NON-MYOPIC ACTIVE FEATURE ACQUISITION VIA PATHWISE POLICY GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Active feature acquisition (AFA) considers prediction problems in which features are costly to obtain and the learner adaptively decides which feature values to acquire for each instance and when to stop and predict. In this paper, we introduce a continuous relaxation of the acquisition process that enables *non-myopic pathwise policy gradients* (NM-PPG) through the full acquisition trajectory, avoiding the high variance of standard score-function policy gradients while allowing end-to-end optimization of the acquisition policy. To better align training with deployment, we develop a straight-through rollout that follows discrete feature acquisitions in the forward pass while backpropagating through the corresponding soft relaxation. We derive an AFA-specific average-case upper bound on the variance of this gradient estimator, which characterizes instability and motivates staged temperature sharpening. Experiments on both synthetic and real-world datasets demonstrate that NM-PPG yields superior performance relative to state-of-the-art AFA baselines.

1 INTRODUCTION

Many predictive systems rely on features that are expensive, slow, invasive, or privacy sensitive to obtain. In medical decision support, ordering all tests for every patient may be costly and burdensome, and can delay treatment (Gorry & Barnett, 1968). In recommender systems and personalized services, collecting many behavioral signals raises both monetary and privacy costs (Jeckmans et al., 2013). In interactive troubleshooting, each diagnostic query consumes time and user effort (Heckerman et al., 1995; Chen et al., 2017). Similar constraints arise in robotics and sensor systems, where measurements require energy, time, or motion (Krause & Guestrin, 2009; Lauri et al., 2023), and in LLM-based decision making, where a model may gather information strategically before answering (Cooper et al., 2025). These settings motivate *active feature acquisition* (AFA), where a learner sequentially decides which feature values to acquire for each instance and when to stop and predict, balancing predictive quality against acquisition cost (Aronsson et al., 2026).

AFA differs fundamentally from static feature selection, where one subset is used for every instance (Guyon & Elisseeff, 2003). This can be suboptimal when informative features vary across instances. For example, a cheap feature may reveal which of several expensive features is relevant for the current instance. A static selector must retain all potentially useful features, whereas an adaptive AFA policy can first acquire the cheap indicator and then query only the relevant expensive feature (Valancius et al., 2024). AFA can therefore achieve a better cost-performance trade-off by tailoring acquisition to realized feature values. AFA can be formulated as a *partially observable Markov decision process* (POMDP) (Åström, 1965). A recent AFA survey organizes existing methods around this formulation (Aronsson et al., 2026), mirroring standard POMDP categories (Åström, 1965; Ross et al., 2008; Silver & Veness, 2010): *(i)* model-based methods estimate the acquisition dynamics, i.e., a distribution over unobserved features given observed ones, and plan using this model (Geman & Jedynak, 1996; Bilgic & Getoor, 2007; Ma et al., 2019). *(ii)* Model-free methods learn acquisition policies directly from experience through oracle-guided supervision or reinforcement learning (RL) (Dulac-Arnold et al., 2011; He et al., 2012; Janisch et al., 2020; Covert et al., 2023; Ghosh & Lan, 2023; Gadgil et al., 2024). Our method belongs to this category. *(iii)* Hybrid methods combine policy learning with a learned model to improve supervision, representations, or stability (Zannone et al., 2019; Li & Oliva, 2021; 2024; Valancius et al., 2024; Guney et al., 2025). Orthogonal to

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

these categories, *myopic* methods evaluate only the immediate benefit of an acquisition, whereas *non-myopic* methods also consider its value through subsequent acquisition decisions.

Existing non-myopic methods make complementary compromises. RL-based methods directly target long-term cost minimization in the AFA-POMDP, but generic value learning and score-function policy gradients can be unstable in its large combinatorial acquisition space (Covert et al., 2023; Gadgil et al., 2024; Schütz et al., 2026). Non-RL non-myopic methods instead exploit AFA structure to construct policies from jointly informative feature groups, improving stability but introducing bias relative to adaptive long-term cost minimization in the full AFA-POMDP (Valancius et al., 2024; Norcliffe et al., 2025). Continuous relaxations have also been considered in AFA, but are myopic: they optimize a one-step truncation of the optimal value function, so gradients span only one acquisition rather than the full trajectory (Ghosh & Lan, 2023; Covert et al., 2023) (see Appendix B). This leaves a crucial gap: *how to directly optimize long-term AFA decisions while retaining stable and computationally efficient AFA-specific learning*. We address this gap with *non-myopic pathwise policy gradients* (NM-PPG). Our contributions are therefore as follows: *(i)* We formulate a general AFA objective covering hard- and soft-budget AFA. We prove that the soft-budget cost-performance frontier dominates the hard-budget frontier (Section 2 and Appendix A). *(ii)* We introduce a continuous relaxation of this objective that enables non-myopic pathwise gradients through the entire acquisition trajectory. To reduce the relaxation gap during training, our straight-through (ST) rollout follows hard feature acquisitions in the forward pass while backpropagating through the corresponding soft relaxation (Sections 3.2–3.3). *(iii)* A straightforward pathwise-gradient estimator for maximally non-myopic planning requires $O(k^2)$ computation, where k is an upper bound on the acquisition-trajectory length. NM-PPG reduces this to $O(k)$ with a single shared forward–backward pass (Section 3.3). *(iv)* We derive an AFA-specific average-case upper bound on the variance of the gradient estimator used by NM-PPG, which characterizes when its gradients become unstable and motivates staged temperature sharpening (Section 3.4). *(v)* We demonstrate that on 12 datasets and seven baselines, NM-PPG is more stable than non-myopic methods, outperforms myopic methods under delayed feature value, and remains competitive otherwise (Section 4).

2 Problem Formulation

Let $p(\mathbf{x}, \mathbf{y})$ be the data distribution, where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d) \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$. Bold symbols denote random variables and regular symbols their realizations; thus (x, y) is sampled from $p(\mathbf{x}, \mathbf{y})$. We consider classification, so $\mathcal{Y}$ is a finite label set, although the method extends directly to regression. Acquiring feature $a \in [d] \triangleq \{1, \ldots, d\}$ costs $c(a) > 0$. For $A \subseteq [d]$, let $x_A \triangleq \{x_a : a \in A\}$ be the corresponding subvector, write $\mathbf{x}_A$ for its random counterpart, and define $c(A) \triangleq \sum_{a \in A} c(a)$.

Acquisition Process. For a single instance $x \in \mathcal{X}$, the acquisition process proceeds as follows. Because the state space grows exponentially with d, we follow prior work and restrict π_θ to at most $k \leq d$ acquisitions (Aronsson et al., 2026). Such a limit is also natural in applications that restrict the number of tests, queries, or measurements per instance. At step t, the mask $m_t \in \{0, 1\}^d$ records observed features, with $m_0 = \mathbf{0}$ and accumulated cost $c(m_t) \triangleq \sum_{a=1}^{d} m_{t,a} c(a)$. Given a per-instance budget $B \geq 0$, the feasible feature actions are $\mathcal{F}_B(m_t; k) \triangleq \{a \in [d] : t < k,\ m_{t,a} = 0,\ c(m_t) + c(a) \leq B\}$, and the available actions are

$$\mathcal{A}_B(m_t; k) \triangleq \mathcal{F}_B(m_t; k) \cup \{d + 1\}. \tag{1}$$

Here, $d + 1$ is the stopping action. Both the policy π_θ and predictor f_ϕ take the currently observed features as input, which we encode as $x(m_t) \triangleq (m_t \odot x, m_t) \in \mathbb{R}^{2d}$, where $\odot$ denotes elementwise multiplication. This representation zeros unobserved entries and concatenates m_t to distinguish them from observed zeros. The policy assigns probability $\pi_\theta(a \mid x(m_t))$ to each $a \in \mathcal{A}_B(m_t; k)$ and zero probability otherwise, while $f_\phi(x(m_t))$ outputs a distribution over $\mathcal{Y}$. Selecting $a_t \in [d]$ reveals x_{a_t} and updates $m_{t+1} = m_t + (1 - m_t) \odot \mathrm{onehot}(a_t)$, whereas selecting $a_t = d + 1$ terminates acquisition and predicts with $f_\phi(x(m_t))$.

Optimization Objective. Let $m_\theta(x)$ denote the terminal mask produced by π_θ for x. Following the general AFA formulation (Aronsson et al., 2026), we minimize

$$\min_{\theta, \phi} \mathbb{E}_{\mathbf{x}, \mathbf{y}, \pi_\theta}\big[\ell(f_\phi(\mathbf{x}(m_\theta(\mathbf{x}))), \mathbf{y}) + \underbrace{\alpha c(m_\theta(\mathbf{x}))}_{\text{soft-budget penalty}}\big] \text{ s.t. } \underbrace{c(m_\theta(x)) \leq B}_{\text{hard-budget}},\ \underbrace{\|m_\theta(x)\|_0 \leq k}_{\text{cardinality constraint}}\ \forall x \in \mathcal{X}. \tag{2}$$

The constraints are enforced directly by $\mathcal{A}_B(m_t; k)$ in (1). Unaffordable features are unavailable, and no feature action remains when $t = k$. *However, existing AFA methods consider two special cases, rather than the full objective in* (2) (Aronsson et al., 2026). We state (2) in general form so that our method can be instantiated for any setting. Following prior work, our experiments consider only these two special cases, with each training run fixing (B, α, k) to one of them. These cases are defined as follows. *(i)* In the *soft-budget* case, $B = c([d])$, making the per-instance cost budget non-binding. The policy learns when to stop for each instance through the trade-off controlled by $\alpha > 0$. *(ii)* In the *hard-budget* case, $\alpha = 0$. Since an optimal predictor can ignore extra features, stopping cannot reduce expected loss while a feature remains feasible. We therefore exclude $d + 1$ until B or k prevents further acquisition. Typically, $k = d$, so cardinality imposes no additional constraint beyond B. At equal average cost, a soft-budget policy may spend less on easy instances and more than B on difficult ones to lower average prediction loss, whereas a hard-budget policy cannot exceed B per instance. Thus, the soft-budget cost-performance frontier weakly dominates the hard-budget frontier at matched costs: an optimal soft-budget solution with the same expected cost has no greater expected loss. A recent AFA benchmark notes that prior work often compares soft-budget methods with hard-budget baselines (Schütz et al., 2026). Any apparent gain may then reflect spending more than B on difficult instances rather than a better learning method. Proposition 3 in Appendix A formalizes the empirical recommendation of the benchmark to compare methods within each budget setting. The appendix also discusses when each objective is appropriate.

3 NM-PPG: Non-Myopic Pathwise Policy Gradients

In this section, we present *non-myopic pathwise policy gradients* (NM-PPG) for full-trajectory AFA.

3.1 AFA as a Finite-Horizon POMDP

The process in Section 2 is a finite-horizon POMDP (Rückstieß et al., 2013; Aronsson et al., 2026). Its latent state is (m_t, x, y), the policy observes $x(m_t)$, and a feature action reveals its value and updates the mask. Define the immediate cost by $C((m_t, x, y), a) \triangleq \alpha c(a)$ for $a \in \mathcal{F}_B(m_t; k)$ and $C((m_t, x, y), d + 1) \triangleq \ell(f_\phi(x(m_t)), y)$. Under π_θ, each action is sampled as $a_t \sim \pi_\theta(\cdot \mid x(m_t))$, with support restricted to $\mathcal{A}_B(m_t; k)$. Let $T_\theta(x) \leq k$ denote the terminal step for x. The full trajectory return and expected objective are

$$G(x, y, \pi_\theta) \triangleq \textstyle\sum_{t=0}^{T_\theta(x)} C((m_t, x, y), a_t), \qquad J(\pi_\theta) \triangleq \mathbb{E}_{\mathbf{x},\mathbf{y},\pi_\theta}[G(\mathbf{x}, \mathbf{y}, \pi_\theta)]. \tag{3}$$

Theorem 1. *For fixed ϕ, minimizing $J(\pi_\theta)$ over θ is equivalent to the corresponding optimization over θ in* (2)*; $\alpha = 0$ and $B = c([d])$ recover its hard- and soft-budget special cases.*

All proofs are in Appendix C. Theorem 1 shows that standard finite-horizon RL tools can be used to optimize the AFA objective. To formalize lookahead, let $1 \leq h \leq k$ denote the planning depth. The deployment horizon k limits how many features may be acquired, whereas h determines how many future acquisition steps are considered when choosing the current action. Let $V_0^*(x(m_t)) \triangleq \mathbb{E}_{\mathbf{y}|x(m_t)}[\ell(f_\phi(x(m_t)), \mathbf{y})]$ be the cost of stopping immediately. For $h \geq 1$, define the optimal value $V_h^*(x(m_t))$ recursively by

$$V_h^*(x(m_t)) = \min\Big\{\underbrace{V_0^*(x(m_t))}_{a=d+1}, \min_{a \in \mathcal{F}_B(m_t;k)} \big[\alpha c(a) + \mathbb{E}_{\mathbf{x}_a|x(m_t)}[V_{h-1}^*(x(m_{t+1}))]\big]\Big\}. \tag{4}$$

For $h \geq 1$, the corresponding optimal state-action values are $Q_h^*(x(m_t), d + 1) \triangleq V_0^*(x(m_t))$ and, for $a \in \mathcal{F}_B(m_t; k)$, $Q_h^*(x(m_t), a) \triangleq \alpha c(a) + \mathbb{E}_{\mathbf{x}_a|x(m_t)}[V_{h-1}^*(x(m_{t+1}))]$.

Definition 1 (Planning depth)**.** For $1 \leq h \leq k$, the optimal receding-horizon policy $\pi_{h,k}^*$ selects at each decision step $0 \leq t < k$ an action in $\arg\min_{a \in \mathcal{A}_B(m_t;k)} Q^*_{\min\{h,k-t\}}(x(m_t), a)$. We call $h = 1$ *myopic*, $h > 1$ *non-myopic*, and $h = k$ *maximally non-myopic*.

At each step, the receding-horizon policy executes only the first action of the truncated planning problem and replans from the resulting state. Example 1 illustrates the benefit of non-myopic planning with $h > 1$.

Example 1 (Delayed feature value)**.** Suppose a feature x_c is marginally independent of the label, but its value identifies which of several remaining features is predictive. Acquiring x_c alone does not reduce V_0^*, so a myopic policy with $h = 1$ has no reason to select it. With $h > 1$, however, $Q_h^*(x(m_t), c)$ accounts for the subsequent acquisition selected after observing x_c, allowing the policy to value x_c through the decision it enables.

We use $h = k$ in NM-PPG for two reasons. *(i)* As Example 1 illustrates, delayed interactions may occur at an unknown depth. Only $h = k$ guarantees that every such interaction can be captured and that $\pi_{k,k}^* \in \arg\min_\pi J(\pi)$ (which is optimal for (2) by Theorem 1). *(ii)* Setting $h = k$ permits the shared-trajectory gradient estimator in Proposition 2, which computes maximally non-myopic gradients in $O(k)$ rather than $O(k^2)$ time. The synthetic datasets considered in our experiments instantiate this structure, making myopic acquisition suboptimal. When no feature has delayed value, a myopic policy may be optimal. Conversely, there exist data distributions on which it performs arbitrarily worse than a non-myopic policy (Chen et al., 2015). Appendix F discusses myopic and non-myopic planning in greater detail.

3.2 Differentiable Full-Trajectory Rollouts

The *policy gradient theorem* (PGT) makes gradients computable despite discrete acquisition actions and mask transitions, using the score function $\nabla_\theta \log \pi_\theta(a_t \mid x(m_t))$ (Sutton & Barto, 2018). Its estimator is unbiased, but often has high variance (Mohamed et al., 2020; Ghosh & Lan, 2023; Voelcker et al., 2026). The broader RL literature instead obtains lower-variance pathwise gradients by backpropagating through reparameterized actions and differentiable environment dynamics, which generally requires a differentiable simulator or learned transition model (Heess et al., 2015; Zhang et al., 2023b; Voelcker et al., 2026). *Crucially, we observe that AFA has unusually favorable structure for pathwise optimization.* The training dataset contains the complete feature vector, so selecting feature a reveals the already available value x_a, while the next mask is the known deterministic update in Section 2. Consequently, pathwise optimization of AFA does not require learning a transition model: only the discrete feature choice and mask update prevent direct differentiation. We exploit this AFA-specific structure by relaxing these two operations over the complete acquisition trajectory.

