Model Cascades with Provable Per-Class Quality

Ashwin Gerard Colaço[1] **Sharad Mehrotra**[1] **Michael De Lucia**[2]
Kevin Hamlen[3] **Murat Kantarcioglu**[4] **Latifur Khan**[3]
Ananthram Swami[2] **Bhavani Thuraisingham**[3] **Unnat Jain**[1]

[1]University of California, Irvine [2]Army Research Laboratory [3]University of Texas at Dallas [4]Virginia Tech
acolaco@uci.edu

Abstract

Model cascades reduce inference cost by sending each input through cheap models first and escalating to expensive ones only when needed. Existing exit rules, such as confidence thresholds, learned scorers, and agreement signals, optimize aggregate quality. A cheap model that is systematically unreliable on a rare or fine-grained class can therefore degrade that class unnoticed, even when it is confident. We introduce NOMAD, whose central idea is that the predicted class label, not the confidence score alone, is the right signal for deciding when a cascade exits. For each model, NOMAD uses held-out data to find the classes on which it loses at most a small, user-chosen fraction of the quality of a role model, typically the most accurate model available. A chain-safety check certifies, under stated conditions, that a sequence of models preserves per-class quality end to end. A greedy selector routes each input through a cheap certified sequence, at an expected cost at most four times that of the best fixed ordering under deterministic exits. On 16 tabular, vision, and text datasets, including cascades of large language models, NOMAD achieves a geometric-mean speedup of $4\times$ and up to $42\times$ over the role model. It is the only method among 11 baselines that stays within this fraction on every class of every dataset. Code, configurations, fold indices, and an interactive results explorer (https://nomad-cascades.surge.sh) are released for reproducibility.

1 Introduction

Classifiers for the same task can differ in inference cost by orders of magnitude. A lightweight CNN labels a bird image in milliseconds, while a Vision Transformer large enough to separate nearly identical species takes fifty times as long. Most inputs are easy, and a cheap model handles them well. *Model cascades* exploit this by calling classifiers from cheap to expensive and stopping as soon as the current one is good enough [50, 52, 49, 8, 22]. The dominant stopping rule is softmax confidence, which stops when the confidence exceeds a threshold and escalates otherwise.

Confidence is a poor proxy for how reliable a model is on a class. On an image of a Black-throated Blue Warbler, MobileNetV3 predicts Cerulean Warbler with softmax confidence 0.91, yet its per-class F1 on the true class is only 0.61 (Fig. 1, left). On a banking query about country support, DistilBERT predicts `card_not_working` with confidence 0.89, yet its per-class F1 on `country_support` is only 0.67 (Fig. 1, right). A cascade that trusts high confidence accepts both predictions and returns the wrong label. Changing the threshold does not help, because confidence is decoupled from per-class reliability, both on the class the model predicts and on the class the input belongs to.

Many applications need a quality floor on every class, which is a stronger requirement than a floor on average quality. A wildlife monitor cannot miss an endangered species, a clinical triage classifier cannot under-call a rare condition, and a network-intrusion detector must catch attack categories that make up under 1% of traffic. No confidence threshold can enforce such a floor, and methods

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

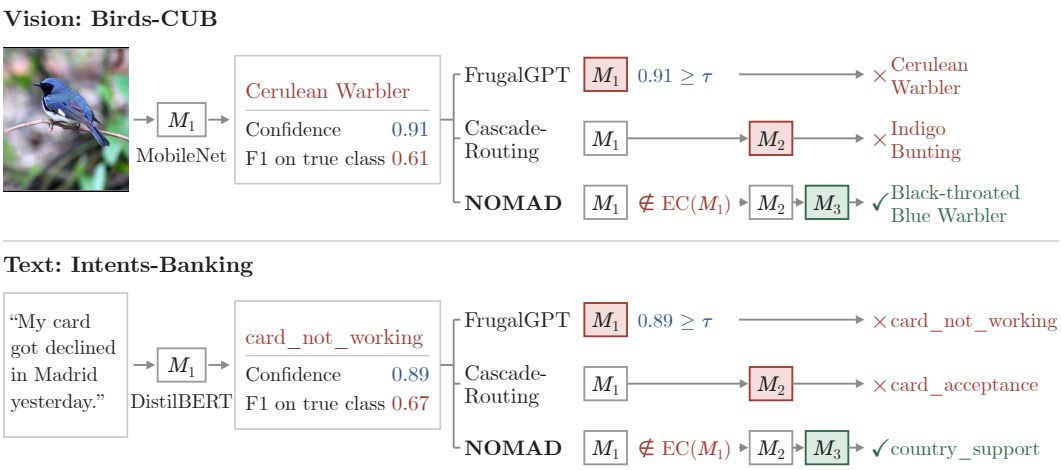


Figure 1: **Confidence is not competence. Two confident errors by cheap models.** In the top row, the cheap model M_1 labels a Black-throated Blue Warbler as a Cerulean Warbler with confidence 0.91, although its F1 on the true class is only 0.61. In the bottom row, M_1 labels a `country_support` query as `card_not_working` with confidence 0.89. FrugalGPT [8] exits because the confidence clears its threshold τ. Cascade-Routing [9] escalates to M_2, which is also wrong. NOMAD escalates because the predicted class lies outside $\mathrm{EC}(M_1)$, the classes on which M_1 meets the quality floor $(1-\epsilon)F_1^{\mathrm{RM}}$, and M_3 returns the correct label. In both rows the cheap model's F1 on the true class is below this floor, so its high confidence says nothing about whether its answer can be trusted. Appendix F.3 gives five non-cherry-picked input traces per dataset, one per behavior pattern.

that optimize quality averaged over all inputs share this limitation. A small drop on a rare class is outweighed by correct predictions on common classes, so failures on rare or fine-grained classes stay invisible to any aggregate measure.

Jitkrittum et al. [20] prove that the optimal deferral rule depends on the predicted class label and not only on a confidence score. Their proposed rules, however, remain functions of confidence statistics, the signal their own theory identifies as suboptimal. NOMAD operationalizes the label-dependence principle of Jitkrittum et al. [20], extending it from a single deferral decision to a certified chain. For each model, NOMAD uses held-out data to find the model's *exit classes*. These are the classes on which the model loses at most a small, user-chosen fraction of the quality of a *role model*, typically the most accurate model. At inference, a cheap model ends the cascade only when it predicts one of its own exit classes. Otherwise the cascade escalates and updates its estimate of the input's true class, so different inputs follow different chains of models. NOMAD needs no access to model internals and no retraining, because it queries each model only for its predicted label and softmax output.

Choosing exit classes for each model separately does not guarantee the floor for the cascade as a whole. An early model can exit on an input that it should have forwarded, and these losses compound down the chain. We therefore introduce *chain safety*, a check on a whole sequence of models. We prove a guarantee for this check under two conditions on how the stages interact. A sequence passing it at population level then loses at most that fraction of the role model's per-class F1 on any class. We also show that ordering the models reduces to Weighted Min-Sum Set Cover [15, 35], the classical problem of ordering sets to cover weighted items at minimum expected cost. The reduction gives greedy selection an expected cost at most four times that of the best fixed ordering of the models, under a deterministic exit model. To our knowledge, this is the first approximation guarantee for greedy model ordering in multiclass cascades.

We evaluate NOMAD on 16 tabular, vision, and text datasets against 11 baselines. NOMAD achieves a geometric-mean speedup of $4\times$ over always calling the role model, and up to $42\times$ on Pendigits. It keeps every class at or above its floor on all 16 datasets. Every baseline lets some class fall below its floor on at least one dataset, and on the three featured datasets the confidence-based cascades do so on 29 to 72% of classes.

Contributions. NOMAD provides a cascade-level, per-class quality certificate using only model outputs. It contributes **(i)** exit classes, an exit signal that conditions on the predicted class rather than on confidence (Sec. 3.1); **(ii)** chain safety, a cascade-level check with a per-class quality guarantee (Sec. 3.2); **(iii)** greedy per-input selection with a 4-approximation to the best fixed ordering under deterministic exits (Sec. 3.3); and **(iv)** an evaluation with zero per-class violations on all 16 datasets at every tested tolerance (Sec. 4). Code, configurations, and fold indices are released for reproducibility.[1]

2 Related Work

NOMAD works with any classifier, because it accesses each model only through its predictions and needs neither model internals nor retraining. White-box methods such as early-exit networks, mixture-of-experts models, and internal-confidence gating [46, 5, 43, 41] operate in a different setting. CalexNet [4], for example, calibrates per-class exit thresholds for early-exit branches trained inside a single network.

Model cascade exit rules. Cascades from Viola–Jones [50] to recent LLM cascades [52, 49, 27, 8, 22, 55, 54, 37] decide when to exit from a score of the model outputs, most often the cheap model's confidence. Post-hoc calibration [17] can make confidence track average accuracy more closely, but it does not close per-class accuracy gaps. A well-calibrated model can still be systematically unreliable on specific classes. Jitkrittum et al. [20] prove that the Bayes-optimal exit rule depends on the predicted class labels of both models, not just on the first model's confidence. In practice their proposed rules approximate this criterion from the cheap model's softmax, so the predicted label itself plays no direct role in the exit decision. NOMAD acts on this result directly, exiting based on which class is predicted rather than on how confident the model is, and provides a cascade-level, per-class quality certificate for black-box cascades.

Routing and aggregate-quality optimization. Routers commit to a single model per input and do not revise that choice [40, 11, 29, 44, 28, 42, 57, 38]. The closest relative of NOMAD is Cascade-Routing [9], which, like NOMAD, can move to any model in the portfolio after observing each response. Its objective, however, is aggregate quality under a cost budget rather than a quality floor on every class, and Sec. 4.2 measures the per-class violations that result.

Quality guarantees in cascades. C3PO [48], Trust or Escalate [21], and Online Cascade [39] give cascades formal guarantees on population coverage, marginal selective risk, and regret, respectively. Each guarantee holds on average over inputs rather than per class. For single models, conformal risk control [1, 2] provides finite-sample risk control, and class-conditional conformal prediction [12] provides per-class coverage. Methods for learning to reject or defer [16, 30, 34, 32] add an abstain or defer option to a single model, and Narasimhan et al. [36] extend rejection to long-tail metrics such as the balanced error. A cascade is the multi-stage form of deferral, but each of these methods certifies a single decision, and per-stage guarantees do not add up to a guarantee for the whole chain. NOMAD's chain-safety check certifies the composed chain. Under two conditions on how the stages interact, Theorem 1 shows that a chain passing it at population level preserves per-class F1 end to end.

Model ordering and approximation guarantees. Cost-sensitive sequential classifiers [47, 18] characterize when cascading helps under budget constraints, but not under per-class quality constraints. NOMAD's reduction of model ordering to Weighted Min-Sum Set Cover [15, 35] yields, to our knowledge, the first approximation guarantee for greedy model selection in multiclass cascades.

3 NOMAD

Setting. We are given a portfolio of n pre-trained classifiers $M_1, \ldots, M_n$ over K classes, indexed in order of increasing per-input inference cost $\mathrm{cost}(M_i)$. The user designates one of them as the *role model* M_r, usually the most accurate and most expensive model, whose per-class quality the cascade must preserve. The user also chooses a tolerance $\epsilon \in [0, 1)$ and supplies a held-out validation set D_{val}. Like prior cascades [50, 52, 8, 21, 22], we fit all statistics on D_{val}, and we report results on a separate test split. We write $Q(M, j) \in [0, 1]$ for the quality of a predictor M on class j, where

[1] Interactive results explorer: https://nomad-cascades.surge.sh.

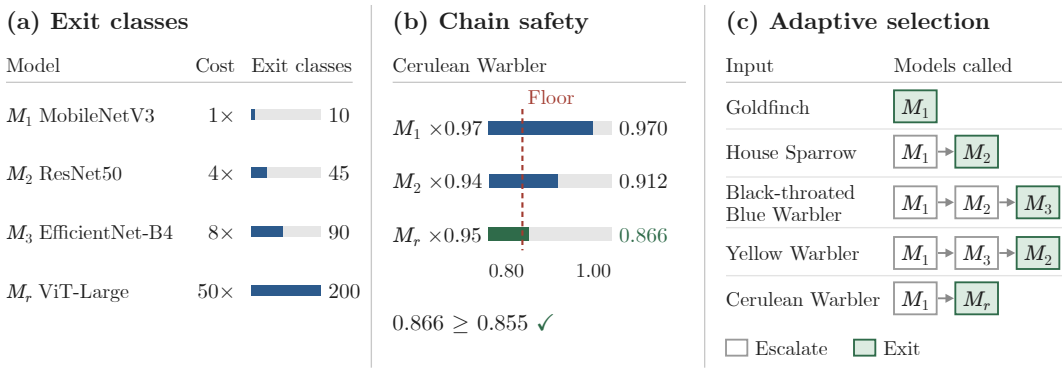


Figure 2: **NOMAD on Birds-CUB.** From left to right, the panels illustrate exit classes (Sec. 3.1), the chain-safety check for the Cerulean Warbler class (Sec. 3.2), and adaptive selection for five inputs (Sec. 3.3). Costs are relative to M_1. In the left panel, the four models have exit sets of 10, 45, 90, and 200 of the $K = 200$ classes, computed once on D_{val}. In the middle panel, the first two factors are the passthroughs 0.97 and 0.94 of the stages before the exit model, and the last is the exit model's per-class F1 of 0.95. Their product, 0.866, must stay at or above the floor $(1-\epsilon)F_1^{\text{RM}} = 0.855$. In the right panel, five birds follow five different chains, each chosen greedily as the belief sharpens after every escalation.

higher is better, and any such per-class quality can be used. Our experiments and Theorem 1 use per-class F1, which combines precision and recall on class j, and per-class recall or accuracy can replace it without changing the method. We write $F_1^{\text{RM}}(j)$ for the role model's per-class F1. The goal is a predictor whose quality on every class j is at least the *floor* $(1-\epsilon)\,Q(M_r, j)$, while calling the role model as rarely as possible. We call this requirement the *per-class contract*, and a class that falls below its floor a *violation*.

NOMAD, our cascade for this problem, works in two phases, which Fig. 2 illustrates on the running example of this section, Birds-CUB. The validation phase runs once and finds each model's exit set, the classes on which the model meets its floor (Sec. 3.1). It also estimates the statistics of a chain-safety check, which accepts a sequence of models only if the whole sequence is projected to meet every floor (Sec. 3.2). The inference phase runs once per input and repeatedly calls the model most likely to resolve the input per unit cost, provided the extended chain passes the check. It stops as soon as a model predicts one of its own exit classes (Sec. 3.3).

3.1 Exit Classes

This subsection finds the classes on which a model is reliable enough to end the cascade. Each model M_i has a $K \times K$ confusion matrix $\text{CM}^{(i)}$ with entries $\text{CM}^{(i)}_{jk} = \Pr[M_i(x) = k \mid \text{true class} = j]$, which NOMAD estimates on D_{val}. The *exit set* of M_i contains the classes on which M_i meets the floor, and we call its members the *exit classes* of M_i:

$$\text{EC}(M_i) = \{j : Q(M_i, j) \geq (1-\epsilon)\,Q(M_r, j)\}. \tag{1}$$

Eq. (1) keeps a class when M_i loses at most a fraction ϵ of the role model's quality on it, and a model with a larger exit set can end the cascade on more inputs. By definition, $\text{EC}(M_r) = \{1, \ldots, K\}$, so the role model can answer any input. At inference, a model that predicts one of its own exit classes ends the cascade with that prediction, and any other prediction sends the input on to another model. The rule reads only the predicted label, never the softmax score, and each exit set is computed once on D_{val}.

An exit set says where a model is reliable, but it does not stop the model from mislabeling inputs of other classes. A model M_i can predict one of its exit classes on an input whose true class j lies outside $\text{EC}(M_i)$. The cascade then stops early with a wrong answer, and we say that M_i *captures* the input. The *passthrough* $\pi(M_i, j)$ is the probability that an input of class $j \notin \text{EC}(M_i)$ leaves M_i unresolved and continues down the chain [47]:

$$\pi(M_i, j) = 1 - \sum_{k \in \text{EC}(M_i)} \text{CM}^{(i)}_{jk}. \tag{2}$$

Eq. (2) subtracts from one the probability that M_i predicts any of its exit classes on a class-j input. A high passthrough is good, because M_i then sends class-j inputs down the chain instead of answering them wrongly. A low passthrough means that M_i often captures class-j inputs, which is the failure that chain safety must bound.

3.2 Chain Safety

This subsection decides whether a candidate chain $\mathcal{S}$, an ordering of some of the n models, meets the per-class contract. Exit sets are chosen one model at a time, but quality is lost along the whole chain. When an early model captures some class-j inputs, those inputs never reach the model that handles class j well, and the losses of successive stages multiply, which is the classic cascading-errors problem [50, 47]. The middle panel of Fig. 2 shows it for the Cerulean Warbler class. The two stages before the exit model forward 97% and 94% of this class, which lowers the exit model's per-class F1 of 0.95 to a projected 0.866, just above the floor of 0.855.

For class j, let $M_{\text{exit}}(j)$ be the first model in $\mathcal{S}$ whose exit set contains j, and write M_ℓ for the ℓ-th model in $\mathcal{S}$. The *projected quality* of $\mathcal{S}$ on class j multiplies the passthroughs of the stages before $M_{\text{exit}}(j)$ by the exit model's own quality:

$$Q_{\text{proj}}(\mathcal{S}, j) = \left(\prod_{\ell : M_\ell \text{ precedes } M_{\text{exit}}(j)} \pi(M_\ell, j) \right) \cdot Q\big(M_{\text{exit}}(j), j\big). \tag{3}$$

The projection is high only when earlier stages rarely capture class j and the exit model is good on j. If no model in $\mathcal{S}$ has j in its exit set, class-j inputs fall through to M_r, whose exit set contains every class. Eq. (3) multiplies per-stage passthrough probabilities as if capture events were independent. When captures are positively associated across stages, as when cheap models fail on the same hard inputs, the true probability of passing every stage is at least this product. The product then understates the share of class-j inputs that reach $M_{\text{exit}}(j)$, and the chain-safety check errs on the strict side. Appendix J.2 reports the measured correlations.

The chain $\mathcal{S}$ passes the *chain-safety check* when every class meets its floor:

$$Q_{\text{proj}}(\mathcal{S}, j) \geq (1 - \epsilon) \cdot Q(M_r, j) \quad \text{for all } j. \tag{4}$$

A single class projected below its floor fails the whole chain.

