a weighted fusion. The third is the two-stage post-hoc estimator PostHoc (Narasimhan et al., 2022), which learns the deferral rule on top of a frozen predictor and

B.1.2 CIFAR-10

Dataset. CIFAR-10 (Krizhevsky, 2009) contains $60{,}000$ colour images of resolution 32×32 evenly distributed over 10 classes ($6{,}000$ images per class). We follow the standard split, dividing the training set into $40{,}000$ training and $10{,}000$ validation images and keeping the $10{,}000$ test images for evaluation, and apply dataset-specific normalisation.

Table 1: Experts on CIFAR-10: accuracy on the test set and consultation cost. ($N = 4$).

Expert	1	2	3	4
Backbone	ResNet-18	ResNet-34	ResNet-50	DenseNet-161
Accuracy (%)	70.83	81.00	82.46	86.07
Cost	1.0	1.6	2.0	3.4

Table 2: Experts on CIFAR-100: accuracy on the test set and consultation cost ($N = 6$).

Expert	1	2	3	4	5	6
Accuracy (%)	46.25	51.68	52.28	51.44	51.37	49.78
Cost	0.0	0.045	0.05	0.04	0.035	0.03

Models. This benchmark uses a configuration that is *not* inherited from the reference work: the shared feature extractor is a ResNet-18 trunk (He et al., 2016) producing a 512-dimensional representation, which feeds the shared selector and the two-layer Transformer encoder described at the beginning of this appendix.

Experts. We build a pool of four experts with heterogeneous backbones and overlapping competence. Each is a ResNet-18, ResNet-34, ResNet-50 or DenseNet-161 (He et al., 2016; Huang et al., 2017) adapted to 32×32 inputs and is assigned a subset of the ten classes on which it is reliable: $\{$truck, automobile, ship$\}$, $\{$bird, horse, deer$\}$, $\{$cat, dog, bird, truck$\}$ and $\{$frog, ship, airplane, cat, horse$\}$. To make an expert unreliable outside its assigned classes, the label of every non-assigned training sample is corrupted with probability 0.8 into a random incorrect class. The experts are trained with Adam (Kingma & Ba, 2015) for at most 100 epochs with early stopping and are kept frozen. Table 1 reports their accuracy on the test set together with their consultation costs.

B.1.3 CIFAR-100

Dataset. CIFAR-100 (Krizhevsky, 2009) consists of $60{,}000$ colour images of resolution 32×32 from 100 classes (600 images per class). We use the standard training/test split and hold out 100 training images per class for validation, which leaves 400, 100 and 100 images per class for training, validation and testing respectively; the same normalisation protocol as for CIFAR-10 is applied.

Models. We adopt the model setting of Montreuil et al. (2026): the shared feature extractor is their rejector backbone, a ResNet-4 (He et al., 2016), followed by the shared selector and the two-layer Transformer fusion network described at the beginning of this appendix. As in the original work, the adaptive Top-$k(x)$ baseline builds its cardinality function on a CLIP-based image encoder (Radford et al., 2021) with a lightweight classification head.

Experts. Following the protocol of Montreuil et al. (2026), we build a pool of six entities with strongly overlapping areas of competence. The first entity is a predictor trained on the full training set with a ResNet-4 backbone (He et al., 2016) for 50 epochs with batch size 256 and the Adam optimiser (Kingma & Ba, 2015), keeping the checkpoint with the lowest validation loss. The remaining five experts are synthesised rather than trained: each is assigned a subset of roughly 55 of the 100 classes on which it predicts correctly with probability $p = 0.94$, and is uniform over the remaining classes, so that the assigned subsets overlap heavily. Table 2 reports the resulting accuracy and cost. The pool is frozen throughout training.

B.1.4 SVHN

Dataset. The Street View House Numbers dataset (Goodfellow et al., 2013) is a large-scale digit classification benchmark with more than $600{,}000$ RGB images of resolution 32×32 cropped from real-world street scenes. We use the standard split of $58{,}606$ training, $14{,}651$ validation and $26{,}032$ test images.