Starting from $m_0 = \mathbf{0}$, we generate a conditional feature-acquisition path through an endpoint $T \le k$, where the deployment horizon k or budget B prevents another acquisition. Learned stopping does not truncate this path; it is represented by the survival weights introduced below so that every possible stopping step can be evaluated differentiably. We replace the binary mask $m_t \in \{0,1\}^d$ with a soft mask $\tilde{m}_t \in [0,1]^d$, initialized by $\tilde{m}_0 = m_0$. At each step $0 \le t < T$, the policy network outputs logits $z_t = z_\theta(x(\tilde{m}_t)) \in \mathbb{R}^{d+1}$, whose first d entries correspond to features and final entry to stopping. Using the hard mask m_t, logits of actions outside $\mathcal{A}_B(m_t; k)$ are set to $-\infty$. Define the corresponding categorical policy as $\pi_t \triangleq \pi_\theta(\cdot \mid x(\tilde{m}_t)) = \mathrm{softmax}(z_t/\tau_{\mathrm{hard}})$, where $\pi_{t,a}$ denotes the probability of action a and $\tau_{\mathrm{hard}} > 0$. We obtain a differentiable relaxed action over feature acquisition and stopping using the Gumbel–Softmax reparameterization (Jang et al., 2017; Maddison et al., 2017):

$$\tilde{a}_t = \mathrm{softmax}((z_t/\tau_{\mathrm{hard}} + \varepsilon_t)/\tau_{\mathrm{soft}}), \qquad \varepsilon_t \sim \mathrm{Gumbel}(0,1)^{d+1}. \tag{5}$$

Here, $\tilde{a}_{t,d+1}$ is the relaxed stopping mass and $\tilde{a}_{t,1:d}$ is the feature-acquisition mass. Conditional on continuing, the relaxed feature distribution and corresponding hard feature action are

$$\tilde{r}_t = \mathrm{softmax}((z_{t,1:d}/\tau_{\mathrm{hard}} + \varepsilon_{t,1:d})/\tau_{\mathrm{soft}}), \qquad r_t = \mathrm{onehot}(\mathrm{argmax}_{a \in \mathcal{F}_B(m_t;k)} \tilde{r}_{t,a}). \tag{6}$$

For finite $\tau_{\mathrm{soft}} > 0$, $\tilde{r}_t = \tilde{a}_{t,1:d}/(1-\tilde{a}_{t,d+1})$. By the Gumbel–Softmax reparameterization, obtaining the hard action in (6) is equivalent to sampling $r_t \sim \mathrm{softmax}(z_{t,1:d}/\tau_{\mathrm{hard}})$, the feature policy conditioned on continuing. Thus, τ_{hard} defines the categorical policy, whereas τ_{soft} controls the smoothness of its continuous relaxation. Smaller values concentrate $\tilde{a}_t$ and $\tilde{r}_t$ near one-hot vectors. The hard mask is updated as $m_{t+1} = m_t + (1-m_t) \odot r_t$ and determines the budget-feasible forward path. When $\alpha > 0$, the policy must also learn the discrete stopping decision. We therefore introduce the survival mass $\tilde{s}_t \in [0,1]$, the relaxed probability that acquisition has not stopped before step t. The relaxed acquisition process evolves according to

$$\tilde{m}_{t+1} = \tilde{m}_t + (1-\tilde{m}_t) \odot \tilde{r}_t, \qquad \tilde{s}_{t+1} = \tilde{s}_t(1-\tilde{a}_{t,d+1}) = \textstyle\prod_{j=0}^{t}(1-\tilde{a}_{j,d+1}), \tag{7}$$

for $0 \le t < T$, with $\tilde{m}_0 = m_0$ and $\tilde{s}_0 = 1$. As discussed in Section 2, the hard-budget case ($\alpha = 0$) does not require learned early stopping. We therefore set the stopping logit to $-\infty$, so $\tilde{a}_{t,d+1} = 0$

and all survival mass continues, giving $\tilde{s}_t = 1$ throughout the rollout. Intuitively, $\tilde{m}_t$ records how strongly each feature has been acquired: the update retains existing acquisition mass and applies $\tilde{r}_t$ only to each feature's unacquired fraction. Related survival-based stopping distributions are used in adaptive computation to learn when to halt an otherwise differentiable internal computation (Chen et al., 2020; Banino et al., 2021). In AFA, however, continuing must also select and reveal a costly feature, update the observation mask, and thereby change all later policy inputs and feasible actions. Let $\boldsymbol{\varepsilon} = (\varepsilon_0, \ldots, \varepsilon_{k-1})$ denote the sequence of independent Gumbel noise vectors in (5). With $\tilde{\ell}_t \triangleq \ell(f_\phi(x(\tilde{m}_t)), y)$ and $c(\tilde{r}_t) \triangleq \sum_a \tilde{r}_{t,a} c(a)$, the relaxed full-trajectory cost is

$$\tilde{G}(x, y, \theta, \boldsymbol{\varepsilon}) \triangleq \sum_{t=0}^{T-1} \tilde{s}_t \big[\alpha(1 - \tilde{a}_{t,d+1}) c(\tilde{r}_t) + \tilde{a}_{t,d+1} \tilde{\ell}_t\big] + \tilde{s}_T \tilde{\ell}_T. \tag{8}$$

The product $\tilde{s}_t \tilde{a}_{t,d+1}$ is the relaxed probability of stopping at step t and weights the current prediction loss. The complementary continuation probability weights the cost of another acquisition and propagates survival to the next decision. The final term assigns any remaining mass to prediction at T. The policy therefore learns both which feature to acquire conditional on continuing and whether its expected future predictive benefit justifies its cost. Define $\tilde{J}(\theta) \triangleq \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}[\tilde{G}(\mathbf{x}, \mathbf{y}, \theta, \boldsymbol{\varepsilon})]$.

Theorem 2 (Relaxed trajectory consistency)**.** *Assume the policy logits and predictor loss are continuous in the relaxed mask, with bounded loss. Then* (i) $\sum_{t<T} \tilde{s}_t \tilde{a}_{t,d+1} + \tilde{s}_T = 1$*;* (ii) *one-hot* $\tilde{a}_t$ *for all* $t < T$ *implies* $\tilde{G} = G$*;* (iii) $\lim_{\tau_{\text{soft}} \to 0} \tilde{G} = G$ *almost surely; and* (iv) $\lim_{\tau_{\text{soft}} \to 0} \tilde{J}(\theta) = J(\pi_\theta)$.

Properties *(i)–(ii)* show that the relaxation defines a normalized stopping distribution and exactly recovers the discrete trajectory in (3) for one-hot actions. Properties *(iii)–(iv)* establish trajectory-level and expected-objective consistency. Thus, $\tilde{G}$ is a principled continuous relaxation of the full AFA trajectory, motivating annealing τ_{soft} toward zero during training.

3.3 Straight-Through Rollouts and Efficient Gradients

The relaxed rollout evaluates later policy and predictor decisions at fractional masks $\tilde{m}_t$, although deployment visits only binary masks. It may therefore exploit partially acquired features that do not exist in the discrete AFA process, creating relaxation bias and a train–deployment mismatch. Straight-through (ST) estimators are widely used to reduce such a mismatch: they use hard discrete variables in the forward pass and a smooth surrogate derivative in the backward pass, and are empirically effective for discrete stochastic computation and categorical latent-variable models (Bengio et al., 2013; Jang et al., 2017). We therefore define

$$\bar{r}_t = r_t - \mathrm{sg}(\tilde{r}_t) + \tilde{r}_t, \qquad \bar{m}_{t+1} = m_{t+1} - \mathrm{sg}(\tilde{m}_{t+1}) + \tilde{m}_{t+1}, \tag{9}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient, i.e., $\mathrm{sg}(u) = u$ and $\partial\, \mathrm{sg}(u)/\partial u = 0$. Thus $\bar{r}_t = r_t$ and $\bar{m}_t = m_t$ in the forward pass. In the ST rollout, we accordingly evaluate $z_t = z_\theta(x(\bar{m}_t))$ and compute $\tilde{a}_t$ and $\tilde{r}_t$ from this logit vector using (5)–(6). Costs are therefore evaluated on discrete feature choices, and both the policy and predictor receive valid binary masks. In the backward pass, $\partial \bar{r}_t / \partial \tilde{r}_t = I$ and $\partial \bar{m}_t / \partial \tilde{m}_t = I$, preserving the differentiable path through the relaxed trajectory. Let $\bar{\ell}_t \triangleq \ell(f_\phi(x(\bar{m}_t)), y)$. We define $\bar{G}(x, y, \theta, \boldsymbol{\varepsilon})$ as in (8), with $(\tilde{r}_t, \tilde{m}_t, \tilde{\ell}_t)$ replaced by $(\bar{r}_t, \bar{m}_t, \bar{\ell}_t)$. Define $\bar{J}(\theta) \triangleq \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}[\bar{G}(\mathbf{x}, \mathbf{y}, \theta, \boldsymbol{\varepsilon})]$ and $\nabla_\theta^{\mathrm{ST}} \bar{J}(\theta) = \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}[\nabla_\theta^{\mathrm{ST}} \bar{G}(\mathbf{x}, \mathbf{y}, \theta, \boldsymbol{\varepsilon})]$. Because $\bar{r}_t$ and $\bar{m}_t$ use stop-gradient, this is an ST surrogate, not the exact gradient of the hard-forward objective.

Proposition 1 (Full-trajectory ST gradient)**.** *For a fixed rollout* $(x, y, \boldsymbol{\varepsilon})$*, the ST surrogate gradient of* $\bar{G}$ *expands as* $\nabla_\theta^{\mathrm{ST}} \bar{G}(x, y, \theta, \boldsymbol{\varepsilon}) = \sum_{t=0}^{T} \Big(\frac{\partial \bar{G}}{\partial \bar{r}_t} \frac{\partial \bar{r}_t}{\partial \tilde{r}_t} \frac{\partial \tilde{r}_t}{\partial \theta} + \frac{\partial \bar{G}}{\partial \tilde{a}_{t,d+1}} \frac{\partial \tilde{a}_{t,d+1}}{\partial \theta} + \frac{\partial \bar{G}}{\partial \bar{m}_t} \frac{\partial \bar{m}_t}{\partial \tilde{m}_t} \frac{\partial \tilde{m}_t}{\partial \theta} + \frac{\partial \bar{G}}{\partial \tilde{s}_t} \frac{\partial \tilde{s}_t}{\partial \theta} \Big)$.

In practice, this gradient is computed directly by automatic differentiation. We state the expansion to show how gradients propagate through feature selection, mask updates, stopping, and survival. The ST derivative remains a biased surrogate, but improves alignment with discrete deployment while retaining lower-variance pathwise credit assignment (Bengio et al., 2013; Jang et al., 2017). Proposition 2 characterizes the computational complexity of the gradient estimator used by NM-PPG.

Proposition 2 (Linear-time NM-PPG gradient)**.** *For* $1 \le h \le k$*, computing a separate* h*-step pathwise gradient from every visited state costs* $O(kh)$*. NM-PPG's full-trajectory estimator* $\nabla_\theta^{\mathrm{ST}} \bar{G}$ *from Proposition 1 instead computes its maximally non-myopic gradient in a single* $O(k)$ *forward–backward pass, rather than* $O(k^2)$.

Proposition 2 gives a second reason to focus on $h = k$. A direct implementation of h-step planning starts a differentiable rollout from each of up to k visited states, costing $O(kh)$ and therefore $O(k^2)$ when $h = k$. The full-horizon case has special structure, however: every decision is optimized against the same remaining trajectory ending at T. NM-PPG represents this trajectory once, and one backward pass propagates each later cost and loss to every earlier decision that can influence it, reducing the computation to $O(k)$. For $h < k$, the rollout beginning at step t instead ends at $T_{t,h} = \min\{t+h, T\}$, so different decisions generally have different cutoffs. No single shared return therefore yields all of these truncated gradients through standard backpropagation. Appendix B presents the direct $O(kh)$ extension using overlapping h-step rollouts. Whether these gradients can instead be computed in $O(k)$ for arbitrary h remains unclear.

3.4 Theoretical Analysis

We next exploit AFA structure to characterize the variance of the full-trajectory estimator. Retain T, let $\mathcal{A}_t$ denote the actions left unmasked at step t. Let $P_t \triangleq I_{|\mathcal{A}_t|} - |\mathcal{A}_t|^{-1}\mathbf{1}\mathbf{1}^\top$ be the centering matrix over these actions, and define the relative-logit sensitivity $\Lambda_t \triangleq \|P_t\, \partial z_{t,\mathcal{A}_t}/\partial \bar{m}_t\|_2$. The feature policy conditioned on continuing is $\pi^F_{t,a} \triangleq \pi_{t,a}/(1 - \pi_{t,d+1})$ for $a \in \mathcal{F}_B(m_t; k)$. Let a^F_t denote the feature index selected by r_t in (6). The pathwise ambiguity $\eta_t \triangleq 1 - \pi^F_{t,a^F_t} + \pi_{t,d+1}(1 - \pi_{t,d+1})$ combines ambiguity among features and between stopping and continuing. Let $\mathcal{B} = \{(x^{(n)}, y^{(n)})\}_{n=1}^N$ be an i.i.d. minibatch from $p(\mathbf{x}, \mathbf{y})$ and $\mathcal{E} = \{\boldsymbol{\varepsilon}^{(n)}\}_{n=1}^N$ independent Gumbel trajectories. Define $\widehat{g} \triangleq N^{-1} \sum_{n=1}^N \nabla^{\mathrm{ST}}_\theta \bar{G}(x^{(n)}, y^{(n)}, \theta, \boldsymbol{\varepsilon}^{(n)})$.

Theorem 3 (AFA-specific average-case ST gradient variance)**.** *For fixed $\tau_{\mathrm{hard}} > 0$, let $\bar{\tau} < 1$ satisfy $0 < \tau_{\mathrm{soft}} \le \bar{\tau}$, and assume the policy-logit and predictor derivatives, loss, and feature costs are uniformly bounded along feasible rollouts. Let $C_1 := 4(1 + \sqrt{2})/(1 - \bar{\tau})$ and $C_2 := 2/(1 - \bar{\tau})$. Then there exists a finite $C_0 > 0$, independent of N, k, and τ_{soft}, such that, with $\rho_t \triangleq C_1\eta_t\Lambda_t/\tau_{\mathrm{hard}} + C_2\eta_t\Lambda_t^2/(\tau^2_{\mathrm{hard}}\tau_{\mathrm{soft}})$,*

$$\mathbb{E}_{\mathcal{B},\mathcal{E}}\left[\|\widehat{g} - \mathbb{E}_{\mathcal{B},\mathcal{E}}[\widehat{g}]\|_2^2\right] \le \frac{C_0 k}{N}\, \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}\left[\sum_{t=0}^{T-1} \tilde{s}_t^2 \eta_t \frac{(T-t)^2}{\tau_{\mathrm{soft}}} \exp\Big(\sum_{j=t+1}^{T-1} \rho_j\Big)\right]. \tag{10}$$

Theorem 3 combines two complementary perspectives from prior theory in a bound tailored to AFA's transition dynamics. *(i) Local ST variance: worst case versus average case.* For one ST decision, the squared norm of the Gumbel–Softmax Jacobian is $O(1/\tau^2_{\mathrm{soft}})$ in the pointwise worst case. This occurs only when perturbed logits are within $O(\tau_{\mathrm{soft}})$ of a tie, an event whose probability is also $O(\tau_{\mathrm{soft}})$. Averaging over relaxation noise therefore improves the local rate to $O(1/\tau_{\mathrm{soft}})$ (Shekhovtsov, 2021), as reflected by each local contribution in (10). *(ii) Sequential pathwise amplification.* General differentiable-MDP analyses replace local policy and transition derivatives by global worst-case Lipschitz bounds; repeating the same gain at every step can yield exponential growth with rollout length (Zhang et al., 2023a;b). AFA's known acquisition and survival transitions have derivative norms at most one, so only the policy's response to the acquired-feature state can amplify a perturbation. This structure permits an average-case bound along realized trajectories rather than a repeated worst-case gain.

The bound separates two effects. First, $\tilde{s}_t^2\eta_t(T-t)^2/\tau_{\mathrm{soft}}$ describes the contribution originating at decision t: survival $\tilde{s}_t^2$ downweights rarely reached decisions, ambiguity η_t captures uncertainty among features or between continuing and stopping, and $1/\tau_{\mathrm{soft}}$ captures sensitivity to sharpening. The remaining-horizon factor $(T-t)^2$ accounts for how far a perturbation can propagate. Small temperatures are therefore most problematic at frequently reached, ambiguous decisions with many downstream consequences. Second, the exponential term bounds amplification through later decisions. Each ρ_j combines ambiguity η_j with sensitivity of relative action preferences to the acquired-feature mask, Λ_j. Since common logit shifts leave action probabilities unchanged, the projection P_j removes their contribution to the sensitivity term without altering action gradients, yielding a potentially tighter variance bound. Such sensitivity can be useful: in Example 1, observing x_c should change the preferred follow-up feature. When it coincides with ambiguity across successive decisions, however, perturbations to earlier acquisitions can be amplified through later masks and policy decisions. The bound thus identifies repeated ambiguity and mask sensitivity at reached decisions as sources of exponential amplification. This motivates staged training: *learn acquisition preferences*

at a smoother temperature, then warm-start sharper stages after those preferences become more decisive. Finally, k reflects summation across steps and $1/N$ minibatch averaging. Appendix E.4.4 empirically confirms substantially lower variance than standard score-function estimators.

3.5 Training and Inference

Although decisive policies reduce the ambiguity terms in Theorem 3, premature policy collapse can prevent exploration. We therefore add a normalized survival-weighted entropy bonus to the ST objective (Ziebart et al., 2008; Haarnoja et al., 2018). Define $\mathcal{H}(\pi_t) \triangleq -\sum_{a\in\mathcal{A}_t} \pi_t(a)\log\pi_t(a)$. Both sums below omit steps with only one available action. Set $\bar{H}(x,\theta,\boldsymbol{\varepsilon}) \triangleq \big(\sum_{t<T} \mathrm{sg}(\tilde{s}_t)\mathcal{H}(\pi_t)/\log|\mathcal{A}_t|\big)/\big(\sum_{t<T}\mathrm{sg}(\tilde{s}_t)\big)$, with zero bonus if the denominator vanishes. The resulting objective is $\bar{J}_{\mathrm{ent}}(\theta) \triangleq \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}[\bar{G}(\mathbf{x},\mathbf{y},\theta,\boldsymbol{\varepsilon}) - \lambda_{\mathrm{ent}}\bar{H}(\mathbf{x},\theta,\boldsymbol{\varepsilon})]$, where $\lambda_{\mathrm{ent}} \geq 0$ controls exploration. Normalization keeps $\bar{H} \in [0,1]$ across action spaces and trajectory lengths. We detach the policy's mask inputs and survival weights so entropy gradients remain local to each decision. Moreover, motivated by Theorem 3, training starts with a smooth relaxation and progressively decreases τ_{soft}, allowing acquisition preferences to develop before sharpening. Each stage starts from the best validation checkpoint across all completed stages, while τ_{hard} remains fixed. The predictor is pretrained on random masks and subsequently trained on states encountered during acquisition. Appendix D details staged training (Algorithm 1), the ST rollout from Section 3.3 (Algorithm 2), and policy inference on test instances (Algorithm 3).

