Capacity-Aware Multi-Fidelity Architecture Evaluation: Best-Arm Identification with Proxy Bias and Costs

Liz Lemma   Future Detective

January 20, 2026

Abstract

Neural architecture search (NAS) in global, high-variance spaces is bottlenecked by evaluation: architectures are typically compared under a single fixed training protocol, despite evidence (including in Efficient Global Neural Architecture Search) that training settings can affect accuracy as much as the architecture itself. We formalize architecture evaluation as a cost-sensitive multi-fidelity best-arm identification problem where each architecture can be queried at varying fidelities (training steps, data fraction, resolution) producing noisy proxy accuracies with a capacity-dependent, fidelity-decaying bias. We prove a lower bound showing that any fixed-fidelity protocol can require linear-in-$|\mathcal{X}|$ compute on heterogeneous learning-curve instances, even when an adaptive multi-fidelity strategy can succeed with sublinear effective cost. We then introduce CASH, a capacity-aware successive-elimination algorithm that chooses per-architecture training budget from a learned/calibrated mapping of architecture statistics to fidelity and uses bias-aware confidence intervals to eliminate suboptimal candidates safely. Under a parametric learning-curve/bias model, CASH is (ε, δ)-correct and achieves near-minimax optimal expected cost up to logarithmic factors. Finally, we outline an experimental validation plan on NAS-Bench/JAHS-Bench and a global macro–micro space to measure rank correlation improvements and compute-to-ε-optimality, turning the source paper's dynamic ranking insight into a principled, provably efficient evaluation primitive for modern NAS.

Table of Contents

2. 2. Related Work: multi-fidelity HPO (successive halving/Hyperband/BOHB), best-arm identification with costs, learning-curve extrapolation, NAS evaluation pitfalls (weight sharing, zero-cost proxies), and how this differs (explicit proxy-bias + capacity-aware fidelity choice).

3. 3. Problem Setup and Preliminaries: define architecture set, fidelities, costs; define proxy-bias model and sub-Gaussian observations; define (ε, δ)-BAI objective and cost minimization.

4. 4. Why Fixed Protocols Fail: formal impossibility/lower bound for fixed-fidelity evaluation under heterogeneous bias/curve rates; interpret in NAS terms (different capacities require different budgets).

5. 5. CASH Algorithm: capacity-aware multi-fidelity successive elimination; calibration stage for budget rule; bias-aware confidence bounds; stopping and output; discussion of design choices and how it maps to NAS evaluation pipelines.

6. 6. Main Upper Bounds: (ε, δ)-correctness proof; expected cost bound; near-minimax optimality (matching lower bound up to logs) for the proxy-bias model.

7. 7. Practical Instantiation for NAS: selecting statistics $s(x)$; setting fidelity grid; estimating/upper-bounding $g(\cdot)$; handling unknown α; incorporating per-architecture wall-clock cost $\kappa(x)$.

8. 8. Experimental Plan (Implementation-Strengthening Section): evaluation on NAS-Bench/JAHS-Bench and a global macro–micro CNN space; metrics (rank correlation vs final, compute-to-ε-optimal, robustness across datasets/seeds); ablations (static vs CASH; learned vs heuristic budget rule).

9. 9. Discussion, Limitations, and Extensions: non-monotone curves, recipe adaptation beyond epochs, multi-objective costs (latency/energy), continuous spaces; open problems.

10. 10. Conclusion: takeaways and the evaluation primitive as a building block for 2026-era global NAS.

1 Introduction and Motivation

Neural architecture search is routinely driven by proxy evaluations: we instantiate an architecture x and train it for a limited budget (say, f epochs), producing an observed score $Y(x, f)$ that is then used to rank candidates and decide which models deserve further compute. The prevalent simplification is to fix a single protocol—a common training schedule, augmentation recipe, and, critically, a common fidelity f_0—and to treat the resulting rankings as stable surrogates for the terminal ranking induced by the converged means $\{\mu_x\}_{x\in\mathcal{X}}$. In practice, however, architecture rankings are not invariant to training budget: the relative order of two architectures can change as f increases, even when all other hyperparameters are held constant. This phenomenon is visible in essentially any heterogeneous search space: smaller or easier-to-optimize architectures often achieve high proxy accuracy early and then saturate, while larger or harder-to-optimize architectures may learn slowly but eventually overtake. Consequently, a fixed-fidelity protocol may systematically mis-rank the set $\mathcal{X}$, promoting "fast starters" and suppressing "late bloomers."

We view this instability as a structural issue rather than an implementation artifact. Even in the idealized setting where each proxy evaluation is an unbiased noisy estimate of a monotone learning curve, a single snapshot f_0 cannot capture heterogeneous rates. More pointedly, in realistic pipelines the proxy is not merely noisy but also biased downward relative to the converged performance: limited training typically underestimates the terminal accuracy, and the magnitude of this underestimation depends on the architecture. Denoting by $\mathbb{E}[Y(x, f)]$ the expected proxy performance at fidelity f, the proxy bias $\beta_x(f) \coloneqq \mu_x - \mathbb{E}[Y(x, f)]$ is nonnegative and decreases with f, but it need not be uniform across arms. If $\beta_x(f_0)$ varies widely with x, then the ranking induced by $\mathbb{E}[Y(x, f_0)]$ can disagree with the ranking induced by μ_x by more than any tolerance relevant to best-arm identification. Importantly, this mis-ranking cannot be repaired by mere repetition at the same fidelity: averaging reduces statistical noise but leaves systematic bias intact.

This observation yields a dilemma for any fixed protocol. If we insist on correctness guarantees for identifying an ε-optimal architecture with high probability, then we must choose f_0 large enough that the worst-case bias at f_0 is below the relevant gaps. In heterogeneous spaces, this "worst-case" fidelity is essentially the near-terminal budget, which forces training many architectures far longer than needed. Conversely, if we choose a modest f_0 to save compute, then there exist plausible instances where the proxy observations are information-theoretically insufficient to distinguish the optimum: two arms may have terminal gap Δ_x yet be indistinguishable at fidelity f_0 because admissible biases can erase the gap at the proxy level. The central thesis of this work is that this dilemma is not incidental; it is minimax. The correct remedy is not to tune a single fidelity, but to allocate different fidelities to different architectures in a way that respects their heterogeneous bias scales.

Our first contribution is to formalize this proxy-evaluation setting as a cost-sensitive multi-fidelity best-arm identification problem. We consider a finite candidate set $\mathcal{X}$ with unknown terminal means $\mu_x \in [0,1]$ and a set of fidelities $\mathcal{F} \subset \mathbb{R}_+$. Querying (x,f) yields $Y(x,f)$ with sub-Gaussian noise and a one-sided bias: $\mathbb{E}[Y(x,f)] \leq \mu_x$ and $0 \leq \mu_x - \mathbb{E}[Y(x,f)] \leq g(s(x))f^{-\alpha}$ for a known statistic $s(x)$ (e.g., parameter count) and known decay exponent $\alpha \in (0,1]$, but unknown nonnegative scale function $g(\cdot)$. Each query incurs cost $c(x,f) = \kappa(x)f$. The goal is to output $\hat{x}$ with $\mu_{\hat{x}} \geq \max_x \mu_x - \varepsilon$ with probability at least $1-\delta$, while minimizing total cost. This model isolates, in a tractable manner, the empirical fact that larger-capacity or otherwise complex architectures can exhibit larger short-budget underestimation.

Our second contribution is a minimax lower bound showing that fixed-fidelity policies are fundamentally inefficient under heterogeneous bias. Informally, for any algorithm that evaluates every arm at (essentially) a single shared fidelity f_0, we construct instances with two "types" of arms having different bias scales such that (ε,δ)-correctness forces the algorithm to increase fidelity to a much larger value $f_{\text{hard}} \gg f_0$ for a linear number of arms. Equivalently, either one pays $\Omega(N)$ near-terminal cost or one cannot guarantee correctness. This establishes a separation: the inability of a fixed protocol to adapt to $g(s(x))$ can be exploited by adversarial (yet model-compliant) learning-curve heterogeneity.