Table 3: Experts on SVHN: accuracy on the test set and consultation cost ($N = 6$).

Expert	1	2	3	4	5	6
Accuracy (%)	69.26	54.84	62.12	64.79	51.76	62.77
Cost	0.0	0.045	0.05	0.04	0.035	0.03

Table 4: Trainable parameter counts of each method's own network on the three benchmarks. The shared frozen predictor pool is excluded.

Method	CIFAR-100	SVHN	CIFAR-10
Top-k (Montreuil et al., 2026)	78,118	78,118	11,170,884
Top-$k(x)$ (Montreuil et al., 2026)†	2,449,196	2,449,196	13,540,936
CET (Hemmer et al., 2022)	80,006	80,006	11,301,188
PostHoc (Narasimhan et al., 2022)	84,217	78,367	78,302
CATABC (ours)	488,354	488,354	11,847,746

†Top-$k(x)$ additionally uses a frozen CLIP ViT-B/32 image encoder (Radford et al., 2021) with 87,848,448 parameters that do not participate in training. Its tabulated trainable parameters comprise the rejector and the cardinality head.

Models. We again follow the model setting of Montreuil et al. (2026), using their rejector backbone, a ResNet-4 (He et al., 2016), as the shared feature extractor, together with the shared selector and fusion network; the Top-$k(x)$ baseline again uses the CLIP-based cardinality function of the original implementation (Radford et al., 2021).

Experts. Following Montreuil et al. (2026), we construct a pool of six experts, each a ResNet-4 classifier (He et al., 2016) trained with the Adam optimiser (Kingma & Ba, 2015) on a randomly sampled, partially overlapping 20% subset of the SVHN training set. This design mimics a realistic scenario in which experts are trained on distinct data partitions because of privacy or ownership constraints, so that each expert is reliable only on part of the input space. Table 3 reports their accuracy and cost; the pool is frozen as above.

B.2 Network Parameter Counts

Table 4 reports the trainable parameters of each method's own network on every benchmark, counted from the implementations used in our experiments. All columns follow this common trainable-parameter accounting; the additional frozen CLIP encoder used by Top-$k(x)$ is disclosed in the table note. The shared frozen predictor pool is excluded. For CATABC the count comprises the feature extractor, the cost-aware selector and the fusion network (77,728, 3,585 and 407,041 parameters on CIFAR-100 and SVHN; 11,168,832, 214,785 and 464,129 on CIFAR-10); CATABC-Top$k$ uses the same network and has the same parameter count. CET combines a feature extractor with an MLP selector (77,728 and 2,278 parameters on CIFAR-100 and SVHN; 11,168,832 and 132,356 on CIFAR-10). Top-k trains a single rejector (ResNet-4 on CIFAR-100 and SVHN, ResNet-18 on CIFAR-10); Top-$k(x)$ keeps that rejector and adds a frozen CLIP ViT-B/32 image encoder together with a four-block residual-MLP cardinality head. PostHoc trains a ResNet-4 backbone with a joint classification and deferral head.

B.3 Training details

B.3.1 Our method

All our models are trained with the Adam optimiser (Kingma & Ba, 2015) and a cosine annealing schedule, with a batch size of 128 on every dataset. The learning rate and the weight decay are chosen by a grid search. Because each dataset comprises several budgets, the search is carried out on a single representative budget and the resulting values are reused for all the other budgets. On CIFAR-10 we search on budget 4 and select a learning rate of 0.001 with a weight decay of 5×10^{-4}; on CIFAR-100 we also search on budget 0.14 and select a learning rate of 0.004 with no

weight decay; on SVHN we search on budget 0.13 and select a learning rate of 0.001 with a weight decay of 5×10^{-4}.