4 Experiments

We present our experimental setup and results, with details and extended results in Appendix E.

Datasets. We use 12 datasets from prior AFA work, covering controlled synthetic, real-world tabular, medical, and image tasks. The synthetic datasets are two *Cube-NM* variants with noise levels $\sigma \in \{0.1, 0.2\}$ (Schütz et al., 2026) (see Appendix E.2) and *Syn1* and *Syn3* (Norcliffe et al., 2025); each contains delayed feature interactions for which myopic acquisition is suboptimal. The real-world datasets are *Connect4* (Tromp, 1995), *Splice* (uci, 1991), *EngineFaultDB* (Vergara et al., 2023), *Metabric* (Curtis et al., 2012; Pereira et al., 2016), *Mortality* (Centers for Disease Control and Prevention, 2026; Erion et al., 2022), *Diabetes* (Centers for Disease Control and Prevention, 2026; Kachuee et al., 2019), *MNIST* (LeCun et al., 1998), and *Fashion-MNIST* (Xiao et al., 2017). Cube-NM, Mortality, and Diabetes have non-uniform acquisition costs, while the remaining datasets use unit costs. Further details are provided in Appendix E.2. **Baselines.** We compare our proposed method, NM-PPG, against *(i)* three myopic methods, *GDFS* (Covert et al., 2023), *DiFA* (Ghosh & Lan, 2023), and *DIME* (Gadgil et al., 2024), *(ii)* two non-myopic RL methods, *GSMRL* (Li & Oliva, 2021) (using PPO (Schulman et al., 2017)) and *OL* (Kachuee et al., 2019) (using DQN (Mnih et al., 2015)), and *(iii)* two approximate non-myopic methods, *AACO* (Valancius et al., 2024) and *SEFA* (Norcliffe et al., 2025). GDFS and DiFA are the only prior AFA methods that consider continuous relaxations of the acquisition process. Appendix B details the contributions of NM-PPG relative to these methods. As motivated after (2) and discussed in more detail in Appendix A, we report hard- and soft-budget results separately. NM-PPG and DIME natively support both objectives. GDFS, DiFA, OL, and SEFA are natively hard-budget, while GSMRL and AACO are natively soft-budget. We evaluate every baseline under both settings by adapting its acquisition and termination rules as needed. GDFS, DiFA, and SEFA remain restricted to uniform feature costs, whereas the remaining methods support non-uniform feature costs. See Appendix E.1 and Table 2 for details.

Hyperparameters. For soft-budget experiments, we use horizon $k = \min\{d, 30\}$. Under hard budgets, we set $k = d$, leaving acquisition limited only by B (see Section 2 for discussion). NM-PPG uses $\tau_{\mathrm{hard}} = 1$ and the staged schedule $\tau_{\mathrm{soft}} \in [0.8, 0.5, 0.2, 0.05, 0.02]$. Both the policy and predictor are neural networks. Their architectures and all remaining hyperparameters, including the entropy coefficient, are specified in Appendix E.3. Each temperature stage runs for up to $2{,}000$ epochs with early-stopping patience 150. Baselines use their published settings and architectures. **Training and Evaluation.** We report mean test-set acquisition cost and predictive performance across five random seeds, using accuracy for balanced datasets and macro-F1 for imbalanced datasets. Hard-budget curves vary B, while soft-budget curves vary α. We summarize each method on each dataset by normalized cost–performance AUC.

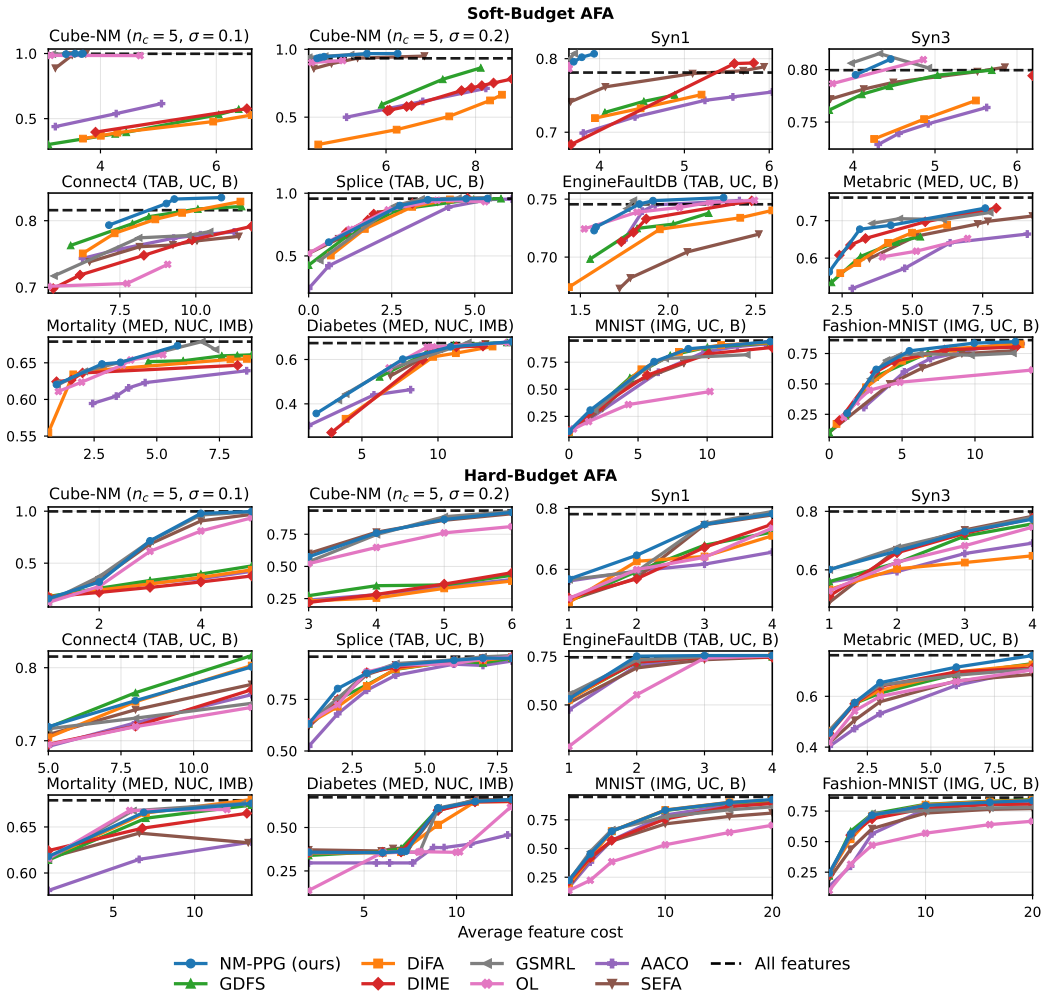

Figure 1: Metrics are accuracy (balanced) and macro-F1 (imbalanced). Real-world titles use TAB/MED/IMG (tabular/medical/image), NUC/UC (non-uniform/uniform cost), and IMB/B (imbalanced/balanced). Appendix E.4.1 provides enlarged plots with seed variance.

Main Results. Figure 1 reports the cost–performance curves, and Table 1 summarizes normalized AUC. NM-PPG has the highest overall AUC under both objectives, with 0.768 under soft budgets and 0.684 under hard budgets. Table 4 in Appendix E.4.2 reports AUC for each dataset and shows that NM-PPG is the most consistently strong method across datasets and objective settings. Moreover, every method has higher overall AUC under soft budgets in Table 1, which motivates reporting the two settings separately, as discussed after (2). For example, on Cube-NM with $\sigma = 0.2$, NM-PPG reaches 0.945 accuracy at average soft-budget cost 4.61, versus 0.759 and 0.865 at hard budgets 4 and 5. Comparable hard-budget accuracy, 0.949, requires budget 7. On Connect4, it reaches 0.832 accuracy at cost 9.29 versus 0.801 at hard budget 12. On Diabetes, it reaches 0.657 macro-F1 at cost 10.84 versus 0.609 and 0.661 at hard budgets 9 and 13, respectively. Thus, even when myopic methods are competitive under hard budgets, NM-PPG can match their performance at substantially lower average cost under soft budgets through non-myopic optimization of instance-adaptive stopping. **Additional Results.** Appendix E.4 provides expanded results and analyses of ablations, gradient variance, acquisition paths, runtime, and training dynamics.

Comparison with Non-Myopic Methods. Existing non-myopic approaches sacrifice either scalability or alignment with the sequential AFA objective. GSMRL and OL use generic PPO and DQN estimators, which prior AFA work reports can become unstable or difficult to scale as dimensionality d grows (Gadgil et al., 2024; Valancius et al., 2024; Norcliffe et al., 2025). On Connect4,

Table 1: Average AUC across datasets (higher is better). Best and second-best results are bold and underlined. Appendix E.4.2 (Table 4) reports dataset-level results and variance across seeds.

Soft-budget AFA								
Average	NM-PPG (ours)	GDFS	DiFA	DIME	GSMRL	OL	AACO	SEFA
Synthetic	<u>0.894</u>	0.664	0.596	0.583	**0.895**	0.876	0.657	0.873
Real-world	**0.705**	0.682	0.675	0.675	<u>0.687</u>	0.623	0.628	0.623
Overall	**0.768**	0.676	0.649	0.645	<u>0.756</u>	0.707	0.638	0.706
Hard-budget AFA								
Average	NM-PPG (ours)	GDFS	DiFA	DIME	GSMRL	OL	AACO	SEFA
Synthetic	**0.667**	0.480	0.436	0.453	<u>0.664</u>	0.597	0.443	0.647
Real-world	**0.689**	<u>0.686</u>	0.681	0.671	0.674	0.594	0.629	0.653
Overall	**0.682**	0.617	0.599	0.598	<u>0.671</u>	0.595	0.567	0.651

soft-budget AUC is 0.780 for NM-PPG, compared with 0.743 for GSMRL and 0.710 for OL. On the 784-dimensional image tasks, it is 0.707 and 0.691 for NM-PPG on MNIST and Fashion-MNIST, versus 0.652 and 0.650 for GSMRL and 0.389 and 0.495 for OL. AACO and SEFA avoid generic RL but optimize non-adaptive proxy scores that are biased with respect to the adaptive long-term objective in (2) (Appendix E.1). NM-PPG instead remains aligned with (2) and robust in higher dimensions, thereby combining the principal strengths of non-myopic and myopic methods. **Comparison with Myopic Methods.** Non-myopic planning should not improve every dataset. As discussed in Section 3.1 and Appendix F, a myopic policy can be optimal when features have no delayed value. This explains why strong myopic methods remain competitive on several real-world datasets. NM-PPG retains this robustness while also assigning credit to delayed feature value when it matters. Appendix E.4.5 visualizes the acquisition paths learned by different methods on synthetic and medical datasets, revealing policy differences not apparent from cost–performance curves alone. On Cube-NM, the optimal policy first acquires context with no immediate label information and then uses its value to select the relevant follow-up features, exactly as in Example 1. NM-PPG learns this context-dependent routing, whereas myopic methods do not identify the context features and therefore cannot condition later acquisitions on them. Similar conclusions hold for Syn1 and Syn3. On Mortality, Diabetes, and METABRIC, NM-PPG similarly uses initial screening measurements to determine which laboratory or molecular tests to acquire next. These screening measurements have no immediate predictive value, so myopic methods overlook them and the conditional policies they enable. Figure 1 shows that the resulting non-myopic policies improve the cost–performance trade-off over the myopic baselines on Mortality and METABRIC and remain competitive on Diabetes. The myopic routes remain reasonably competitive because several measurements are individually predictive, but they are less adaptive and may acquire tests that are irrelevant for a particular patient. Although myopic methods often offer stable training and consequently perform well in high-dimensional settings such as image datasets, their inability to exploit delayed value can be consequential in medical AFA: a non-myopic policy may avoid unnecessary tests and thereby reduce financial cost, patient burden, and procedural risk.

5 Conclusion

We introduced NM-PPG for direct long-term cost minimization in AFA. Our contributions are *(i)* a general objective for hard-, soft-, and jointly constrained AFA and a proof of soft-budget dominance (Section 2), *(ii)* a full-trajectory relaxation and bias-reducing ST rollout that aligns relaxed training with discrete deployment (Sections 3.2–3.3), *(iii)* an $O(k)$ shared-trajectory gradient estimator that avoids the $O(k^2)$ cost of a straightforward maximally non-myopic estimator (Section 3.3), and *(iv)* consistency guarantees and an AFA-specific gradient-variance bound that explains when full-trajectory gradients become unstable and motivates staged temperature sharpening (Sections 3.2 and 3.4). Experiments (Section 4) show that NM-PPG *(i)* is more stable than non-myopic baselines, *(ii)* matches strong myopic baselines when myopic planning ($h = 1$) suffices, and *(iii)* outperforms them when the dataset exhibits genuine non-myopic structure. Overall, NM-PPG provides a practical way to obtain non-myopic AFA policies while retaining much of the robustness that makes myopic methods attractive, and establishes important foundations for future work on pathwise-gradient methods for AFA.

AI USE STATEMENT

In this work, we have used generative AI tools for: help develop theoretical models or conceptual frameworks and provide feedback on research methodology or experiments. We have not used generative AI tools for: cleaning and reformatting datasets, implement methods, support qualitative and thematic data analysis, assist in the writing of proofs, propose or refine hypotheses, provide critical ingredients for proving mathematical claims, formulate mathematical claims, generate synthetic data sets, interpret results. Assist with translation are not applicable to this work. Additionally, we have used generative AI tools for: creating or modifying scientific figures or images, creating or editing software code, creation of artifacts, brainstorming, searching for information, and identify relevant literature. We have reviewed all AI-assisted work. We have checked the LLM-generated ideas, formulations, and statements to be correct or corrected them. All the code and content have been verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We support reproducibility through complete theoretical, algorithmic, and experimental documentation. All assumptions and proofs for the main theoretical results are provided in Appendix C, and Appendix D gives pseudocode for training, straight-through rollouts, and inference. Appendix E documents the baselines and their adaptations, datasets and preprocessing, hyperparameters and model selection, evaluation protocol, and additional analyses. The source code and configurations required to reproduce the experiments will be made publicly available upon publication.

REFERENCES

Molecular Biology (Splice-junction Gene Sequences). UCI Machine Learning Repository, 1991. DOI: https://doi.org/10.24432/C5M888.

Linus Aronsson, Arman Rahbar, and Morteza Haghir Chehreghani. A survey on active feature acquisition strategies. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, 2026.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Mustafa Bilgic and Lise Getoor. Voila: efficient feature-value acquisition for classification. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, 2007.

Centers for Disease Control and Prevention. National health and nutrition examination survey (nhanes). https://www.cdc.gov/nchs/nhanes/, 2026. Accessed: 2026-04-08.

Xinshi Chen, Hanjun Dai, Yu Li, Xin Gao, and Le Song. Learning to stop while learning to predict. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1520–1530, 2020.

Yuxin Chen, S Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In *Proceedings of The 28th Conference on Learning Theory*. PMLR, 2015.

Yuxin Chen, Jean-Michel Renders, Morteza Haghir Chehreghani, and Andreas Krause. Efficient online learning for optimizing value of information: Theory and application to interactive troubleshooting. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, 2017.

Michael Cooper, Rohan Wadhawan, John Michael Giorgi, Chenhao Tan, and Davis Liang. The curious language model: Strategic test-time information acquisition. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025.

Ian Covert, Wei Qiu, Mingyu Lu, Nayoon Kim, Nathan White, and Su-In Lee. Learning to maximize mutual information for dynamic feature selection. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Christina Curtis, Sohrab P. Shah, Suet-Feung Chin, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012. doi: 10.1038/nature10983.

Gabriel Dulac-Arnold, Ludovic Denoyer, Philippe Preux, and Patrick Gallinari. Datum-wise classification: A sequential approach to sparsity. In *Machine Learning and Knowledge Discovery in Databases*, 2011.

Gabriel Erion, Joseph D. Janizek, Carly Hudelson, Richard B. Utarnachitt, Andrew M. McCoy, Michael R. Sayre, Nathan J. White, and Su-In Lee. A cost-aware framework for the development of ai models for healthcare applications. *Nature Biomedical Engineering*, 6:1384–1398, 2022. doi: 10.1038/s41551-022-00872-8.

Soham Gadgil, Ian Connick Covert, and Su-In Lee. Estimating conditional mutual information for dynamic feature selection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oju2Qu9jvn.

Donald Geman and Bruno Jedynak. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, 1996.

Aritra Ghosh and Andrew Lan. Difa: Differentiable feature acquisition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

G. A. Gorry and G. O. Barnett. Experience with a model of sequential diagnosis. *Computers and Biomedical Research*, 1(5):490–507, 1968.

Osman Berke Guney, Ketan Suhaas Saichandran, Karim Elzokm, Ziming Zhang, and Vijaya B Kolachalama. Active feature acquisition via explainability-driven ranking. In *International Conference on Machine Learning*, 2025.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003. URL https://jmlr.org/papers/v3/guyon03a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

He He, Jason Eisner, and Hal Daume. Imitation learning by coaching. In *Advances in Neural Information Processing Systems*, 2012.

David Heckerman, John S. Breese, and Koos Rommelse. Decision-theoretic troubleshooting. *Commun. ACM*, 38(3):4957, 1995.

Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Jaromír Janisch, Tomás Pevný, and Viliam Lisý. Classification with costly features as a sequential decision-making problem. *Mach. Learn.*, 109(8):1587–1615, 2020.

Arjan J. P. Jeckmans, Michael Beye, Zekeriya Erkin, Pieter Hartel, Reginald L. Lagendijk, and Qiang Tang. *Privacy in Recommender Systems*. Springer London, 2013.