Our third contribution is an adaptive algorithm, CASH, that uses the known statistic $s(x)$ to choose per-arm fidelities and employs bias-aware confidence bounds within a successive-elimination framework. At a high level, CASH maintains an active set and, in each round, selects for each active arm x a fidelity $f_r(x)$ just large enough to reduce the bias upper bound below a round-dependent tolerance. It then queries the arm at that fidelity, forms upper and lower confidence bounds that explicitly account for one-sided bias, and eliminates arms whose (bias-corrected) upper bounds fall below the incumbent's lower bound. The design ensures that, on a single high-probability event, the true terminal means remain sandwiched between the bounds for all arms and rounds, so the optimal arm is never discarded.

Our fourth contribution is a matching (up to logarithmic factors) expected-cost analysis. We show that the total cost decomposes into two unavoidable components: a *bias-resolution* term scaling like $\kappa(x)\big(g(s(x))/\Delta_x\big)^{1/\alpha}$, which is the cost needed to push the proxy bias below the gap for arm x, and a *statistical* term scaling like $\kappa(x)\sigma^2/\Delta_x^2$, which is the cost needed to overcome sub-Gaussian noise. Summed over suboptimal arms, these terms yield a near-minimax upper bound, and we complement it with a lower bound indicating that no adaptive policy can do uniformly better (up to logs) under the stated bias model.

Finally, we outline an empirical plan to validate the theory against practical NAS evaluation. We measure (i) rank instability as a function of f (e.g., Kendall τ across fidelities), (ii) compute-to-ε-optimality, and (iii) the distribution of per-architecture allocated budgets. We compare CASH to fixed-fidelity baselines and standard multi-fidelity heuristics, using controlled synthetic instances (where $g(\cdot)$ and α are known) and realistic search spaces (where they are not), thereby testing whether capacity-aware fidelity selection delivers the predicted compute savings without sacrificing identification accuracy.

2 Related Work

Multi-fidelity hyperparameter optimization. A large body of work in hyperparameter optimization (HPO) exploits the fact that partial training can be used as a cheap proxy for full training. Early-stopping and racing methods instantiate many configurations at small budgets and progressively allocate more resources to a shrinking subset. Canonical examples include successive halving and its budget-adaptive variant Hyperband **??**. These methods are designed to be simple, parallelizable, and robust when learning curves are informative, and they provide worst-case guarantees for identifying good configurations under stylized assumptions (e.g., stochastic rewards at each budget). BOHB **?** combines Hyperband-style resource allocation with model-based sampling (KDE-based Bayesian optimization), targeting improved sample efficiency in large configuration spaces. More recent multi-fidelity Bayesian optimization frameworks incorporate fidelity as an explicit input and model correlations across budgets via Gaussian processes or related surrogates **??**. Our setting shares the same operational primitive (querying a configuration at a chosen budget), but the emphasis is different: we study finite candidate sets with an explicit *one-sided* proxy bias relative to a terminal objective, and we make the cost model and the fidelity-dependent bias constraints central to the identification guarantee.

Best-arm identification with costs and structured sampling. The best-arm identification (BAI) literature provides information-theoretic and algorithmic foundations for identifying an ε-optimal arm with probability at least $1-\delta$ under noisy observations **???**. Cost-sensitive variants allow different arms to have different sampling costs, leading to policies that trade off information gain and expenditure **??**. There is also work on BAI under additional structure (e.g., correlated arms, contextual information, or side observations), where the goal is to reduce sample complexity by leveraging known relationships between arms **???**. Our contribution is orthogonal to most of this literature: the principal obstruction we address is not purely statistical noise but *systematic underestimation* that depends on fidelity and varies across arms. In particular, repeated evaluation at a fixed fidelity cannot remove this bias. Consequently, the relevant resource is not only the number of samples but the fidelity required to make observations *informative* about μ_x at the desired resolution.

Learning-curve modeling and extrapolation. Another line of work seeks to predict terminal performance from partial learning curves via parametric fits, extrapolation, or meta-modeling across tasks and architectures **???**. These approaches can be effective empirically when curves conform to a family of shapes and when sufficient meta-data are available. However, their guarantees typically depend on modeling assumptions that can be violated by architecture-dependent optimization dynamics, regularization, or training instabilities. By contrast, our proxy model is deliberately conservative: we assume only that proxy evaluations are biased downward and that the bias admits an upper bound that decays with fidelity. This perspective treats partial training as an *admissible but systematically distorted* observation of the terminal objective, and it suggests that one should allocate fidelity until the distortion is provably dominated by the gap scale relevant for elimination.

Neural architecture search evaluation pitfalls and proxy metrics. Practical NAS pipelines commonly rely on proxy evaluation mechanisms beyond early stopping, including weight sharing (one-shot NAS), low-resolution or shortened training schedules, and a variety of "zero-cost" predictors based on architecture or gradient statistics **???**. These proxies are motivated by extreme computational constraints, but their relationship to true converged accuracy can be unstable, and the induced ranking can vary substantially with training protocol and budget **??**. In particular, weight sharing introduces interference between candidates and can yield proxy scores that are not comparable to standalone training **?**. Zero-cost predictors can correlate with final accuracy in some regimes but may fail under distribution shift, strong regularization, or changes in optimizer and augmentation **?**. Our work does not attempt to replace these proxies; rather, we isolate a failure mode that persists even when the proxy is "honest" partial training with independent noise: heterogeneous, fidelity-dependent underestimation can force a fixed evaluation protocol to either spend near-terminal compute broadly or suffer irreparable mis-ranking.

How our formulation differs. The closest conceptual intersection is multi-fidelity racing (e.g., Hyperband) combined with statistical elimination. The key distinction is that we model proxy evaluations as *biased* estimators of μ_x with a one-sided, fidelity-decaying envelope $\beta_x(f) \leq g(s(x))f^{-\alpha}$, where $s(x)$ is known and $g(\cdot)$ is unknown. This yields two consequences that are not addressed explicitly in standard multi-fidelity HPO analyses. First, correctness depends on resolving bias below arm-dependent gaps, leading to a fidelity requirement of order $(g(s(x))/\Delta_x)^{1/\alpha}$ that cannot be bypassed by repetition at low fidelity. Second, the presence of a known statistic $s(x)$ motivates *capacity-aware* fidelity assignment, which is neither purely bandit-style cost weighting nor purely budget scheduling: the budget rule is chosen to control bias as a function of architecture complexity. In this sense, we treat fidelity not only as a resource but as an instrument for *debiasing* proxy observations to the accuracy required for (ε, δ)-BAI.

3 Problem Setup and Preliminaries

We formalize neural architecture evaluation as a cost-sensitive, multi-fidelity best-arm identification problem over a finite candidate set. Let $\mathcal{X}$ denote a finite collection of architectures (arms) with $|\mathcal{X}| = N$. Each $x \in \mathcal{X}$ has an associated (unknown) *terminal* or *converged* mean performance $\mu_x \in [0, 1]$, which we interpret as the expected validation accuracy (or any bounded score) obtained by training x to completion under a fixed protocol. We let $x^* \in \arg\max_{x \in \mathcal{X}} \mu_x$ denote an optimal architecture and define the suboptimality gap of an arm x by

$$\Delta_x \;:=\; \mu_{x^*} - \mu_x \;\in\; [0, 1].$$

The goal is to identify an architecture whose terminal performance is near-optimal while spending as little compute as possible.

Fidelities and evaluation costs. We model partial training via a fidelity variable. Let $\mathcal{F} \subset \mathbb{R}_+$ be a set of allowed fidelities (e.g., epochs, gradient steps, tokens processed). A query consists of selecting a pair $(x, f) \in \mathcal{X} \times \mathcal{F}$ and observing a random proxy evaluation $Y(x, f) \in [0, 1]$. The cost of a query is known and additive:

$$c(x, f) \;=\; \kappa(x)\, f,$$

where $\kappa(x) > 0$ is a known per-unit-fidelity cost for architecture x (e.g., seconds per epoch). For a (possibly adaptive) sequence of queries $\{(x_t, f_t)\}_{t \geq 1}$, the accumulated cost is $\sum_t c(x_t, f_t)$. We will either (i) impose a hard budget constraint $\sum_t c(x_t, f_t) \leq C$ and study the best achievable identification accuracy under that budget, or (ii) more commonly, treat C as implicit and study the minimal (or expected) cost needed to attain a target identification guarantee.