Theorem 1 (Chain Safety Guarantee)**.** *Fix a chain $\mathcal{S}$ and a class j, let e be the position of $M_{exit}(j)$ in $\mathcal{S}$, and let $a_j = \prod_{\ell<e} \pi(M_\ell, j)$. Let $E_1, \ldots, E_{e-1}$ be the events that the stages before $M_{exit}(j)$ forward a class-j input, and let E_e be the event that $M_{exit}(j)$ labels it correctly. For an input x with true class y, assume that (a) $E_1, \ldots, E_e$ are positively associated [14], and (b) $\Pr[$a stage after $M_{exit}(j)$ outputs $j,\ y \neq j] \leq (1 - a_j) \Pr[M_{exit}(j)(x) = j,\ y = j]$. If the chain-safety check (4) holds with π and Q at their population-level values, then the realized per-class F1 of the cascade on class j is at least $(1 - \epsilon)\, Q(M_r, j)$.*

Corollary 1.1. *Hypothesis (a) holds whenever $E_1, \ldots, E_e$ are independent. If Q is per-class recall instead of F1, the conclusion of Theorem 1 holds under hypothesis (a) alone.*

Theorem 1 lifts reliability from single models to the whole chain. Hypothesis (a) controls the class-j inputs lost before $M_{\text{exit}}(j)$. False positives on j need no assumption up to $M_{\text{exit}}(j)$, because no earlier stage can output j and $M_{\text{exit}}(j)$ sees a subset of the inputs it would see alone. Hypothesis (b) bounds the false positives that later stages can add. The theorem does not account for estimation error, because the check is evaluated at population values. It concerns a fixed chain, and the loop in Sec. 3.3 applies the check to every chain it runs. Appendix B.1 gives the proof.

Estimating π and Q from D_{val}. We estimate π and Q marginally on all of D_{val}. The passthrough $\pi(M_i, j)$ is the fraction of class-j validation inputs whose predicted label lies outside $\text{EC}(M_i)$, and $Q(M_i, j)$ is the empirical per-class F1. Conditional estimation on each chain's forwarded subpopulation would need no dependence assumption, but it is high-variance for rare classes and must be redone for every chain. Hypothesis (a) of Theorem 1 is the price of the marginal choice. Sparse data raise a second issue, since a point estimate from few samples can overstate a model's quality on a rare class. A class is therefore added to a cheaper model's exit set exactly when the Wilson lower confidence bound of the class's estimated quality clears the floor, and no separate point-estimate rule exists. With few samples the bound lies far below the empirical value and keeps

the class out, and with many samples it tightens to the point estimate. The Wilson interval treats F1 as a binomial proportion, which it is not, so this rule is a heuristic without a formal coverage guarantee (Appendix A.2). This admission rule is the frequentist counterpart of NOMAD-BA, the Bayesian online variant in Appendix K, which tracks the rows of the same confusion matrices with Dirichlet posteriors.

3.3 Inference with Adaptive Cascade Selection

This subsection decides, for one input, the order in which to call the models. NOMAD tracks a *belief* $p_{\text{cur}} \in \Delta^{K-1}$, the probability distribution it currently assigns to the input's true class, where Δ^{K-1} is the probability simplex over the K classes. The belief starts at the empirical class prior p, where p_j is the fraction of validation inputs in class j. At each step NOMAD calls the model most likely to resolve the input per unit cost, and if that model does not resolve the input, NOMAD updates the belief and chooses again.

Greedy selector. At each step, the selector picks the untried model with the highest utility

$$U(M_i) = \frac{\sum_{j \in \mathrm{EC}(M_i)} p_{\text{cur}}(j)}{\mathrm{cost}(M_i)}. \tag{5}$$

The numerator is the belief mass on the exit classes of M_i, which estimates the chance that M_i resolves the input. The denominator is the cost of M_i. A high utility means that M_i is cheap and likely to resolve the input.

Belief update. When the selected model M_{sel} predicts a class outside its exit set, NOMAD updates the belief in two steps. First, it reweights the belief by the softmax output $\mathbf{s}$ of M_{sel}, which moves belief toward the classes M_{sel} found plausible,

$$\tilde{p}(j) = \frac{p_{\text{cur}}(j)\, s(j)}{\sum_{j'} p_{\text{cur}}(j')\, s(j')}. \tag{6}$$

Second, because the prediction of M_{sel} lies outside $\mathrm{EC}(M_{\text{sel}})$, NOMAD sets the belief on $\mathrm{EC}(M_{\text{sel}})$ to zero and renormalizes over the remaining classes:

$$p_{\text{new}}(j) = \begin{cases} 0 & j \in \mathrm{EC}(M_{\text{sel}}), \\ \tilde{p}(j)/Z & j \notin \mathrm{EC}(M_{\text{sel}}), \end{cases} \qquad Z = \sum_{j' \notin \mathrm{EC}(M_{\text{sel}})} \tilde{p}(j'). \tag{7}$$

This step reflects that a model reliable on a class usually recognizes inputs of that class, so a non-exit prediction points away from its exit classes. The right panel of Fig. 2 shows five Birds-CUB inputs that follow five different chains as their beliefs sharpen.

Per-input inference loop. NOMAD sets p_{cur} to the class prior and repeats four steps.

1. Select $M_t = \arg\max_i U(M_i)$ among the models not yet tried, using Eq. (5). If none remains, return the role model's prediction.
2. Check the running chain extended by M_t against Eq. (4). If the check fails, discard M_t and return to step 1.
3. Run M_t. If its predicted label is in $\mathrm{EC}(M_t)$, return that label.
4. Otherwise, update p_{cur} by Eqs. (6) and (7) and return to step 1.

Alg. 1 gives the pseudocode.

Cost guarantee. Choosing $\arg\max_i U(M_i)$ is exactly the greedy step of Weighted Min-Sum Set Cover [15, 35], whose items here are the classes with weights p_j and whose sets are the exit sets $\mathrm{EC}(M_i)$ with costs $\mathrm{cost}(M_i)$. The role model's exit set covers every class, so this instance is always feasible and every input terminates. Consider a deterministic exit model, in which every input exits at the first model whose exit set contains its class. Under this model, a non-exit reveals only that the class lies outside the exit set, so the greedy rule with the update of Eq. (7) is the WMSSC greedy algorithm. Its expected cost is at most four times that of the cost-optimal fixed ordering of the models it runs (Theorem 2). The bound does not cover the softmax reweighting of Eq. (6), which adapts the order to each input, and Appendix A.1 compares greedy selection with an optimal adaptive policy (Theorem 3).

Table 1: **NOMAD is the only method with zero per-class violations on every dataset.** Each dataset is evaluated independently, with its own model portfolio and role model. Values are 5-fold cross-validation means at $\epsilon = 0.10$. Speedup is relative to M_r. Bold marks the best non-bound value per column. The next-best baseline, Conformal-CC, still violates 1.3 to 1.6% of classes. Confidence-based cascades reach their speed by violating 29 to 72% of classes, and Cascade-Routing reaches $10.6\times$ on Birds-CUB against NOMAD's $5.0\times$ while violating 27.5% of classes. [†]The rarest Intrusions-NSL class has 9 or 10 validation samples per fold, too few for a meaningful class-conditional conformal quantile (Appendix G.3). Tabs. A.5 to A.7 cover all 16 datasets, and Tab. A.8 gives per-fold standard deviations.

		Intrusions-NSL *Tabular*			**Birds-CUB** *Vision*			**Intents-Banking** *Text*		
Method	Key Technique	F1↑	Spd↑	Viol%↓	F1↑	Spd↑	Viol%↓	F1↑	Spd↑	Viol%↓
Bounds										
Role Model	Most-expensive-only	90.1	1.0×	0.0	83.4	1.0×	0.0	92.7	1.0×	0.0
Cheapest Model	Cheapest-only	75.1	429.0×	76.0	68.6	16.6×	94.1	75.0	3.0×	87.3
Cascades										
FrugalGPT	Learned exit scorer	81.8	4.7×	52.0	72.4	**12.4**×	65.1	83.9	2.6×	48.3
Cost Cascade	Fixed confidence threshold	80.3	4.5×	60.0	81.8	10.3×	43.6	85.0	**2.7**×	45.2
ABC	Class-balanced confidence	83.8	4.1×	36.0	76.1	12.2×	46.1	87.8	2.5×	28.8
Trust or Escalate	Marginal UCB	85.3	3.9×	32.0	82.7	9.3×	38.3	86.9	2.5×	32.2
Online Cascade	Imitation-learned thresholds	76.8	4.9×	72.0	76.4	10.4×	70.9	82.7	2.6×	57.7
Conformal-CC	Class-cond. conformal sets	degenerate calibration[†]			75.9	4.9×	1.6	82.8	2.0×	1.3
Routers										
RouteLLM	Pairwise logistic router	78.9	**5.1**×	64.0	78.8	10.9×	58.7	80.2	2.6×	74.5
AutoMix	Entropy self-verification	84.0	4.5×	48.0	79.7	8.8×	53.8	86.3	2.3×	35.3
Cascade-routing hybrids										
C3PO	Population coverage	85.8	3.7×	36.0	78.1	9.7×	33.9	86.0	2.3×	37.7
Cascade-Routing	Aggregate-optimal re-routing	84.2	4.1×	40.0	79.7	10.6×	27.5	86.4	2.6×	35.8
Gating (white-box reference)										
MoE	Mixture-of-Experts gating	85.9	4.0×	32.0	83.8	8.1×	32.8	87.9	2.2×	25.7
Ours										
NOMAD	Class-cond. exit + dynamic order	**86.5**	3.5×	**0.0**	**84.5**	5.0×	**0.0**	**89.3**	2.1×	**0.0**

4 Experiments

4.1 Setup

Datasets. We evaluate on 16 datasets across tabular, vision, and text data. Three of them, one per modality, are featured in Tab. 1 and Fig. 3: Intrusions-NSL [45] with normal traffic and 4 attack categories, two of them rarer than 1% of traffic; Birds-CUB [51] with 200 bird species in look-alike groups of 5 to 15 closely related species; and Intents-Banking [7] with 77 banking intents whose vocabulary overlaps, such as `card_arrival` and `card_delivery_estimate`. All three are imbalanced, with per-class counts spanning more than a factor of ten. This is the regime where the per-class contract becomes more apparent, because aggregate metrics hide failures on rare classes. Tab. A.2 lists all 16 datasets, and Fig. A.2 gives an overview by modality and difficulty.

Base classifiers. Tabular portfolios contain up to 8 classical models, vision portfolios range from pretrained CNNs to ViT, and text portfolios range from DistilBERT to DeBERTa-v3-large. LLM portfolios use Qwen2.5-7B, 32B, and 72B and LLaMA-3.1-8B and 70B, served with vLLM, whose per-input costs span a $16\times$ range. To decorrelate errors across models, each non-LLM model is trained on an independent random subsample.

Baselines. We compare against 11 baselines, grouped by routing mechanism and listed in Tab. A.4. The cascades FrugalGPT [8], Cost Cascade [52], ABC [22], Trust or Escalate [21], Online Cascade [39], and Conformal-CC [12] call the cheapest model first and escalate when an exit signal fires, such as confidence, agreement between models, or the size of a conformal prediction set. The routers RouteLLM [40] and AutoMix [29] pick one model per input up front and commit to it. The cascade-router hybrids C3PO [48] and Cascade-Routing [9] pick the next model again after each call.

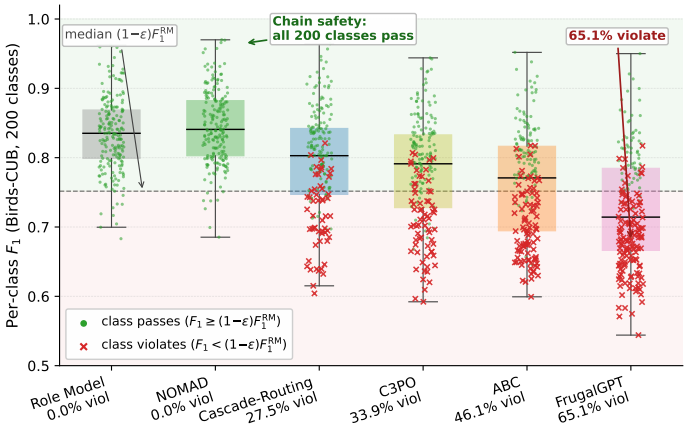


Figure 3: **NOMAD is the only method with no per-class violations.** Per-class F1 on Birds-CUB at $\epsilon = 0.10$. Each box shows the per-class F_1 of one method over the 200 classes, averaged across 5 folds. Green dots are classes at or above their floor $(1-\epsilon)F_1^{\mathrm{RM}}$, and red crosses are violations. Every NOMAD class lies inside the passing region, while the baseline boxes spill below it.

	F1	Spd	Viol%
NOMAD	**84.5**	5.0×	**0.0**
w/o per-class	82.2	9.1×	22.5
w/o dynamic	83.1	4.0×	**0.0**
C3PO	78.1	**9.7**×	33.9

Table 2: **Removing per-class scope reproduces the baseline failure mode.** Ablations on Birds-CUB at $\epsilon = 0.10$, with C3PO for reference. W/o per-class uses an aggregate chain-safety check and violates 22.5% of classes, close to C3PO's 33.9%. W/o dynamic fixes the model order to cheapest first and keeps the contract but loses 20% of the speedup. Bold marks the best value per column.

Mixture-of-experts gating is a white-box reference, and the role model alone and the cheapest model alone are reference bounds. Two ablations isolate NOMAD's components: NOMAD w/o per-class replaces the per-class chain-safety check with an aggregate one, and NOMAD w/o dynamic fixes the model order to cheapest first. All methods except ABC and the two bounds are fit on the same labeled validation split as NOMAD. This includes our adaptation of C3PO, which is self-supervised in its original form.

Metrics. We report F1 as the mean per-class F1 in percent, Speedup as $\mathrm{cost}(M_r)$ divided by the method's average cost per input, and Viol% as the share of classes whose F1 falls below the floor $(1-\epsilon)F_1^{\mathrm{RM}}$. Viol% measures the per-class contract directly. Tab. A.3 defines three further metrics.

4.2 Per-Class Quality

NOMAD has no per-class violations on any of the 16 datasets, while every baseline has violations on at least one (Tabs. 1 and A.7). The closest baseline, Conformal-CC, violates 1.6% of classes on Birds-CUB and 1.3% on Intents-Banking and has no reported value on Intrusions-NSL. This agrees with Theorem 1, although the theorem concerns population-level quantities and the tables report held-out test data. Fig. 3 shows how the violations are spread across the 200 classes of Birds-CUB at $\epsilon = 0.10$. Every NOMAD class lies at or above its floor, and not by a thin margin. Across the 200 classes and 5 folds, the smallest gap between a class's F1 and its floor is 0.013 and the median gap is 0.095. Methods that optimize aggregate quality leave many classes below it, from C3PO at 33.9% and ABC at 46.1% of classes to FrugalGPT at 65.1%. Cascade-Routing is the baseline closest to NOMAD in design, because it also re-orders models for each input, but its objective averages quality over classes. Its median per-class F1 is close to NOMAD's, yet on Birds-CUB it violates 27.5% of classes while reaching 10.6× speedup against NOMAD's 5.0×. On ImageNet it reaches 10.6× against 8.9× while violating 35.7% of classes (Tab. A.9). The ablation in Tab. 2 reproduces this failure. Replacing NOMAD's per-class check with an aggregate one raises its violations on Birds-CUB from zero to 22.5% of classes, the same failure mode the aggregate baselines show. Figs. A.3 to A.5 give per-dataset views, and Appendix F.3 gives non-cherry-picked per-input traces.

4.3 Cost: Speedup Under the Per-Class Contract

NOMAD's geometric-mean speedup over the role model is 4.0× across the 16 datasets, reaching 41.8× on Pendigits and 8.9× on ImageNet with no per-class violations (Tab. A.6). Greedy selection comes much closer to the optimum than its factor-4 bound requires. On three datasets small enough

Table 3: **Speedup does not come from deferring to the role model, and it persists across portfolio sizes and cost structures.** Left, the share of test inputs resolved at each stage. Middle, speedup with 2, 4, and all n models. Right, speedup with measured costs c replaced by $\sqrt{c}$ or c^2. All at $\epsilon = 0.10$ with zero violations in every cell. Pendigits speedups are fold medians.

	Inputs resolved at			Speedup by portfolio size				Speedup by cost structure		
Dataset	first	intermediate	M_r	2	4	full	n	c	$\sqrt{c}$	c^2
Birds-CUB	50%	46%	4%	1.8×	3.6×	5.0×	6	5.0×	4.7×	4.9×
Intents-Banking	87%	3%	10%	2.1×	2.0×	2.1×	5	2.1×	1.3×	4.5×
Intrusions-NSL	76%	12%	12%	4.2×	4.0×	3.5×	8	3.5×	2.1×	4.9×
Pendigits	80%	19.5%	0.5%	5×	5×	30.3×	8	—	—	—

to enumerate every model ordering, NOMAD's cost is 1.07 to 1.17 times the optimum. On five datasets, greedy selection keeps 93 to 98% of the speedup of the optimal adaptive policy defined in Appendix A.1 (Tabs. A.13 and A.14). Fixing the model order to cheapest first keeps every Birds-CUB class above its floor but costs 20% of the speedup (Tab. 2).

The speedup does not come from sending inputs to the role model. Tab. 3 shows that the role model answers 4% of Birds-CUB inputs and at most 12% on any of the four datasets shown. On Intrusions-NSL, the traffic that reaches it is the two rare attack classes, R2L and U2R, which no cheap model's exit set contains. The speedup also persists as the portfolio grows. An added model raises the speedup when it certifies new exit classes, which is why Birds-CUB rises from $1.8\times$ with two models to $5.0\times$ with six. A model that fails certification is skipped and never paid for, so Intents-Banking stays near $2\times$. A certified model also adds a stage that forwarded traffic must pass through, so Intrusions-NSL eases from $4.2\times$ to $3.5\times$. Costs enter only the visit order of Eq. (5), never the chain-safety check, so changing them moves the speedup but cannot create a violation. Appendix G.2 adds three hardware profiles.

The largest speedups occur when models differ in which classes they handle well. Fig. 5 shows that on Birds-CUB and Intents-Banking a cheap model can be the only model other than M_r that is reliable on certain classes, even where more expensive intermediate models are not. NOMAD sends those classes to the cheap model directly. On Pendigits, the cheapest model is a KNN classifier that is reliable on most digit classes, so most inputs exit at the first model (Fig. A.12). KNN is $1279\times$ cheaper than the role model, and the gap to NOMAD's $41.8\times$ is the price of the conservative chain-safety check. The Pendigits speedup ranges from $7\times$ to $90\times$ across folds with the class balance of each split, with a median of $30.3\times$ (Tab. A.8). Speedup shrinks when each cheap model is about equally accurate across classes, because a cheap model below the floor on one class is then below it on all of them (Sec. 5).