Mohammad Kachuee, Orpaz Goldstein, Kimmo Kärkkäinen, and Majid Sarrafzadeh. Opportunistic learning: Budgeted cost-sensitive learning from data streams. In *International Conference on Learning Representations*, 2019.

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

Andreas Krause and Carlos Guestrin. Optimal value of information in graphical models. *J. Artif. Int. Res.*, 35:557591, 2009.

Mikko Lauri, David Hsu, and Joni Pajarinen. Partially observable markov decision processes in robotics: A survey. *IEEE Transactions on Robotics*, 39(1):21–40, 2023.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yang Li and Junier Oliva. Active feature acquisition with generative surrogate models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Yang Li and Junier Oliva. Distribution guided active feature acquisition. *arXiv preprint arXiv:2410.03915*, 2024.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial VAE. In *International Conference on Machine Learning*. PMLR, 2019.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 2020.

Alexander Luke Ian Norcliffe, Changhee Lee, Fergus Imrie, Mihaela van der Schaar, and Pietro Lio. Stochastic encodings for active feature acquisition. In *International Conference on Machine Learning*, 2025.

Bernard Pereira, Suet-Feung Chin, Oscar M. Rueda, et al. The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature Communications*, 7: 11479, 2016. doi: 10.1038/ncomms11479.

Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-draa. Online planning algorithms for pomdps. *J. Artif. Int. Res.*, 32:663704, 2008.

T. Rückstieß, C. Osendorfer, and P. van der Smagt. Minimizing data consumption with sequential online feature selection. *International Journal of Machine Learning and Cybernetics*, 4:235–243, 2013.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Valter Schütz, Han Wu, Reza Rezvan, Linus Aronsson, and Morteza Haghir Chehreghani. AFABench: A generic framework for benchmarking active feature acquisition. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026.

Alexander Shekhovtsov. Bias-variance tradeoffs in single-sample binary gradient estimators. *arXiv preprint arXiv:2110.03549*, 2021.

David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems*, 2010.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning series. The MIT Press, 2 edition, 2018.

John Tromp. Connect-4. UCI Machine Learning Repository, 1995. DOI: https://doi.org/10.24432/C59P43.

Michael Valancius, Maxwell Lennon, and Junier B. Oliva. Acquisition conditioned oracle for non-greedy active feature acquisition. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Mary Vergara, Leo Ramos, Néstor Diego Rivera-Campoverde, and Francklin Rivas-Echeverría. Enginefaultdb: A novel dataset for automotive engine fault classification and baseline results. *IEEE Access*, 11:126155–126171, 2023. doi: 10.1109/ACCESS.2023.3331316.

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

Claas A Voelcker, Axel Brunnbauer, Marcel Hussing, Michal Nauman, Pieter Abbeel, Radu Grosu, Eric Eaton, Amir massoud Farahmand, and Igor Gilitschenski. Relative entropy pathwise policy optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4vmm8mlHkS.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Sara Zannone, Jose Miguel Hernandez Lobato, Cheng Zhang, and Konstantina Palla. Odin: Optimal discovery of high-value information using model-based deep reinforcement learning. In *Real-world Sequential Decision Making Workshop, ICML*, 2019.

Shenao Zhang, Wanxin Jin, and Zhaoran Wang. Adaptive barrier smoothing for first-order policy gradient with contact dynamics. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 41219–41243. PMLR, 2023a.

Shenao Zhang, Boyi Liu, Zhaoran Wang, and Tuo Zhao. Model-based reparameterization policy gradient methods: Theory and practical algorithms. In *Advances in Neural Information Processing Systems*, volume 36, 2023b.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pp. 1433–1438, 2008.

K.J Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

A Hard- and Soft-Budget Objectives

Proposition 3 (Soft-budget dominance)**.** *Fix k and B. Define the expected acquisition cost as $\mathcal{C}(\theta) \triangleq \mathbb{E}_{\mathbf{x},\pi_\theta}[c(m_\theta(\mathbf{x}))]$ and the expected prediction loss as $\mathcal{L}(\theta,\phi) \triangleq \mathbb{E}_{\mathbf{x},\mathbf{y},\pi_\theta}[\ell(f_\phi(\mathbf{x}(m_\theta(\mathbf{x}))),\mathbf{y})]$. Let $(\theta_{\text{hard}},\phi_{\text{hard}})$ be an optimal hard-budget solution, and assume its policy can be represented in the soft-budget policy class. If some $\alpha > 0$ yields an optimal soft-budget solution $(\theta_{\text{soft}},\phi_{\text{soft}})$ with $\mathcal{C}(\theta_{\text{soft}}) = \mathcal{C}(\theta_{\text{hard}})$, then $\mathcal{L}(\theta_{\text{soft}},\phi_{\text{soft}}) \leq \mathcal{L}(\theta_{\text{hard}},\phi_{\text{hard}})$.*

Intuitively, at the same expected acquisition cost, a soft-budget policy may spend less on easy instances and more on difficult ones, including more than B, while a hard-budget policy cannot exceed B on any instance. The soft-budget cost-performance frontier therefore weakly dominates the hard-budget frontier. The appropriate objective depends on the application. A hard budget is necessary when every instance must obey a known resource limit, while a soft budget is appropriate when cost may be allocated across instances. A recent AFA benchmark observes that prior work often compares a proposed soft-budget method with hard-budget baselines without separating the two objectives (Schütz et al., 2026). Such comparisons may attribute an advantage from the more flexible soft-budget policy class to the learning method itself. The benchmark therefore recommends separate evaluation within each setting, and Proposition 3 formally justifies this recommendation.

B Arbitrary Planning Depth and Prior Pathwise AFA

The main method uses one shared full-trajectory rollout with $h = k$. Here we define a *hard-rooted* h-step estimator as a reference for comparing NM-PPG with prior pathwise methods. For $1 \le h \le k$, this estimator first samples the global conditional feature path and Gumbel noise as in the main method, then constructs a local rollout at every visited step t. Let $T_{t,h} \le \min\{t + h, T\}$ be the local endpoint. Each local graph is initialized by $\tilde{m}_t = \bar{m}_t = m_t$ and $\tilde{s}_{t,t} = 1$, and applies the relaxed and ST updates from Sections 3.2–3.3 until $T_{t,h}$, with $\tilde{s}_{t,i+1} = \tilde{s}_{t,i}(1 - \tilde{a}_{i,d+1})$. It reuses the global hard actions and Gumbel samples because every local ST mask equals m_i in the forward pass; resetting $\tilde{m}_t = m_t$ changes only the backward relaxation. All quantities inside a local graph are recomputed for root t; we suppress this root index except on $\tilde{s}_{t,i}$. Its ST return is

$$\bar{G}_{t,h} \triangleq \sum_{i=t}^{T_{t,h}-1} \tilde{s}_{t,i}\big[\alpha(1 - \tilde{a}_{i,d+1})c(\bar{r}_i) + \tilde{a}_{i,d+1}\bar{\ell}_i\big] + \tilde{s}_{t,T_{t,h}}\bar{\ell}_{T_{t,h}}. \tag{11}$$

The relaxed return $\tilde{G}_{t,h}$ replaces $(\bar{r}_i, \bar{m}_i, \bar{\ell}_i)$ with $(\tilde{r}_i, \tilde{m}_i, \tilde{\ell}_i)$. Let $\nabla_\theta^{(t),\mathrm{ST}}$ retain direct parameter gradients only through the policy evaluation at root t. At every later step $i > t$, the policy parameters are treated as constants while derivatives through the mask input are retained. Later decisions and losses can therefore affect the root update through the local state sequence, but the policy evaluations at those later states do not receive a separate direct parameter gradient from the same window. Because $\tilde{s}_{t,t} = 1$, $\bar{G}_{t,h}$ is conditional on reaching root t. Multiplying by the detached global survival $\mathrm{sg}(\tilde{s}_t)$ accounts for the probability of reaching that root without sending this local update through earlier stopping decisions. The resulting estimator is

$$g_h^{\mathrm{root}}(x, y, \boldsymbol{\varepsilon}; \theta) \triangleq \sum_{t=0}^{T-1} \mathrm{sg}(\tilde{s}_t)\nabla_\theta^{(t),\mathrm{ST}}\bar{G}_{t,h}. \tag{12}$$

There are at most k roots and each local graph contains at most h policy evaluations, so this estimator costs $O(kh)$. Proposition 2 contrasts this standard rooted computation with NM-PPG's shared estimator for $h = k$. Because all temporal contributions are derivatives of one full-trajectory return, NM-PPG differentiates a single global rollout and evaluates each of the k policy states only once, giving $O(k)$ computation. Planning depth changes training only; inference still follows Algorithm 3 with deployment horizon k.

B.1 Relation to GDFS and DiFA

To our knowledge, DiFA (Ghosh & Lan, 2023) and GDFS (Covert et al., 2023) are the only prior AFA methods that use continuous relaxations to obtain pathwise gradients. Both score a feature through its immediate effect on prediction. The hard-rooted estimator above makes their relationship to our formulation precise.

Proposition 4 (GDFS and DiFA as rooted special cases)**.** *Assume $\alpha = 0$, uniform unit costs ($c(a) = 1$ for all $a \in [d]$), no learned stopping, and $h = 1$. Under these shared conditions, the hard-rooted h-step estimator corresponds to* (i) *GDFS (Covert et al., 2023) when its successor mask remains relaxed and* (ii) *DiFA (Ghosh & Lan, 2023) when the ST correction from Section 3.3 makes this mask discrete in the forward pass.*

The $h > 1$ case requires an additional construction: neither DiFA nor GDFS develops pathwise updates through the AFA-POMDP's multi-step return. Our POMDP formulation (Section 3.1) provides the basis for this extension. We introduce the hard-rooted estimator as one possible way to train policies with arbitrary planning depth h. This new construction is used for theoretical and computational comparison, not as an experimental baseline, since our baseline comparisons focus on existing methods from the literature.

Proposition 4 identifies the precise overlap and the extensions provided by NM-PPG. *(i) General objectives and stopping.* NM-PPG optimizes (2) for arbitrary (B, α, k) and non-uniform costs. When $\alpha > 0$, the survival recursion in (7) makes stopping differentiable, so the policy jointly learns what to acquire and when further acquisition is not worth its cost. *(ii) Full-trajectory non-myopic planning.* GDFS and DiFA use myopic planning ($h = 1$), whereas NM-PPG uses $h = k$, the only depth in Definition 1 guaranteed to align with the complete AFA objective in (2). *(iii) Linear-time gradients.* Extending the estimators of DiFA and GDFS to $h > 1$ is non-trivial and is not developed in either paper. Our POMDP formulation in Section 3.1 makes such an extension explicit through the hard-rooted estimator, which costs $O(kh)$. Both this estimator with $h = k$ and NM-PPG account for

the complete remaining trajectory. The former constructs a separate rollout from every visited state, costing $O(k^2)$. NM-PPG instead uses one shared ST trajectory from $t = 0$, allowing all temporal gradient contributions to be computed in a single $O(k)$ forward–backward pass (Proposition 2). The different relaxed-mask initializations mean that these are distinct estimators, not two implementations of the same gradient. *(iv) AFA-specific guarantees.* We formally connect the relaxation to the discrete trajectory and objective (Theorem 2). We also study the gradient estimator's variance theoretically (Theorem 3), rather than relying only on empirical evidence of stable training. The average-case bound links gradient stability to policy ambiguity and sequential acquisition dynamics and motivates staged temperature training.

C Proofs

C.1 Proof of Proposition 3

The hard-budget solution is feasible in the soft-budget problem because the latter uses the non-binding budget $B = c([d])$ and the same cardinality limit k. Optimality of $(\theta_{\text{soft}}, \phi_{\text{soft}})$ therefore gives $\mathcal{L}(\theta_{\text{soft}}, \phi_{\text{soft}}) + \alpha\mathcal{C}(\theta_{\text{soft}}) \leq \mathcal{L}(\theta_{\text{hard}}, \phi_{\text{hard}}) + \alpha\mathcal{C}(\theta_{\text{hard}})$. Since $\mathcal{C}(\theta_{\text{soft}}) = \mathcal{C}(\theta_{\text{hard}})$, the acquisition-cost terms cancel, proving the result. □

C.2 Proof of Theorem 1

The finite-horizon POMDP has latent state (m_t, x, y), observation $x(m_t)$, available actions $\mathcal{A}_B(m_t; k)$, initial mask $m_0 = \mathbf{0}$, and an absorbing terminal state. A feature action reveals x_a and applies the deterministic mask update from Section 2; action $d + 1$ terminates the trajectory. The immediate cost is $\alpha c(a)$ for a feature action and $\ell(f_\phi(x(m_t)), y)$ for stopping. The hidden pair (x, y) makes this a POMDP, while $x(m_t)$ records all acquired indices and values and therefore determines the corresponding belief under the fixed data distribution.

Every feature selected from $\mathcal{F}_B(m_t; k)$ preserves $c(m_t) \leq B$, and $\mathcal{F}_B(m_k; k) = \varnothing$, so the terminal mask also satisfies $\|m_{T_\theta(x)}\|_0 \leq k$. The realized return is

$$G(x, y, \pi_\theta) = \sum_{t=0}^{T_\theta(x)-1} \alpha c(a_t) + \ell\big(f_\phi\big(x(m_{T_\theta(x)})\big), y\big)\,. \tag{13}$$

Because the terminal mask contains exactly the acquired features, $\sum_{t<T_\theta(x)} c(a_t) = c(m_\theta(x))$. Taking expectations yields the objective in (2), and the available-action set enforces both constraints. Setting $\alpha = 0$ gives the hard-budget case; setting $B = c([d])$ gives the soft-budget case. □

C.3 Proof of Theorem 2

Couple the relaxed and discrete rollouts using the same Gumbel sequence, with the discrete rollout taking the corresponding Gumbel–Max actions. Let $M < \infty$ satisfy $|\ell| \leq M$. Write $\tau = \tau_{\text{soft}}$, and use a superscript τ when the dependence of a rollout variable on temperature must be explicit.

For property *(i)*, the survival recursion in (7) gives

$$\tilde{s}_t \tilde{a}_{t,d+1} = \tilde{s}_t - \tilde{s}_t(1 - \tilde{a}_{t,d+1}) = \tilde{s}_t - \tilde{s}_{t+1}. \tag{14}$$

Consequently,

$$\textstyle\sum_{t=0}^{T-1} \tilde{s}_t \tilde{a}_{t,d+1} + \tilde{s}_T = \sum_{t=0}^{T-1} (\tilde{s}_t - \tilde{s}_{t+1}) + \tilde{s}_T = (\tilde{s}_0 - \tilde{s}_1) + \cdots + (\tilde{s}_{T-1} - \tilde{s}_T) + \tilde{s}_T = \tilde{s}_0 = 1. \tag{15}$$

Thus, the stopping masses $\tilde{s}_t \tilde{a}_{t,d+1}$ together with the residual terminal mass $\tilde{s}_T$ form a normalized distribution over stopping steps.

For property *(ii)*, let L be the first $t < T$ for which $\tilde{a}_{t,d+1} = 1$, or let $L = T$ if there is no such step. For every $t < L$, the one-hot action selects a feature, and the identity $\tilde{r}_t = \tilde{a}_{t,1:d}/(1 - \tilde{a}_{t,d+1})$ gives $\tilde{r}_t = r_t$. Since $\tilde{m}_0 = m_0$, induction through the mask recursion yields $\tilde{m}_t = m_t$ for $0 \leq t \leq L$. Moreover, $\tilde{s}_t = 1$ for $t \leq L$, while $\tilde{s}_t = 0$ for $t > L$ when $L < T$. Terms after L therefore vanish, and (8) reduces to $\tilde{G} = \sum_{t=0}^{L-1} \alpha c(r_t) + \ell(f_\phi(x(m_L)), y) = G$. No claim about the relaxed masks after a hard stop is required.

For property *(iii)*, fix (x, y) and the Gumbel sequence $\boldsymbol{\varepsilon}$. At step t, ε_t is independent of the state generated by earlier perturbations. Its continuous density therefore makes both the joint and feature-only perturbed logits at every visited limiting state have unique maximizers almost surely; this holds simultaneously at all steps because k is finite. We now induct on t. At $t = 0$, $m_0^\tau = \tilde{m}_0^\tau = m_0$. Suppose that, for all sufficiently small τ, $m_t^\tau = m_t^0$ and that $\tilde{m}_t^\tau \to m_t^0$. The feasible action set is then fixed, and continuity of the unmasked logits gives $z_t^\tau \to z_\theta(x(m_t^0))$ on that set. Stability of a unique maximizer implies that the hard conditional feature action is eventually equal to its limiting Gumbel–Max action, while the Gumbel–Softmax limit gives $\tilde{a}_t^\tau \to a_t^0$ and $\tilde{r}_t^\tau \to r_t^0$. The hard and relaxed mask recursions consequently give $m_{t+1}^\tau = m_{t+1}^0$ eventually and $\tilde{m}_{t+1}^\tau \to m_{t+1}^0$; the survival recursion similarly gives $\tilde{s}_{t+1}^\tau \to s_{t+1}^0$. This completes the induction. Because the endpoint is determined by the hard mask, budget, and finite horizon, it also stabilizes: $T^\tau = T^0$ for all sufficiently small τ. Continuity of the loss and the finite return in (8) now imply $\tilde{G}^\tau \to G^0$ almost surely, where G^0 is the coupled discrete return.