The optimiser is given three parameter groups. The feature extractor is initialized from a pre-trained backbone and fine-tuned at one tenth of the learning rate selected above ($0.1\times$), so that the pre-trained representation is adapted to the expert pool and the deferral task at a smaller step size rather than held fixed or re-learned from scratch; the selector and the fusion network are trained from scratch at the full learning rate. Every run starts with a linear warm-up over the first five epochs and then follows a cosine decay, and the same weight decay is applied to all three groups.

The budget constraint $\mathbb{E}[C(x)] \leq B$, where $C(x)$ is the consultation cost of an input, is handled by a Lagrangian multiplier λ that is not optimised by the gradient descent but acts as an external dual variable. Each epoch we minimise the objective $\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda\,(\bar{C} - B)$, in which $\mathcal{L}_{\text{cls}}$ is the cross-entropy of the fused prediction and $\bar{C}$ is the average cost realised over the epoch, and we then update the multiplier on the realised violation with the projected rule $\lambda \leftarrow \max\{0, \lambda + \eta\,(\bar{C} - B)\}$, where $\eta = 0.01$ is the learning rate of the dual variable. The multiplier is initialised at 0.1 and clamped to $[0, 50]$, so that it increases while the budget is exceeded and decreases once the constraint is satisfied.

The costs are rescaled during training on the datasets where they are small. On CIFAR-100 and SVHN we follow the cost setting of Montreuil et al. (2026), whose costs are much smaller than those of CIFAR-10, and therefore multiply both the predictor costs and the budget by 100 during training, so that the cross-entropy loss and the Lagrangian constraint remain on the same order of magnitude. On CIFAR-100, for instance, each predictor is assigned one hundred times the cost reported in Table 2 and the budget is enlarged by the same factor. The test split is nevertheless evaluated with the original costs, so all the numbers reported in the main text and in Appendix B.4 are expressed in the original units. CATABC-Topk and the ablation variants are trained and tested with the hyperparameters of the corresponding configuration in the main experiments.

B.3.2 TOP-K AND TOP-K(X) (MONTREUIL ET AL., 2026)

Both methods follow the two-stage procedure of the reference implementation. In the first stage, Top-k trains a rejector with the Adam optimiser (Kingma & Ba, 2015) and a cosine annealing schedule. On CIFAR-100 the rejector is a ResNet-4 trained for 200 epochs with a batch size of 2048, a learning rate of 0.01 and no weight decay; on CIFAR-10 the rejector is a CIFAR-style ResNet-18 trained with the same optimiser, schedule, batch size, learning rate and number of epochs; on SVHN the rejector is a ResNet-4 trained for 50 epochs with a batch size of 256 and a learning rate of 0.01. The checkpoint achieving the lowest deferral loss on the validation set is selected, and at test time Top-k defers to the k predictors with the highest rejector scores.

In the second stage, Top-$k(x)$ freezes the first-stage rejector and trains a four-block residual-MLP cardinality head on features of a frozen CLIP ViT-B/32 image encoder (Radford et al., 2021), using the Adam optimiser and a cosine annealing schedule. On CIFAR-100 and CIFAR-10 the head is trained for 15 epochs with a batch size of 128, a learning rate of 0.001 and a weight decay of 10^{-4}; on SVHN it is trained for 10 epochs with a batch size of 256, a learning rate of 0.001 and a weight decay of 10^{-5}. For every dataset we sweep a grid of the cardinality penalty Λ together with the decision rule (weighted-voting), and the configuration selected on the validation set at each budget is evaluated on the test split. The grid reaches $\Lambda = 0.3$ on CIFAR-100 and extends down to $\Lambda = 10^{-9}$ on CIFAR-10 and SVHN, so that the low-budget regime, in which the optimal cardinality is small, is covered as well.

B.3.3 CET (HEMMER ET AL., 2022)

CET is trained with the Adam optimiser (Kingma & Ba, 2015) and a cosine annealing schedule, with a batch size of 128 on every dataset, a learning rate of 0.001 and a weight decay of 5×10^{-4}. The checkpoint achieving the lowest loss on the validation set is selected for testing.