Robustness across ϵ. NOMAD has no violations on any dataset at any $\epsilon \in \{0.05, 0.10, 0.15, 0.20\}$, and its geometric-mean speedup grows from $2.8\times$ to $7.5\times$ across this range (Tab. A.10). Every baseline still has violations on at least 12 of the 16 datasets at $\epsilon = 0.20$. A looser tolerance makes the floor easier to meet, but it does not repair the routing signal. Fig. 4 traces the resulting cost-quality curve on Birds-CUB and Intents-Banking. Each ϵ buys a speedup at a guaranteed floor of $(1 - \epsilon)$ times the role model's per-class F1. Every baseline that is faster than NOMAD on these datasets violates at least 25% of classes.

Online setting without validation data. NOMAD's exit sets come from a labeled validation set, which some deployments do not have. NOMAD-BA (Appendix K) learns them online instead, with a Dirichlet posterior on each confusion-matrix row updated from the role model's predictions. On simulated streams for UCI-HAR, Intents-Banking, and ImageNet, NOMAD-BA reaches 93 to 98% of offline NOMAD's speedup with zero cumulative violations, and it converges after about 1,800 inputs (Fig. A.15).

5 Conclusion

NOMAD lets the predicted class, rather than the confidence score, decide when a cascade stops. Each model ends the cascade only on classes where it keeps the role model's quality within a tolerance ϵ. Under stated conditions, the chain-safety check certifies that per-class quality survives the whole

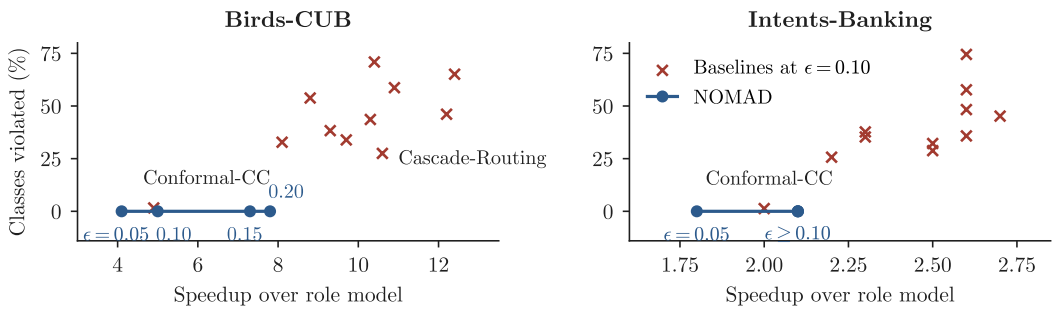


Figure 4: **Cost-quality curves.** NOMAD's speedup at $\epsilon \in \{0.05, 0.10, 0.15, 0.20\}$, with zero violations at every ϵ, against the 11 baselines at their Tab. 1 operating points for $\epsilon = 0.10$. The vertical axis is the share of classes below the floor. On Intents-Banking, NOMAD's speedup is $2.1\times$ for every $\epsilon \geq 0.10$. Tab. A.10 gives the sweep on all 16 datasets.

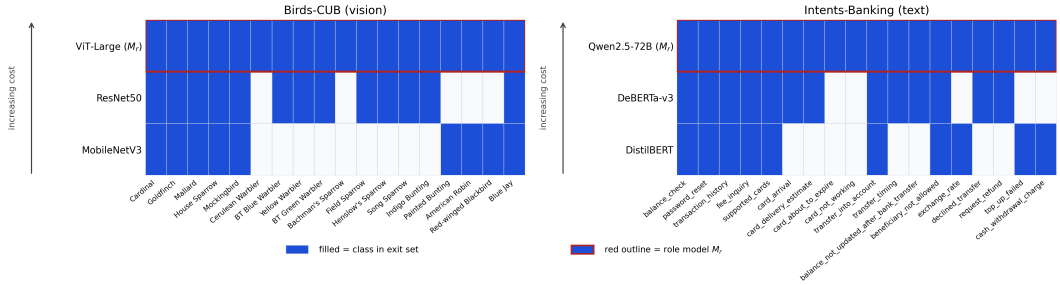


Figure 5: **NOMAD picks a cheap model for classes that more expensive models cannot reliably handle.** Exit sets on Birds-CUB and Intents-Banking at $\epsilon = 0.10$. Rows are models ordered by cost, with M_r on top and the cheapest model at the bottom, and columns are classes. A filled cell marks a class on which the model's per-class F1 is at least $(1-\epsilon)F_1^{\mathrm{RM}}$. On both datasets some classes are in a cheap model's exit set but in no intermediate model's, and NOMAD routes those classes to the cheap model directly.

chain. Greedy selection orders the models at an expected cost at most four times that of the best fixed ordering under deterministic exits. Across 16 datasets, NOMAD reaches a geometric-mean speedup of $4\times$ and up to $42\times$ on Pendigits with no per-class violations. The ablations show that the per-class check and dynamic ordering are both needed (Tab. 2).

Limitations and complementary directions.

Distribution shift. Like methods that calibrate per-stage statistics on a validation set [50, 21, 22], NOMAD's exit classes assume D_{val} matches deployment. The NOMAD-BA extension sketches Thompson-sampling-based recalibration under drift (Appendix K, preliminary convergence in Fig. A.15); finite-horizon regret bounds remain open.

Beyond greedy selection. The 4-approximation compares against the best fixed ordering and assumes deterministic exits. Learned routing policies, such as contextual bandits or policy gradients over chain prefixes [40, 9], could complement the greedy rule and tighten the bound under stochastic or non-stationary regimes.

Classifier-set composition. When each cheap classifier is about equally accurate across classes (Letter, Satimage in our suite), no cheap chain passes the per-class check and NOMAD degrades to $\approx 1\times$ speedup, safe but unhelpful (Tab. A.15). Specialization-aware classifier-set search would broaden the cost-saving regime. Tightening the chain-safety bound with the measured correlations and allowing per-class tolerances ϵ_j are discussed in Appendix J.2.

Acknowledgments and Disclosure of Funding

The research reported herein was supported by Army Research Office (ARO) award W911NF-24-2-0114. The views and opinions presented herein are those of the authors and do not necessarily represent the views of ARO or its components. Ashwin was also supported in part by HPI@UCI and NSF grant 2536802. The authors declare no competing interests.

References

[1] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2022.

[2] Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *ICLR*, 2024.

[3] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones, 2013.

[4] Yehudit Aperstein and Alexander Apartsin. CalexNet: Soft cascade-aligned training and calibration for lightweight early-exit branches. *arXiv preprint arXiv:2509.08318*, 2025.

[5] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *ICML*, pages 527–536, 2017.

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[7] Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020.

[8] Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research (TMLR)*, 2024.

[9] Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs. In *ICML*, 2025.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[11] Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *ICLR*, 2024.

[12] Tiffany Ding, Anastasios N. Angelopoulos, Stephen Bates, Michael I. Jordan, and Ryan J. Tibshirani. Class-conditional conformal prediction with many classes. In *NeurIPS*, 2023.

[13] Dheeru Dua and Casey Graff. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml, 2019.

[14] J. D. Esary, F. Proschan, and D. W. Walkup. Association of random variables, with applications. *The Annals of Mathematical Statistics*, 38(5):1466–1474, 1967.

[15] Uriel Feige, László Lovász, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.

[16] Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *ICML*, 2019.

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[18] Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. *arXiv preprint arXiv:2404.10136*, 2024.

[19] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021.

[20] Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice? In *NeurIPS*, 2023.

[21] Jaehun Jung, Faeze Brahman, and Yejin Choi. Trust or escalate: LLM judges with provable guarantees for human agreement. In *ICLR*, 2025.

[22] Steven Kolawole, Don Dennis, Ameet Talwalkar, and Virginia Smith. Agreement-based cascading for efficient inference. *Transactions on Machine Learning Research (TMLR)*, 2025.

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[24] Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *EMNLP-IJCNLP*, 2019.

[25] E. L. Lehmann. Some concepts of dependence. *The Annals of Mathematical Statistics*, 37(5): 1137–1153, 1966.

[26] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[27] Lei Li, Yankai Lin, Deli Chen, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. CascadeBERT: Accelerating inference of pre-trained language models via calibrated complete models cascade. In *Findings of EMNLP*, 2021.

[28] Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In *NAACL*, pages 1964–1974, 2024.

[29] Aman Madaan, Pranjal Aggarwal, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Mausam, and Manaal Faruqui. AutoMix: Automatically mixing language models. In *NeurIPS*, 2024.

[30] David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *NeurIPS*, 2018.

[31] Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796, 2014.

[32] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Two-stage learning to defer with multiple experts. In *NeurIPS*, 2023.

[33] Nour Moustafa and Jill Slay. UNSW-NB15: A comprehensive data set for network intrusion detection systems, 2015.

[34] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *ICML*, pages 7076–7087, 2020.

[35] Kamesh Munagala, Shivnath Babu, Rajeev Motwani, and Jennifer Widom. The pipelined set cover problem. In *International Conference on Database Theory (ICDT)*, pages 83–98, 2005.

[36] Harikrishna Narasimhan, Aditya Krishna Menon, Wittawat Jitkrittum, Neha Gupta, and Sanjiv Kumar. Learning to reject meets long-tail learning. In *International Conference on Learning Representations*, 2024.

[37] Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Seungyeon Kim, Neha Gupta, Aditya Krishna Menon, and Sanjiv Kumar. Faster cascades via speculative decoding. In *ICLR*, 2025.

[38] Quang H. Nguyen, Thinh Dao, Duy C. Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V. Chawla, and Khoa D. Doan. MetaLLM: A high-performant and cost-efficient dynamic framework for wrapping LLMs. *arXiv preprint arXiv:2407.10834*, 2024.

[39] Lunyiu Nie, Zhimin Ding, Erdong Hu, Christopher Jermaine, and Swarat Chaudhuri. Online cascade learning for efficient inference over streams. In *ICML*, 2024.

[40] 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*, 2025.

[41] Stephan Rabanser, Nathalie Rauschmayr, Achin Kulshrestha, Petra Poklukar, Wittawat Jitkrittum, Sean Augenstein, Congchao Wang, and Federico Tombari. Gatekeeper: Improving model cascades through confidence tuning. *arXiv preprint arXiv:2502.19335*, 2025.

[42] Marija Šakota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *WSDM*, 2024.

[43] Tal Schuster, Adam Fisch, Jai Prakash Gupta, Mostafa Dehghani, Dara Bahri, Vinh Quoc Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *NeurIPS*, 2022.

[44] Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. In *COLM*, 2024.

[45] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009.

[46] Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2016.

[47] Kirill Trapeznikov and Venkatesh Saligrama. Supervised sequential classification under budget constraints. In *AISTATS*, pages 581–589, 2013.

[48] Antonios Valkanas, Soumyasundar Pal, Pavel Rumiantsev, Yingxue Zhang, and Mark Coates. C3PO: Optimized large language model cascades with probabilistic cost constraints for reasoning. In *NeurIPS*, 2025.

[49] Neeraj Varshney and Chitta Baral. Model cascading: Towards jointly improving efficiency and accuracy of NLP systems. In *EMNLP*, pages 11007–11021, 2022.

[50] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[51] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[52] Xin Wang, Yujia Luo, Daniel Crankshaw, Alexei Tumanov, Fisher Yu, and Joseph E. Gonzalez. IDK cascades: Fast deep learning by learning not to overthink. In *UAI*, 2018.

[53] Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

[54] Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *ICLR*, 2024.

[55] Michael J. Zellinger, Rex Liu, and Matt Thomson. Cost-saving LLM cascades with early abstention. *arXiv preprint arXiv:2502.09054*, 2025.

[56] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, 2015.

[57] Xuechen Zhang, Zijian Huang, Ege Onur Taga, Carlee Joe-Wong, Samet Oymak, and Jiasi Chen. TREACLE: Thrifty reasoning via context-aware LLM and prompt selection. In *NeurIPS*, 2024.

Appendix Outline

Eleven sections, listed below as a table of contents. Each section extends a specific main-paper element; each subsection states its own extension target in its title so a reader landing mid-appendix can immediately backtrack.

- **Appendix A: Theoretical Guarantees** *(extends Sec. 3)*
  - Appendix A.1: Near-optimal cost via WMSSC reduction
  - Appendix A.2: Validation-set estimation and Wilson smoothing
  - Fig. A.1: Confidence Trapeze (the per-class quality contract, visualized)
- **Appendix B: Proofs** *(extends Theorems 1 and 2)*
  - Appendix B.1: Chain safety
  - Proof of Theorem 2 (4-approximation)
  - Proof of Theorem 4 (stochastic-cost bound)
  - Wilson-score lower bound for exit-class admission
- **Appendix C: Implementation Details** *(extends Secs. 3.3 and 4.1)*
  - Per-input algorithm pseudocode (Alg. 1)
  - Hardware and cost model
  - Hyperparameters (Tab. A.1)
- **Appendix D: Datasets, Metrics, and Baselines** *(extends Sec. 4.1)*
  - Fig. A.2: Dataset universe (16-bubble overview by modality, K, speedup)
  - Full dataset and metric listings (Tabs. A.2 and A.3)
  - Full baseline listing (Tab. A.4)
- **Appendix E: Per-Metric Headline Tables** *(extends Tab. 1)*
  - Per-metric tables across 16 datasets (Tabs. A.5 to A.7)
- **Appendix F: Per-Dataset Results** *(extends Tab. 1 and Fig. 3)*
  - Per-dataset offline results
  - Per-fold standard deviations (Tab. A.8)
  - Appendix F.3: Per-input cascade trace examples – five flagship datasets (Birds-CUB, Intents-Banking77, Intrusions-NSL, Foods-Food101, MMLU), each as a multi-row trajectory grid (Figs. A.6 to A.10) showing how five inputs (one per behavior pattern) traverse the dataset's portfolio; Tab. A.9 summarises the NOMAD vs Cascade-Routing trade-off on the three main-paper featured datasets
- **Appendix G: Robustness and Sensitivity** *(extends Sec. 4.1 and Tab. 2)*
  - Sensitivity to ϵ (Tab. A.10, Fig. A.11)
  - Class-conditional conformal baseline (Tab. A.11)
  - Static-order ablation per-dataset
- **Appendix H: Mechanism Visualizations** *(extends Fig. 5)*
  - Per-class cascade-depth trace
  - UCI-HAR exit-class matrix and depth trace
- **Appendix I: Greedy Selection Analyses** *(extends Theorem 2)*
  - MDP vs greedy selection
  - Empirical approximation ratio
- **Appendix J: Chain Safety Analyses** *(extends Sec. 3.2)*
  - Safety-check fallback rate (Tab. A.15)
  - Empirical support for the positive-association assumption (Appendix J.2)
- **Appendix K: NOMAD-BA Online Extension** *(extends Sec. 5 future work; relaxes Sec. 3's validation-set assumption)*

Code and supplementary materials. The companion code release (training scripts, configs, classifier checkpoints, 5-fold split indices) and the interactive results site (`https://nomad-cascades.surge.sh`) are described in Sec. 4.1 and accompany this paper.

A Theoretical Guarantees

This section supplements Sec. 3.3, which states the greedy WMSSC reduction informally, and Sec. 3.2, which describes how π and Q are estimated. It covers two topics. First, greedy utility-based selection is provably near-optimal for cost (§A.1). Second, we describe how the chain-safety check is estimated from finite validation data, and which part of that estimation is heuristic (§A.2). Full proofs are in Appendix B.

Notation. The appendix writes the class-label set as $\mathcal{C} = \{1, \ldots, K\}$, so that C_j is class j, and the class prior as $\text{Prob}(C_j) = p_j$. All other symbols (M_i, EC, cost, Q, π) are as defined in Sec. 3.

A.1 Near-Optimal Cost via WMSSC Reduction (extending Sec. 3.3, Theorem 2)

Why does the greedy utility score (Eq. 5) produce good cascade orderings? We show it is equivalent to a classical algorithm with known approximation guarantees, by reducing the cascade cost minimization problem to *Weighted Min-Sum Set Cover* (WMSSC) [15, 35].

Deterministic exit model. We write a chain of L models as $\mathcal{S} = (M_{\sigma(1)}, \ldots, M_{\sigma(L)})$, where $\sigma(\ell)$ is the index of the model at position ℓ. For the analysis, we consider a simplified model where an event of class C_j exits deterministically at the first model in the chain whose exit classes contain C_j: i.e., it exits at position $\ell^*(j) = \min\{\ell : C_j \in \text{EC}(M_{\sigma(\ell)})\}$. Under this model, the expected per-event cost under chain $\mathcal{S} = (M_{\sigma(1)}, \ldots, M_{\sigma(L)})$ is:

$$\mathbb{E}[\text{cost}(\mathcal{S})] = \sum_{j=1}^{K} \text{Prob}(C_j) \sum_{\ell=1}^{\ell^*(j)} \text{cost}(M_{\sigma(\ell)}) \tag{8}$$

Each class C_j incurs the cumulative cost of all models it must traverse before reaching its exit model. This is exactly the WMSSC objective with universe $\mathcal{C}$, element weights $w(C_j) = \text{Prob}(C_j)$, sets $S_i = \text{EC}(M_i)$, and set costs $c_i = \text{cost}(M_i)$.

The reduction works because the role model fallback ensures every class is eventually covered ($\text{EC}(M_r) = \mathcal{C}$), so the WMSSC instance always has a feasible solution. NOMAD's utility maximization (selecting $\arg\max_i U(M_i)$) is equivalent to the standard WMSSC greedy algorithm (selecting the set minimizing cost per unit of uncovered weight), since maximizing exit-probability-per-cost is the reciprocal of minimizing cost-per-uncovered-weight. The zero-and-renormalize step of NOMAD's belief update (Eq. (7)) zeroes out probabilities for exited classes, corresponding exactly to removing covered elements in WMSSC. The analysis uses this step alone, since under the deterministic exit model a non-exit reveals only that the class lies outside the exit set. The following theorem then follows from the greedy analysis of Feige et al. [15], which Munagala et al. [35] extend to sets with costs:

Theorem 2 (4-Approximation)**.** *Under the deterministic exit model, the greedy strategy $\mathcal{S}^G$ of Lemma 6 satisfies $\mathbb{E}[\text{cost}(\mathcal{S}^G)] \leq 4 \cdot \mathbb{E}[\text{cost}(\mathcal{S}^*)]$, where $\mathcal{S}^*$ is the cost-optimal fixed chain over the same models.*

When the chain-safety check discards a model, the greedy rule skips it without running it, so the residual weights do not change. Each model that is run was the utility maximizer among all untried models, and hence among the untried models that are eventually run. The run models therefore appear in WMSSC greedy order, and Theorem 2 holds with $\mathcal{S}^*$ the cost-optimal fixed chain over the models in the greedy order.