For property *(iv)*, let $c_{\max} = \max_{a \in [d]} c(a)$. The prediction-loss weights are nonnegative and sum to one by property *(i)*, while $c(\tilde{r}_t) \le c_{\max}$, $\tilde{s}_t(1 - \tilde{a}_{t,d+1}) \le 1$, and $T \le k$. Hence $|\tilde{G}^\tau| \le M + |\alpha| k c_{\max}$ for every τ, providing an integrable bound independent of temperature. Dominated convergence and property *(iii)* give $\lim_{\tau \to 0} \tilde{J}(\theta) = \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}[G^0]$. Conditional on every discrete history, the Gumbel–Max identity gives $a_t^0 \sim \pi_\theta(\cdot \mid x(m_t^0))$. Thus the coupled limiting trajectory has exactly the law of the discrete policy rollout, so $\mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}[G^0] = \mathbb{E}_{\mathbf{x},\mathbf{y},\pi_\theta}[G] = J(\pi_\theta)$. □

C.4 Proof of Proposition 1

Proof. Fix $(x, y, \boldsymbol{\varepsilon})$. The ST rollout is then a deterministic computation graph from θ to $\bar{G}$. At each step, θ can affect the return through four intermediate quantities: the feature action $\bar{r}_t$, stopping mass $\tilde{a}_{t,d+1}$, mask $\bar{m}_t$, and survival mass $\tilde{s}_t$. To separate their roles, write $v_t = (\bar{r}_t, \tilde{a}_{t,d+1}, \bar{m}_t, \tilde{s}_t)$ and regard the outer return as a function $F(v_0, \dots, v_T)$. Each partial derivative below holds the other intermediate variables fixed and therefore measures only the direct effect of one variable on the return. Its derivative with respect to θ is instead a total derivative that includes all dependencies through the preceding policy evaluations and state updates. The multivariate chain rule gives

$$\nabla_\theta^{\mathrm{ST}} \bar{G} = \sum_{t=0}^{T} \left(\frac{\partial \bar{G}}{\partial \bar{r}_t} \frac{\partial \bar{r}_t}{\partial \theta} + \frac{\partial \bar{G}}{\partial \tilde{a}_{t,d+1}} \frac{\partial \tilde{a}_{t,d+1}}{\partial \theta} + \frac{\partial \bar{G}}{\partial \bar{m}_t} \frac{\partial \bar{m}_t}{\partial \theta} + \frac{\partial \bar{G}}{\partial \tilde{s}_t} \frac{\partial \tilde{s}_t}{\partial \theta} \right). \tag{16}$$

The ST definitions in (9) are numerically hard in the forward pass but satisfy $\partial \bar{r}_t / \partial \tilde{r}_t = I$ and $\partial \bar{m}_t / \partial \tilde{m}_t = I$ in the backward pass. Hence $\partial \bar{r}_t / \partial \theta = (\partial \bar{r}_t / \partial \tilde{r}_t)(\partial \tilde{r}_t / \partial \theta)$, with the analogous identity for $\bar{m}_t$. Substituting these identities gives the expansion in Proposition 1. The total derivatives of $\tilde{r}_t, \tilde{a}_{t,d+1}, \tilde{m}_t$, and $\tilde{s}_t$ recursively include every earlier policy evaluation through (5) and (7); consequently, losses and costs at later steps contribute gradients to all earlier decisions that can affect them. At $t = T$, no action is sampled, so the terms involving $\bar{r}_T$ and $\tilde{a}_{T,d+1}$ are zero. □

C.5 Proof of Proposition 2

Treat each policy, transition, loss, and corresponding backward computation as constant time with respect to k. The h-step window beginning at step t contains at most $\min\{h, k - t\}$ decisions. Computing a separate gradient from every visited state therefore costs $\sum_{t=0}^{k-1} \min\{h, k - t\} = kh - h(h-1)/2 = O(kh)$, which becomes $O(k^2)$ when $h = k$. For NM-PPG, temporarily replace the shared policy parameters at the k trajectory steps by copies $\theta_0, \dots, \theta_{k-1}$, all equal to θ in the forward pass. Tying them again gives $\nabla_\theta^{\mathrm{ST}} \bar{G} = \sum_{t=0}^{k-1} \nabla_{\theta_t}^{\mathrm{ST}} \bar{G}$. Each term contains the effect of the decision at step t on every later outcome it can influence. The shared computation graph has $O(k)$ operations, so one backward pass computes their sum in $O(k)$ time. □

C.6 Proof of Theorem 3

Proof. The proof has three parts: conditional relaxation moments, AFA transition sensitivity, and trajectory-gradient variance. For one rollout, let $\mathcal{P} = (a_0^F, \dots, a_{T-1}^F)$ be the sampled hard feature

path. At each visited step, let $\mathcal{A}_t$ be the set of unmasked actions, write $P_t \coloneqq I_{|\mathcal{A}_t|} - |\mathcal{A}_t|^{-1}\mathbf{1}\mathbf{1}^\top$, and let $\Lambda_t \coloneqq \|P_t\, \partial z_{t,\mathcal{A}_t}/\partial \bar{m}_t\|_2$. Let $L_\theta, M_\ell, G_\ell < \infty$ bound, respectively, $\|P_t\, \partial z_{t,\mathcal{A}_t}/\partial \theta\|_2$, the prediction loss, and its mask derivative along visited hard states, and assume Λ_t is bounded. Let $\bar{\tau} < 1$ be a fixed upper bound satisfying $0 < \tau_{\text{soft}} \le \bar{\tau}$, and assume feasible feature costs are bounded. For the full-trajectory estimator, set $T_t \triangleq T$ and $H_t \triangleq T - t$. We analyze each temporally untied contribution of the single global graph while conditioning on its incoming relaxed state. Temporal untying replaces the shared parameter at step i by a copy $\theta_i = \theta$ only for differentiation; tying the copies again sums these partial derivatives into the gradient of the shared policy.

1. Conditional relaxation moments. *Conditioning on the hard path.* The ST construction satisfies $\bar{m}_t = m_t$ numerically in the forward pass. Consequently, conditional on the data (x, y) and hard feature path $\mathcal{P}$, every hard mask, feasible action set, policy logit vector, and categorical policy is fixed:

$$m_t, \quad \mathcal{A}_t, \quad P_t, \quad z_{t,\mathcal{A}_t}, \quad \pi_t, \quad \Lambda_t. \tag{17}$$

The event that a particular hard feature path is realized factorizes as

$$E(\mathcal{P}) = \bigcap_{t<T} E_t(a_t^F), \tag{18}$$

where $E_t(a_t^F)$ is the event that a_t^F is the largest Gumbel-perturbed feasible feature logit at step t. Each event $E_t(a_t^F)$ depends only on the feature Gumbels at that step. Since the Gumbel vectors are independent across steps,

$$p(\varepsilon_{0:T-1} \mid x, y, \mathcal{P}) = \prod_{t<T} p(\varepsilon_t \mid x, y, \mathcal{P}). \tag{19}$$

Thus, after conditioning on the hard path, the residual Gumbel randomness remains independent across time. This property will allow us to control products of future state Jacobians without assuming that the random AFA states are unconditionally independent.

Feature-action relaxation. We first record a bound for a generic categorical Gumbel–Softmax variable. Let $u_b = \log w_b + \varepsilon_b$, $W = \sum_b w_b$, $\pi_b = w_b/W$, $A = \arg\max_b u_b$, and $y = \mathrm{softmax}(u/\tau_{\text{soft}})$. Conditional on $A = a$, use the exponential-race representation $u_b = -\log T_b$, where the winning time $T_a \sim \mathrm{Exp}(W)$ and, by memorylessness, $T_b = T_a + R_b$ with $R_b \sim \mathrm{Exp}(w_b)$ for each $b \neq a$. For the pairwise perturbed-logit gap $\Delta_{ab} = u_a - u_b = \log((T_a + R_b)/T_a)$,

$$\Pr(\Delta_{ab} > x \mid A = a) = \frac{1}{1 + \pi_b(e^x - 1)}, \qquad f_{\Delta_{ab}|A=a}(x) \le \pi_b e^x. \tag{20}$$

Consequently,

$$\mathbb{E}\Big[e^{-\Delta_{ab}/\tau_{\text{soft}}} \mid A = a\Big] \le \frac{\pi_b \tau_{\text{soft}}}{1 - \tau_{\text{soft}}}. \tag{21}$$

Because $1 - y_a \le \sum_{b\neq a} e^{-\Delta_{ab}/\tau_{\text{soft}}}$, summing (21) over the losing actions gives

$$\mathbb{E}[1 - y_a \mid A = a] \le \frac{\tau_{\text{soft}}}{1 - \bar{\tau}}(1 - \pi_a), \tag{22}$$

which contains no factor depending on the number of actions.

For $S(y) \coloneqq \mathrm{Diag}(y) - yy^\top$, the categorical covariance satisfies $\|S(y)\|_2 \le 2(1 - y_a)$ when a is the largest component and $\|S(y)\|_2 \le 1/2$. Hence $\|S(y)\|_2^2 \le 1 - y_a$. Applying (22) to the conditional feature relaxation, whose hard winner is a_t^F, gives $R_t \coloneqq \partial \tilde{r}_t/\partial z_{t,\mathcal{F}_B(m_t;k)} = S(\tilde{r}_t)/(\tau_{\text{hard}}\tau_{\text{soft}})$ and

$$\mathbb{E}[\|R_t\|_2 \mid x, y, \mathcal{P}] \le \frac{2[1 - \pi^F_{t,a_t^F}]}{\tau_{\text{hard}}(1 - \bar{\tau})}, \tag{23}$$

$$\mathbb{E}[\|R_t\|_2^2 \mid x, y, \mathcal{P}] \le \frac{1 - \pi^F_{t,a_t^F}}{\tau^2_{\text{hard}}\tau_{\text{soft}}(1 - \bar{\tau})}. \tag{24}$$

Stopping relaxation. When stopping is masked, set $q_t = 0$ and $Q_t = 0$, so the bounds below hold trivially with $p = 0$. Otherwise, let $q_t \coloneqq \tilde{a}_{t,d+1}$ and

$$Q_t \coloneqq \nabla_{z_{t,\mathcal{A}_t}} q_t = \frac{S(\tilde{a}_{t,\mathcal{A}_t})e_{d+1}}{\tau_{\text{hard}}\tau_{\text{soft}}}. \tag{25}$$

Writing $q = q_t$, direct calculation gives

$$\|S(\tilde{a}_{t,\mathcal{A}_t})e_{d+1}\|_2^2 = q^2\left[(1-q)^2 + \sum_{b\neq d+1}\tilde{a}_{t,b}^2\right] \leq 2q^2(1-q)^2. \tag{26}$$

Set $p \coloneqq \pi_{t,d+1}$, $r \coloneqq \pi^F_{t,a_t^F}$, and $\eta_t = (1-r) + p(1-p)$. We next bound $q(1-q)$ conditional on the sampled feature winner a_t^F.

Represent the joint feature and stopping Gumbels by independent exponential clocks with rates $w_b = \exp(z_{t,b}/\tau_{\text{hard}})$. Let A be the feature-clock winner and $A^\star$ the winner among all feature and stopping clocks. Conditional on $A = a$, the joint winner is either a or $d+1$, and $\Pr(A^\star = d+1 \mid A = a) = p$. If $A^\star = a$, (22) applied to the joint race gives

$$\mathbb{E}[1 - \tilde{a}_{t,a} \mid A^\star = a] \leq \frac{\tau_{\text{soft}}}{1-\bar{\tau}}[1-(1-p)r]. \tag{27}$$

If stopping wins first at clock time T, memorylessness leaves independent feature residual clocks. Conditional also on $A = a$, their minimum has rate equal to the sum of the feature rates, while each remaining residual after this minimum has its original feature rate. Applying the pairwise calculation in (21) to these residual clocks yields

$$\begin{aligned}\mathbb{E}[1-q \mid A^\star = d+1, A = a] &\leq \frac{\tau_{\text{soft}}}{1-\bar{\tau}}\left[(1-p) + \sum_{b\in\mathcal{F}_B(m_t;k)\setminus\{a\}}\pi_{t,b}\right] \\ &= \frac{\tau_{\text{soft}}}{1-\bar{\tau}}(1-p)(2-r).\end{aligned} \tag{28}$$

In either case, $q(1-q) \leq 1 - \tilde{a}_{t,A^\star}$. Combining (27)–(28) therefore gives

$$\begin{aligned}\mathbb{E}[q(1-q) \mid x, y, \mathcal{P}] &\leq \frac{\tau_{\text{soft}}}{1-\bar{\tau}}(1-p)(1-r+2p) \\ &\leq \frac{2\tau_{\text{soft}}}{1-\bar{\tau}}\eta_t.\end{aligned} \tag{29}$$

Using (26), and $[q(1-q)]^2 \leq q(1-q)/4$, now gives

$$\mathbb{E}[\|Q_t\|_2 \mid x, y, \mathcal{P}] \leq \frac{2\sqrt{2}\,\eta_t}{\tau_{\text{hard}}(1-\bar{\tau})}, \tag{30}$$

$$\mathbb{E}[\|Q_t\|_2^2 \mid x, y, \mathcal{P}] \leq \frac{\eta_t}{\tau_{\text{hard}}^2\tau_{\text{soft}}(1-\bar{\tau})}. \tag{31}$$

2. AFA transition sensitivity. *Acquisition and survival dynamics.* Its conditional survival satisfies $\tilde{s}_{t,t} = 1$ and $\tilde{s}_{t,j+1} = \tilde{s}_{t,j}(1-q_j)$. Parts 2–3 bound the corresponding temporally untied contribution b_t, defined at the end of Part 3. Define the augmented relaxed AFA state and action by

$$\xi_j \coloneqq (\tilde{m}_j, \tilde{s}_{t,j}), \qquad U_j \coloneqq (\tilde{r}_j, q_j). \tag{32}$$

Their conditional relaxed dynamics are

$$\tilde{m}_{j+1} = \tilde{m}_j + (1-\tilde{m}_j)\odot\tilde{r}_j, \qquad \tilde{s}_{t,j+1} = \tilde{s}_{t,j}(1-q_j). \tag{33}$$

Holding U_j fixed, the state Jacobian is

$$F_{\xi,j} = \begin{bmatrix} I - \text{Diag}(\tilde{r}_j) & 0 \\ 0 & 1-q_j \end{bmatrix}. \tag{34}$$

Since $\tilde{r}_j$ lies in the probability simplex and $q_j \in [0,1]$,

$$\|F_{\xi,j}\|_2 \leq 1. \tag{35}$$

Similarly, the derivative of the next state with respect to U_j satisfies

$$F_{U,j} = \begin{bmatrix} \text{Diag}(1-\tilde{m}_j) & 0 \\ 0 & -\tilde{s}_{t,j} \end{bmatrix}, \tag{36}$$

in the natural block representation, and hence

$$\|F_{U,j}\|_2 \le 1. \tag{37}$$

Thus neither the mask transition nor the survival transition can by itself amplify a perturbation.

Policy feedback. Because the ST construction satisfies $\partial \bar{m}_j / \partial \tilde{m}_j = I$, the action sensitivity to the relaxed mask is obtained by differentiating through the feasible logits. Let $\widehat{R}_j = [R_j \ 0]$ append a zero stopping-action column to R_j, and define

$$D_j \coloneqq \begin{bmatrix} \widehat{R}_j \\ Q_j^\top \end{bmatrix} = \frac{\partial U_j}{\partial z_{j,\mathcal{A}_j}}. \tag{38}$$

Softmax is invariant to a common shift of its logits, so $R_j \mathbf{1} = 0$ and $Q_j^\top \mathbf{1} = 0$. Hence $D_j \mathbf{1} = 0$ and, exactly,

$$D_j = D_j P_j. \tag{39}$$

Since P_j is an orthogonal projection, $\Lambda_j \le \|\partial z_{j,\mathcal{A}_j} / \partial \bar{m}_j\|_2$. The action–mask feedback therefore satisfies

$$A_j \coloneqq \frac{\partial U_j}{\partial \tilde{m}_j} = D_j P_j \frac{\partial z_{j,\mathcal{A}_j}}{\partial \bar{m}_j}, \qquad \|A_j\|_2 \le \Lambda_j \sqrt{\|R_j\|_2^2 + \|Q_j\|_2^2}. \tag{40}$$

Thus common logit shifts, which cannot alter any action probability, make no contribution to policy feedback. Define the scalar envelope

$$\Gamma_j \coloneqq \Lambda_j \sqrt{\|R_j\|_2^2 + \|Q_j\|_2^2}. \tag{41}$$

Since the policy does not take survival mass as input, $\partial U_j / \partial \xi_j = [A_j \ \mathbf{0}]$, with $\|[A_j \ \mathbf{0}]\|_2 = \|A_j\|_2 \le \Gamma_j$. The complete closed-loop state Jacobian is therefore

$$J_j = F_{\xi,j} + F_{U,j}[A_j \ \mathbf{0}]. \tag{42}$$

Using (35) and (37),

$$\|J_j\|_2 \le 1 + \Gamma_j. \tag{43}$$

By (23)–(31), $1 - \pi_j^F(a_j^F) \le \eta_j$, and $\sqrt{u^2 + v^2} \le u + v$ for $u, v \ge 0$. Set

$$C_1 \coloneqq \frac{4(1+\sqrt{2})}{1-\bar{\tau}}, \qquad C_2 \coloneqq \frac{2}{1-\bar{\tau}}. \tag{44}$$

These constants do not depend on the number of feasible actions. The feedback gain satisfies

$$\mathbb{E}[\Gamma_j \mid x, y, \mathcal{P}] \le \frac{C_1}{2} \eta_j \frac{\Lambda_j}{\tau_{\text{hard}}}, \tag{45}$$

$$\mathbb{E}[\Gamma_j^2 \mid x, y, \mathcal{P}] \le C_2 \eta_j \frac{\Lambda_j^2}{\tau_{\text{hard}}^2 \tau_{\text{soft}}}. \tag{46}$$

Consequently,

$$\begin{aligned} \mathbb{E}[(1+\Gamma_j)^2 \mid x, y, \mathcal{P}] &= 1 + 2\mathbb{E}[\Gamma_j \mid x, y, \mathcal{P}] + \mathbb{E}[\Gamma_j^2 \mid x, y, \mathcal{P}] \\ &\le 1 + \rho_j, \end{aligned} \tag{47}$$

where ρ_j is defined in Theorem 3.