B.3.4 POSTHOC (NARASIMHAN ET AL., 2022)

PostHoc is evaluated with a two-stage procedure built on a frozen feature extractor. The extractor has a ResNet-4 backbone producing 64-dimensional features and is trained with the Adam optimiser (Kingma & Ba, 2015) and a cosine annealing schedule for 60 epochs, with a batch size of 256, a

learning rate of 0.05, a weight decay of 10^{-4} and gradient clipping at norm 5; the checkpoint with the highest validation accuracy of the base predictor is kept. The first stage caches the predictions of the predictors that are free of charge; since no predictor is free on CIFAR-10, the least expensive predictor serves as the base predictor on that dataset. In the second stage, a linear head with sigmoid outputs estimates the error probability of each predictor from the frozen features and is trained with binary cross-entropy, using the Adam optimiser with a batch size of 1024, a learning rate of 0.05 and a weight decay of 10^{-4} for 200 epochs. An input is deferred to the predictor k that maximises the gain $\hat{\varepsilon}_{\text{base}}(x) - \hat{\varepsilon}_k(x) - t\, c_k$ over the base predictor whenever this gain is positive, where $\hat{\varepsilon}_k(x)$ is the estimated error probability of predictor k and c_k its cost; the trade-off parameter t is selected on the validation set from the grid $\{0, 0.25, 0.5, 1, 2, 4, 8, 16, 32\}$ for each budget.

Table 5: Comparison results with the baseline on CIFAR-10.

Budget	Metric	Method					
		CATABC (Ours)	**CATABC-Topk (Ours)**	**Top-K**	**Top-K(x)**	**CET**	**PostHoc**
2.0	ACC	**90.84±0.11**	86.38±0.07	72.27±1.24	80.44±0.58	90.68±0.13	87.20±0.27
	Cost	1.83±0.10	2.00±0.02	1.05±0.04	2.01±0.09	2.49±0.04	2.04±0.09
3.0	ACC	**91.59±0.09**	84.96±0.74	72.27±1.24	82.16±0.33	/	/
	Cost	2.73±0.12	2.99±0.39	2.71±0.17	2.46±0.10	/	/
4.0	ACC	**91.69±0.13**	/	/	84.76±0.34	/	/
	Cost	2.81±0.08	/	/	3.64±0.15	/	/
5.0	ACC	**91.58±0.07**	87.71±0.35	80.51±5.59	86.23±0.30	/	/
	Cost	2.85±0.16	4.99±0.03	4.60±0.01	4.71±0.33	/	/
8.0	ACC	**91.31±0.03**	89.77±0.35	82.79±6.90	87.03±0.02	/	/
	Cost	7.45±0.60	8.00±0.00	8.00±0.00	6.98±0.34	/	/

B.4 Additional Results

B.4.1 Additional main results

Table 5, 6, and 7 present the detailed experimental results of our proposed method compared with the baselines on CIFAR-10, CIFAR-100, and SVHN (mean and standard deviation over four runs). For the CET baseline (Hemmer et al., 2022), the original method lacks explicit parameters to control costs, resulting in only a single reported value across each dataset. In contrast, PostHoc (Narasimhan et al., 2022) and Top-k(x) (Montreuil et al., 2026) incorporate hyperparameters for cost control; we performed a hyperparameter sweep to identify the configuration that yields the highest accuracy-cost metric within the specified budget. On SVHN, however, both Top-k(x) and PostHoc had already converged, so subsequent values are omitted, consistent with the curves in Figure 2. Additionally, on CIFAR-10, CATABC-Topk and Top-k (Montreuil et al., 2026) each report one fewer data point due to the constraint of having only four entities.