The formal reduction and greedy equivalence are stated as Lemmas 5 and 6 in Appendix B. The constant 4 is tight for WMSSC in general; on the three datasets where brute-force enumeration over classifier orderings is tractable, NOMAD achieves cost ratios in $[1.07, 1.17]$ (Tab. A.14), well below the worst case but evaluated on a deliberately narrow regime.

MDP dominance. Greedy selection looks one step ahead. An adaptive policy could instead plan the whole sequence of calls for each input, and we formalize such policies as a Markov decision process (MDP). The MDP uses NOMAD's exit classes and belief. Its state is $s = (E, k, h)$, where E is the sequence of models already executed, which the chain-safety check needs because it depends on their order, k is the predicted class, and h is a discretized belief-entropy bucket. Each action runs one model not in E, subject to the chain-safety check, and costs that model's inference cost. We solve the

MDP by value iteration, with the entropy-based exit calibrated on D_{val}. Greedy is one feasible policy of this MDP, so the optimal policy can only improve on it:

Theorem 3 (MDP Dominance)**.** *Let ϕ^* be the optimal MDP policy with chain-safety action constraints, and ϕ^G the greedy policy of Lemma 6 with models that fail the chain-safety check skipped. Both satisfy the chain-safety check at every step, by construction. Their expected costs satisfy $\mathbb{E}[\text{cost}(\phi^*)] \leq \mathbb{E}[\text{cost}(\phi^G)] \leq 4 \cdot \mathbb{E}[\text{cost}(\mathcal{S}^*)]$, where $\mathcal{S}^*$ is the cost-optimal fixed chain over the models in the greedy order, under the deterministic exit model.*

The first inequality holds because greedy is a feasible (myopic) MDP policy; the second is Theorem 2. Tab. A.13 compares the two policies on five datasets where the MDP is tractable. The inequality still leaves room for non-obvious orderings to help when belief updates are informative.

Stochastic extension. In practice, exit decisions are stochastic: an event of class C_j may prematurely exit at a model where $C_j \notin \text{EC}$, or may fail to exit even when $C_j \in \text{EC}$ (if the model makes an error on this particular instance). We bound the gap between stochastic and deterministic costs:

Theorem 4 (Stochastic Cost Bound)**.** *Let $\mu(M_i, C_j)$ be the probability that, on an input of class C_j, the exit decision of M_i differs from the deterministic exit model, that is, $1 - \pi(M_i, j)$ if $C_j \notin \text{EC}(M_i)$ and the probability that M_i predicts a class outside $\text{EC}(M_i)$ if $C_j \in \text{EC}(M_i)$. Let $\bar{c}_\ell(\mathcal{S})$ be the total cost of the stages after position ℓ, including the role-model fallback. Then*

$$|\mathbb{E}[\text{cost}_{stoch}(\mathcal{S})] - \mathbb{E}[\text{cost}_{det}(\mathcal{S})]| \leq \sum_{j=1}^{K} \text{Prob}(C_j) \sum_{\ell} \mu(M_{\sigma(\ell)}, C_j) \cdot \bar{c}_\ell(\mathcal{S}).$$

When models are accurate on their exit classes (small off-diagonal confusion mass), as enforced by chain safety which rejects models with high misclassification rates, the stochastic gap is small and the 4-approximation remains a useful bound.

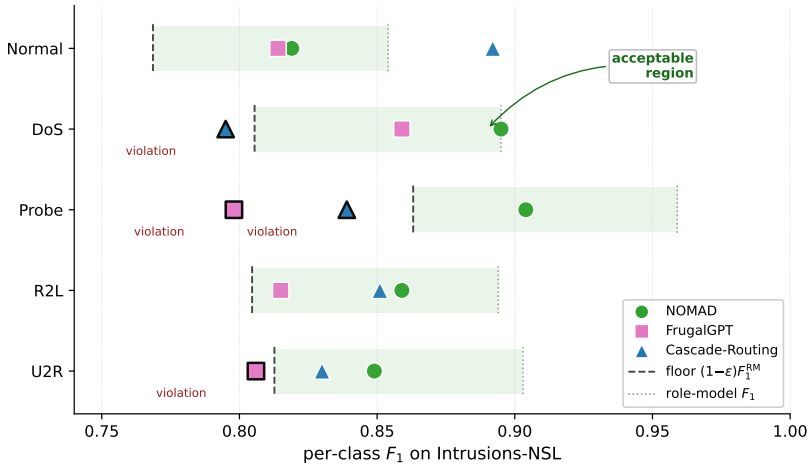


Figure A.1: **The per-class quality contract, visualized.** For each class on Intrusions-NSL the green band marks the acceptable region: per-class F1 lies between the floor $(1-\epsilon)F_1^{\text{RM}}$ (dashed) and the role-model F1 (dotted right edge). Markers show NOMAD, FrugalGPT, and Cascade-Routing per-class F1 at $\epsilon = 0.10$; markers below the floor are violations. NOMAD admits a chain only if the projected per-class F1 (Eq. (3)) sits at or above the floor for every class (Eq. (4) and Theorem 1). FrugalGPT violates two classes (Probe, U2R); Cascade-Routing violates two classes (DoS, Probe).

A.2 Validation-Set Estimates and Smoothing for Sparse CM Cells (extending Sec. 3.2 estimation paragraph)

Chain safety (Sec. 3.2) plugs in passthrough $\pi(M_i, j)$ and per-class F1 $Q(M_i, j)$ estimated from each classifier's confusion matrix on D_{val}. Rare classes are the difficult case, because their CM cells hold few samples and their point estimates are noisy.

Smoothing. NOMAD applies one interval-based rule to every class. A class is added to a cheaper model's exit set exactly when the Wilson-score lower bound on its per-class F1 clears the floor, with a Bonferroni correction across the $n \cdot K$ admission decisions (Appendix B.4). When the number N_j of class-j samples is small, the bound sits well below the point estimate, so an underpowered class is filtered out of cheap-model exit sets. As N_j grows, the bound tightens to the point estimate. The smoothed values decide exit-set membership, and the chain-safety check (4) then uses the empirical estimates of π and Q. A formal PAC-style passthrough+F1 LCB construction with explicit finite-sample rates (Hoeffding plus bootstrap, or Clopper-Pearson) is a future direction; the smoothing-based approach is what we report empirically and what gives zero per-class violations across 16 datasets at $\epsilon = 0.10$.

B Proofs

B.1 Proof of Theorem 1 (Chain Safety)

Fix class j, let $M_e = M_{\text{exit}}(j)$ be the first classifier in $\mathcal{S}$ with $j \in \text{EC}(M_e)$, at chain position e, and let $a_j = \prod_{\ell<e} \pi(M_\ell, j)$. Write $p_j = \Pr[y = j]$ and, for M_e run alone on all inputs, $T_e = \Pr[M_e(x) = j,\ y = j]$ and $F_e = \Pr[M_e(x) = j,\ y \neq j]$ for its true- and false-positive rates on j. Its per-class recall is T_e/p_j, and its per-class F1 is

$$Q(M_e, j) = \frac{2T_e}{2T_e + F_e + (p_j - T_e)} = \frac{2T_e}{T_e + p_j + F_e}.$$

Let T and F be the corresponding rates of the cascade's output. The realized per-class F1 is then $2T/(T + p_j + F)$, which increases in T and decreases in F.

True positives. A class-j input that every stage before M_e forwards and that M_e labels correctly leaves the cascade with the correct label, because $j \in \text{EC}(M_e)$. Given $y = j$, the events $E_1, \ldots, E_{e-1}$ have probabilities $\pi(M_\ell, j)$ and E_e has probability T_e/p_j. By hypothesis (a) and the product inequality for associated events [14],

$$T \;\geq\; p_j \Pr[E_1 \cap \cdots \cap E_e \mid y = j] \;\geq\; p_j \cdot a_j \cdot \frac{T_e}{p_j} \;=\; a_j T_e.$$

Stages after M_e can only add further correct answers.

False positives. No stage before M_e outputs j, because M_e is the first classifier whose exit set contains j. M_e makes the same prediction on an input whether or not it runs inside the cascade, and inside the cascade it sees only the inputs that earlier stages forward. Its false positives in the cascade are therefore a subset of its false positives alone, with rate at most F_e. By hypothesis (b), stages after M_e add a false-positive rate of at most $(1 - a_j)T_e$. Hence $F \leq F_e + (1 - a_j)T_e$.

Combining. Substituting both bounds,

$$\frac{2T}{T + p_j + F} \;\geq\; \frac{2a_j T_e}{a_j T_e + p_j + F_e + (1 - a_j)T_e} \;=\; a_j \cdot \frac{2T_e}{T_e + p_j + F_e} \;=\; Q_{\text{proj}}(\mathcal{S}, j).$$

The chain-safety check (4) enforces $Q_{\text{proj}}(\mathcal{S}, j) \geq (1 - \epsilon)\, Q(M_r, j)$, which proves Theorem 1. □

Proof of Corollary 1.1. Independent random variables are associated [14], which gives the first claim. When Q is per-class recall, the true-positive bound alone gives a realized recall $T/p_j \geq a_j T_e/p_j = Q_{\text{proj}}(\mathcal{S}, j)$, and false positives do not enter recall, so hypothesis (b) is not needed. With a single stage before M_e, positive association of the two events reduces to positive quadrant dependence [25]. Appendix J.2 reports the measured correlations between stages. □

Remark (false positives on class j). The two factors of Eq. (3) split the accounting for class j. The passthrough product counts the class-j inputs that reach $M_{\text{exit}}(j)$, a recall-type quantity. $Q(M_{\text{exit}}(j), j)$ is the exit model's per-class F1 and therefore already contains the false positives it commits on j, which the cascade can only reduce. No stage before $M_{\text{exit}}(j)$ can emit label j, since $M_{\text{exit}}(j)$ is the first model whose exit set contains j. The only false positives outside this accounting come from stages after $M_{\text{exit}}(j)$, which hypothesis (b) bounds.

Remark (relation to Jitkrittum et al. [20]). The connection has two parts. The first concerns what the deferral rule reads. Jitkrittum et al. [20] prove that the Bayes-optimal deferral rule depends on the

predicted label, yet their proposed rules score confidence-vector statistics. In NOMAD the predicted label itself triggers the exit, through exit sets estimated from held-out data. The second part concerns what the guarantee covers. Jitkrittum et al. [20] analyze a single deferral decision. Theorem 1 extends the label-based principle to whole chains, and Theorem 2 adds a cost guarantee for the label-driven greedy order.

B.2 Proof of Theorem 2 (4-Approximation)

The proof proceeds in two steps: a reduction to WMSSC, then showing NOMAD's greedy is equivalent to the WMSSC greedy.

Lemma 5 (Reduction to WMSSC)**.** *Under the deterministic exit model, NOMAD's cost minimization is an instance of WMSSC with universe $\mathcal{C}$, element weights $w(C_j) = \mathrm{Prob}(C_j)$, sets $S_i = \mathrm{EC}(M_i)$, and set costs $c_i = \mathrm{cost}(M_i)$.*

Proof. An event of class C_j traverses all models in the chain until the first model $M_{\sigma(\ell^*(j))}$ with $C_j \in \mathrm{EC}(M_{\sigma(\ell^*(j))})$. The cost incurred for this event is $\sum_{\ell=1}^{\ell^*(j)} \mathrm{cost}(M_{\sigma(\ell)})$, which equals $\mathrm{cover_cost}(C_j, \sigma)$ in the WMSSC formulation. Taking the expectation over the class distribution:

$$\mathbb{E}[\mathrm{cost}(\mathcal{S})] = \sum_{j=1}^{K} \mathrm{Prob}(C_j) \cdot \mathrm{cover_cost}(C_j, \sigma) = \mathrm{WMSSC}(\sigma).$$

The role model fallback ensures all elements are covered, since $\mathrm{EC}(M_r) = \mathcal{C}$. □

Lemma 6 (Greedy Equivalence)**.** *Under the deterministic exit model, with initial beliefs equal to the true class distribution and beliefs updated by Eq. (7), NOMAD's utility maximization (Eq. (5)) is equivalent to the standard WMSSC greedy algorithm.*

Proof. In the WMSSC greedy algorithm, at each step we select the set minimizing the ratio of cost to weight of newly covered elements:

$$S^* = \arg\min_{S_i} \frac{c_i}{w(S_i \setminus \mathrm{covered})}.$$

In NOMAD's formulation, $c_i = \mathrm{cost}(M_i)$ and $w(S_i \setminus \mathrm{covered}) = \sum_{C_j \in \mathrm{EC}(M_i) \setminus \mathrm{exited}} \mathrm{Prob}(C_j)$. Maximizing utility $U(M_i) = [\sum_{C_j \in \mathrm{EC}(M_i)} \mathrm{Prob}_{\text{current}}(C_j)]/\mathrm{cost}(M_i)$ over remaining classes is equivalent to minimizing the cost-effectiveness ratio. The update of Eq. (7) zeroes out probabilities for exited classes, corresponding exactly to removing covered elements, and renormalization scales every utility by the same constant, which leaves the argmax unchanged. □

By Lemmas 5 and 6, NOMAD's greedy strategy is an instance of the WMSSC greedy algorithm. The 4-approximation follows from the greedy analysis of Feige et al. [15], extended to sets with costs by Munagala et al. [35]. □

B.3 Proof of Theorem 4 (Stochastic Cost Bound)

Run the stochastic and deterministic processes on the same class-C_j input. They pay identical costs up to and including the first stage ℓ at which the stochastic exit decision differs from the deterministic one, and this stage is at position ℓ with probability at most $\mu(M_{\sigma(\ell)}, C_j)$. After it, one process stops and the other pays at most the cost $\bar{c}_\ell(\mathcal{S})$ of the remaining stages, so the two costs differ by at most $\bar{c}_\ell(\mathcal{S})$. A union bound over ℓ and an expectation over classes yield the bound. □

B.4 Wilson-Score Lower Bound for Exit-Class Admission (extending Sec. 3.2 estimation paragraph)

For sparse CM cells (rare classes with few class-j validation samples), the empirical F1 has high variance and can pass the exit-class threshold by luck. Our implementation applies the Wilson-score formula [53] to the empirical F1 with N_j as the effective sample size and Bonferroni correction across the $n \cdot K$ admission decisions, admitting class j to $\mathrm{EC}(M_i)$ exactly when the resulting lower

bound is at least $(1-\epsilon)\,F_1^{\mathrm{RM}}(j)$. Strictly, the Wilson interval is defined for binomial proportions and F1 is not a proportion, so this construction is a heuristic without a formal coverage guarantee: it pulls estimates downward when N_j is small and recovers the point estimate as N_j grows, but its calibration is empirical rather than provable. A principled per-class F1 lower bound (e.g., separate Wilson intervals on per-class precision and recall composed via the F1 functional, or a bootstrap LCB) is left to future work; the heuristic is empirically conservative – across 16 datasets at $\epsilon = 0.10$, NOMAD records zero per-class violations. The smoothed values decide exit-set membership, and the chain-safety check (4) then uses the empirical estimates of π and Q (Sec. 3.2). Because the construction has no formal coverage guarantee, Theorem 1 does not extend to finite samples through it; the zero violations we report are measured on held-out test data.

C Implementation Details

This section supplements Sec. 4.1 with the per-input algorithm, hardware and cost model, and full dataset, metric, and baseline listings.

C.1 Per-Input Algorithm (extending Sec. 3.3)

Algorithm 1: NOMAD per-input processing

Function *ProcessEvent*$(x, \{M_i\}, M_r, p_{init}, \epsilon)$
  $\mathcal{M}_{\text{avail}} \leftarrow \{M_1, \ldots, M_n\}$;;
  $p_{\text{cur}} \leftarrow p_{\text{init}}$;;
  $\mathcal{S}_{\text{cur}} \leftarrow \emptyset$;
  while $\mathcal{M}_{avail} \neq \emptyset$ **do**
    $M_{\text{sel}} \leftarrow \arg\max_{M_i \in \mathcal{M}_{\text{avail}}} U(M_i)$; `// greedy utility`
    if *CheckChainSafety*$(M_{sel}, \mathcal{S}_{cur}, M_r, \epsilon)$ **then**
      $(C_{\text{pred}}, \mathbf{s}) \leftarrow \text{Execute}(M_{\text{sel}}, x)$;
      $\mathcal{S}_{\text{cur}} \leftarrow \mathcal{S}_{\text{cur}} \circ (M_{\text{sel}})$;
      $p_{\text{cur}} \leftarrow \text{UpdateBelief}(p_{\text{cur}}, \mathbf{s}, \text{EC}(M_{\text{sel}}))$;
      if $C_{pred} \in \text{EC}(M_{sel})$ **then return** C_{pred};
    $\mathcal{M}_{\text{avail}} \leftarrow \mathcal{M}_{\text{avail}} \setminus \{M_{\text{sel}}\}$;
  return *Execute*(M_r, x); `// fallback`

C.2 Hardware and Cost Model (extending Sec. 4.1)

Hardware. Intel i7-12th Gen, 64 GB RAM, Ubuntu 22.04 LTS. All non-LLM inference is single-threaded and sequential; reported wall-clock times include the forward pass through a loaded model but exclude one-time model-loading cost (models are loaded once into memory at the start of evaluation). LLM inference uses vLLM with its default scheduler, batch size 32, and tensor parallelism degree matched to the model size (1 for 7B/8B, 2 for 32B, 4 for 70B/72B); reported costs are per-input under steady-state throughput.

Model cost tiers. Simple models (LDA, CART): 2 to 10 ms; medium (RF, XGB): 10 to 30 ms; deep learning: 30 to 100 ms; ensembles: above 100 ms per inference. LLM inference costs grow with parameter count, from $\sim$ 40 ms per classified input for 7B to $\sim$ 640 ms for 72B under our batch/GPU configuration, a $16\times$ range across the five LLMs.