Proxy observations: noise and one-sided bias. A central feature of our setting is that low-fidelity training underestimates terminal performance in a systematic way. Formally, for each (x, f) the random variable $Y(x, f)$ has mean $\mathbb{E}[Y(x, f)]$ and satisfies two conditions.

First, the noise is light-tailed: we assume that $Y(x, f) - \mathbb{E}[Y(x, f)]$ is σ^2-sub-Gaussian uniformly over x and f. In particular, for any m i.i.d. samples $\{Y_i(x, f)\}_{i=1}^m$ at the same pair (x, f), the empirical mean $\bar{Y}(x, f) = m^{-1} \sum_{i=1}^m Y_i(x, f)$ concentrates around $\mathbb{E}[Y(x, f)]$ at rate $O(\sqrt{\sigma^2/m})$.

Second, the proxy is biased downward relative to the terminal mean. Define the *proxy bias*

$$\beta_x(f) \;:=\; \mu_x - \mathbb{E}[Y(x, f)].$$

We assume a one-sided constraint $\mathbb{E}[Y(x, f)] \le \mu_x$, equivalently $\beta_x(f) \ge 0$, and we assume that the bias decreases with fidelity (higher budgets produce less underestimation). Crucially, we assume only an *upper envelope* on this bias of the form

$$0 \;\le\; \beta_x(f) \;\le\; g\big(s(x)\big)\, f^{-\alpha}, \tag{1}$$

where $s(x)$ is a known architecture statistic (e.g., parameter count, FLOPs, depth, or any scalar summary computed from the architecture), $g(\cdot)$ is an unknown nonnegative function, and the exponent $\alpha \in (0, 1]$ is known. The envelope (1) is deliberately permissive: it asserts only that larger $s(x)$ may entail larger bias, and that increasing f reduces the bias at a polynomial rate controlled by α. This model is compatible with the empirical observation that different architectures may require different training budgets for their rankings to stabilize, while still allowing substantial heterogeneity in learning-curve shape.

Algorithms and access model. An algorithm operates sequentially. At each round t, it selects (x_t, f_t) as a (possibly randomized) function of past observations and costs, then receives $Y(x_t, f_t)$. The algorithm has black-box oracle access to $Y(\cdot, \cdot)$, but may freely inspect $\kappa(\cdot)$ and $s(\cdot)$ without cost. The principal resource is evaluation cost; memory and arithmetic costs are secondary and will be controlled at the level of $O(N)$ bookkeeping in our constructions.

(ε, δ)-best-arm identification. Given $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$, we seek an output $\hat{x} \in \mathcal{X}$ such that

$$\mathbb{P}\left(\mu_{\hat{x}} \;\ge\; \max_{x \in \mathcal{X}} \mu_x - \varepsilon \right) \;\ge\; 1 - \delta.$$

We call an algorithm satisfying this guarantee (ε, δ)-*correct*. The performance criterion we optimize is the total evaluation cost required to achieve

(ε, δ)-correctness. In particular, we will study upper bounds on the expected cost of specific adaptive strategies and lower bounds showing that certain costs are unavoidable.

Fixed-fidelity protocols as a restricted class. To articulate the benefit of multi-fidelity adaptivity, it is convenient to define a restricted baseline class. We say an algorithm is *fixed-fidelity* if, up to constant factors, it evaluates each arm using a single common fidelity f_0 (possibly with repetitions) before producing an output. Such protocols encompass standard practices in which every candidate architecture is trained for the same number of epochs (or steps) and then compared by its resulting proxy score. Our lower-bound construction in the next section shows that, under heterogeneous bias envelopes (1), fixed-fidelity protocols can be forced to spend near-terminal budgets on many arms, whereas an adaptive multi-fidelity algorithm can concentrate high fidelities only where bias resolution is information-theoretically necessary.

4 Why Fixed Protocols Fail

We now formalize a limitation of the common "train every candidate for the same number of epochs and pick the best" protocol. The key point is that, under the one-sided bias envelope (1), *repeating low-fidelity evaluations cannot remove systematic underestimation*, and if different architectures admit different bias scales $g(s(x))$, then any single globally chosen fidelity must be large enough to accommodate the worst case. This creates instances where fixed-fidelity evaluation is either incorrect or incurs essentially linear-in-N high-fidelity cost.

A restricted class: fixed-fidelity evaluation. Fix $f_0 \in \mathcal{F}$. We call an algorithm f_0*-fixed* if all of its oracle calls use the same fidelity f_0 (possibly adaptively choosing which arm to query next and how many times to repeat, but never changing f). This models the standard NAS practice in which every candidate is trained for a fixed budget and compared by its resulting proxy score, optionally with multiple seeds.

The difficulty is that, for any arm x, the observation distribution at fidelity f_0 is centered at $\mathbb{E}[Y(x, f_0)] = \mu_x - \beta_x(f_0)$, and $\beta_x(f_0)$ is a deterministic (non-random) downward shift that does *not* average out with repetitions. Consequently, if two arms have proxy means that are nearly equal at f_0 due to different biases, then no amount of repeated sampling at f_0 can reliably recover their ordering in terms of μ_x.

A hard instance via heterogeneous bias scales. The following theorem (stated informally here, with a complete proof deferred to Appendix **??**)

captures the impossibility: for any globally fixed choice of fidelity, one can construct an instance consistent with (1) in which many suboptimal arms appear competitive at f_0 because their bias is large, forcing the evaluator either to increase the common fidelity to a much larger "hard" value or to fail (ε, δ)-correctness.

Theorem 4.1 (Fixed-fidelity lower bound under heterogeneous proxy bias). *Fix any $f_0 \in \mathcal{F}$ and consider the class of f_0-fixed algorithms. For any $N \geq 2$, there exists an instance satisfying the sub-Gaussian noise assumption and the one-sided bias envelope* (1) *(with two groups of arms having different bias scales $g(s(x))$) such that any (ε, δ)-correct f_0-fixed algorithm must incur expected total cost at least*

$$\Omega\big(N\, \kappa_{\min}\, f_{\text{hard}} \log(1/\delta)\big) \qquad \textit{for some } f_{\text{hard}} \gg f_0.$$

Equivalently, if the evaluator insists on using a single common fidelity for all arms, then there are admissible instances where correctness forces a near-terminal common fidelity for every *arm, leading to linear-in-N high-fidelity cost.*

Proof idea (indistinguishability at low fidelity). We sketch the construction and the information-theoretic argument. Split the arms into two groups. Let $x^\star$ be the true optimal arm with terminal mean $\mu_{x^\star} = \frac{1}{2} + 2\varepsilon$ and a small bias scale $g(s(x^\star)) = g_{\text{low}}$. Let the remaining $N-1$ arms be "decoys" with terminal means $\mu_x = \frac{1}{2} + \varepsilon$ (so $\Delta_x = \varepsilon$) but much larger bias scale $g(s(x)) = g_{\text{high}} \gg g_{\text{low}}$. Choose g_{high} and f_{hard} so that the envelope permits

$$\beta_x(f) \approx g_{\text{high}} f^{-\alpha} \quad \text{and} \quad \beta_{x^\star}(f) \approx g_{\text{low}} f^{-\alpha},$$

and in particular the decoys can realize a bias at f_0 large enough to cancel their true disadvantage:

$$\mathbb{E}[Y(x, f_0)] \;=\; \mu_x - \beta_x(f_0) \;\approx\; \mu_{x^\star} - \beta_{x^\star}(f_0) \;=\; \mathbb{E}[Y(x^\star, f_0)].$$

Since the noise is centered and sub-Gaussian, this makes the distributions of observations at fidelity f_0 (nearly) identical across the optimal arm and each decoy. Therefore an f_0-fixed algorithm, regardless of how many repetitions it allocates, cannot confidently distinguish whether a particular arm is truly optimal or merely appears optimal due to bias at f_0. Formally, we compare two instances that differ only in which arm has terminal mean $\frac{1}{2} + 2\varepsilon$ while maintaining identical proxy means at fidelity f_0; by a Le Cam or Fano change-of-measure argument, any decision rule based only on f_0 observations has error probability bounded away from δ.