Comparison results on Cifar-10. Table 5 reports the full accuracy and cost of every method and budget on CIFAR-10. CATABC attains the highest accuracy at all budgets, and against Top-$k(x)$, the baseline with the closest setting, it leads by 9.4, 6.9 and 5.4 points at budgets 3, 4 and 5. It is also cheaper where it is more accurate: at budget 5 it reaches 91.58% at a cost of 2.85, against 86.23% at a cost of 4.71 for Top-$k(x)$. CATABC and its variant CATABC-Top$k$ both respect the prescribed budget.

Comparison results on Cifar-100. Table 6 shows that the ranking depends on the budget. Our method is not competitive in the low-budget regime—at budget 0.05 CATABC reaches 60.4%, against 71.0% for Top-$k(x)$—but from budget 0.13 onwards it overtakes the baseline by a wide margin, leading Top-$k(x)$ by 9.1 points and producing the highest accuracy of the table, 93.71% at budget 0.17, obtained by the variant CATABC-Top$k$. The cost side shows the same advantage: at budget 0.17 CATABC reaches 93.25% at a cost of 0.1617, which is below the budget and below the 0.1745 spent by Top-$k(x)$, i.e. a higher accuracy at a lower cost.

Table 6: Comparison results with the baseline on CIFAR-100.

Budget	Metric	Method					
		CATABC (Ours)	CATABC-Topk (Ours)	Top-K	Top-K(x)	CET	PostHoc
0.05	ACC	60.41±0.63	56.82±1.52	67.68±0.35	70.98±0.65	**71.47±0.10**	62.14±0.89
	Cost	0.0452±0.0018	0.0402±0.0038	0.0356±0.0005	0.0506±0.0035	0.0376±0.0001	0.0352±0.0036
0.07	ACC	65.88±2.19	61.19±0.99	67.68±0.35	**74.43±0.83**	/	62.11±0.93
	Cost	0.0629±0.0040	0.0463±0.0022	0.0707±0.0011	0.0704±0.0055	/	0.0345±0.0036
0.11	ACC	68.53±1.22	68.75±0.61	**74.77±0.26**	/	/	/
	Cost	0.0783±0.0057	0.0775±0.0042	0.1030±0.0017	/	/	/
0.14	ACC	**90.94±0.08**	90.84±0.24	79.72±0.49	81.83±1.47	/	/
	Cost	0.1100±0.0000	0.1100±0.0000	0.1345±0.0013	0.1226±0.0119	/	/
0.17	ACC	93.25±0.55	**93.71±0.09**	84.42±0.49	87.05±0.36	/	/
	Cost	0.1617±0.0000	0.1600±0.0000	0.1675±0.0009	0.1745±0.0044	/	/
0.20	ACC	**92.06±2.07**	90.64±3.54	88.02±0.19	88.53±0.03	/	/
	Cost	0.2000±0.0000	0.2000±0.0000	0.2000±0.0000	0.1976±0.0013	/	/

Comparison results on SVHN. Table 7 reports the results on SVHN. CATABC attains the highest accuracy at every budget; against Top-$k(x)$ it leads by 0.68 and 0.45 points at budgets 0.03 and 0.06, the only two budgets at which Top-$k(x)$ still reports a value, and at budget 0.20 it leads Top-k by 6.1 points. With respect to cost control, at budget 0.17 CATABC spends only 0.1178, about 69% of the budget, and is still the most accurate method at that budget, which indicates that it consults additional entities only when their contribution pays for their cost.

Table 7: Comparison results with the baseline on SVHN.