C.3 Hyperparameters (extending Sec. 4.1)

This subsection supplements Sec. 4.1. Tab. A.1 lists every hyperparameter used to produce the results in this paper, organized by component (NOMAD core, base-classifier training per modality, evaluation protocol). Values are taken directly from the released code; we use a single configuration across all 16 datasets except where the value depends on K or the modality.

Table A.1: **All hyperparameters used in this paper.** Values fixed across all 16 datasets unless noted. Sourced from the released code (Sec. 4.1); "–" = sklearn/torchvision default.

Hyperparameter	Value
NOMAD core	
Quality tolerance ϵ (default)	0.10
Quality tolerance ϵ (sweep, Tab. A.10)	$\{0.05, 0.10, 0.15, 0.20\}$
Wilson-score family-wise confidence δ	0.05
Bonferroni correction count	$n \cdot K$ (per-classifier $\times$ per-class admissions)
Belief initialization $p_{\text{cur}}^{(0)}$	empirical class prior on D_{val}
Belief update (Eqs. (6) and (7))	soft Bayes + zero-and-renormalize
Tabular base classifiers (sklearn / xgboost)	
Logistic Regression	`max_iter`$= 1000$
Decision Tree	default
k-Nearest Neighbors	$k = 5$
Random Forest	`n_estimators`$= 100$
XGBoost	`n_estimators`$= 100$, `eval_metric=mlogloss`
MLP variants	layers $\in \{(64), (128, 64), (256, 128, 64)\}$, `max_iter`$= 200$, early stopping
Vision base classifiers (PyTorch / torchvision)	
Backbones (cheap $\rightarrow$ expensive)	MobileNetV2/V3, ResNet18/50, EfficientNet-B0/B4, ViT-B/16, ViT-Large
Optimizer	Adam
Learning rate	1e$-$3
Batch size	64
Fine-tune epochs	10
Input normalization	ImageNet mean/std
Text base classifiers (HuggingFace transformers)	
Backbones (cheap $\rightarrow$ expensive)	DistilBERT, BERT-base, RoBERTa-base, DeBERTa-v3-base/large
Optimizer	AdamW
Learning rate	2e$-$5
Batch size	32
Fine-tune epochs	5
Max sequence length	128 tokens
LLM base classifiers (vLLM-served, used for Intents-Banking, CLINC150, FPB, MMLU)	
Models (cheap $\rightarrow$ expensive)	Qwen2.5-7B, LLaMA-3.1-8B, Qwen2.5-32B, LLaMA-3.1-70B, Qwen2.5-72B
vLLM batch size	32
Tensor parallelism	1 ($\leq$8B) / 2 (32B) / 4 ($\geq$70B)
Decoding	greedy (temperature 0)
Evaluation protocol	
Pre-split (stratified, fixed across folds)	90% CV pool / 10% held-out test
Inner cross-validation (stratified)	5-fold $\rightarrow$ 80% train / 20% val per fold
Random seed	42
Cost-measurement sample size N_{cost}	500 inputs

D Datasets, Metrics, and Baselines

D.1 Full Dataset and Metric Listings (extending Sec. 4.1)

Tab. A.2 lists all 16 datasets evaluated in this paper, organized by modality (a compact one-glance overview is in Fig. A.2). Bal/Imb is by class-distribution Gini at threshold 0.25. Tab. A.3 extends the F1 / Speedup / Viol% triple used in Tab. 1 with the three additional metrics (worst-class F1, violation severity, geometric-mean speedup) referenced in main-paper prose.

D.2 Full Baseline Listing (extending Sec. 4.1)

Tab. A.4 lists the full set of comparison methods. All baselines are evaluated on the same 5-fold CV splits as NOMAD; Online Cascade is adapted to the offline regime by training on validation data rather than a live stream.

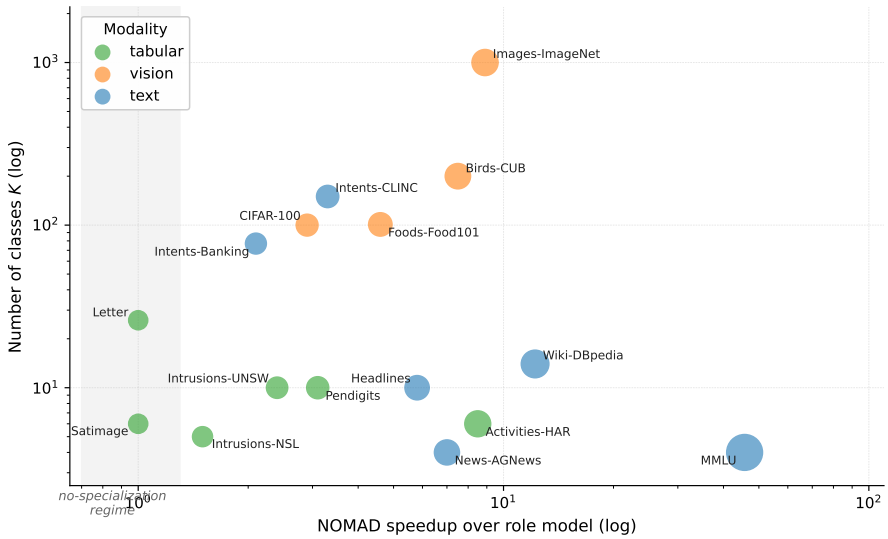


Figure A.2: **Dataset universe.** 16 evaluation datasets positioned by NOMAD speedup over the role model (x, log) and number of classes K (y, log). Bubble area is proportional to speedup; color encodes modality. The shaded band on the left marks the no-specialization regime named in Sec. 5 (Letter, Satimage): when every classifier is roughly equally accurate on every class, no cheap chain passes the per-class check and NOMAD reduces to $\approx 1\times$. Numbers match Tabs. A.2 and A.6.

Table A.2: **16 datasets across 3 modalities, with K ranging from 3 to 1000 classes.** Featured in Tab. 1 and Fig. 3: starred ($\star$). Bal/Imb is by class-distribution Gini at threshold 0.25.

Dataset	K	N	Bal
Tabular			
UCI-HAR [3]	6	10K	BAL
Pendigits [13]	10	11K	BAL
Letter [13]	26	20K	BAL
Satimage [13]	6	6K	BAL
Intrusions-NSL (NSL-KDD) [45] $\star$	5	150K	IMB
UNSW-NB15 [33]	10	250K	IMB
Vision			
CIFAR-100 [23]	100	60K	BAL
ImageNet [10]	1000	1.28M	BAL
Food-101 [6]	101	101K	BAL
Birds-CUB (CUB-200) [51] $\star$	200	12K	IMB
Text			
AG-News [56]	4	120K	BAL
DBpedia [26]	14	630K	BAL
Intents-Banking (BANKING77) [7] $\star$	77	13K	IMB
CLINC150 [24]	150	23K	IMB
FPB [31]	3	5K	IMB
MMLU [19]	4	16K	BAL

E Per-Metric Headline Tables

This section supplements Tab. 1 (which reports the three featured datasets) with full per-dataset, per-method results across all 16 datasets, plus per-fold variance, ϵ-sensitivity, the class-conditional conformal baseline, and the static-order ablation.

E.1 Per-Metric Results Across All Datasets (extending Tab. 1)

This subsection supplements Tab. 1 (which reports the three featured datasets). The tables below give per-method, per-dataset breakdowns across all 16 datasets, one table per metric: F1 (Tab. A.5),

Table A.3: **Six metrics in total: F1 / Speedup / Viol% in Tab. 1, plus three further metrics referenced in main-paper prose and the appendix.** Per-class quality is defined on Viol%; F1 averages over classes and so cannot detect a per-class violation on its own.

Metric		Definition
F1	↑	mean per-class F1
Speedup	↑	$\text{cost}(M_r)/\text{cost}$
Viol%	↓	percentage of K classes with $F_1 < (1-\epsilon)F_1^{\text{RM}}$
Worst F1	↑	$\min_j F_1(j)$ (per-class fairness floor)
Severity	↓	mean gap $\left[(1-\epsilon)F_1^{\text{RM}}(j) - F_1(j)\right]_+$
GeoSpd	↑	geometric mean of Speedup across datasets

Table A.4: **11 baselines plus 2 NOMAD ablations and 2 reference bounds.** NOMAD w/o per-class and NOMAD w/o dynamic are ablations differing from NOMAD only in safety scope or selection policy, respectively. Val. labels marks the methods fit on the labeled validation split, which NOMAD also uses. *The original C3PO is self-supervised; our adaptation calibrates on the labeled validation split like the other methods.

Method	Type	Guarantees	Training	Val. labels
Role Model Only	Upper bound	N/A	No	No
Cheapest Model Only	Lower bound	No	No	No
Cost Cascade [52]	Cascade	No	No	Yes
ABC [22]	Cascade	No	No	No
FrugalGPT [8]	Cascade	No	Yes	Yes
RouteLLM [40]	Router	No	Yes	Yes
AutoMix [29]	Router	No	Yes	Yes
C3PO [48]	Cascade	Population-level	Yes	Yes*
Cascade-Routing [9]	Cascade+Router	Cost-optimal	Yes	Yes
Online Cascade [39]	Cascade	Regret (adapted offline)	Online	Yes
Trust or Escalate [21]	Cascade	Marginal	Yes	Yes
Conformal-CC [12]	Cascade (new)	Per-class coverage	Yes	Yes
MoE (white-box)	Gating	No	Yes (joint)	Yes
NOMAD w/o per-class *[ablation]*	Cascade	Aggregate (chain safety)	No	Yes
NOMAD w/o dynamic *[ablation]*	Cascade	Per-class (chain safety)	No	Yes

speedup (Tab. A.6), and violation count (Tab. A.7). The same ten baselines and three NOMAD variants appear in each table, so the reader can scan across metrics column-by-column. NOMAD and NOMAD w/o dynamic are the only methods with zero violations on every dataset; NOMAD also recovers most of NOMAD w/o per-class's speedup on the imbalanced side without losing the per-class contract.

F Per-Dataset Results

F.1 Per-Dataset Offline Results (extending Tab. 1 and Fig. 3)

Figs. A.3 to A.5 give the full per-dataset breakdowns underlying Tab. 1. Across all 16 datasets, NOMAD holds the per-class quality threshold on every dataset and is the only method sitting on the zero-violation line across every dataset; NOMAD w/o per-class and the confidence-based baselines drop on the imbalanced side.

F.2 Per-Fold Standard Deviations (extending Tab. 1)

This subsection supplements the NOMAD row of Tab. 1. Tab. A.8 reports per-dataset mean and sample standard deviation across the 5-fold cross-validation for NOMAD, the main-paper row. F1 std is at most 4.3pp across all 16 datasets and at most 2.0pp on 15 of 16 (FPB at 4.3pp, driven by its 3-class polarity setup); violation-count std is exactly zero across all folds for all datasets (the per-class guarantee holds pointwise, not on average). Speedup varies more across folds than F1, most of all on

Table A.5: **NOMAD reaches the highest F1 on every one of 16 datasets.** Per-dataset F1 (%) at $\epsilon = 0.10$. Bold: per-column max among non-bound methods.

Method	UCI-HAR	Pendigits	Letter	Satimage	Intrusions-NSL	UNSW-NB15	CIFAR-100	ImageNet	Food-101	Birds-CUB	AG-News	DBpedia	Intents-Banking	CLINC150	FPB	MMLU
Bounds																
Role Model	95.7	99.1	97.2	91.1	90.1	87.6	93.4	86.0	92.5	83.4	95.4	98.5	92.7	95.0	84.6	84.2
Cheapest Model	84.8	95.6	86.2	82.8	75.1	69.4	74.7	69.5	73.9	68.6	75.9	71.6	75.0	75.3	67.2	59.4
Cascades & routers (10 baselines; Conformal-CC in Tab. A.11)																
FrugalGPT	85.1	89.1	85.2	79.8	81.8	79.1	84.4	78.1	81.5	72.4	87.4	88.7	83.9	86.1	74.0	76.0
Cost Cascade	87.4	89.9	90.3	82.1	80.3	81.1	84.8	76.1	81.5	81.8	84.4	89.3	85.0	87.6	77.4	76.6
ABC	89.5	94.7	91.4	86.5	83.8	83.6	86.6	80.1	87.2	76.1	89.7	93.0	87.8	87.7	78.2	77.7
Trust or Escalate	89.2	93.4	89.5	83.5	85.3	82.7	84.8	80.5	84.4	82.7	90.2	89.8	86.9	87.8	78.0	77.4
RouteLLM	83.7	85.9	86.7	82.4	78.9	76.6	80.1	73.9	83.6	78.8	83.5	84.2	80.2	83.9	73.6	75.2
AutoMix	88.7	91.6	86.9	82.3	84.0	80.0	86.3	80.0	83.7	79.7	88.9	92.2	86.3	89.4	78.1	77.3
C3PO	89.0	94.9	93.3	87.4	85.8	82.4	86.5	80.5	86.7	78.1	88.5	94.4	86.0	89.8	78.5	78.0
Cascade-Routing	88.4	88.3	87.9	83.3	84.2	78.9	83.9	79.6	82.4	79.7	89.0	89.2	86.4	85.4	77.7	76.3
Online Cascade	86.4	88.6	83.8	82.2	76.8	76.9	80.5	76.1	81.6	76.4	83.6	86.3	82.7	84.9	74.7	74.9
MoE	89.7	96.9	93.4	86.7	85.9	84.4	90.7	83.0	88.4	83.8	91.1	94.1	87.9	91.2	78.7	78.6
NOMAD (ours)																
NOMAD w/o per-class	90.5	96.8	93.9	88.9	86.5	84.9	91.1	83.0	89.2	82.2	92.7	96.3	89.2	92.0	79.3	78.7
NOMAD w/o dynamic	90.2	96.2	93.3	87.5	85.9	83.8	89.8	83.4	88.5	83.1	92.5	94.8	88.8	91.5	79.0	78.5
NOMAD	**90.7**	**99.1**	**94.1**	**88.9**	**86.5**	**85.0**	**91.2**	**83.9**	**90.6**	**84.5**	**93.0**	**96.5**	**89.3**	**92.0**	**79.5**	**79.1**

Table A.6: **Baselines that reach higher raw speed than NOMAD do so by violating the per-class contract; see Tab. A.7.** Per-dataset speedup ($\times$ vs M_r) at $\epsilon = 0.10$. Bold: per-column max among non-bound methods. NOMAD w/o dynamic sub-1$\times$ on Letter and Satimage reflects the static cheapest-first order paying for failed intermediate stages on degenerate-per-class-spread datasets where dynamic NOMAD also reduces to $\approx 1\times$.

Method	UCI-HAR	Pendigits	Letter	Satimage	Intrusions-NSL	UNSW-NB15	CIFAR-100	ImageNet	Food-101	Birds-CUB	AG-News	DBpedia	Intents-Banking	CLINC150	FPB	MMLU
Bounds																
Role Model	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×	1.0×
Cheapest Model	143.9×	1279.1×	392.8×	933.4×	429.0×	428.2×	16.7×	16.8×	16.7×	16.6×	2.2×	3.6×	3.0×	3.0×	3.0×	8.3×
Cascades & routers (10 baselines; Conformal-CC in Tab. A.11)																
FrugalGPT	5.2×	54.7×	1.2×	1.2×	4.7×	4.6×	11.4×	11.8×	10.8×	**12.4**×	2.8×	4.4×	2.6×	3.1×	**2.9**×	9.3×
Cost Cascade	4.5×	57.5×	1.3×	1.3×	4.5×	5.1×	11.3×	11.4×	10.6×	10.3×	2.4×	**4.9**×	**2.7**×	3.4×	2.6×	8.8×
ABC	4.1×	46.7×	1.1×	1.1×	4.1×	4.4×	9.7×	10.8×	9.3×	12.2×	2.4×	4.0×	2.5×	2.9×	2.3×	7.3×
Trust or Escalate	4.6×	50.5×	1.2×	1.2×	3.9×	4.2×	10.8×	10.7×	9.1×	9.3×	2.5×	3.9×	2.5×	3.0×	2.5×	8.0×
RouteLLM	**5.4**×	**58.7**×	1.4×	1.3×	**5.1**×	**5.3**×	12.2×	**11.8**×	11.1×	10.9×	2.9×	4.4×	2.6×	3.3×	2.7×	8.6×
AutoMix	4.2×	50.6×	1.2×	1.2×	4.5×	4.3×	10.4×	10.0×	10.1×	8.8×	2.3×	4.2×	2.3×	2.8×	2.4×	8.4×
C3PO	4.2×	48.3×	1.1×	1.1×	3.7×	4.1×	9.2×	10.6×	8.9×	9.7×	2.3×	4.1×	2.3×	2.7×	2.4×	7.8×
Cascade-Routing	4.5×	48.4×	1.2×	1.3×	4.1×	4.6×	11.3×	10.6×	10.3×	10.6×	2.6×	4.1×	2.6×	3.0×	2.6×	8.4×
Online Cascade	5.2×	54.6×	**1.5**×	1.4×	4.9×	4.9×	**12.4**×	11.7×	**12.1**×	10.4×	**3.0**×	4.6×	2.6×	**3.6**×	2.8×	**9.5**×
MoE	3.9×	46.2×	1.1×	1.1×	4.0×	4.2×	9.1×	9.7×	9.2×	8.1×	2.3×	3.6×	2.2×	2.7×	2.4×	7.4×
NOMAD (ours)																
NOMAD w/o per-class	4.6×	49.4×	1.2×	**1.4**×	4.1×	4.3×	11.2×	10.7×	9.8×	9.1×	2.5×	4.4×	2.4×	3.1×	2.6×	8.1×
NOMAD w/o dynamic	2.9×	37.2×	0.8×	0.9×	2.3×	2.4×	7.8×	6.4×	6.2×	4.0×	1.8×	2.3×	1.4×	1.6×	1.2×	4.8×
NOMAD	3.7×	41.8×	1.0×	1.0×	3.5×	3.8×	8.6×	8.9×	8.3×	5.0×	2.0×	3.5×	2.1×	2.5×	2.1×	6.7×

Pendigits, where it ranges from 7$\times$ to 90$\times$ depending on which classes dominate the test split; the mean is 41.8$\times$, the median 30.3$\times$, and the per-class guarantee holds on every fold.