3. Trajectory-gradient variance. *Local derivatives.* For the suffix rooted at t, define the local stage contribution

$$\psi_j(\xi_j, U_j) = \tilde{s}_{t,j} \left[\alpha(1-q_j) c(\bar{r}_j) + q_j \bar{\ell}_j \right], \qquad j < T_t, \tag{48}$$

and the terminal contribution

$$\psi_{T_t}(\xi_{T_t}) = \tilde{s}_{t,T_t} \bar{\ell}_{T_t}. \tag{49}$$

Let

$$c_{\max} = \max_{a \in [d]} c(a), \tag{50}$$

and let C_c be a finite upper bound on the Euclidean norm of the feasible feature-cost vector. Holding U_j fixed, the local state derivative is bounded by

$$\left\| \frac{\partial \psi_j}{\partial \xi_j} \right\|_2 \le B_\alpha, \tag{51}$$

where one valid finite choice is

$$B_\alpha^2 := G_\ell^2 + (\alpha c_{\max} + M_\ell)^2. \tag{52}$$

Similarly, the local derivative with respect to the relaxed feature/stopping action is bounded by

$$\left\| \frac{\partial \psi_j}{\partial U_j} \right\|_2 \le C_\alpha, \tag{53}$$

where, for example,

$$C_\alpha^2 := \alpha^2 C_c^2 + (\alpha c_{\max} + M_\ell)^2. \tag{54}$$

Finally,

$$\left\| \nabla_{\xi_{T_t}} \psi_{T_t} \right\|_2 \le B_T, \qquad B_T := \sqrt{G_\ell^2 + M_\ell^2}. \tag{55}$$

Define

$$L_\alpha := B_\alpha + C_\alpha + B_T. \tag{56}$$

This quantity contains only local one-step smoothness terms and does not hide products over the future trajectory.

Future propagation. Let p_j denote the derivative of the remaining suffix with respect to the augmented state ξ_j. At the terminal step,

$$p_{T_t} = \nabla_{\xi_{T_t}} \psi_{T_t}. \tag{57}$$

For $j < T_t$, the chain rule gives

$$p_j = \frac{\partial \psi_j}{\partial \xi_j} + [A_j \ \mathbf{0}]^\top \frac{\partial \psi_j}{\partial U_j} + J_j^\top p_{j+1}. \tag{58}$$

Using (51), (53), and (43),

$$\|p_j\|_2 \le B_\alpha + C_\alpha \Gamma_j + (1 + \Gamma_j)\|p_{j+1}\|_2. \tag{59}$$

Since

$$B_\alpha + C_\alpha \Gamma_j \le L_\alpha (1 + \Gamma_j), \tag{60}$$

unrolling (59) gives

$$\|p_{t+1}\|_2 \le L_\alpha H_t \prod_{j=t+1}^{T_t - 1} (1 + \Gamma_j). \tag{61}$$

Average propagation. Conditional on $(x, y, \mathcal{P})$, the quantities Λ_j are fixed, and Γ_j depends only on the residual Gumbel vector at step j. By (19), the Γ_j are therefore conditionally independent across time. Squaring (61) and taking the conditional expectation gives

$$\begin{aligned} \mathbb{E}\left[\|p_{t+1}\|_2^2 \mid x, y, \mathcal{P} \right] &\le L_\alpha^2 H_t^2 \prod_{j=t+1}^{T_t - 1} \mathbb{E}\left[(1 + \Gamma_j)^2 \mid x, y, \mathcal{P} \right] \\ &\le L_\alpha^2 H_t^2 \prod_{j=t+1}^{T_t - 1} (1 + \rho_j). \end{aligned} \tag{62}$$

Using $1 + x \le e^x$,

$$\mathbb{E}\left[\|p_{t+1}\|_2^2 \mid x, y, \mathcal{P} \right] \le L_\alpha^2 H_t^2 \exp\left\{ \sum_{j=t+1}^{T_t - 1} \rho_j \right\}. \tag{63}$$

Local variance injection. Let d_t denote the derivative of the root-t truncated suffix with respect to the current relaxed action U_t. Then

$$d_t = \frac{\partial \psi_t}{\partial U_t} + F_{U,t}^\top p_{t+1}. \tag{64}$$

Since $\|F_{U,t}\|_2 \le 1$,

$$\|d_t\|_2 \le C_\alpha + \|p_{t+1}\|_2. \tag{65}$$

Because $H_t \ge 1$, $C_\alpha \le L_\alpha$, and the exponential factor is at least one, $(a+b)^2 \le 2a^2 + 2b^2$ and (63) give

$$\mathbb{E}[\|d_t\|_2^2 \mid x, y, \mathcal{P}] \le 4L_\alpha^2 H_t^2 \exp\left\{ \sum_{j=t+1}^{T_t - 1} \rho_j \right\}. \tag{66}$$

Define

$$U_{z,t} \coloneqq \frac{\partial U_t}{\partial z_{t,\mathcal{A}_t}}. \tag{67}$$

By Equation (39), $U_{z,t} = U_{z,t} P_t$ and hence $b_t = (P_t\, \partial z_{t,\mathcal{A}_t}/\partial \theta)^\top U_{z,t}^\top d_t$. Equations (24) and (31) imply

$$\mathbb{E}[\|U_{z,t}\|_2^2 \mid x, y, \mathcal{P}] \le \frac{2\eta_t}{\tau_{\text{hard}}^2 \tau_{\text{soft}} (1 - \bar{\tau})}. \tag{68}$$

The direct term $\partial \psi_t / \partial U_t$ may depend on the current Gumbel vector, so no independence between d_t and $U_{z,t}$ is assumed. Instead, (61) gives a pointwise upper bound on the future term that depends only on residual Gumbels after step t. This upper bound is conditionally independent of $U_{z,t}$ given $(x, y, \mathcal{P})$. Therefore, using $(a+b)^2 \le 2a^2 + 2b^2$, $\|F_{U,t}\|_2 \le 1$, and $\|P_t\, \partial z_{t,\mathcal{A}_t}/\partial \theta\|_2 \le L_\theta$,

$$\mathbb{E}[\|b_t\|_2^2 \mid x, y, \mathcal{P}] \le 2L_\theta^2 \mathbb{E}[\|U_{z,t}\|_2^2 \mid x, y, \mathcal{P}] \left(C_\alpha^2 + L_\alpha^2 H_t^2 \prod_{j=t+1}^{T_t - 1} \mathbb{E}[(1 + \Gamma_j)^2 \mid x, y, \mathcal{P}] \right) \tag{69}$$

$$\le C_0 \eta_t \frac{H_t^2}{\tau_{\text{soft}}} \exp\left\{ \sum_{j=t+1}^{T_t - 1} \rho_j \right\}, \tag{70}$$

where

$$C_0 \coloneqq \frac{8 L_\theta^2 L_\alpha^2}{\tau_{\text{hard}}^2 (1 - \bar{\tau})}. \tag{71}$$

The second inequality also uses $C_\alpha \le L_\alpha$, $H_t \ge 1$, and $\prod_{j=t+1}^{T_t - 1} (1 + \rho_j) \ge 1$.

Survival weighting and minibatch averaging. Let $g = \nabla_\theta^{\text{ST}} \bar{G}$, and let b_t be the derivative of the conditional segment beginning at step t with respect to the temporal parameter copy θ_t, holding its inherited state and later parameter copies fixed. The chain rule for tying $\theta_0, \dots, \theta_{T-1}$ back to the shared θ gives

$$g = \sum_{t < T} \text{sg}(\tilde{s}_t) b_t. \tag{72}$$

Since $T \le k$, Cauchy–Schwarz gives

$$\|g\|_2^2 \le k \sum_{t < T} \tilde{s}_t^2 \|b_t\|_2^2. \tag{73}$$

The incoming relaxed state and global survival at step t depend only on the prefix Gumbels. Conditioning on $(x, y, \mathcal{P})$ and this prefix fixes both quantities while preserving the conditional independence of current and future residual Gumbels. Applying (70) and then the tower property yields

$$\mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}[\|g\|_2^2] \le C_0 k\, \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}} \left[\sum_{t < T} \tilde{s}_t^2 \eta_t \frac{H_t^2}{\tau_{\text{soft}}} \exp\left\{ \sum_{j=t+1}^{T_t - 1} \rho_j \right\} \right]. \tag{74}$$

Finally, the variance over the data and Gumbel noise is at most the corresponding second moment. Independence of the N minibatch rollouts then gives

$$\mathbb{E}_{\mathcal{B},\mathcal{E}} \left[\|\widehat{g} - \mathbb{E}_{\mathcal{B},\mathcal{E}}[\widehat{g}]\|_2^2 \right] = \frac{1}{N} \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}} \left[\|g - \mathbb{E}_{\mathbf{x},\mathbf{y},\boldsymbol{\varepsilon}}[g]\|_2^2 \right]. \tag{75}$$

Substituting $T_t = T$ and $H_t = T - t$ proves (10). □

Algorithm 1 Efficient Full-Trajectory NM-PPG Training

Input: Dataset $\mathcal{D}$, B, α, parameters (θ, ϕ), deployment horizon k, temperature τ_{hard} and decreasing schedule $\tau_{\text{soft}}^{(1)} > \cdots > \tau_{\text{soft}}^{(R)} > 0$, entropy coefficient λ_{ent}, step sizes (η_θ, η_ϕ), epochs E, batch size N.
Split $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$; pretrain f_ϕ on random masks.
$(\theta^\star, \phi^\star) \leftarrow (\theta, \phi)$.
for $j = 1, \ldots, R$ **do**
  $\tau_{\text{soft}} \leftarrow \tau_{\text{soft}}^{(j)}$.
  for $e = 1, \ldots, E$ **do**
    for each minibatch $\mathcal{B} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ **do**
      for $n = 1, \ldots, N$ **do** `// Computed in parallel.`
        $(\bar{G}_n, \bar{H}_n, L_{\text{pred},n}, T_n) \leftarrow \text{ST-Rollout}(x^{(n)}, y^{(n)}; \theta, \phi, B, \alpha, k, \tau_{\text{hard}}, \tau_{\text{soft}})$.
      end for
      $\bar{J}_{\text{ent}} \leftarrow N^{-1} \sum_{n=1}^{N} (\bar{G}_n - \lambda_{\text{ent}} \bar{H}_n)$.
      $L_{\text{pred}} \leftarrow \big(\sum_{n=1}^{N} L_{\text{pred},n}\big) / \big(\sum_{n=1}^{N} (T_n + 1)\big)$.
      $\theta \leftarrow \theta - \eta_\theta \nabla_\theta^{\text{ST}} \bar{J}_{\text{ent}}$; $\phi \leftarrow \phi - \eta_\phi \nabla_\phi L_{\text{pred}}$.
    end for
    Evaluate $L_{\text{val}}(\theta, \phi)$ on $\mathcal{D}_{\text{val}}$ using Algorithm 3.
    If $L_{\text{val}}(\theta, \phi) < L_{\text{val}}(\theta^\star, \phi^\star)$, set $(\theta^\star, \phi^\star) \leftarrow (\theta, \phi)$.
    Perform the early-stopping check for stage j.
  end for
  $(\theta, \phi) \leftarrow (\theta^\star, \phi^\star)$.
end for
Return $(\theta^\star, \phi^\star)$.

C.7 Proof of Proposition 4

Proof. Setting $h = 1$ gives $T_{t,1} = t + 1$. Under $\alpha = 0$ and no learned stopping, all acquisition-cost terms vanish and $\tilde{s}_{t,t} = \tilde{s}_{t,t+1} = 1$, so the rooted return in (11) reduces to

$$\bar{G}_{t,1} = \ell\left(f_\phi\left(x(\bar{m}_{t+1})\right), y\right). \tag{76}$$

The local rollout begins at the observed mask, $\tilde{m}_t = m_t$, and forms the relaxed successor $\tilde{m}_{t+1} = m_t + (1 - m_t) \odot \tilde{r}_t$. Applying the ST correction from (9) gives $\bar{m}_{t+1} = m_{t+1} - \text{sg}(\tilde{m}_{t+1}) + \tilde{m}_{t+1}$. Its forward value is the discrete successor m_{t+1}, while its derivative is that of $\tilde{m}_{t+1}$. Consequently, the root contribution in (12) is $\nabla_\theta^{(t),\text{ST}} \ell(f_\phi(x(\bar{m}_{t+1})), y)$, exactly the one-step estimator used by DiFA (Ghosh & Lan, 2023). Without the ST correction, the contribution is $\nabla_\theta^{(t)} \ell(f_\phi(x(\tilde{m}_{t+1})), y)$, exactly the fully relaxed one-step estimator used by GDFS (Covert et al., 2023). Since stopping is unavailable, the global survival factor in (12) equals one at every visited step, so summing these contributions recovers the sequential training estimator of each method. Uniform unit costs make the hard budget equivalent to their native fixed-cardinality setting with $K = \min\{k, \lfloor B \rfloor\}$. □

D Training and Inference Algorithms

Algorithms 1–3 give the full-trajectory NM-PPG procedure used in the main paper. They support arbitrary (B, α, k), including the full objective and its hard- and soft-budget special cases. Model selection uses the empirical objective in (2) under the discrete inference policy: $L_{\text{val}}(\theta, \phi) \triangleq |\mathcal{D}_{\text{val}}|^{-1} \sum_{(x,y) \in \mathcal{D}_{\text{val}}} [\ell(f_\phi(x(m_\theta(x))), y) + \alpha c(m_\theta(x))]$.

Entropy regularization and temperature sharpening serve complementary roles. Entropy discourages premature policy collapse, whereas sharpening brings the relaxation closer to discrete acquisition. Theorem 3 motivates starting with a smoother relaxation because low temperatures magnify variance contributions while decisions remain ambiguous. In the entropy term, detaching the mask inputs prevents gradients from changing earlier acquisitions merely to increase later policy entropy, and detaching the survival weights prevents them from changing stopping decisions merely to collect more entropy. The task-return gradients still propagate through the full trajectory.

Algorithm 2 General-Budget Full-Trajectory ST Rollout

Input: $x, y, k, \theta, \phi, B, \alpha, \tau_{\text{hard}}, \tau_{\text{soft}}$.
$m_0 \leftarrow \mathbf{0}$, $\tilde{m}_0 \leftarrow m_0$, $\bar{m}_0 \leftarrow m_0$, $\tilde{s}_0 \leftarrow 1$, and $T \leftarrow k$.
Set $\bar{G} = \bar{H} = L_{\text{pred}} = W = 0$.
for $t = 0, \dots, T-1$ **do**
  $\mathcal{F}_B(m_t; k) \leftarrow \{a : m_{t,a} = 0,\ c(m_t) + c(a) \le B\}$.
  if $\mathcal{F}_B(m_t; k) = \varnothing$ **then**
    $T \leftarrow t$; **break**.
  end if
  $z_t \leftarrow z_\theta(x(\bar{m}_t))$; mask infeasible features; if $\alpha = 0$, set $z_{t,d+1} \leftarrow -\infty$.
  $\pi_t \leftarrow \pi_\theta(\cdot \mid x(\text{sg}(\bar{m}_t)))$; sample $\varepsilon_t \sim \text{Gumbel}(0,1)^{d+1}$.
  $\tilde{a}_t \leftarrow \text{softmax}((z_t/\tau_{\text{hard}} + \varepsilon_t)/\tau_{\text{soft}})$.
  $\tilde{r}_t \leftarrow \text{softmax}((z_{t,1:d}/\tau_{\text{hard}} + \varepsilon_{t,1:d})/\tau_{\text{soft}})$.
  $r_t \leftarrow \text{onehot}(\text{argmax}_{a \in \mathcal{F}_B(m_t;k)} \tilde{r}_{t,a})$.
  $\bar{r}_t \leftarrow r_t - \text{sg}(\tilde{r}_t) + \tilde{r}_t$; $\bar{\ell}_t \leftarrow \ell(f_\phi(x(\bar{m}_t)), y)$.
  $\Delta_t \leftarrow \alpha(1 - \tilde{a}_{t,d+1})c(\bar{r}_t) + \tilde{a}_{t,d+1}\bar{\ell}_t$.
  $\bar{G} \leftarrow \bar{G} + \tilde{s}_t\Delta_t$; $L_{\text{pred}} \leftarrow L_{\text{pred}} + \bar{\ell}_t$.
  if $|\mathcal{A}_B(m_t; k)| > 1$ **then**
    $\bar{H} \leftarrow \bar{H} + \text{sg}(\tilde{s}_t)\mathcal{H}(\pi_t)/\log|\mathcal{A}_B(m_t; k)|$; $W \leftarrow W + \text{sg}(\tilde{s}_t)$.
  end if
  $\tilde{m}_{t+1} \leftarrow \tilde{m}_t + (1 - \tilde{m}_t) \odot \tilde{r}_t$; $m_{t+1} \leftarrow m_t + (1 - m_t) \odot r_t$.
  $\bar{m}_{t+1} \leftarrow m_{t+1} - \text{sg}(\tilde{m}_{t+1}) + \tilde{m}_{t+1}$.
  $\tilde{s}_{t+1} \leftarrow \tilde{s}_t(1 - \tilde{a}_{t,d+1})$.
end for
$\bar{\ell}_T \leftarrow \ell(f_\phi(x(\bar{m}_T)), y)$; $\bar{G} \leftarrow \bar{G} + \tilde{s}_T\bar{\ell}_T$.
$L_{\text{pred}} \leftarrow L_{\text{pred}} + \bar{\ell}_T$.
Set $\bar{H} \leftarrow \bar{H}/W$ if $W > 0$, and $\bar{H} \leftarrow 0$ otherwise.
Return $(\bar{G}, \bar{H}, L_{\text{pred}}, T)$.