Budget	Metric	Method					
		CATABC (Ours)	CATABC-Topk (Ours)	Top-K	Top-K(x)	CET	PostHoc
0.03	ACC	76.58±0.12	**76.60±0.10**	75.84±0.26	75.90±0.27	76.10±0.05	73.57±0.19
	Cost	0.0281±0.0007	0.0295±0.0010	0.0206±0.0018	0.0356±0.0014	0.0300±0.0001	0.0268±0.0030
0.06	ACC	**76.48±1.86**	74.61±1.57	75.84±0.26	76.03±0.27	/	73.46±0.23
	Cost	0.0559±0.0042	0.0456±0.0096	0.0527±0.0012	0.0531±0.0067	/	0.0227±0.0032
0.09	ACC	**76.59±0.57**	76.43±0.42	75.97±0.21	/	/	/
	Cost	0.0763±0.0090	0.0750±0.0071	0.0885±0.0006	/	/	/
0.13	ACC	**77.87±0.41**	77.20±0.12	75.02±0.14	/	/	/
	Cost	0.1034±0.0191	0.1050±0.0000	0.1249±0.0004	/	/	/
0.17	ACC	**77.83±0.47**	77.66±0.42	73.35±0.06	/	/	/
	Cost	0.1178±0.0305	0.1575±0.0075	0.1612±0.0006	/	/	/
0.20	ACC	**78.03±0.31**	77.56±0.22	71.92±0.06	/	/	/
	Cost	0.1841±0.0276	0.2000±0.0000	0.2000±0.0000	/	/	/

B.4.2 Additional ablation results

Table 8 presents the detailed experimental results of our proposed full method alongside the selector-related variants on CIFAR-10 under three budget settings, serving to validate the effectiveness of the selector. Table 9 similarly reports the results for the fusion network-related variants under the same conditions, aiming to verify the contribution of the fusion network. For each configuration, the mean and standard deviation over four independent runs are reported.

Ablation of the selector. Table 8 confirms that the cost-related components of the selector are necessary: removing them lowers the accuracy at every budget while raising the cost, and the degradation worsens as the budget grows. At budget 5, for instance, *score w/o cost* drops from 91.58% to 91.11% while its cost rises from 2.85 to 4.53; *w/o cost feature* behaves in the same direction with a smaller magnitude. The variant CATABC-Top$k$ fails differently: because it forces a fixed number of experts, it spends almost the whole budget yet attains the lowest accuracy of every configuration in the table.

Table 8: Ablation results of selector

Method	Budget-3.0		Budget-4.0		Budget-5.0	
	ACC	Cost	ACC	Cost	ACC	Cost
CATABC (Ours)	**91.59±0.09**	2.73±0.12	**91.69±0.13**	2.81±0.08	**91.58±0.07**	2.85±0.16
CATABC-Topk (Ours)	84.96±0.74	2.99±0.39	/	/	87.71±0.35	4.99±0.03
score w/o cost	91.51±0.03	2.81±0.03	91.37±0.04	2.99±0.03	91.11±0.20	4.53±0.30
w/o cost feature	91.44±0.28	2.86±0.12	91.31±0.09	3.11±0.40	91.33±0.15	3.89±0.71

Table 9: Ablation results of fusion network.

Method	Budget-3.0		Budget-4.0		Budget-5.0	
	ACC	Cost	ACC	Cost	ACC	Cost
CATABC (Ours)	**91.59±0.09**	2.73±0.12	**91.69±0.13**	2.81±0.08	**91.58±0.07**	2.85±0.16
w/o Transformer	91.42±0.08	2.75±0.02	91.24±0.05	2.82±0.03	91.34±0.13	3.03±0.35
w/o auxiliary tokens	91.50±0.22	2.75±0.15	91.40±0.12	3.04±0.23	91.28±0.12	3.73±0.28
uniform average	91.32±0.33	2.80±0.02	91.34±0.09	2.93±0.05	91.32±0.12	2.81±0.07
Best-RandomWeights	85.09	2.6	86.73	3.6	89.21	5
Mean-RandomWeightse	79.02±5.52	1.90±0.76	80.77±5.34	2.26±0.90	82.29±5.11	3.00±1.35

Ablation of the fusion network. Table 9 shows that the complete method attains the highest accuracy at every budget and that each replacement lowers it. Replacing the Transformer by a plain linear layer as well as averaging the experts with uniform weights degrades both metrics at all three budgets, and removing the embeddings of the cost and of the selector output (*w/o auxiliary tokens*) weakens the cost awareness of the model: at budget 5 the cost grows to 3.73 while the accuracy falls to 91.28%. The random-weight control performs far below all of the above configurations. The fusion network therefore needs the Transformer architecture, the cost-aware embeddings and adaptive weight learning together to reach the best accuracy/budget trade-off.