The only way to restore identifiability is to query at a fidelity where the worst-case bias is smaller than the relevant gap. Under (1), ensuring

$\beta_x(f) \le \varepsilon/10$ for a high-bias decoy requires

$$f \;\gtrsim\; \left(\frac{g_{\text{high}}}{\varepsilon}\right)^{1/\alpha} \;=:\; f_{\text{hard}}.$$

But fixed-fidelity evaluation applies the same f to all N arms, hence it must pay cost on the order of $\sum_x \kappa(x)\, f_{\text{hard}}$, yielding the stated $\Omega(N)$ scaling (up to $\log(1/\delta)$ factors arising from the confidence requirement).

Interpretation for NAS practice. In NAS terms, the statistic $s(x)$ captures a capacity proxy (parameters, FLOPs, depth), while $g(s(x))$ quantifies how much low-budget training underestimates the eventual performance of that capacity class. The lower bound formalizes two empirical phenomena: (i) larger or more difficult-to-optimize architectures often have learning curves that rise more slowly, so their early-epoch validation scores can be systematically pessimistic; and (ii) this pessimism is not removable by averaging over seeds at the same short budget, because it is a deterministic training-budget effect rather than a stochastic fluctuation.

Consequently, a single global training budget faces an unavoidable trade-off. If it is set small, it can mis-rank high-capacity candidates relative to smaller ones (and, in the worst case, cannot be made (ε, δ)-correct). If it is set large enough to be fair to the worst-case bias scale, it wastes compute on the many arms whose bias would have resolved at much smaller fidelities. This motivates a capacity-aware multi-fidelity policy: we should *allocate fidelity as a function of $s(x)$ and refine only when needed to eliminate an arm*, rather than enforcing a uniform protocol across a heterogeneous candidate set.

5 CASH: Capacity-Aware Multi-Fidelity Successive Elimination

We now describe an adaptive policy that *varies* the fidelity as a function of architecture capacity and of the current evidence for suboptimality. The guiding constraint is the one-sided bias envelope (1): since $\mathbb{E}[Y(x, f)] \le \mu_x$, any upper confidence bound for μ_x must explicitly account for the (unknown) downward shift induced by finite fidelity. Our algorithm, which we call CASH, implements a successive-elimination template in which each arm x is evaluated only at the smallest fidelity sufficient to make its bias commensurate with the current elimination tolerance.

Calibration and the budget rule. The envelope (1) depends on the unknown function $g(\cdot)$, which scales the worst-case bias as a function of the observed statistic $s(x)$. In the idealized analysis below we may assume access to an *upper envelope* $\hat{g}$ such that $\hat{g}(s(x)) \ge g(s(x))$ for all $x \in \mathcal{X}$; in practice we obtain $\hat{g}$ by a lightweight calibration stage. Concretely, we select a small subset $\mathcal{X}_0 \subset \mathcal{X}$, evaluate each $x \in \mathcal{X}_0$ at a short ladder of fidelities $f_1 < \cdots < f_K$, and fit a conservative curve that upper-bounds the empirical bias estimates as a function of $s(x)$. Any procedure that returns a high-probability upper envelope is admissible; a convenient choice is monotone regression in s together with a union bound over the calibration points to ensure that the fitted curve is conservative simultaneously for all x.

Given $\hat{g}$, we define a *budget rule* by choosing, for each arm and each round-specific bias tolerance $\tau > 0$, the smallest fidelity

$$b_\tau(s(x)) \ := \ \min\bigl\{f \in \mathcal{F} : \ \hat{g}(s(x))\, f^{-\alpha} \le \tau\bigr\}, \tag{2}$$

with the convention that $b_\tau(s(x)) = \max \mathcal{F}$ if the set is empty. The role of (2) is purely to ensure that the bias term entering the confidence bounds is controlled at the scale τ, while respecting that different arms may require vastly different fidelities because $\hat{g}(s(x))$ may differ by orders of magnitude.

Rounds and adaptive fidelities. CASH maintains an active set $A_r \subseteq \mathcal{X}$ at round r, initialized as $A_0 = \mathcal{X}$. In each round we choose three schedules: (i) a bias tolerance τ_r decreasing in r; (ii) a statistical sample size m_r (number of repeated evaluations per arm at the chosen fidelity); and (iii) an elimination tolerance ε_r decreasing to the target ε. For each active arm $x \in A_r$ we set

$$f_r(x) \ := \ b_{\tau_r}(s(x)),$$

and we obtain m_r independent samples $Y_1(x, f_r(x)), \ldots, Y_{m_r}(x, f_r(x))$ (e.g., distinct random seeds). Let $\bar{Y}_r(x)$ denote their empirical mean.

Bias-aware confidence bounds. Because the proxy is biased downward, we treat $\bar{Y}_r(x)$ as a conservative estimate for μ_x, and we only inflate the *upper* bound by the worst-case bias. Specifically, writing

$$\mathrm{rad}_r \ := \ \sqrt{\frac{2\sigma^2 \log\bigl(4Nr^2/\delta\bigr)}{m_r}},$$

we define

$$L_r(x) \ := \ \bar{Y}_r(x) - \mathrm{rad}_r, \qquad U_r(x) \ := \ \bar{Y}_r(x) + \hat{g}(s(x))\, f_r(x)^{-\alpha} + \mathrm{rad}_r. \tag{3}$$

The choice (3) reflects the model: the sub-Gaussian noise yields a two-sided deviation term rad_r, while the one-sided bias yields an additional nonnegative term on the upper end only. The defining invariant we exploit later is that, on a suitable high-probability event (union-bounded over r and x), we have simultaneous coverage $L_r(x) \le \mu_x \le U_r(x)$ for all active arms and all rounds.

Elimination rule and termination. Given $(L_r(x), U_r(x))_{x\in A_r}$, we compute a provisional best arm according to the lower bound,

$$x_r^{\text{best}} \in \arg\max_{x\in A_r} L_r(x),$$

and we eliminate any arm whose optimistic performance is separated from this candidate by more than the round tolerance:

$$A_{r+1} \ := \ \Big\{x \in A_r :\ U_r(x) \geq L_r\big(x_r^{\text{best}}\big) - \varepsilon_r\Big\}.$$

The algorithm stops when $|A_r| = 1$ (or when the remaining arms are mutually ε-indistinguishable under the current tolerances), and outputs any $\hat{x} \in A_r$. In the analysis, the schedules are chosen so that $\varepsilon_r \downarrow \varepsilon$ and the confidence coverage holds uniformly, ensuring that (i) the optimal arm is never removed, and (ii) every arm with gap Δ_x is removed once $\tau_r + \mathrm{rad}_r + \varepsilon_r$ falls below a constant fraction of Δ_x.

Design choices and relation to NAS pipelines. Several implementation details map directly to common NAS evaluation workflows. First, the statistic $s(x)$ may be taken as parameter count, FLOPs, depth, or any scalarized embedding; CASH requires only that $s(x)$ is known before training. Second, the fidelity set $\mathcal{F}$ is typically discrete (epochs, steps, or wall-clock checkpoints); (2) then becomes a lookup to the smallest admissible checkpoint meeting the bias tolerance. Third, the unit cost $\kappa(x)$ may be measured in GPU-seconds per epoch for each architecture; incorporating $\kappa(x)$ permits the policy to avoid over-training architectures that are intrinsically slow. Finally, the per-round structure admits parallelism: at round r we evaluate all $x \in A_r$ at their selected fidelities $f_r(x)$ concurrently, aggregate $\bar{Y}_r(x)$, and then apply the elimination rule. In this sense, CASH can be viewed as a capacity-aware variant of successive halving in which the resource allocated to each configuration is not a single shared budget but a statistic-dependent budget chosen to control proxy bias.

6 Main Upper Bounds and Near-Minimax Optimality

We now record the two main analytical guarantees for CASH: (ε, δ)-correctness and an expected evaluation-cost bound that is instance-dependent through the gaps Δ_x and the bias scales $g(s(x))$. We then explain why this upper bound is unimprovable (up to logarithmic factors) under our proxy-bias model.