Algorithm 3 General-Budget NM-PPG Inference for One Instance

Input: Instance x, f_ϕ, θ, budget B, cost coefficient α, horizon k.
$m_0 \leftarrow \mathbf{0}$.
for $i = 0, \dots, k-1$ **do**
  $\mathcal{F}_B(m_i; k) \leftarrow \{a : m_{i,a} = 0,\ c(m_i) + c(a) \le B\}$.
  if $\mathcal{F}_B(m_i; k) = \varnothing$ **then**
    Return $\arg\max f_\phi(x(m_i))$.
  end if
  $z_i \leftarrow z_\theta(x(m_i))$; mask infeasible features; if $\alpha = 0$, set $z_{i,d+1} \leftarrow -\infty$.
  $a_i \leftarrow \arg\max_{a \in \mathcal{A}_B(m_i;k)} z_{i,a}$.
  if $a_i = d + 1$ **then**
    Return $\arg\max f_\phi(x(m_i))$.
  end if
  $m_{i+1} \leftarrow m_i + (1 - m_i) \odot \text{onehot}(a_i)$.
end for
Return $\arg\max f_\phi(x(m_k))$.

Algorithm 2 implements the call in line 9. The hard action mask enforces B and k, while the acquisition-cost and survival terms optimize the general objective in (2). During training, the rollout follows the hard feature path conditional on continuing until T, rather than terminating at a sampled stopping action. The survival weights evaluate all possible stopping times differentiably; Algorithm 3 applies the learned stopping decision discretely at inference.

One backward pass through Algorithm 2 computes the gradient in Proposition 1. The predictor is pretrained on random masks and refined by averaging its loss over all rollout-visited masks in each minibatch, including the empty masks.

Table 2: Setting of each method relative to (2).

Method	Planning	Native setting	Termination	Acquisition costs
GDFS	Myopic ($h = 1$)	Hard	Budget	Uniform
DiFA	Myopic ($h = 1$)	Hard	Budget	Uniform
DIME	Myopic ($h = 1$)	Hard and soft	Learned stopping or budget	Non-uniform
GSMRL	Non-myopic ($h = k$)	Soft	Learned stopping	Non-uniform
OL	Non-myopic ($h = k$)	Hard	Budget	Non-uniform
NM-PPG (ours)	Non-myopic ($h = k$)	Full, hard, and soft	Learned stopping or budget	Non-uniform
AACO	Approx. non-myopic (no explicit h)	Soft	Learned stopping	Non-uniform
SEFA	Approx. non-myopic (no explicit h)	Hard	Budget	Uniform

E Experimental Details and Additional Results

This section describes the baseline methods and their adaptations, datasets, hyperparameters, model-selection and evaluation procedures, and additional experimental results. Each run uses one method, dataset, trade-off value (hard budget B or soft-budget cost coefficient α), and random seed, and was trained on one NVIDIA A40 GPU.

E.1 Baseline Methods

We implement each baseline according to its original paper and released code and group them by their planning interpretation from Definition 1: myopic methods optimize one-step decisions with $h = 1$, while non-myopic RL methods optimize the complete acquisition episode with $h = k$. AACO and SEFA instead use approximate non-myopic, non-adaptive proxy scores without recursively optimizing an adaptive h-step policy, so they have no explicit planning depth. Table 2 summarizes these properties.

E.1.1 Myopic Methods

GDFS. GDFS (Covert et al., 2023) amortizes the greedy conditional-mutual-information policy. It jointly trains a predictor and acquisition network by replacing the next discrete feature with a soft mask and minimizing the prediction loss after that single acquisition. It therefore uses $h = 1$: at each step it estimates which feature is most useful if prediction follows immediately. At inference, GDFS repeatedly acquires the highest-scoring unobserved feature until a fixed number of features has been selected. The original formulation corresponds to the hard-budget case of (2) with a cardinality budget and uniform feature costs; it has no stopping action or cost-dependent acquisition rule.

DiFA. DiFA (Ghosh & Lan, 2023) also uses $h = 1$, training the acquisition policy from the prediction loss after one successor acquisition. It uses a straight-through Gumbel–Softmax sample so that the forward pass acquires one discrete feature while the backward pass differentiates through its relaxation. The learned policy is applied sequentially until exactly K features have been acquired. Thus, the main DiFA method is a myopic, hard-budget method with uniform costs and no learned stopping action.

DIME. DIME (Gadgil et al., 2024) trains a predictor together with a value network that estimates the conditional mutual information between each unobserved feature and the label, equivalently the expected one-step reduction in cross-entropy under an optimal predictor. It greedily selects the feature with largest estimated value, or largest value per unit cost when costs are non-uniform, and therefore also uses $h = 1$. The original paper provides three termination rules: a fixed acquisition budget, a confidence threshold, and a penalized rule that stops when no feature's estimated information per unit cost exceeds a threshold. DIME natively supports both the hard-budget and soft-budget settings, as well as non-uniform feature costs, without retraining the value network for each termination rule.

E.1.2 Non-Myopic Methods

GSMRL. GSMRL (Li & Oliva, 2021) is a non-myopic hybrid RL method. It uses PPO (Schulman et al., 2017) for model-free policy training, while an ACFlow-style generative surrogate estimates

conditional feature and label distributions from the current partial observation. These estimates provide information-gain reward shaping and side information to the actor–critic policy. The policy is optimized using the return over the complete acquisition episode, corresponding to $h = k$. Acquiring feature a incurs $\alpha c(a)$, while a learned stopping action terminates acquisition and incurs the prediction loss. GSMRL therefore directly targets the soft-budget special case of (2) and supports non-uniform feature costs.

OL. Opportunistic Learning (OL) (Kachuee et al., 2019) is a non-myopic, model-free RL method based on a DQN (Mnih et al., 2015). Its action-value network learns the cumulative return over the acquisition episode, corresponding to $h = k$, from rewards based on the expected change in predictive uncertainty per unit acquisition cost. The network shares intermediate representations with the predictor. The original procedure acquires the highest-valued affordable feature until a user-specified per-instance budget, an external stopping condition, or exhaustion of the feature set. Since stopping is not an action learned from the cost–loss objective, OL corresponds to the hard-budget setting of (2). Its value-per-cost reward and budget accounting natively support non-uniform feature costs.

E.1.3 Approximate Non-Myopic Methods

AACO. AACO is the practical approximation to the acquisition-conditioned oracle introduced in (Valancius et al., 2024). Given the current observation, it uses nearby training instances to estimate the expected prediction loss for sampled subsets of unobserved features and minimizes this loss plus an acquisition-cost penalty. If the best subset is empty, acquisition stops; otherwise, one feature from that subset is acquired and the procedure repeats. AACO's subset score is non-myopic, but the candidate subset is evaluated as a fixed joint acquisition rather than an adaptive sequence that responds to intermediate feature values. It therefore has no explicit planning depth h. The cost penalty and instance-dependent stopping place it in the soft-budget setting. The oracle objective permits a general subset-cost function, although the original experiments use equal feature costs.

SEFA. SEFA (Norcliffe et al., 2025) learns stochastic latent encodings of partially observed instances and scores candidate acquisitions by how their possible latent realizations affect the predictive distribution. It is designed to identify jointly informative features without training an RL policy, but its score is not obtained by recursively optimizing an adaptive h-step return. SEFA therefore has no explicit planning depth h. Its original evaluation follows the resulting acquisition ordering and measures predictive performance after each fixed number of acquisitions. Consequently, the implemented method has no learned stopping action and corresponds to a hard cardinality budget with uniform acquisition costs.

E.1.4 Cross-Setting Adaptations

For comparisons under a common objective, we additionally consider simple cross-setting variants and report them separately from the original methods. *(i) Entropy-stopping adaptations.* GDFS, DiFA, and SEFA can be evaluated in the soft-budget setting by applying their acquisition rules sequentially and stopping when the normalized entropy of the predictive label distribution, $\mathcal{H}(f_\phi(x(m_t)))/\log|\mathcal{Y}|$, falls below α (Gadgil et al., 2024). This gives instance-dependent stopping without changing their acquisition scores or training them directly on the soft-budget objective. Such entropy-threshold rules are a common heuristic for equipping myopic AFA methods with instance-adaptive stopping (Gadgil et al., 2024; Aronsson et al., 2026). *(ii) RL methods.* GSMRL and OL can be moved between the hard- and soft-budget settings by retaining their respective PPO and DQN estimators while changing the return and available actions as in Section 3.1. The hard-budget variant sets $\alpha = 0$, masks the stopping action while a feature remains feasible, and enforces B; the soft-budget variant makes B non-binding, charges $\alpha c(a)$ for each acquisition, and permits stopping with terminal prediction loss. DIME already supports both settings natively. *(iii) AACO.* Its hard-budget adaptation sets $\alpha = 0$, restricts candidate subsets to those that fit the remaining budget, and excludes the empty subset while an affordable feature remains. This preserves AACO's subset-loss estimate while enforcing the per-instance budget.

Table 3: Summary of benchmark datasets. For Cube-NM we use $n_c = 5$.

Dataset	Type	Train	Validation	Test	# Features	# Classes	Imbalanced	Non-uniform cost
Cube-NM ($\sigma = 0.1$)	Synthetic	7,000	1,500	1,500	55	8	No	Yes
Cube-NM ($\sigma = 0.2$)	Synthetic	7,000	1,500	1,500	55	8	No	Yes
Syn1	Synthetic	60,000	10,000	10,000	11	2	No	No
Syn3	Synthetic	60,000	10,000	10,000	11	2	No	No
Connect4	Real-world tabular	47,290	10,133	10,134	42	3	No	No
Splice	Real-world tabular	2,233	478	479	60	3	No	No
EngineFaultDB	Real-world tabular	39,200	8,400	8,399	14	4	No	No
Metabric	Real-world medical	1,329	285	284	489	6	No	No
Mortality	Real-world medical	9,409	2,016	2,017	26	2	Yes	Yes
Diabetes	Real-world medical	64,443	13,809	13,810	33	3	Yes	Yes
MNIST	Real-world image	50,000	10,000	10,000	784	10	No	No
Fashion-MNIST	Real-world image	50,000	10,000	10,000	784	10	No	No

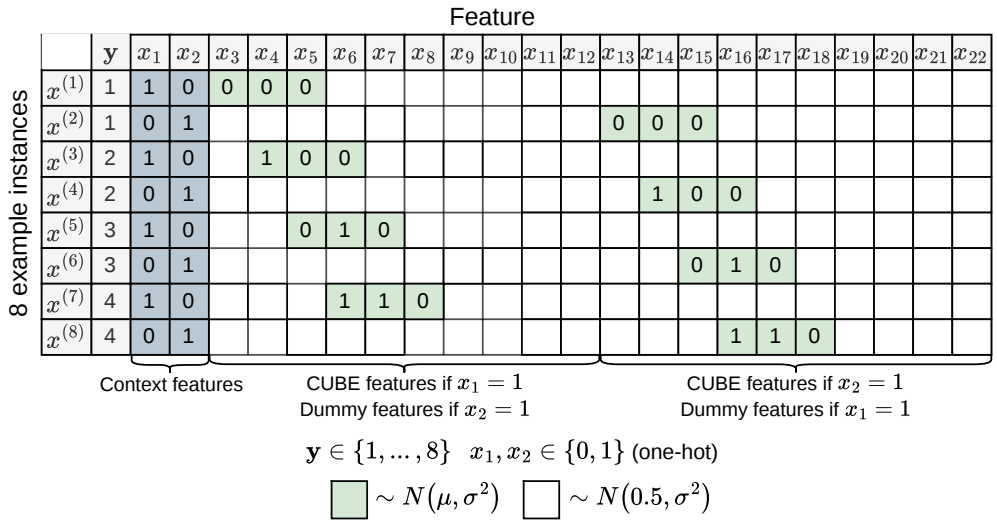

Figure 2: Illustration of the Cube-NM context mechanism for $n_c = 2$. The first two features x_1, x_2 form the one-hot context. If $x_1 = 1$, features $x_3, \ldots, x_{12}$ are the CUBE block and $x_{13}, \ldots, x_{22}$ are dummy features; if $x_2 = 1$, the roles are reversed. Blue entries are sampled from $\mathcal{N}(\mu, \sigma^2)$, where μ is the corresponding coordinate of the class prototype p_y for label y. White entries are sampled from the dummy distribution $\mathcal{N}(0.5, \sigma^2)$. The experiments use $n_c = 5$ and $\sigma \in \{0.1, 0.2\}$.

E.2 Datasets

This subsection provides additional details for each benchmark dataset used in our experiments. Table 3 provides summary information about each dataset. The # Features column reports the number of acquisition feature groups used by the AFA policies. Some processed features, such as one-hot encoded categorical variables, are treated as a single acquisition because they effectively correspond to the same underlying feature; this avoids forcing the policy to learn to acquire all components of such a group separately. We specify below how feature groups are formed for each dataset.

Cube-NM. Cube-NM (Schütz et al., 2026) is a synthetic AFA dataset designed to isolate the benefit of non-myopic acquisition. Each instance has a latent context that selects which one of several feature blocks contains label information; the context features reveal this block identity but do not directly reveal the label. We use x_i to denote the scalar value of feature i. Formally, with n_c possible contexts, the first n_c features, $x_1, \ldots, x_{n_c}$, are one-hot context indicators. The remaining features are split into n_c feature blocks of size 10: the block for context j consists of $x_{n_c+10(j-1)+1}, \ldots, x_{n_c+10j}$. In Figure 2, we illustrate this construction for $n_c = 2$: x_1, x_2 are context features, the first candidate block is $x_3, \ldots, x_{12}$, and the second candidate block is $x_{13}, \ldots, x_{22}$.

An instance is generated by sampling a label $y \in \{1, \ldots, 8\}$ and an active context $r \in \{1, \ldots, n_c\}$. The context features are set to $x_r = 1$ and $x_j = 0$ for $j \leq n_c$, $j \neq r$. Because r is sampled

independently of y, these context features do not directly predict the label; they only reveal which feature block of size 10 contains label information.

The label controls the mean of the active block through a class prototype p_y. Here $p_y \in \mathbb{R}^{10}$ is not an observed feature; it is the length-10 mean vector used to generate the informative block for class y. To define it, set $q = y - 1$, write q in binary as (q_0, q_1, q_2), initialize all ten entries of p_y to 0.5, and replace entries $q+1, q+2, q+3$ by (q_0, q_1, q_2). For example, if $y = 6$, then $q = 5$, the bits are $(1, 0, 1)$, and p_y has entries $1, 0, 1$ in positions $6, 7, 8$ and 0.5 elsewhere.

Finally, features are sampled. If r is the active context, then for local coordinate $\ell = 1, \ldots, 10$, the feature $x_{n_c+10(r-1)+\ell}$ in the active block is sampled from $\mathcal{N}(p_y^{(\ell)}, \sigma^2)$. For every inactive context $j \neq r$, the dummy-block feature $x_{n_c+10(j-1)+\ell}$ is sampled from $\mathcal{N}(0.5, \sigma^2)$. Therefore, only the active block has a label-dependent mean pattern; all dummy blocks are centered at 0.5 and carry no label information. In Figure 2, this means that if $x_1 = 1$, then $x_3, \ldots, x_{12}$ are sampled from the class prototype and $x_{13}, \ldots, x_{22}$ are dummy features; if $x_2 = 1$, the roles are reversed.

In our experiments we use $n_c = 5$, noise levels $\sigma \in \{0.1, 0.2\}$, and $5 + 5 \cdot 10 = 55$ processed features, each treated as a separate acquisition action. Each context indicator costs 0.2, while each feature in the candidate blocks costs 1. This creates the non-myopic structure: the context indicators have no immediate predictive value, but jointly identify which feature block should be queried next. The paper that introduced Cube-NM proves that, in the noiseless setting, non-myopic selection can require substantially fewer acquisitions than a myopic policy:

Theorem 4 (Informal Cube-NM result (Schütz et al., 2026))**.** *Consider the noiseless Cube-NM dataset with n_c contexts and $\sigma = 0$, so that 100% prediction accuracy is achievable. Then a myopic policy requires, in expectation over instances, $13(2n_c+1)/16$ feature acquisitions to achieve 100% accuracy. In contrast, there exists an optimal non-myopic policy that achieves 100% accuracy after acquiring only $\mathbf{1}\{n_c \geq 2\} + 9/4$ features in expectation.*

The exact constants in Theorem 4 assume that the context is acquired as a single categorical feature, equivalently as one acquisition group containing the one-hot context indicators. In our experiments, the context indicators are separate actions with lower cost than the remaining features. The same qualitative non-myopic structure applies, but the exact expected acquisition counts and costs differ.

Theorem 4 also illustrates how the advantage of non-myopic selection depends on the number of contexts. As n_c increases, the myopic acquisition cost grows linearly in n_c, while the non-myopic policy remains constant after $n_c \geq 2$. This reflects the general principle that non-myopic AFA is most useful when early acquisitions reveal which later acquisitions are valuable; see Appendix F for a more detailed discussion.

Syn1. Syn1 is one of the synthetic datasets used in previous AFA work to evaluate non-myopic acquisition methods (Norcliffe et al., 2025). These datasets are useful because they are constructed so that the optimal acquisition order is known. Each instance consists of 11 independent standard-normal features $x_1, \ldots, x_{11}$, with x_{11} acting as a context feature that determines which feature interaction generates the label. If $x_{11} < 0$, the label probability is determined by the interaction $\ell_1 = 4x_1x_2$, while if $x_{11} \geq 0$, it is determined by the quadratic block $\ell_2 = 1.2\sum_{j=3}^{6} x_j^2 - 4.2$. In both cases, $p(y = 1) = 1/(1 + \exp(\ell))$, where ℓ is the selected logit. Thus, x_{11} does not directly define the label by itself; instead, it indicates whether the useful information is contained in the pair (x_1, x_2) or in the block $x_3, \ldots, x_6$. This makes Syn1 a controlled test of whether an AFA method can acquire a context feature before selecting the features that are predictive under that context.