Table 10: Ablation results of threshold.

Method	Budget-3.0		Budget-4.0		Budget-5.0	
	ACC	Cost	ACC	Cost	ACC	Cost
CATABC (Ours)	**91.59±0.09**	2.73±0.12	**91.69±0.13**	2.81±0.08	**91.58±0.07**	2.85±0.16
threshold= -0.5	91.38±0.35	2.56±0.36	91.48±0.13	2.97±0.07	91.37±0.07	3.51±0.54
threshold= -0.3	91.52±0.08	2.66±0.10	91.33±0.12	2.94±0.18	91.24±0.09	3.96±0.40
threshold= -0.2	91.48±0.10	2.69±0.05	91.30±0.30	2.94±0.24	91.18±0.21	3.73±0.65
threshold= 0.2	91.53±0.14	2.84±0.09	91.34±0.11	2.89±0.09	91.23±0.21	3.27±0.58
threshold= 0.3	91.47±0.10	2.85±0.10	91.36±0.14	2.94±0.08	91.20±0.06	3.34±0.31
threshold= 0.5	91.48±0.50	2.78±0.14	91.40±0.11	3.06±0.40	91.27±0.05	3.33±0.36

Ablation of the threshold within selector. Our method itself operates at a fixed threshold of 0, the shadow price of the Lagrangian formulation; Table 10 varies only this threshold, leaving every other setting unchanged, in order to examine how the selector behaves at different operating points. The complete method at threshold 0 attains the highest accuracy at all three budgets, and the six threshold variants remain 0.06 to 0.40 points below it: at budgets 3 and 5 the best variant is still 0.06 and 0.21 points lower, and at budget 4 the best variant is 0.21 points lower. The accuracy is not monotone in the threshold—negative and positive thresholds average the same accuracy (91.36% against 91.36%)—but it peaks exactly at 0, which indicates that placing the shadow price at 0 allows the selector to play its role most effectively. Deviating from 0 brings no cost advantage either: at budget 5 the negative-threshold variants cost 3.51 and 3.73, more than the 2.85 of the threshold-0

method, while being less accurate, so in several settings our method is simultaneously more accurate and cheaper.

Ablation of the Lagrangian multiplier. Our method adaptively adjusts the Lagrangian multiplier, outperforming fixed multipliers across all budgets and achieving the best accuracy-cost trade-off. (Table 11) Increasing a fixed multiplier reduces cost but sacrifices accuracy (e.g., at budget 4, raising it from 0.05 to 0.20 drops accuracy from 91.36% to 91.18%, while the cost decreases from 2.87 to 2.58). The adaptive multiplier responds to average-cost deviation via the constraint-feedback principle of Lagrangian updates (Nedić & Ozdaglar, 2009), coupling prediction to the consultation budget.

Table 11: Ablation study of Lagrange multipliers on the CIFAR-10 dataset under 3 budgets.

Method	Budget-3.0		Budget-4.0		Budget-5.0	
	ACC	**Cost**	**ACC**	**Cost**	**ACC**	**Cost**
CATABC (Ours)	**91.59±0.09**	2.73±0.12	**91.69±0.13**	2.81±0.08	**91.58±0.07**	2.85±0.16
lambda=0.05	91.34±0.19	2.68±0.17	91.36±0.19	2.87±0.17	91.32±0.20	3.01±0.85
lambda=0.10	91.27±0.10	2.65±0.23	91.31±0.10	2.74±0.28	91.21±0.10	2.86±0.23
lambda=0.15	91.16±0.10	2.57±0.17	91.22±0.10	2.66±0.13	91.19±0.10	2.72±0.22
lambda=0.20	91.10±0.17	2.45±0.25	91.18±0.17	2.58±0.36	91.09±0.17	2.63±0.14