(ε, δ)-correctness. Fix any schedules $(\tau_r, m_r, \varepsilon_r)_{r \geq 1}$ with $\varepsilon_r \downarrow \varepsilon$ and $m_r \geq 1$. Define the concentration event

$$\mathcal{E} \ := \ \Big\{ \forall r \geq 1, \ \forall x \in \mathcal{X} : \ \big| \bar{Y}_r(x) - \mathbb{E}[Y(x, f_r(x))] \big| \leq \mathrm{rad}_r \Big\},$$

where $\mathrm{rad}_r = \sqrt{2\sigma^2 \log(4Nr^2/\delta)/m_r}$ as in (3). By sub-Gaussianity and a union bound over arms and rounds (using $\sum_{r \geq 1} r^{-2} < \infty$), we obtain $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$.

On $\mathcal{E}$, for every round r and arm x, we have simultaneously

$$\mathbb{E}[Y(x, f_r(x))] \in [\bar{Y}_r(x) - \mathrm{rad}_r, \ \bar{Y}_r(x) + \mathrm{rad}_r].$$

Since $\mathbb{E}[Y(x, f)] \leq \mu_x$ and $\mu_x - \mathbb{E}[Y(x, f)] \leq g(s(x)) f^{-\alpha} \leq \hat{g}(s(x)) f^{-\alpha}$, it follows that on $\mathcal{E}$,

$$L_r(x) = \bar{Y}_r(x) - \mathrm{rad}_r \ \leq \ \mathbb{E}[Y(x, f_r(x))] \ \leq \ \mu_x \ \leq \ \bar{Y}_r(x) + \mathrm{rad}_r + \hat{g}(s(x)) f_r(x)^{-\alpha} = U_r(x).$$

Thus (3) yields simultaneous coverage of μ_x for all arms and rounds.

We next show that the elimination rule is safe on $\mathcal{E}$. Let $x^* \in \arg\max_x \mu_x$. Consider any round r, and let $x_r^{\mathrm{best}} \in \arg\max_{x \in A_r} L_r(x)$. Since $x^* \in A_r$ until (possibly) eliminated, we have

$$L_r\big(x_r^{\mathrm{best}}\big) \ \geq \ L_r(x^*) \ \geq \ \mu_{x^*} - \big(\mu_{x^*} - L_r(x^*)\big) \ \geq \ \mu_{x^*} - 0,$$

where the last inequality uses $L_r(x^*) \leq \mu_{x^*}$. Now suppose, for contradiction, that x^* is eliminated at round r, i.e.,

$$U_r(x^*) < L_r\big(x_r^{\mathrm{best}}\big) - \varepsilon_r.$$

Using $\mu_{x^*} \leq U_r(x^*)$ and $L_r(x_r^{\mathrm{best}}) \leq \mu_{x_r^{\mathrm{best}}}$ on $\mathcal{E}$, we would obtain

$$\mu_{x^*} \ < \ \mu_{x_r^{\mathrm{best}}} - \varepsilon_r,$$

contradicting optimality of x^*. Hence x^* is never removed on $\mathcal{E}$. Finally, when the algorithm terminates, either a single arm remains or all remaining arms are ε-indistinguishable under the final tolerances; in either case, since x^* is still present and $\varepsilon_r \downarrow \varepsilon$, any output $\hat{x} \in A_r$ satisfies $\mu_{\hat{x}} \geq \mu_{x^*} - \varepsilon$ on $\mathcal{E}$. Therefore CASH is (ε, δ)-correct.

Expected evaluation-cost upper bound. We now specialize to schedules that make the preceding argument effective for elimination and yield a cost bound that decomposes into a *bias-resolution* term and a *statistical* term. For a fixed suboptimal arm $x \neq x^*$, elimination becomes possible once both (i) the worst-case proxy bias at the chosen fidelity and (ii) the statistical radius fall below a constant fraction of the gap $\Delta_x = \mu_{x^*} - \mu_x$.

Concretely, define the minimal fidelity needed to make the bias negligible at scale Δ_x:

$$f_x^{\text{bias}} := \inf\Big\{f \in \mathcal{F} : \hat{g}(s(x))\, f^{-\alpha} \le \tfrac{1}{4}\Delta_x\Big\}, \qquad \text{so that} \qquad f_x^{\text{bias}} \asymp \Big(\tfrac{\hat{g}(s(x))}{\Delta_x}\Big)^{1/\alpha}.$$

Similarly, choose m_r so that eventually $\text{rad}_r \le \Delta_x/4$, which requires

$$m_x^{\text{stat}} \asymp \frac{\sigma^2}{\Delta_x^2} \log\Big(\frac{N}{\delta}\Big)$$

up to constant and iterated-log factors arising from the round union bound. Because CASH increases fidelity only as τ_r decreases, arm x is evaluated at fidelities no larger than a constant multiple of f_x^{bias} before it is eliminated; likewise it receives only $O(m_x^{\text{stat}})$ total repeated samples before elimination. Since each evaluation at fidelity f costs $c(x, f) = \kappa(x)f$, we obtain the instance-dependent decomposition

$$\mathbb{E}[\text{Cost}] \le \tilde{O}\left(\sum_{x\neq x^*} \kappa(x)\, f_x^{\text{bias}} + \sum_{x\neq x^*} \kappa(x)\, \frac{\sigma^2}{\Delta_x^2}\right) = \tilde{O}\left(\sum_{x\neq x^*} \kappa(x)\Big(\frac{g(s(x))}{\Delta_x}\Big)^{1/\alpha} + \sum_{x\neq x^*} \kappa(x)\, \frac{\sigma^2}{\Delta_x^2}\right),$$

where in the last step we use $\hat{g} \ge g$ and absorb schedule-dependent constants and polylogarithmic factors into $\tilde{O}(\cdot)$. The first term is the compute required to shrink proxy bias below the gap, while the second is the unavoidable sampling cost needed to overcome sub-Gaussian noise.

Near-minimax optimality (matching lower bound up to logs). It remains to justify that the above cost bound is essentially tight under our assumptions. The statistical term $\sum_{x\neq x^*} \kappa(x)\sigma^2/\Delta_x^2$ follows from standard best-arm identification lower bounds via change-of-measure: for each competing arm x, distinguishing an instance where x is optimal from one where x^* is optimal requires $\Omega(\sigma^2\Delta_x^{-2}\log(1/\delta))$ effective samples, hence proportional cost when each sample incurs $\kappa(x)$-scaled expenditure.

The bias-resolution term is specific to the multi-fidelity proxy setting. We construct pairs of instances that agree on all low-fidelity proxy distributions but differ in terminal means by Δ_x, made possible by the one-sided nature of the bias constraint. If an algorithm never queries arm x at fidelity $f \gtrsim (g(s(x))/\Delta_x)^{1/\alpha}$, then the admissible bias can hide the Δ_x separation in μ by shifting $\mathbb{E}[Y(x, f)]$ downward so that observations are (nearly) indistinguishable from those of x^*. Consequently, no amount of repetition at too-small f can certify suboptimality with error probability $\le \delta$. This yields a per-arm lower bound of order $\kappa(x)(g(s(x))/\Delta_x)^{1/\alpha}$ (up to logarithms), which matches the first term of the CASH upper bound.

Taken together, these two arguments establish that our upper bound is minimax-optimal up to polylogarithmic factors in N and $1/\delta$, and that adaptively assigning heterogeneous fidelities is not merely beneficial but necessary to avoid worst-case linear-in-N over-training under heterogeneous proxy bias.

Practical instantiation in neural architecture search. To deploy CASH in a NAS pipeline, we must instantiate four ingredients that are abstract in the model: the statistic $s(x)$ used to modulate the bias bound, a discrete fidelity grid $\mathcal{F}$ compatible with the training protocol, a high-probability upper envelope $\hat{g}(\cdot)$, and the cost model $c(x, f) = \kappa(x)f$ (including the fact that $\kappa(x)$ is typically architecture-dependent and not known a priori).