F.3 Per-Input Trace Examples (extending Figs. 2 and 3 and Sec. 4.2)

This subsection accompanies the live cascade-trace explorer linked in the appendix outline. Each figure below shows one flagship dataset; rows are five real test inputs (one per behavior pattern: easy graduation, confidence trap, mid-depth exit, rare fall-through, honest failure) and columns are the dataset's portfolio of base classifiers from cheapest (left) to role model (right), with NOMAD's verdict at the far right. Cell color encodes the action taken on that (input, classifier) pair: **green** = predicted class is in the classifier's exit set and the cascade exits; **red** = predicted class is outside the exit set and chain-safety escalates; **blue** = outside the exit set and the cascade continues without a safety rejection; *beige* = NOMAD's adaptive selector skipped this column for this input. Empty cells ('—') indicate models past the point where NOMAD already exited. The verdict column reports NOMAD's final prediction, depth, and speedup vs. the role model.

Table A.7: **NOMAD and NOMAD w/o dynamic are zero on every dataset; every other method violates somewhere.** Per-dataset Viol% at $\epsilon = 0.10$. Bold: per-column min among the 13 non-bound methods.

Method	UCI-HAR	Pendigits	Letter	Satimage	Intrusions-NSL	UNSW-NB15	CIFAR-100	ImageNet	Food-101	Birds-CUB	AG-News	DBpedia	Intents-Banking	CLINC150	FPB	MMLU
Bounds																
Role Model	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cheapest Model	76.7	0.0	57.7	33.3	76.0	88.0	94.6	93.0	93.3	94.1	95.0	97.1	87.3	93.5	86.7	100.0
Cascades & routers (10 baselines; Conformal-CC in Tab. A.11)																
FrugalGPT	70.0	48.0	70.0	50.0	52.0	54.0	46.8	45.9	61.2	65.1	40.0	42.9	48.3	46.3	60.0	50.0
Cost Cascade	33.3	28.0	24.6	40.0	60.0	34.0	45.8	59.3	62.0	43.6	55.0	37.1	45.2	38.4	46.7	35.0
ABC	**0.0**	**0.0**	**0.0**	13.3	36.0	28.0	33.6	32.2	23.0	46.1	15.0	15.7	28.8	37.1	46.7	20.0
Trust or Escalate	20.0	2.0	28.5	26.7	32.0	28.0	43.6	30.4	38.6	38.3	15.0	32.9	32.2	36.0	40.0	35.0
RouteLLM	80.0	100.0	55.4	36.7	64.0	64.0	77.4	72.9	48.1	58.7	65.0	85.7	74.5	63.7	66.7	55.0
AutoMix	26.7	8.0	51.5	36.7	48.0	44.0	33.6	32.4	45.5	53.8	20.0	15.7	35.3	28.5	40.0	50.0
C3PO	26.7	**0.0**	1.5	16.7	36.0	46.0	33.0	29.4	25.5	33.9	35.0	5.7	37.7	23.7	46.7	20.0
Cascade-Routing	26.7	68.0	42.3	26.7	40.0	56.0	51.0	35.7	55.6	27.5	30.0	40.0	35.8	50.1	26.7	40.0
Online Cascade	53.3	64.0	86.9	40.0	72.0	62.0	75.0	58.9	62.6	70.9	75.0	68.6	57.7	56.1	60.0	50.0
MoE	16.7	**0.0**	8.5	16.7	32.0	20.0	11.6	17.9	18.2	32.8	20.0	5.7	25.7	17.5	40.0	35.0
NOMAD (ours)																
NOMAD w/o per-class	16.7	2.0	9.2	20.0	32.0	24.0	11.8	16.6	13.7	22.5	10.0	2.9	23.4	15.9	46.7	30.0
NOMAD w/o dynamic	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
NOMAD	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

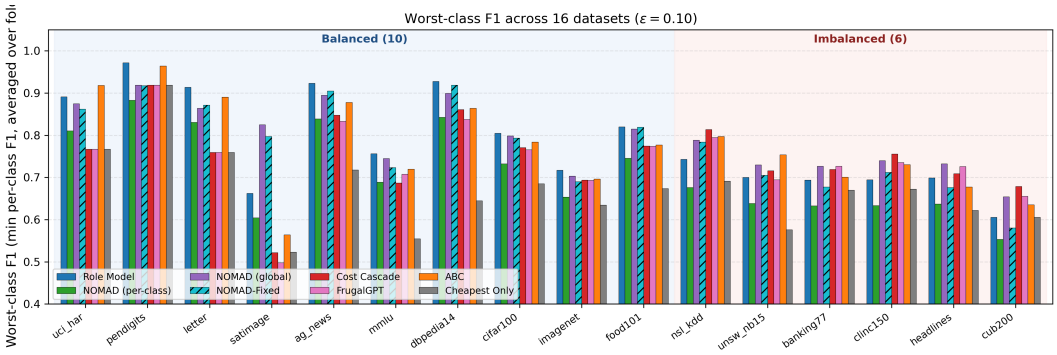


Figure A.3: **NOMAD tracks the role model on every dataset; baselines drop on imbalanced.** Worst-class F1 at $\epsilon = 0.10$ across 16 datasets, 8 methods. Datasets grouped balanced (blue) / imbalanced (red). Y-axis starts at 0.4.

Cascade-Routing comparison summary

The case studies above are concrete instances of a quantitative trend: Cascade-Routing reaches higher speedup than NOMAD on the three featured datasets but commits per-class violations on a non-trivial share of classes. Tab. A.9 summarises the trade-off; per-dataset full numbers are in Tabs. A.6 and A.7.

G Robustness and Sensitivity

G.1 Sensitivity to ϵ (extending Sec. 4.1)

Tab. A.10 backs the ϵ-sweep claim in §4.3; Fig. A.11 plots the same sweep. Across the 16 datasets, geometric-mean NOMAD speedup grows monotonically with ϵ, and every dataset retains zero per-class violations at every ϵ in the sweep. The max-speedup column reflects the Pendigits regime where Cheapest Model becomes increasingly eligible at looser tolerances.

G.2 Model Costs, Portfolio Size, and Where Inputs Resolve (extending Tab. 3)

Costs change the speedup but not the guarantee. Exit sets in Eq. (1) and the chain-safety check in Eq. (4) contain no cost term. Costs enter only the utility of Eq. (5), so they decide the order in which certified models are visited and never which chains are certified. Any change to the cost vector can

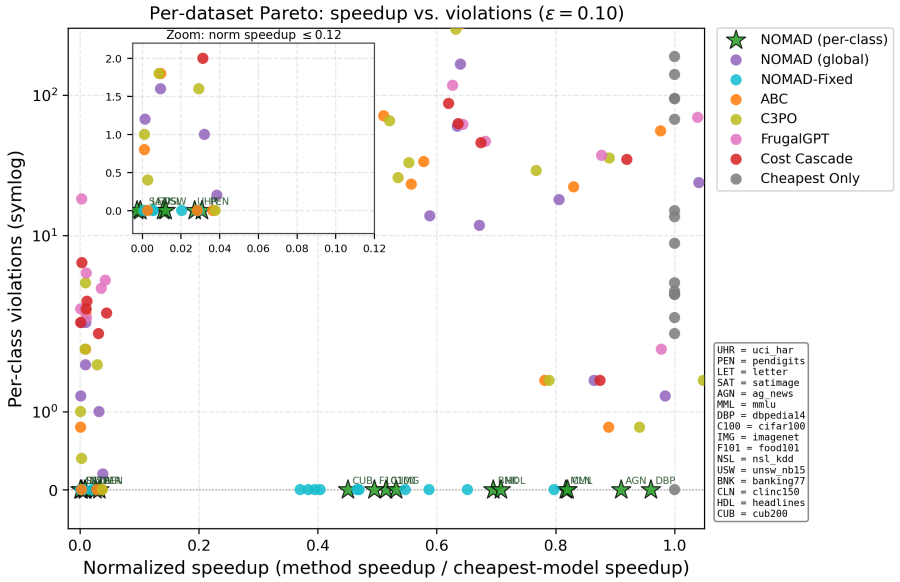


Figure A.4: **NOMAD is the only method on the zero-violation line for every dataset; baselines and NOMAD w/o per-class scatter upward on imbalanced data.** Pareto view at $\epsilon = 0.10$: normalized speedup (method/cheapest) vs per-class violations across 8 methods. Y-axis symlog (linthresh= 2).

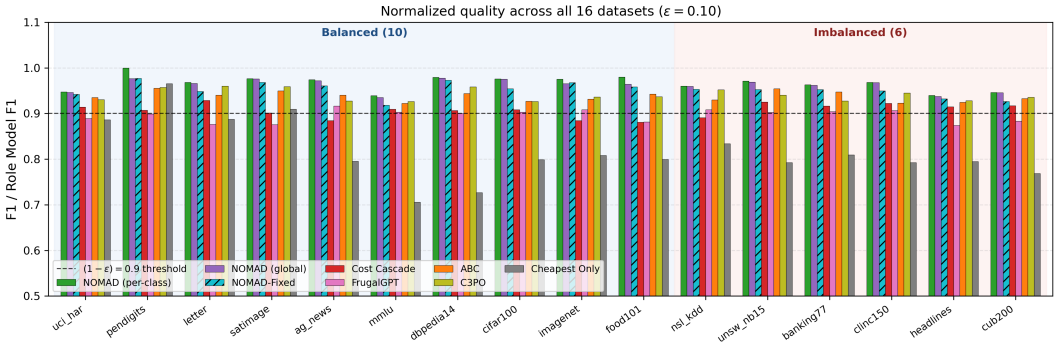


Figure A.5: **NOMAD holds the $(1 - \epsilon)$ threshold on every dataset, even where NOMAD w/o per-class fails individual classes on imbalanced data.** Normalized quality (F1 / role-model F1) at $\epsilon = 0.10$ across 16 datasets, 8 methods. Dashed line: $(1 - \epsilon) = 0.9$. NOMAD w/o per-class stays above in mean F1 but violates individual classes on imbalanced data (Fig. A.3).

therefore move the speedup, but it cannot create a per-class violation. We measure how much the speedup moves, starting from the wall-clock per-model costs of Appendix C.2, under three hardware profiles a deployment might see.

- *Uniformly faster hardware.* If every model gets faster by the same factor, every speedup is unchanged, because speedup is a ratio of costs.
- *Faster low-memory models*, for example on hardware with better caching. The cheap tier gets cheaper and speedup rises. On Intrusions-NSL, making the low-memory models three times faster gives $3.6\times$ against the measured $3.5\times$.
- *Faster GPU-parallelizable models*, with tree and nearest-neighbor models unchanged. The selector reorders the chain around the new costs. On Intrusions-NSL, a five-times-faster neural tier gives $4.2\times$.

The right block of Tab. 3 stresses the cost structure further by replacing every measured cost c with $\sqrt{c}$, which compresses the differences between models, or with c^2, which amplifies them. On

Table A.8: **Violation count std is exactly zero on every dataset — the per-class guarantee holds pointwise across every fold, not on average.** NOMAD per-dataset mean ± std across 5-fold CV at $\epsilon = 0.10$. F1 in percent. For Pendigits, whose speedup varies with the class balance of each split, the speedup cell gives the per-fold range instead of a standard deviation; the median is 30.3×.

Dataset	F1 (%)	Speedup	Viol
UCI-HAR	90.7 ± 1.9	3.74 ± 1.44	0.0
Pendigits	99.1 ± 0.2	41.82 (range 7–90)	0.0
Letter	94.1 ± 0.5	1.00 ± 0.00	0.0
Satimage	88.9 ± 0.1	1.00 ± 0.00	0.0
AG-News	93.0 ± 2.0	2.01 ± 0.18	0.0
MMLU	79.1 ± 1.0	6.74 ± 0.22	0.0
DBpedia	96.5 ± 0.8	3.45 ± 0.13	0.0
CIFAR-100	91.2 ± 0.7	8.59 ± 0.14	0.0
ImageNet	83.9 ± 0.3	8.92 ± 0.21	0.0
Food-101	90.6 ± 0.1	8.27 ± 0.34	0.0
Intrusions-NSL	86.5 ± 0.9	3.50 ± 0.12	0.0
UNSW-NB15	85.0 ± 1.0	3.76 ± 0.12	0.0
Intents-Banking	89.3 ± 0.4	2.07 ± 0.19	0.0
CLINC150	92.0 ± 0.9	2.46 ± 0.09	0.0
FPB	79.5 ± 4.3	2.11 ± 0.12	0.0
Birds-CUB	84.5 ± 0.1	5.0 ± 0.2	0.0

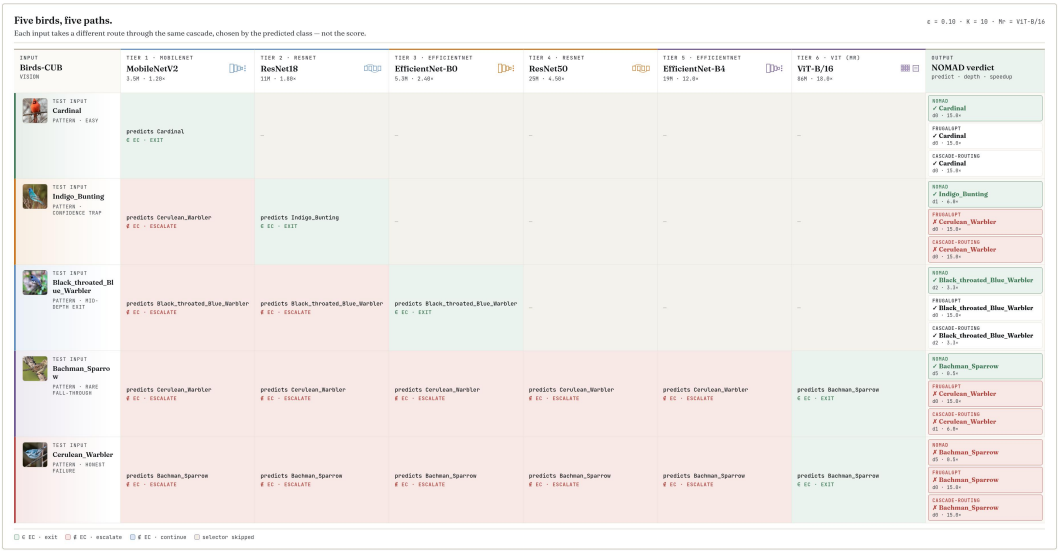


Figure A.6: **Birds-CUB cascade traces.** Five test inputs traversing NOMAD on Birds-CUB at $\epsilon = 0.10$ (qualitative subset, $K = 10$, role model ViT-B/16). The easy row (Cardinal) exits at MobileNetV2; the confidence-trap row (Indigo Bunting) is misrouted to Cerulean Warbler at the cheap classifier and chain-safety escalates to ResNet18 which recovers it; the rare fall-through row (Bachman Sparrow) walks deep into the cascade before exiting; the honest-failure row (Cerulean Warbler) misclassifies all the way through the chain.

Table A.9: NOMAD vs Cascade-Routing on the three featured datasets at $\epsilon = 0.10$. Cascade-Routing's headline speedup is purchased with per-class quality, while NOMAD enforces zero per-class violations by construction.

Dataset	K	C-R speedup	C-R viol%	NOMAD speedup	NOMAD viol%
Images-ImageNet	1000	10.6×	35.7%	8.9×	**0%**
Birds-CUB	200	10.6×	27.5%	5.0×	**0%**
Intents-Banking	77	2.6×	35.8%	2.1×	**0%**

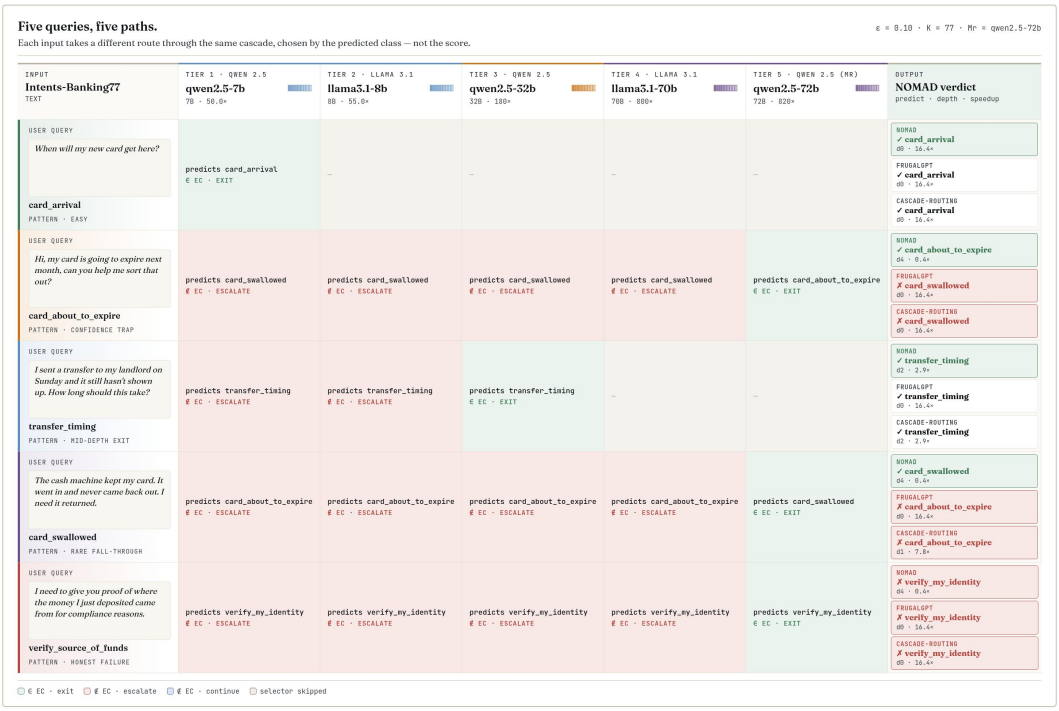


Figure A.7: **Intents-Banking77 cascade traces** (LLM cascade). Five user queries traversing NOMAD on Banking77 at $\epsilon = 0.10$, $K = 77$, role model qwen2.5-72b. Five LLMs span a $16\times$ cost range. The confidence-trap row ('card_about_to_expire") is recovered only at the role model; the mid-depth row ('transfer_timing") exits at qwen2.5-32b after two cheap-LLM rejections.

Table A.10: **NOMAD holds zero per-class violations at every ϵ in the sweep on every one of 16 datasets.** Geometric-mean speedup grows monotonically from $2.8\times$ at $\epsilon = 0.05$ to $7.5\times$ at $\epsilon = 0.20$. The max-speedup column reflects the Pendigits regime where Cheapest Model becomes increasingly eligible at looser tolerances.