Choosing the statistic $s(x)$. The role of $s(x)$ is to summarize the aspects of x that control the magnitude of proxy bias at low fidelity. In standard NAS settings, proxy bias is largely induced by optimization and learning-curve effects; empirically it correlates with model capacity and per-step optimization difficulty. We therefore choose $s(x)$ from quantities available without training, such as parameter count $\mathrm{Params}(x)$, FLOPs per example $\mathrm{FLOPs}(x)$, depth, width, or a low-dimensional vector of such features mapped to a scalar. A simple and robust choice is

$$s(x) = \log(1 + \mathrm{Params}(x)),$$

or, when data-loading dominates and compute scales superlinearly, $s(x) = \log(1 + \mathrm{FLOPs}(x))$. When multiple statistics are informative, we may set $s(x) = w^\top \phi(x)$ with $\phi(x)$ containing $(\log(1{+}\mathrm{Params}), \log(1{+}\mathrm{FLOPs}), \mathrm{depth})$ and $w \geq 0$ fixed by a small calibration fit; this preserves the required observability of $s(x)$.

Because $\hat{g}(\cdot)$ is fitted from finite data, we typically discretize s into bins and enforce monotonicity across bins. Concretely, define bins $B_1 < \cdots < B_K$ and let $b(x) \in [K]$ be the bin index of $s(x)$. We then fit a nondecreasing sequence $(\hat{g}_1, \ldots, \hat{g}_K)$ and set $\hat{g}(s(x)) = \hat{g}_{b(x)}$. This reduces variance of the calibration step and makes the envelope property easier to enforce uniformly over $\mathcal{X}$.

Selecting the fidelity grid $\mathcal{F}$. In practice, we evaluate at a discrete set of fidelities corresponding to training budgets (epochs, steps, or wall-clock minutes). We recommend a geometric grid

$$\mathcal{F} = \{f_{\min}\gamma^j : \ j = 0, 1, \ldots, J\}, \qquad \gamma > 1,$$

with $f_{\max} = f_{\min}\gamma^J$ equal to the maximum budget for which we trust μ_x to be effectively reached (or the benchmark provides the terminal score). The geometric grid is aligned with the power-law form $f^{-\alpha}$ and ensures that successive rounds increase f by a constant factor, avoiding an excessive number of near-duplicate fidelities. When the training process has an initialization transient, we choose $f_{\min}$ beyond that transient so that $Y(x, f)$ is meaningful (e.g., at least one full epoch for supervised CNNs). If the benchmark provides intermediate checkpoints at specific epochs, we set $\mathcal{F}$ to that native grid.

Estimating and upper-bounding $g(\cdot)$. The calibration step must produce $\hat{g}$ such that $g(s(x)) \leq \hat{g}(s(x))$ holds simultaneously for all $x \in \mathcal{X}$ with high probability. Since μ_x is unobserved, we avoid direct estimation of $\beta_x(f) = \mu_x - \mathbb{E}[Y(x,f)]$ and instead use differences across fidelities. For any $f_1 < f_2$, monotonicity of the bias implies

$$\mathbb{E}[Y(x,f_2)] - \mathbb{E}[Y(x,f_1)] = \beta_x(f_1) - \beta_x(f_2) \leq g(s(x))\bigl(f_1^{-\alpha} - f_2^{-\alpha}\bigr).$$

Thus, for a fixed pair (f_1, f_2), an admissible upper bound is

$$g(s(x)) \ \geq \ \frac{\mathbb{E}[Y(x,f_2)] - \mathbb{E}[Y(x,f_1)]}{f_1^{-\alpha} - f_2^{-\alpha}}.$$

We estimate the numerator by repeated evaluations at f_1, f_2 on a calibration subset $\mathcal{X}_0$ and add a concentration slack term derived from sub-Gaussianity. Writing $\bar{Y}(x,f)$ for the sample mean from m_{cal} repeats, we set

$$\widehat{g}_x := \max_{(f_1,f_2)\in\mathcal{P}} \frac{\bigl(\bar{Y}(x,f_2) - \bar{Y}(x,f_1)\bigr)_+ + \eta}{f_1^{-\alpha} - f_2^{-\alpha}}, \qquad \eta := 2\sqrt{\frac{2\sigma^2 \log(4|\mathcal{X}_0||\mathcal{P}|/\delta_{\text{cal}})}{m_{\text{cal}}}},$$

where $\mathcal{P}$ is a set of adjacent fidelity pairs (typically consecutive elements of $\mathcal{F}$) and $(\cdot)_+$ enforces nonnegativity. We then fit the binned monotone envelope by isotonic regression: among nondecreasing sequences $(g_1, \ldots, g_K)$, choose the smallest g_k such that $g_{b(x)} \geq \widehat{g}_x$ for all $x \in \mathcal{X}_0$. Finally, to guard against sampling sparsity in extreme bins, we apply a small inflation factor, e.g. $\hat{g}_k \leftarrow (1+\rho)\hat{g}_k$ with $\rho \in [0.05, 0.2]$, and use a union bound to allocate δ_{cal}.

Handling unknown α. When α is unknown, we treat it as a nuisance parameter controlling how aggressively fidelity must grow to resolve bias. Two conservative approaches are effective. First, we may fix $\alpha = 1$, which yields a valid (though potentially loose) envelope because $f^{-1} \leq f^{-\alpha}$ for $\alpha \leq 1$ and $f \geq 1$ after rescaling $f_{\min} = 1$. Second, we may run a small grid search over $\alpha \in \{\alpha^{(1)}, \ldots, \alpha^{(L)}\}$ during calibration and select the smallest α consistent with the observed learning-curve increments, which is the most conservative for bias decay. Concretely, for each candidate $\alpha^{(\ell)}$ we compute $\hat{g}^{(\ell)}$ as above and choose the pair $(\alpha^{(\ell)}, \hat{g}^{(\ell)})$ that minimizes a validation upper envelope criterion while maintaining coverage; a union bound over ℓ preserves the global δ budget. In settings where calibration data are rich (e.g. tabular NAS benchmarks), we may additionally fit α by a log–log slope of estimated increments across f, but we do not rely on such estimation for correctness.

Incorporating architecture-dependent wall-clock cost $\kappa(x)$. The factor $\kappa(x)$ is observable via short timing runs and can vary substantially with depth, width, and operator choice. We estimate $\kappa(x)$ online by timing the first mini-batch or first epoch and normalizing by fidelity units, then setting $c(x, f) = \widehat{\kappa}(x)f$. This affects CASH only through the budget accounting and through any optional cost-aware prioritization of queries within a round. If strict per-round parallelism is not required, we may further reduce expected wall-clock by ordering evaluations in increasing $\widehat{\kappa}(x)$, so that eliminations occur earlier and expensive arms are evaluated at higher fidelities only when necessary. When $\widehat{\kappa}(x)$ itself is noisy, we treat it as a deterministic upper bound by taking a high quantile of observed timings for that architecture class, ensuring that the realized spend does not exceed the planned budget except with small probability.

These design choices yield an implementable instantiation: $s(x)$ is computed from the architecture encoding, $\mathcal{F}$ is a geometric grid of training budgets, $\hat{g}$ is a monotone high-probability envelope from a small calibration set, α is fixed conservatively or selected over a finite grid, and costs use online estimates of $\kappa(x)$. Under these instantiations, the bias-aware confidence bounds remain valid and the algorithmic tradeoff between bias resolution and statistical uncertainty is realized in wall-clock compute rather than in abstract sample counts.

7 Experimental Plan

Our experiments are designed to test the two claims implicit in the theory: (i) that a fidelity-adaptive, bias-aware elimination strategy reduces total cost to achieve a target identification accuracy, and (ii) that the reduction is driven specifically by heterogeneity in proxy bias across architectures (as summarized by $s(x)$), rather than by incidental implementation choices. We therefore emphasize settings in which terminal scores μ_x and intermediate-fidelity evaluations are available (tabular benchmarks), together with a complementary "in-the-wild" global CNN search space in which we must actually train networks and measure wall-clock costs.

Benchmarks and search spaces. We consider two classes of NAS problems. First, we use tabular multi-fidelity benchmarks where, for each architecture, validation/test performance is recorded at multiple training budgets. Concretely, we evaluate on NAS-Bench-style spaces (cell-based micro search with a finite $\mathcal{X}$) and on JAHS-Bench-style spaces where architecture and (a small number of) training hyperparameters are jointly varied. In these benchmarks, the fidelity set $\mathcal{F}$ is given by recorded epoch/step checkpoints, and the terminal mean μ_x is operationally taken to be the final-budget metric provided by the benchmark (or the largest available $f_{\max} \in \mathcal{F}$). The availability of μ_x allows us to compute ground-truth identification error and compute-to-ε curves without ambiguity.