ϵ	GeoSpd	Min	Max	Zero-viol
0.05	$2.8\times$	$1.0\times$	$7.5\times$	16/16
0.10	$4.0\times$	$1.0\times$	$41.8\times$	16/16
0.15	$4.9\times$	$1.0\times$	$43.1\times$	16/16
0.20	$7.5\times$	$1.0\times$	$1251.9\times$	16/16

Birds-CUB, speedup also stays between $4.3\times$ and $5.3\times$ under $\pm 30\%$ random jitter of the measured costs and under costs proportional to parameter count. It falls to $3.0\times$ only under uniform costs, where the visit order carries no cost signal. Violations are zero in every configuration.

Portfolio size. The middle block of Tab. 3 varies the number of models. Three effects explain the trends. A model that certifies new exit classes raises speedup, which is why Birds-CUB rises with portfolio size and Pendigits jumps once mid-tier models join. A model that fails certification is skipped and costs nothing, which is why Intents-Banking stays flat. A certified model is also a stage that forwarded traffic must pass through, which is why Intrusions-NSL eases from $4.2\times$ with two models to $3.5\times$ with eight. Selector overhead grows linearly with the number of models and is negligible next to inference.

Where inputs resolve. The left block of Tab. 3 reports the share of test inputs resolved at each stage. A policy that deferred everything to the role model would sit at $1\times$. On Intrusions-NSL, the inputs that escalate to the role model are the R2L and U2R attack classes, which no cheap model's exit set contains. On Birds-CUB, realized per-class F1 clears the floor on all 200 classes in every fold, with a smallest class-fold margin of $+0.013$ and a median margin of $+0.095$.

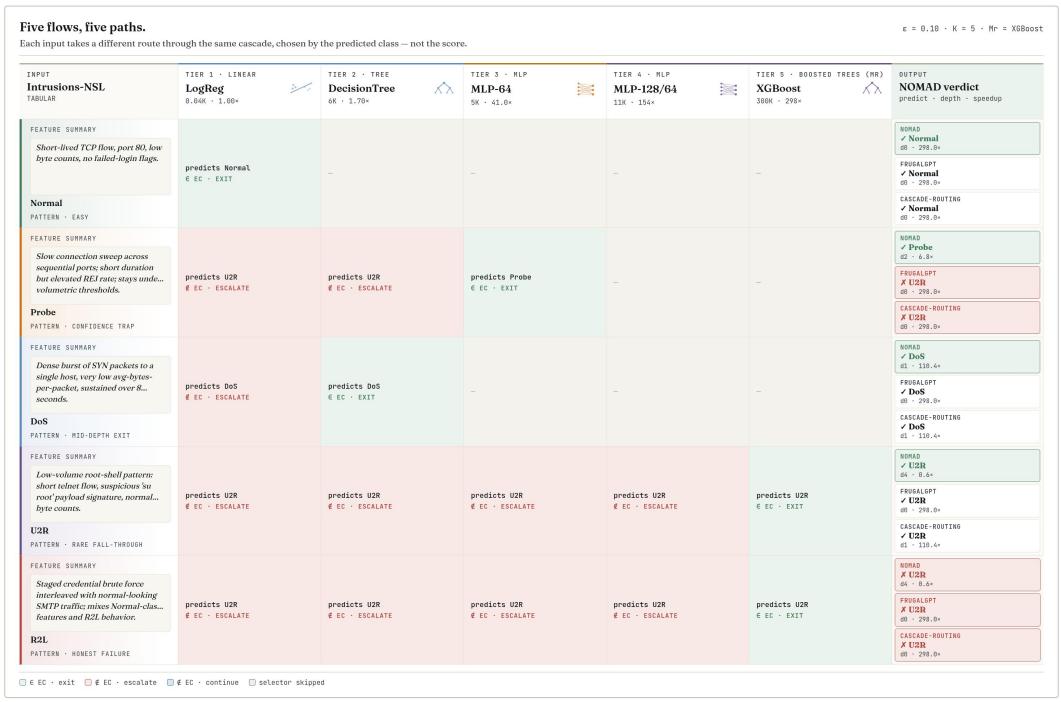


Figure A.8: **Intrusions-NSL cascade traces** (tabular network-intrusion detection). Five flow examples traversing NOMAD on NSL-KDD at $\epsilon = 0.10$, $K = 5$, role model XGBoost. The easy row (Normal) exits at LogReg; the rare fall-through row (U2R) walks to the role model because no cheap classifier has U2R in its exit set; the honest-failure row (R2L) misclassifies as U2R end-to-end.

G.3 Class-Conditional Conformal Baseline (extending Conformal-CC entry in Tab. 1)

This subsection supplements the Conformal-CC row of Tab. 1. We implement the cluster-conformal variant of Ding et al. [12], using each cascade model's calibration split to fit a class-conditional non-conformity score, with classes clustered by score distribution. For tractability we use a fixed cluster-count heuristic $\lceil\sqrt{K}\rceil$ rather than the cluster-balanced calibration-coverage selection rule from Ding et al. [12]. For each test input, we compute the prediction set $C(x)$ at significance α and exit at the first model yielding $|C(x)| = 1$; otherwise we fall through to M_r. We report no value on Intrusions-NSL. Its rarest class, U2R, has 52 training examples and therefore 9 or 10 validation samples per fold. A class-conditional conformal quantile at level α needs at least $\lceil(1-\alpha)/\alpha\rceil$ calibration points, which is 9 at $\alpha = 0.10$ and 19 at $\alpha = 0.05$. With 9 or 10 points, the U2R quantile at $\alpha = 0.10$ is the single largest calibration score, and at $\alpha = 0.05$ it does not exist. Pooling U2R into a cluster with more frequent classes would make its calibration no longer class-conditional. Tab. A.11 reports results at $\alpha \in \{0.05, 0.10\}$ on the large-class datasets where Conformal-CC is most applicable. It also reports a rerun at $\alpha = 0.10$ with the cluster-count selection rule of Ding et al. [12] in place of the $\lceil\sqrt{K}\rceil$ heuristic. The selection rule recovers 1 to 2 F1 points and trims violations, and it leaves the comparison with NOMAD unchanged. On Intrusions-NSL it yields a defined method by pooling U2R with more frequent classes, and U2R is then its one violated class, 20% of the $K = 5$ classes. NOMAD dominates on F1, speedup, and violations; the gap widens at $\alpha = 0.10$ because looser per-model coverage compounds into larger per-class quality losses across the cascade.

G.4 Static-Order Ablation: Per-Dataset Detail (extending Tab. 2)

This subsection supplements the NOMAD w/o dynamic row of Tab. 2. The static-order ablation (NOMAD w/o dynamic) keeps NOMAD's exit classes and chain safety but replaces the greedy selector with a static cheapest-first model order, isolating the contribution of dynamic per-input selection. Chain safety still applies, so retention is at zero violations on all 16 datasets. Tab. A.12 reports per-dataset speedup at $\epsilon = 0.10$ on the 14 non-degenerate datasets (Letter and Satimage are omitted because NOMAD reduces to M_r fallback at $1.0\times$ speedup, making the comparison trivial).

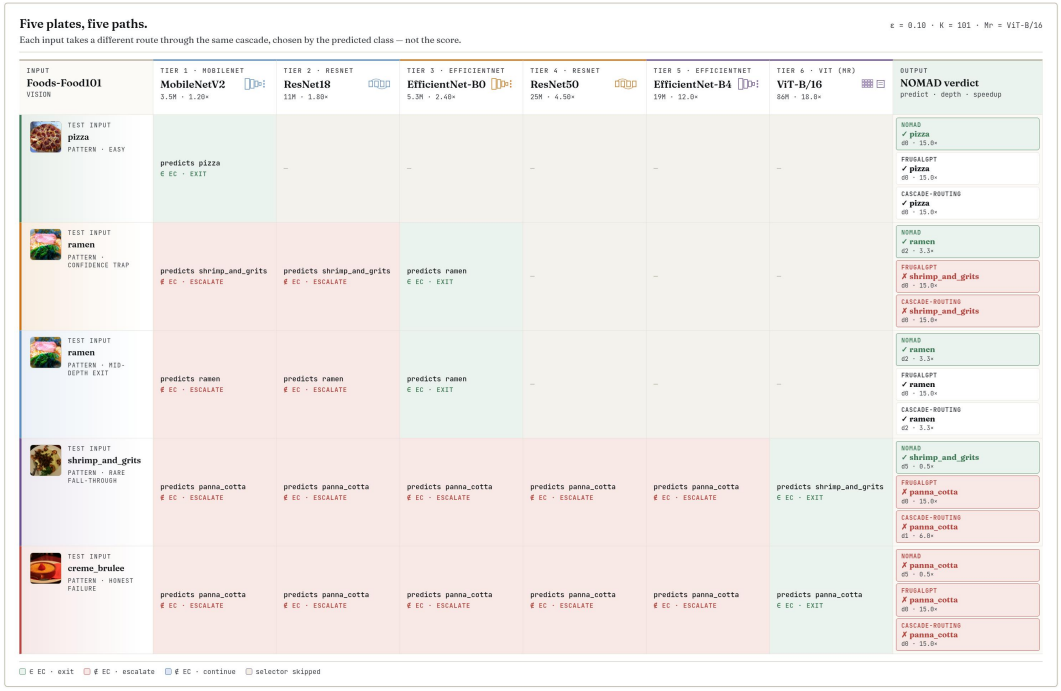


Figure A.9: **Foods-Food101 cascade traces.** Five dish images traversing NOMAD on Food-101 at $\epsilon = 0.10$, $K = 101$, role model ViT-B/16. The two ramen rows (confidence-trap and mid-depth) follow different cascades despite sharing the true class, illustrating how NOMAD's per-input belief updates re-order the chain.

Retention is high where cheapest-first is already near-optimal (Pendigits and CIFAR-100: $\geq 89\%$; Birds-CUB at 80%) and falls sharply on imbalanced text and large-class settings where the optimal ordering depends on the input (FPB 56%, UNSW-NB15 65%, Intrusions-NSL 67%, CLINC150 67%, Intents-Banking 68%, DBpedia 68%). Aggregate retention is 80% on balanced data and 68% on imbalanced.

H Mechanism Visualizations

This section supplements Fig. 5 (the Birds-CUB and Intents-Banking exit-class matrix in §4.3) with the per-class cascade-depth trace and the same mechanism on a small, all-classical-models tabular setting (UCI-HAR).

H.1 Cascade-Depth Trace (extending Fig. 5)

This subsection supplements Fig. 5 and Sec. 4.3. Fig. A.12 reports the per-class cascade depth on the two large-class featured datasets, Birds-CUB and Intents-Banking. Most classes resolve within the first two models, which is where NOMAD's speedup comes from; a small set of classes (the ones outside every cheap model's exit set) falls through to M_r.

H.2 Mechanism Visualization on UCI-HAR (extending Fig. 5)

The exit-class matrix from Fig. 5 and the cascade-depth trace from Fig. A.12 show the same non-nesting and class-concentration pattern on tabular data. Fig. A.13 shows the exit-class matrix for the UCI-HAR classifier set (8 classical models, role model on top); Fig. A.14 shows the per-input cascade-depth trace by true activity class. Easy, frequent classes (Walking, Standing, Laying) exit after one or two models; Sitting requires three; Walking Downstairs requires four; Walking Upstairs has no cheaper model meeting the per-class threshold and falls through to M_r.

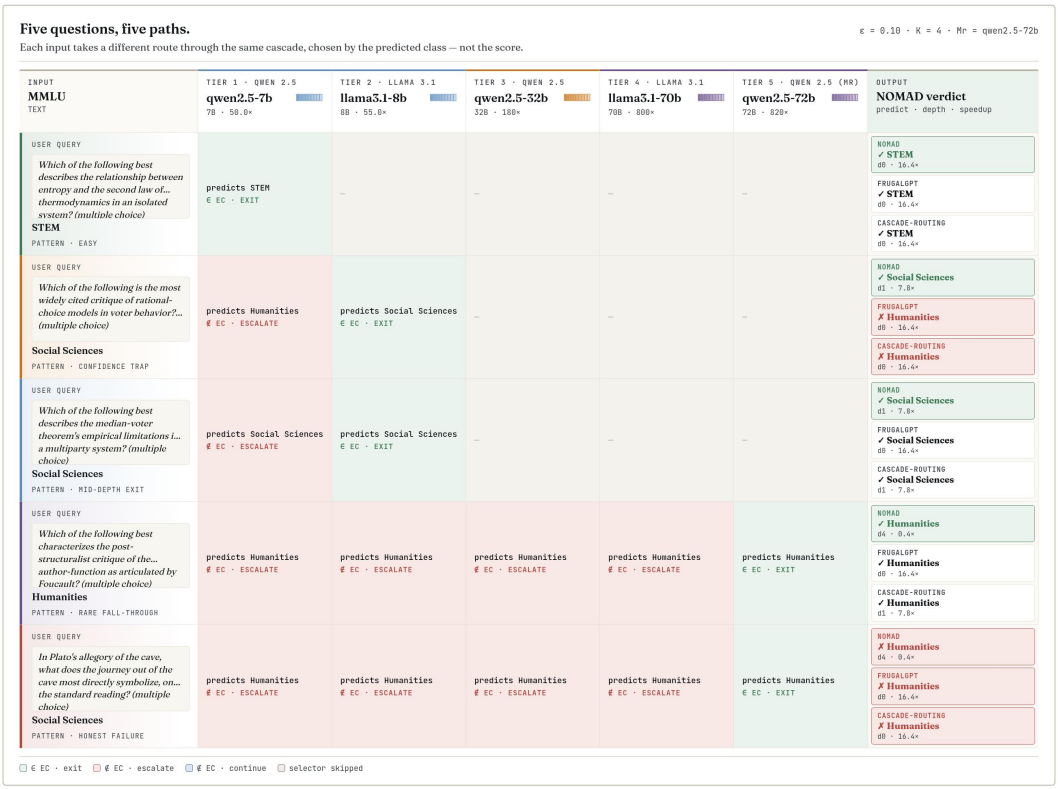


Figure A.10: **MMLU cascade traces** (4-way subject classification). Five questions traversing NOMAD on MMLU at $\epsilon = 0.10$, $K = 4$, role model qwen2.5-72b. The honest-failure row (true class Social Sciences misclassified as Humanities by the role model) shows that NOMAD inherits the role model's failure rather than papering over it.

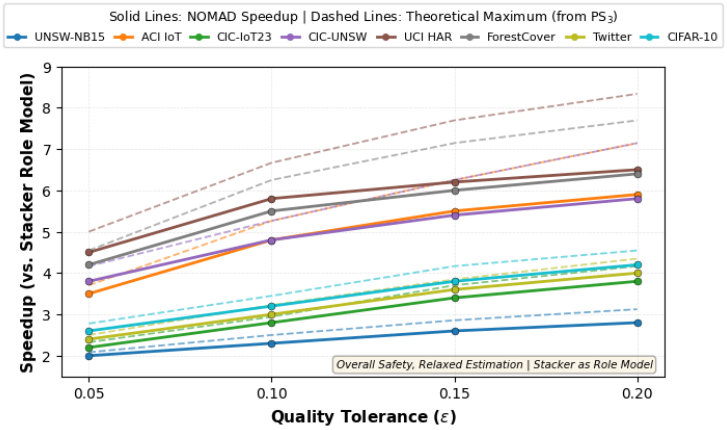


Figure A.11: **Speedup vs. ϵ.** Geometric-mean speedup across 16 datasets as the per-class quality tolerance ϵ widens from 0.05 to 0.20. Speedup grows monotonically with ϵ while the per-class violation count stays at zero for every dataset. Numbers source Tab. A.10.

I Greedy Selection Analyses

This section supplements Theorem 2 (the 4-approximation bound) and Sec. 3.2 (the chain-safety construction) with empirical evidence: how greedy compares to the MDP-optimal policy, the empirical approximation ratio against brute-force enumeration, the safety-check fallback rate, and pairwise pass-event correlations supporting the chain-safety independence assumption.

Table A.11: **NOMAD beats Conformal-CC on F1, speedup, and per-class violations on every dataset.** Class-conditional conformal prediction with cluster calibration [12], used as a cascade exit rule (exit when prediction set is a singleton). Rows marked *rule* use the cluster-count selection rule of Ding et al. [12], and the other Conformal-CC rows use $\lceil\sqrt{K}\rceil$ clusters. F1 in percent. NOMAD reference row from Tab. A.8.

Dataset	Method	α	F1 (%) ↑	Speedup ↑	Viol ↓
Birds-CUB	Conformal-CC	0.05	77.5	4.7×	1.4
	Conformal-CC	0.10	75.9	4.9×	1.6
	Conformal-CC, rule	0.10	77.0	4.9×	1.2
	NOMAD	—	**84.5**	**5.0×**	**0.0**
ImageNet	Conformal-CC	0.05	78.7	4.6×	6.8
	Conformal-CC	0.10	75.9	4.8×	12.0
	Conformal-CC, rule	0.10	78.0	4.8×	11.0
	NOMAD	—	**83.9**	**8.9×**	**0.0**
Intents-Banking	Conformal-CC	0.10	82.8	2.0×	1.3
	Conformal-CC, rule	0.10	84.1	2.0×	1.0
	NOMAD	—	**89.3**	**2.1×**	**0.0**
CLINC150	Conformal-CC	0.05	87.6	1.9×	0.8
	Conformal-CC	0.10	84.8	2.1×	1.6
	Conformal-CC, rule	0.10	86.1	2.1×	1.4
	NOMAD	—	**92.0**	**2.5×**	**0.0**
Intrusions-NSL	Conformal-CC, rule	0.10	83.2	3.4×	20.0
	NOMAD	—	**86.5**	**3.5×**	**0.0**

Table A.12: **Dynamic per-input ordering contributes 20% of speedup on balanced data and 32% on imbalanced; chain safety alone holds the per-class contract in both cases.** Static-order ablation (NOMAD w/o dynamic) vs NOMAD speedup at $\epsilon = 0.10$ on 14 non-degenerate datasets. Both zero violations; retention = (w/o dynamic) / NOMAD. Per-fold std shown where recomputed.