Second, we construct a global macro–micro CNN space in which an architecture x specifies both a macro skeleton (e.g. depth, stage widths, downsampling pattern) and a micro cell/operator pattern within stages. This space is continuous/large in principle; to match our finite-arm model, we instantiate $\mathcal{X}$ by sampling a large candidate pool (e.g. $N \in [10^3, 10^4]$) from the space and then run best-arm identification over this pool. Fidelity f is the number of training epochs (or optimizer steps) and cost is measured in wall-clock seconds; we estimate $\kappa(x)$ online as described previously.

Protocols and budgets. For each task/benchmark we fix (ε, δ) and a maximum budget C expressed in the same units as $\sum_t c(x_t, f_t)$. On tabular benchmarks, we take $c(x, f)$ from the benchmark-provided training-time surrogate when available; otherwise we use $c(x, f) = f$ and report results in "epoch equivalents". On real training runs, we log realized wall-clock and enforce the budget constraint with an abort rule when the cumulative spend reaches C. To isolate the effect of adaptivity, we use the same fidelity grid $\mathcal{F}$ (typically geometric) across methods, and we control for total cost rather than the number of queries.

Primary metrics. We report three families of metrics aligned with the stated objectives. *(1) Rank correlation to terminal performance.* At any time t the algorithm has produced a set of proxy observations $\{Y(x_i, f_i)\}_{i \le t}$, and hence a score for each evaluated arm (e.g. the latest $\bar{Y}(x, \cdot)$, or the upper/lower confidence bounds). For tabular benchmarks (where all μ_x are known), we compute Spearman correlation

$$\rho_t \;=\; \mathrm{Spearman}\big(\{\widehat{s}_t(x)\}_{x \in \mathcal{X}_t}, \{\mu_x\}_{x \in \mathcal{X}_t}\big),$$

where $\mathcal{X}_t$ is the set of arms evaluated up to time t and $\widehat{s}_t(x)$ is the method-specific proxy score. We report ρ_t as a function of cumulative cost, as well as top-k overlap with the true top-k arms. This metric quantifies whether the chosen fidelities improve *ordering quality*, not merely final selection.

(2) Compute-to-ε-optimal identification. Because μ_x is known on tabular benchmarks, we can compute the stopping cost

$$T_\varepsilon \;=\; \inf\Big\{t : \; \mu_{\hat{x}_t} \ge \max_{x \in \mathcal{X}} \mu_x - \varepsilon\Big\}, \qquad \mathrm{Cost}_\varepsilon \;=\; \sum_{i \le T_\varepsilon} c(x_i, f_i),$$

where $\hat{x}_t$ denotes the method's recommendation at time t. We report the empirical distribution of $\mathrm{Cost}_\varepsilon$ across random seeds. On real training tasks (where $\max_x \mu_x$ is unknown), we approximate it by the best fully-trained architecture found across all methods and seeds, and we additionally report the fully-trained performance of the final recommendation under a standardized training recipe.

(3) Robustness across datasets and seeds. For each benchmark we run multiple independent seeds controlling (a) the algorithmic randomness and (b) stochastic training noise when applicable. We report mean and standard error of the above metrics and, crucially, the *failure rate*

$$\widehat{p}_{\text{fail}} \;=\; \frac{1}{S}\sum_{s=1}^{S} \mathbf{1}\Big\{\mu_{\widehat{x}^{(s)}} < \max_x \mu_x - \varepsilon\Big\},$$

to check empirical alignment with the desired confidence level $1-\delta$.

Baselines. We compare CASH to cost-matched alternatives that isolate the role of fidelity adaptivity and bias modeling: (i) *static fixed-fidelity elimination*, which evaluates all arms at a single $f_0 \in \mathcal{F}$ (with repetitions) and performs successive elimination using standard sub-Gaussian confidence radii (no bias term); (ii) *static with high fidelity*, setting $f_0 = f_{\max}$ to represent the "train-everything" regime; and (iii) *multi-fidelity without capacity awareness*, which uses a common fidelity schedule across arms (e.g. a Hyperband-style ladder) but does not modulate f by $s(x)$.

Ablations: separating the sources of gain. We perform targeted ablations. *(A) Static versus CASH.* We hold the elimination rule fixed and only vary the fidelity assignment: constant $f_r(x) \equiv f_r$ versus $f_r(x)$ chosen by the bias tolerance condition $\hat{g}(s(x))f^{-\alpha} \leq \tau_r$. *(B) Learned versus heuristic budget rule.* We compare (i) calibration-based $\hat{g}$ (binned isotonic envelope) to (ii) a heuristic mapping $\hat{g}(s) \propto s$ (or $\propto \log(1 + \text{Params})$) and (iii) an oracle variant on tabular benchmarks where $g(s(x))$ is replaced by an empirical upper quantile of realized bias increments for that arm/bin. This isolates whether performance depends on accurately learning the envelope or merely on using any monotone capacity proxy. *(C) Sensitivity to α and to $s(x)$.* We sweep conservative choices of α and alternative statistics (Params, FLOPs, depth) to test whether the predicted compute savings persist under misspecification.

Collectively, these experiments test not only whether CASH is faster, but whether it is faster for the structural reason posited by the model: heterogeneous, capacity-dependent proxy bias that cannot be resolved by repetition at a fixed, low fidelity.

8 Discussion, Limitations, and Extensions

Our formal model isolates a single structural feature of multi-fidelity NAS: the presence of a *one-sided* proxy bias that decreases with fidelity, with magnitude modulated by an architecture statistic $s(x)$. This abstraction is intentionally narrow. It gives a clean separation between two sources of error—statistical noise (handled by concentration) and systematic underestimation (handled by paying fidelity until a bias tolerance is met). In practice, both the bias model and the cost model can fail in ways that matter for algorithmic design. We discuss the main limitations and how one might extend the framework while preserving the basic identification guarantee.

Non-monotone and non-uniform learning curves. We assume $\beta_x(f)$ is nonnegative and decreasing in f (equivalently, $\mathbb{E}[Y(x,f)]$ increases to μ_x). Empirically, learning curves can be non-monotone due to regularization schedules, optimizer instabilities, or early overfitting; moreover, validation accuracy can fluctuate even when training loss decreases. When $\mathbb{E}[Y(x,f)]$ is not monotone, a naive use of low-fidelity observations can spuriously eliminate arms that would recover at higher budgets. Two directions appear viable. First, one may replace monotonicity with a *one-sided envelope* assumption of the form $\mathbb{E}[Y(x,f)] \le \mu_x$ for all f but without requiring improvement in f. This preserves safety if we always treat $Y(x,f)$ as a lower proxy and only use it to form lower bounds, at the price of potentially slower elimination (since we cannot infer that larger f reduces bias). Second, one may explicitly model *shape constraints* on the curve, e.g. piecewise monotonicity beyond a warmup fidelity f_{warm}, or Hölder/Lipschitz regularity in f. These assumptions would permit interpolation and more aggressive scheduling, but the associated guarantees would depend on additional, less standard concentration arguments for correlated observations across fidelities.

What counts as "fidelity" beyond epochs. The present exposition takes f to be training epochs or steps and cost $c(x,f) = \kappa(x)f$. Modern training pipelines offer many alternative fidelity knobs: input resolution, dataset subsampling, augmentation strength, optimizer precision (FP16/FP8), width multipliers, or early-stopping criteria determined on the fly. These dials often produce proxies $Y(x,f)$ with biases that are *not* ordered solely by compute: e.g. increasing resolution increases cost and may reduce bias for some architectures but not others; conversely, stronger augmentation may increase cost and also change the target metric distribution. A principled generalization is to let fidelity be a *vector* $f \in \mathcal{F} \subset \mathbb{R}_+^d$ with a partial order and cost $c(x,f)$ defined on that set, together with a bias bound $\beta_x(f) \le g(s(x))\psi(f)$ for a known decreasing shape ψ. The elimination mechanism remains conceptually the same—pay cost until a target bias tolerance is reached—but the algorithmic subproblem becomes: find a near-minimal-cost f such that $\psi(f) \le \tau_r/\hat{g}(s(x))$. Even for moderate d, this becomes a constrained optimization problem that interacts with systems considerations (e.g. discrete kernel availability and hardware-dependent throughput).