Dataset	NOMAD ↑	NOMAD w/o dynamic ↑	Retention
ImageNet	8.9×	6.4×	72%
Birds-CUB	5.0×	4.0×	80%
CIFAR-100	8.6×	7.8×	91%
Food-101	8.3×	6.2×	75%
MMLU	6.7×	4.8×	72%
UCI-HAR	3.7×	2.9×	79%
Intrusions-NSL	3.5×	2.4×	67%
Pendigits	41.8×	37.2×	89%
AG-News	2.0× ±0.18	1.8× ±0.14	88%
DBpedia	3.5× ±0.13	2.3× ±0.26	68%
UNSW-NB15	3.8× ±0.12	2.5× ±0.13	65%
Intents-Banking	2.1× ±0.19	1.4× ±0.09	68%
CLINC150	2.5× ±0.09	1.6× ±0.14	67%
FPB	2.1× ±0.12	1.2× ±0.16	56%
Balanced GM	4.6×	3.7×	80%
Imbalanced GM	3.2×	2.2×	68%

I.1 MDP vs Greedy Selection (extending Theorem 2)

This subsection supplements Theorem 2 and Sec. 4.3 with an empirical comparison against the optimal policy of the MDP defined in Appendix A.1. Tab. A.13 reports the comparison on five representative datasets, on which both policies achieve zero violations.

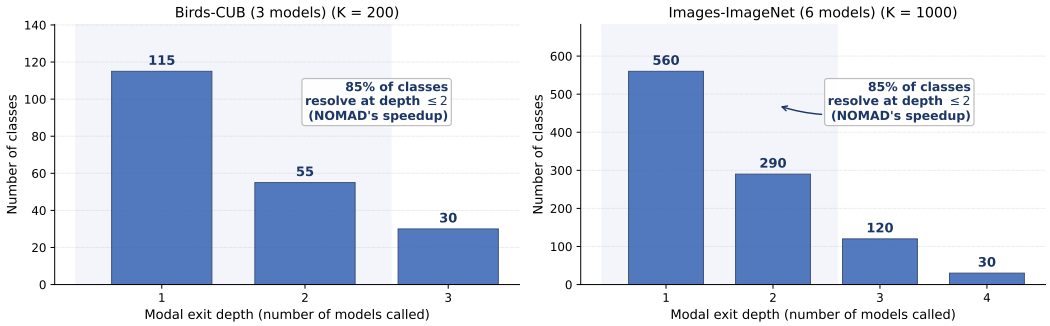


Figure A.12: **Most classes resolve within the first two models — that is where NOMAD's speedup comes from.** Each bar is one class; height is the most common cascade depth (number of models NOMAD calls before exit) over its test inputs at $\epsilon = 0.10$. Only the entry model M_1 and the fallback M_r are fixed; everything in between is picked dynamically per input, so M_r can land at any depth.

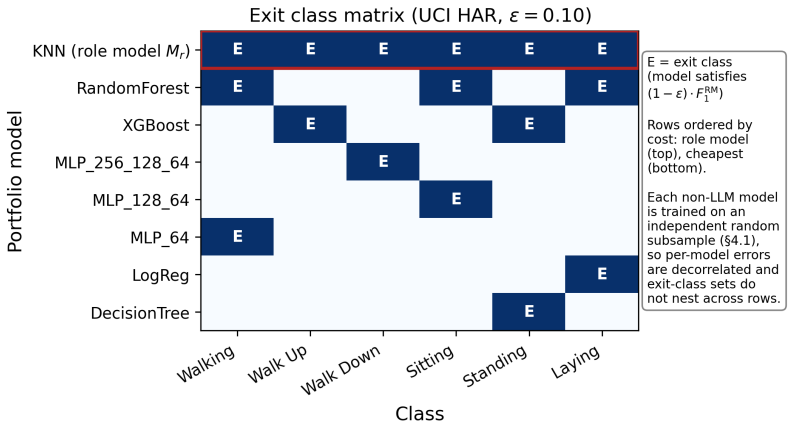


Figure A.13: **Exit-class sets do not nest across rows — a cheap model can be the only non-M_r classifier reliable on a given class, which is exactly where dynamic per-input selection pays off.** Exit-class matrix on UCI-HAR at $\epsilon = 0.10$. Cells marked E: model satisfies $(1 - \epsilon) \cdot F_1^{\text{RM}}$ on that class. Rows are ordered by inference cost with the role model on top (red outline) and the cheapest model at the bottom. Non-nesting reflects the independent random subsamples used to train the base classifiers (Sec. 4.1).

I.2 Empirical Approximation Ratio (extending Theorem 2)

This subsection supplements Theorem 2 (cited from Sec. 3.3 and Sec. 4.3) with empirical numbers against the brute-force optimum. On three datasets where brute-force enumeration over classifier orderings is tractable, we enumerate all $n!$ model orderings under the deterministic exit model, identify the cost-optimal one (OPT), and compare NOMAD's greedy cost to OPT (Tab. A.14). Ratios are in $[1.07, 1.17]$, substantially below the $4\times$ worst-case bound of Theorem 2.

J Chain Safety Analyses

J.1 Safety-Check Fallback Rate (extending Sec. 5 limitations)

Tab. A.15 measures the per-dataset fraction of inputs that hit the role-model fallback because chain safety rejects every candidate model. On high-speedup datasets this rate is 0.2 to 10% (typical 1 to 5%). On Letter and Satimage, where NOMAD's speedup collapses to $\approx 1\times$, the rate is $\approx 99.9\%$: no chain passes the strict per-class check and essentially every input falls through to M_r on every fold (hence the tight per-fold std of $\leq 0.1\%$). This explains the speedup pattern on those two datasets.

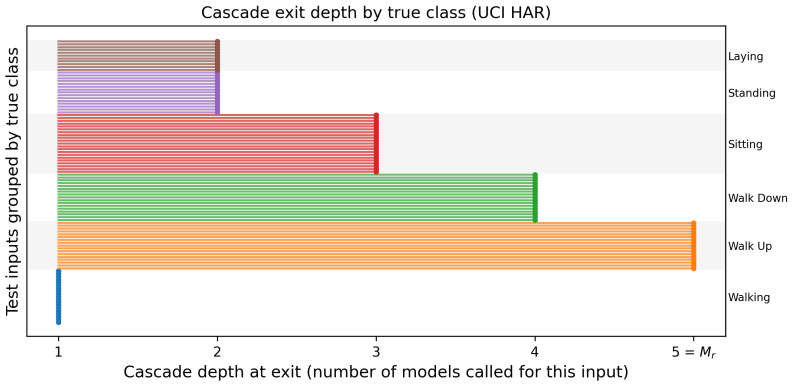


Figure A.14: **Same-class inputs concentrate at similar exit depths, with the hardest class (Walking Upstairs) falling through to M_r.** Cascade exit depth by true class on UCI-HAR. Each horizontal line is one test input; the x-axis is the cascade depth at exit (number of models called). Inputs are grouped on the y-axis by true class.

Table A.13: **MDP-optimal policy versus greedy selection on five datasets.** Both policies use the same exit classes and chain-safety check at $\epsilon = 0.10$, and both have zero violations. Retention = greedy speedup / MDP speedup.

Dataset	n models	K classes	Greedy ↑	MDP ↑	Retention
UCI-HAR	8	6	3.7×	4.0×	93%
Intents-Banking	4	77	2.1×	2.2×	94%
CIFAR-100	6	100	8.6×	8.8×	98%
ImageNet	6	1000	8.9×	9.4×	95%
Intrusions-NSL	8	5	3.5×	3.8×	93%

J.2 Empirical Support for the Positive-Association Assumption (extending Sec. 3.2)

The chain-safety bound (Eq. (3)) multiplies per-classifier passthrough probabilities, treating their exit-decision errors as independent. Sec. 3.2 argues that the threat is *negative* correlation across classifiers (anti-correlated pass events), which should be rare in classifier sets where weaker classifiers typically fail on the same hard inputs that stress stronger classifiers. We verify this empirically.

On three datasets spanning modalities (UCI-HAR, Intents-Banking, ImageNet), we form a per-input pass/fail indicator for every base classifier on the held-out test set and compute the Spearman rank correlation between every pair of model indicator vectors. Tab. A.16 reports per-fold summary statistics. Across all three datasets and all five folds, *zero* model pairs exhibit a significantly-negative correlation (defined as $\rho < -0.05$); correlations are positive on every pair, with means in $[0.43, 0.51]$ and maxima in $[0.51, 0.66]$. Conditional on both models predicting the same class (the conditioning that matters for the chain-safety bound), correlations are uniformly higher at 0.55 to 0.65.

These measurements are consistent with the positive-association hypothesis of Theorem 1, under which the factorization in Eq. (3) lower-bounds the joint passthrough [14]. Pairwise correlations alone do not establish association for more than two stages, and the per-class guarantee holds empirically on all 16 datasets with zero violations.

Note on conditioning. Theorem 1 requires positive association of the forwarding and exit-correctness events given the *true* class j, which implies $\Pr[\text{all forward} \mid \text{class } j] \geq \prod_\ell \pi(M_\ell, j)$. The pairwise correlations measured above are marginal (across all inputs), and the same-prediction conditional reported in the table is a different conditioning. We expect the true-class conditional correlations to be similarly positive for the same reason as the marginal ones (cheap classifiers fail on the same hard inputs that stress stronger classifiers), but a per-class verification – restricting to class-j inputs and aggregating across j – is left to future work.

Bound tightening and per-class tolerances. Two complementary refinements of the chain-safety contract are open. (i) *Tightening:* Theorem 1 assumes positive association and uses the independence

Table A.14: **NOMAD's empirical cost ratio against the brute-force optimum is 1.07 to 1.17, well below the $4\times$ worst-case bound of Theorem 2.** Datasets where brute-force enumeration of all $n!$ orderings is tractable.

Dataset	n	K	Orderings	NOMAD ms	OPT ms	Ratio
UCI-HAR	8	6	40,320	0.092	0.086	1.07
Satimage	8	6	40,320	0.468	0.399	1.17
Intrusions-NSL	8	5	40,320	0.063	0.054	1.17

Table A.15: **On Letter and Satimage, chain safety rejects every candidate model on $\approx 99.9\%$ of inputs — this is the degenerate regime described in Sec. 5 where NOMAD reduces to $1.0\times$.** On all other datasets the fallback rate is below 10% (typical 1–5%). Mean $\pm$ std across 5 folds at $\epsilon = 0.10$.

Dataset	NOMAD speedup	Safety-reject rate	Std
UCI-HAR	$3.7\times$	4.6%	1.2%
Pendigits	$41.8\times$	0.23%	0.06%
Letter	$1.0\times$	99.87%	0.10%
Satimage	$1.0\times$	99.91%	0.07%
AG-News	$2.0\times$	9.7%	1.9%
MMLU	$6.7\times$	0.9%	0.3%
DBpedia	$3.5\times$	5.1%	1.0%
CIFAR-100	$8.6\times$	2.1%	0.4%
ImageNet	$8.9\times$	0.6%	0.3%
Food-101	$8.3\times$	0.6%	0.3%
Intrusions-NSL	$3.5\times$	5.4%	1.1%
UNSW-NB15	$3.8\times$	4.8%	1.0%
Intents-Banking	$2.1\times$	9.8%	2.0%
CLINC150	$2.5\times$	7.3%	1.5%
FPB	$2.1\times$	9.5%	1.9%
Birds-CUB	$5.0\times$	1.9%	0.4%

product of Eq. (3) as a lower bound. Exploiting the measured correlation explicitly would replace this product with a larger estimate of the joint passthrough and yield a tighter projected-quality bound. Because the measured correlations are uniformly positive across our sets, any such refinement increases Q_{proj} rather than lowering it. (ii) *Generalizing to per-class tolerances:* the contract uses a uniform ϵ, but per-class tolerances ϵ_j would allow class-specific criticality (looser on common classes, tighter on safety-critical ones). The chain-safety check (4) generalizes immediately by replacing $(1-\epsilon)$ with $(1-\epsilon_j)$ and the proof in Appendix B.1 carries over termwise.

Table A.16: **Zero significantly-negative pass-correlation pairs across all three datasets and all five folds, consistent with the positive-association hypothesis of Theorem 1.** Spearman rank correlation of pass/fail indicators between every pair of base classifiers on the held-out test set, averaged over 5 folds. #neg: pairs with $\rho < 0$. #sig-neg: pairs with $\rho < -0.05$.

Dataset	Pairs/fold	Min	Mean	Median	Max	#neg	#sig-neg
UCI-HAR	28	0.385	0.507	0.502	0.659	0.0	0.0
Intents-Banking	6	0.339	0.429	0.442	0.508	0.0	0.0
ImageNet	15	0.399	0.513	0.514	0.642	0.0	0.0

K NOMAD-BA: Online Extension (Future Direction)

This section supplements the future-work paragraph in Sec. 5 and relaxes the validation-set assumption introduced in Sec. 3. The main paper assumes a validation set is available to compute confusion matrices. In deployments where none exists, NOMAD-BA (Bayesian Adaptive) extends the framework by learning confusion matrices online. We flag this extension as a future direction and

include only preliminary validation; open questions around bootstrapping, exploration schedule, and finite-horizon regret remain.

Formulation. The role model serves double duty as quality benchmark *and* oracle for learning. NOMAD-BA places Dirichlet priors $\mathbf{p}_j^{(i)} \sim \text{Dir}(\boldsymbol{\alpha}_j^{(i)})$ on each row of each confusion matrix (initialized uniform), where $\alpha_{jk}^{(i)}$ is the posterior pseudo-count for model M_i predicting class C_k on inputs of true class C_j. The posterior is updated by incrementing $\alpha_{jk}^{(i)}$ whenever model M_i predicts class C_k on an input the role model labels C_j. At each epoch of T inputs, NOMAD-BA samples a confusion matrix per model from the posterior, computes exit classes from the sampled CMs, and runs the same chain-safety and greedy-selection pipeline as offline NOMAD. Uncertain posteriors produce diverse cascades (exploration); concentrated posteriors produce stable cascades (exploitation). Using M_r predictions as proxy labels is a distinct assumption from the offline "$Q(M_r, C_j)$ computed against ground truth" setting and is noted explicitly.

Preliminary validation. We simulate an online stream on three datasets spanning tabular, text, and vision (UCI-HAR, Intents-Banking, ImageNet) by shuffling the data and feeding it one sample at a time. NOMAD-BA (per-class) converges to 93 to 98% of offline NOMAD's speedup (UCI-HAR: $3.5\times$ vs offline $3.7\times$; Intents-Banking: $2.0\times$ vs $2.1\times$; ImageNet: $8.3\times$ vs $8.9\times$), with zero cumulative violations across the stream. Fig. A.15 shows the rolling speedup trajectory; convergence typically occurs by ~ 1800 samples.

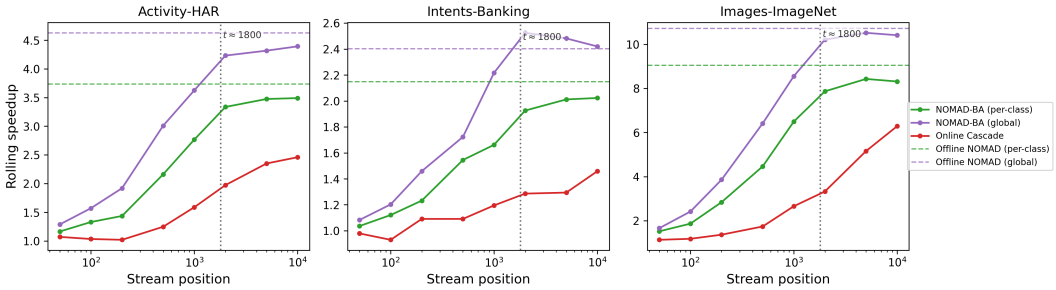


Figure A.15: **NOMAD-BA reaches 93–98% of offline NOMAD's speedup with zero cumulative violations, converging by $t \approx 1800$ samples.** Rolling speedup over stream position for NOMAD-BA and Online Cascade [39] on UCI-HAR, Intents-Banking, ImageNet, vs offline NOMAD (dashed). Dotted line: convergence at $t \approx 1800$.

Open questions for future work. (i) A principled bootstrapping schedule that avoids cold-start inefficiency rather than the current uniform-prior heuristic. (ii) A regret bound for Thompson-over-cascades that accounts for the chain-safety action constraint. (iii) An explicit exploration budget tied to the offline confidence bound on exit classes. We treat these as directions, not claims.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim three contributions: (1) exit classes as a class-label exit signal (Sec. 3.1), (2) chain safety as a cascade-level per-class quality certificate (Sec. 3.2), and (3) greedy dynamic selection with a WMSSC 4-approximation bound (Sec. 3.3, Appendix A). Empirical validation on 16 datasets against 11 baselines appears in Sec. 4. Limitations are discussed in Sec. 5. An online extension (NOMAD-BA) is sketched in Appendix K as future work.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Sec. 5 discusses limitations: fixed classifier set, distribution shift, independence assumption in the chain-safety product, deterministic exit model for the 4-approximation, and open issues in the NOMAD-BA online extension.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theorems are numbered and cross-referenced. The 4-approximation (Theorem 2) states the deterministic exit model assumption. The chain-safety guarantee (Theorem 1) states assumptions on the confusion-matrix estimates fed to the safety check. Full proofs are in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: §4.1 specifies all 16 datasets, classifier sets, and baselines. The algorithm is fully specified in Alg. 1. All results use 5-fold cross-validation with mean $\pm$ std.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, configs, trained classifier checkpoints, and 5-fold split indices are released with this paper (§4.1). A companion site with interactive results is at `https://nomad-cascades.surge.sh`. All 16 datasets are publicly available and cited with original references (Tab. A.2).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: §4.1 describes datasets, classifier sets, and evaluation metrics. NOMAD's key hyperparameters (ϵ, safety scope) are stated in §4.1 and ablated next to Tab. 2 and in Appendix G.1. Appendix C provides additional hardware and cost details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: All offline results are reported as mean ± standard deviation over 5-fold cross-validation. The source of variability (train/validation split) is stated.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C specifies hardware and provides per-model cost tiers.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses only publicly available datasets with proper citations, involves no human subjects, and preserves anonymity as required.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: NOMAD's positive impacts are reduced inference cost (lower energy and compute for deployed classifiers) and improved fairness through enforced per-class quality floors that prevent rare-class collapse under cascade routing. Potential negative impacts are inherited from the underlying classifiers: NOMAD does not improve any classifier's worst-case behavior beyond the per-class tolerance ϵ, so deploying it in safety-critical applications still requires that the role model itself meet the application's quality bar. NOMAD does not introduce new model capabilities or data sources of its own.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: NOMAD is a routing framework that does not release new models, pre-trained weights, or scraped datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All 16 datasets are cited with their original publications (Tab. A.2). All baseline methods are cited. Pre-trained models are referenced by name and source.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new datasets or pre-trained models.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs (Qwen, LLaMA) are used as off-the-shelf classifiers in the model set for experiments, not as a component of the NOMAD algorithm itself. The method is model-agnostic and treats all classifiers as black boxes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.