Multi-objective and constrained costs (latency, energy, memory). We model cost as a single additive scalar. In deployment-oriented NAS, however, one often optimizes performance subject to constraints, e.g. inference latency $\leq L$ and peak memory $\leq M$, or one trades off multiple costs (GPU-seconds during search, energy during training, and latency at inference). There are at least two natural extensions. The first is *constrained best-arm identification*: restrict $\mathcal{X}$ to feasible arms (as estimated from $s(x)$ or measured) and run identification within that feasible set, acknowledging that constraint violation probabilities must be controlled jointly with identification error. The second is *Pareto-front identification*: treat (μ_x, ℓ_x) as a vector of objectives/costs (e.g. accuracy and latency) and aim to return an ε-approximate Pareto set. Multi-fidelity observations complicate this because low-fidelity training affects μ_x but does not directly reveal deployment costs, which may be deterministic functions of x. In such settings, $s(x)$ can serve dual roles: as a predictor of proxy bias and as a proxy for feasibility; the main open question is how to share information across these tasks without invalidating the high-probability safety event used in elimination.

Continuous or extremely large search spaces. Our theory is stated for finite $\mathcal{X}$, which matches tabular benchmarks and the "sample-a-pool-then-identify" protocol. In global NAS, $\mathcal{X}$ is effectively continuous and the pool size N may itself be a decision variable. One direction is to treat pool construction as an outer loop: sample candidates according to a proposal distribution, apply CASH to allocate training budgets efficiently, and adapt the proposal using the surviving set (a hybrid of best-arm identification and bandit optimization). Another direction is to incorporate a parametric or kernel model over architectures and share statistical strength across arms, which could reduce the $\sum_x \sigma^2/\Delta_x^2$-type dependence. Doing so while retaining robustness to misspecification is nontrivial: if we use a surrogate to "borrow" information, then confidence bounds must include model error terms, and the one-sided bias property must be preserved under aggregation.

Open problems suggested by the model. Several theoretical questions remain. (i) *Learning the envelope* $g(\cdot)$ with minimal overhead: our calibration step is heuristic, and the optimal exploration strategy to estimate a valid upper envelope under budget constraints is unclear. (ii) *Unknown* α: conservative misspecification can waste cost, but aggressive choices can break safety; adaptive, high-probability estimation of α under one-sided bias is open. (iii) *Instance-dependent scheduling*: the upper bound separates a "bias-resolution" term and a "noise-resolution" term; designing schedules that provably balance these terms without hand-tuned round parameters would strengthen the practical story. (iv) *Correlations*: evaluations of different architectures can share randomness (data order, augmentations, initialization) and thereby reduce effective noise; exploiting correlations without compromising the union-bound style guarantees is a promising but technically delicate direction.

These limitations do not negate the central message: when low-fidelity proxies are systematically pessimistic in a capacity-dependent way, compute should be allocated to *resolve bias where it matters.* The extensions above aim to bring that message closer to the full complexity of modern NAS pipelines while maintaining the explicit safety guarantees that motivate our approach.

9 Conclusion

We close by distilling what the preceding development establishes and by clarifying how the resulting *evaluation primitive* can be used as a modular component inside global NAS systems as they are plausibly engineered in the 2026 regime.

The technical message is that *multi-fidelity NAS is not merely a speedup heuristic*, but a setting in which the choice of training budget is information-theoretically coupled to the correctness of selection. Under the one-sided proxy model, low-fidelity measurements are not noisy estimates of μ_x; they are *systematically pessimistic* lower proxies whose bias can vary substantially across architectures. This observation forces a structural conclusion: there are instances where any policy that commits to a single fidelity (even if it repeats evaluations) must either pay near-maximal fidelity for essentially all arms or fail to be (ε, δ)-correct. In other words, heterogeneous learning-curve rates can turn "fair" fixed protocols into worst-case suboptimal compute allocations.

Conversely, when we allow fidelity to depend on an observable statistic $s(x)$ (and we only assume that $s(x)$ indexes an upper envelope for the bias magnitude), we can separate the cost of identification into two qualitatively different requirements. One requirement is *bias resolution*: for a suboptimal arm x, no amount of repetition at too small a fidelity can rule it out if its proxy remains within $O(\Delta_x)$ of the optimum due to admissible bias. The other requirement is *noise resolution*: once bias is made negligible relative to the gap, we still need the familiar σ^2/Δ_x^2 sampling complexity to certify elimination. CASH is designed precisely to address these two requirements in the least committal way: it increases fidelity only until the bias bound is below a round-dependent tolerance and allocates repetitions only until the statistical radius is commensurate. The resulting upper bound, and its matching lower bound up to logarithmic factors, indicate that the algorithm is not merely consistent but essentially optimal under the stated abstraction.

From a practical perspective, we view the main contribution as a *contract* for evaluation rather than a specific elimination schedule. The contract is:

for any query (x, f), the system returns an observation $Y(x, f)$ with known cost $c(x, f)$, whose expectation is a lower bound on μ_x with a one-sided bias bounded by a known shape in f multiplied by an unknown but capacity-indexed scale $g(s(x))$. The specific choice of $s(x)$ is intentionally left open: it may be parameter count, FLOPs, depth, width, token budget, an embedding of the architecture graph, or any statistic for which one can plausibly upper bound proxy pessimism. Once such a contract is accepted, we can plug it into many higher-level decision rules while retaining a clean safety argument: we only eliminate when a bias-aware upper confidence bound falls below a competing lower bound, and we ensure simultaneous coverage by a standard concentration-plus-union-bound event.

This suggests a design principle for 2026-era NAS pipelines: *treat evaluation as a first-class primitive with explicit uncertainty and bias accounting*, rather than as an opaque "train-and-score" call. In large-scale distributed settings, this matters because evaluation is typically the dominant cost and the primary failure mode. Systems already multiplex heterogeneous workloads (different architectures, batch sizes, precision modes, and data pipelines). Our model provides a way to make such heterogeneity algorithmically meaningful: it says when it is legitimate to terminate training early, and when early termination is informationally useless because bias has not been resolved. In particular, it clarifies that the relevant question is not whether an early score correlates with the final score on average, but whether the early score admits a *valid one-sided error bar* that is small enough to support elimination.

We also emphasize what the theory does *not* require. It does not require that low-fidelity ranking be good, nor that learning curves be well fit by a parametric model, nor that we can predict μ_x accurately from $s(x)$. The only place where $s(x)$ enters is through a conservative upper envelope on the bias. This is compatible with deployment realities: it is often easier to overestimate worst-case pessimism for certain architecture classes than it is to build a calibrated global predictor of final accuracy. In this sense, the approach aligns with robust optimization: we trade sharpness for validity, and we let adaptivity recover efficiency by spending additional fidelity only where conservative bounds would otherwise block elimination.

Finally, we interpret the results as a statement about modularity. Global NAS systems increasingly combine (i) proposal mechanisms that generate candidates (via evolutionary operators, LLM-guided mutation, diffusion over graphs, or learned generators), (ii) surrogate models that predict performance, and (iii) schedulers that allocate compute across candidates. The present work primarily informs component (iii): it gives a correctness-preserving scheduler for biased, noisy proxies, and it identifies the minimal information that the scheduler needs from the rest of the system. If a generator proposes a large pool and a surrogate provides a prior ranking, CASH can be used to *certify* the top of that ranking under explicit budget constraints, rather than to replace the surrogate. Conversely, if one wants to integrate surrogate predictions more aggressively, the burden is clear: any borrowed information must be incorporated into confidence bounds without violating the high-probability event that protects the optimal arm from elimination.

The broader takeaway is therefore simple and operational: when proxies are systematically pessimistic in a way that depends on architecture capacity, we must allocate compute to *resolve bias where it matters*, and we can do so while preserving (ε, δ) identification guarantees. This provides a principled foundation for the common empirical practice of training different candidates for different durations, and it isolates the assumptions under which that practice can be justified as more than a heuristic.