Syn3. Syn3 is another synthetic dataset used in previous AFA work to evaluate non-myopic acquisition methods (Norcliffe et al., 2025). It uses the same 11-feature construction and the same context variable x_{11}, but switches between two higher-order feature blocks. If $x_{11} < 0$, the label is generated from the quadratic block $\ell_2 = 1.2\sum_{j=3}^{6} x_j^2 - 4.2$. If $x_{11} \geq 0$, it is generated from the nonlinear block $\ell_3 = -10\sin(0.2x_7) + |x_8| + x_9 + \exp(-x_{10}) - 2.4$. As in Syn1, $p(y = 1) = 1/(1 + \exp(\ell))$ for the selected logit. Syn3 is therefore a harder context-dependent benchmark with a known optimal acquisition order: the policy must first identify which branch applies, and then acquire features from either $x_3, \ldots, x_6$ or $x_7, \ldots, x_{10}$. All Syn1 and Syn3 features have unit acquisition cost.

Connect4. Connect4 (Tromp, 1995) is a board-state classification dataset. The prediction task is to classify the eventual game outcome, i.e., loss, draw, or win, from the current Connect Four board.

The board has 42 cells, each taking one of three values: empty, player x, or player o. We one-hot encode each cell, giving 126 processed binary features, and treat each board cell as one acquisition group so that acquiring a cell reveals its full categorical state. This is useful for AFA because only a subset of board positions may be needed to determine the outcome, mimicking decision settings where an agent should inspect only the most informative parts of a structured state.

Splice. Splice (uci, 1991) is a DNA splice-junction classification dataset. The prediction task is to classify a length-60 DNA sequence as a non-splice example, an exon-intron junction, or an intron-exon junction. We one-hot encode the nucleotide at each sequence position, yielding 480 processed binary features, and treat each sequence position as one acquisition group. This is relevant for AFA because biological sequence assays can be costly, and the predictive signal may be concentrated in a small subset of positions around the junction.

EngineFaultDB. EngineFaultDB (Vergara et al., 2023) is a real-world tabular engine fault diagnosis dataset. The prediction task is to identify one of four engine fault classes from 14 numeric sensor measurements. We normalize the measurements using training-set statistics and treat each sensor value as one acquisition. This is a natural AFA setting because diagnostic systems may be able to query additional sensors or tests sequentially, but each measurement can consume time, energy, or hardware resources.

Metabric. Metabric (Curtis et al., 2012; Pereira et al., 2016) is a real-world medical breast cancer dataset. The prediction task is to classify the PAM50 plus claudin-low molecular subtype, giving six classes after removing invalid labels. We use the molecular feature block from the processed METABRIC table, resulting in 662 gene-expression and mutation features. Expression features are winsorized and standardized using training-set statistics, and each molecular measurement is treated as one acquisition. This is important for AFA because molecular profiling can be expensive, and an adaptive policy may reduce the number of assays needed for accurate subtype prediction.

Mortality. Mortality is derived from the National Health and Nutrition Examination Survey (NHANES) (Centers for Disease Control and Prevention, 2026), and was processed into an outpatient mortality benchmark in (Erion et al., 2022). The prediction task is binary 10-year mortality prediction from demographic variables, laboratory measurements, examination results, and questionnaire-derived variables. The raw clinical variables are expanded into 118 processed columns, including continuous measurements, binary indicators, missingness indicators, test-status indicators such as unacceptable or test-not-done flags, and one-hot encodings for categorical measurements. In the AFA setup, acquisition is defined at the level of the underlying clinical variable or measurement panel: acquiring a group reveals all processed columns derived from that variable or panel. For example, a laboratory measurement group can include the measured value together with its missingness and test-status indicators, and categorical urine-test groups include all corresponding one-hot indicators. This gives 26 acquisition groups with non-uniform costs. The feature costs follow the CoAI setup, where costs were assigned to reflect monetary burden and patient inconvenience (Erion et al., 2022). We standardize numeric features and handle missing values using training-set statistics. This is an important AFA benchmark because clinical risk assessment often involves deciding which patient information or lab tests are worth collecting before making a prediction.

Diabetes. Diabetes is derived from NHANES (Centers for Disease Control and Prevention, 2026) in the OL paper (Kachuee et al., 2019), a prior AFA work that used this dataset as a benchmark. The feature set contains demographic variables, laboratory results, examination measurements, and questionnaire answers, including variables such as age, gender, ethnicity, total cholesterol, triglycerides, weight, height, smoking, alcohol use, and sleep habits. The prediction task is three-class diabetes status prediction, where fasting glucose values define the classes normal, pre-diabetes, and diabetes according to standard threshold values. The original benchmark contains 92,062 samples and 45 processed feature columns. In our AFA setup, one-hot encoded categorical variables are treated as acquisition groups, so acquiring the original categorical variable reveals all of its one-hot indicators; this gives 33 acquisition groups. We impute missing values using training-set statistics, winsorize and standardize continuous features, and keep binary features unchanged. The non-uniform feature costs follow the OL setup, where a medical expert assigned costs based on financial burden, patient privacy, and patient inconvenience. This dataset is relevant for AFA because diabetes screening combines cheap background variables with potentially more costly clinical measurements, making adaptive test selection practically meaningful.

MNIST. MNIST (LeCun et al., 1998) is a 10-class handwritten digit recognition dataset. The prediction task is to classify the digit identity from a 28×28 grayscale image. As for Fashion-MNIST, we flatten each image into 784 normalized pixel features and treat pixels as individually acquirable features. This provides a canonical image benchmark for evaluating whether AFA methods can classify accurately while observing only a subset of pixels. The AFA motivation is again fast visual decision-making: in applications such as real-time detection or embedded recognition, acquiring or processing fewer image locations can reduce latency and computation (Viola & Jones, 2004).

Fashion-MNIST. Fashion-MNIST (Xiao et al., 2017) is a 10-class image classification dataset of clothing items. The prediction task is to classify the object category from a 28×28 grayscale image. We flatten each image into 784 normalized pixel features and treat each pixel as one acquisition. In the AFA setting, this tests whether a method can identify an object while observing only a subset of pixels. This is motivated by real-time visual recognition settings, where fast inference can require focusing computation on informative image regions rather than processing every pixel, as in classical cascaded detection systems (Viola & Jones, 2004).

E.3 Hyperparameters and Model Selection

Operating points and model selection. We distinguish the parameters that trace a cost–performance curve from hyperparameters selected for training. Hard-budget curves vary the budget B, while soft-budget curves vary the cost coefficient α. Checkpoints and ordinary hyperparameters are selected on the validation split; the test split is evaluated only after this selection.

Shared training setup. All methods use the same train/validation/test splits, preprocessing, feature groups, acquisition costs, prediction loss, and evaluation metric on each dataset. Unless a method's original architecture requires otherwise, predictors and policies are two-hidden-layer ReLU MLPs of width 128. Predictors use dropout 0.2 on Cube-NM, Connect4, EngineFaultDB, Splice, MNIST, and Fashion-MNIST, 0.3 on Syn1 and Syn3, and 0.5 on Metabric, Mortality, and Diabetes; policy networks use no dropout. We use cross-entropy loss and Adam (Kingma & Ba, 2015). During method training, all methods use inverse-frequency class-weighted cross-entropy on Mortality and Diabetes and unweighted cross-entropy otherwise. Shared predictor pretraining uses unweighted cross-entropy, batches of 512, learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at 5. It runs for at most 500 fully observed epochs followed by 4,000 random-mask epochs.

NM-PPG hyperparameters. The categorical-policy temperature is $\tau_{\text{hard}} = 1$, and staged training uses $\tau_{\text{soft}} \in [0.8, 0.5, 0.2, 0.05, 0.02]$ in this order. The default entropy coefficient is $\lambda_{\text{ent}} = 0.2$; we use 0.1 on Connect4, Syn1, and Syn3, and 0.05 for soft-budget Mortality. The policy and predictor are optimized with batches of 256, learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at 5. The policy additionally receives detached predictor class probabilities on every dataset except Syn1 and Syn3. Each temperature stage runs for at most 2,000 epochs with early-stopping patience 150, and each stage is warm-started from the best validation checkpoint across all completed stages. As described in Appendix D, the predictor is pretrained on random masks and refined on states visited by the acquisition policy.

Baseline hyperparameters. We use the hyperparameters and implementation choices recommended by each baseline's paper and released implementation, tuning from these defaults only on the validation split. Where the method permits a common setup, we use width 128, training batches of 256, learning rate 10^{-3}, weight decay 10^{-4}, gradient clipping at 1, and early-stopping patience 150. GDFS and DiFA use five relaxation stages, $[1.0, 0.5623, 0.3162, 0.1778, 0.1]$, with at most 2,000 epochs per stage. DIME uses exploration probability 0.2. For GSMRL, we use PPO with five update epochs, discount 0.95, GAE parameter 0.95, clipping ratio 0.2, entropy coefficient 0.01, rollout batches of 256, and PPO minibatches of 512. OL uses discount 1, rollout batches of 256, replay minibatches of 512, a buffer of 50,000 transitions, soft target updates with coefficient 10^{-3}, and exploration decreasing from 0.2 to 0.05. AACO uses 5–20 nearest neighbors and 1,000 candidate subsets. SEFA uses latent dimension 8, $\beta = 10^{-3}$, 100 samples during training, and 200 samples for prediction and acquisition.

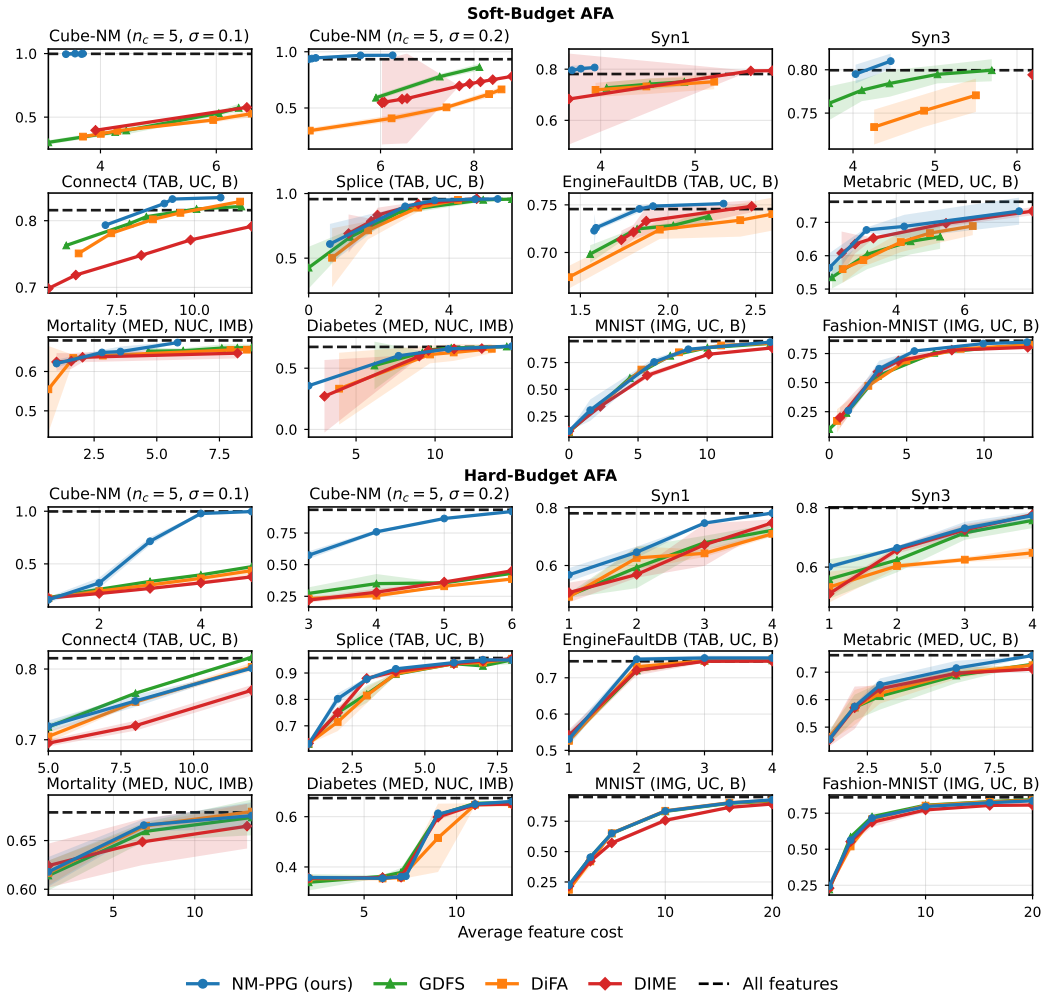

Figure 3: Cost–performance curves for NM-PPG and the myopic baselines GDFS, DiFA, and DIME under soft and hard budgets. Shaded bands show one standard deviation across seeds, and dashed lines show full-feature performance.

E.4 Additional Results

E.4.1 Method-Group Result Grids

Figures 3 and 4 provide enlarged cost–performance comparisons with the myopic and non-myopic baselines, respectively, under both objective settings. The dashed lines indicate full-feature performance, and the shaded bands show one standard deviation across seeds. Section 4 discusses the main findings.

E.4.2 Full Quantitative Results

Table 4 reports normalized AUC for every dataset under both objective settings and shows that NM-PPG is the most consistently strong method across datasets and settings.

E.4.3 Ablations

Figure 5 isolates three components of NM-PPG. *Without ST* replaces the hard-forward ST rollout with fully relaxed masks and actions during training. *Without entropy* sets the entropy coefficient to zero while retaining the ST rollout and staged temperatures. *Fixed* τ_{soft} removes temperature staging

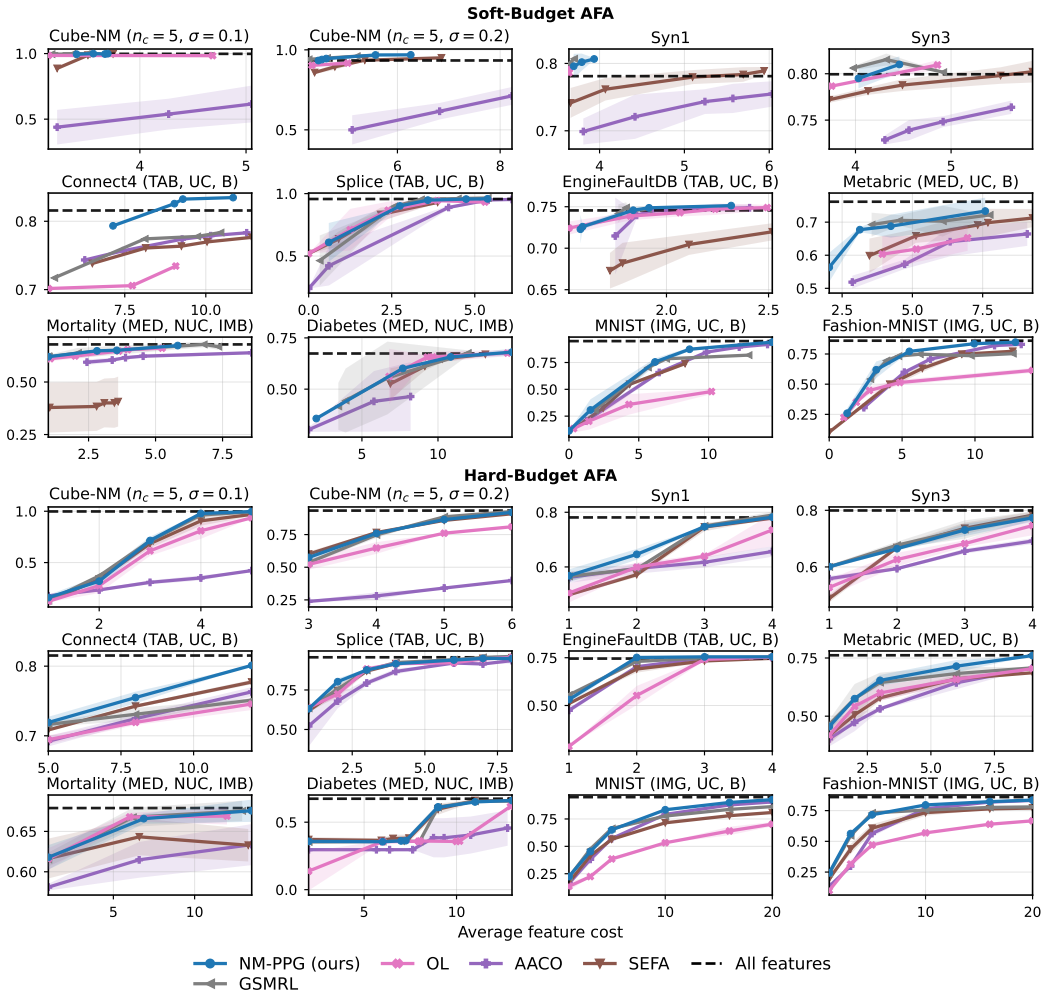

Figure 4: Cost–performance curves for NM-PPG and the non-myopic baselines GSMRL, OL, AACO, and SEFA under soft and hard budgets. Shaded bands show one standard deviation across seeds, and dashed lines show full-feature performance.

and trains at the temperature of the stage containing the best validation checkpoint of standard NM-PPG for the same operating point. Overall, NM-PPG is consistent with or outperforms these variants across datasets and settings. Removing entropy causes the clearest degradations, particularly on the hard-budget Cube-NM tasks, while removing ST or temperature staging occasionally matches NM-PPG but provides no consistent improvement.

E.4.4 Gradient Variance Analysis

Figure 6 compares gradient estimators over the first two temperature stages of NM-PPG training for CUBE-NM1, Syn3, Connect4, Metabric, and MNIST under hard budgets, and for CUBE-NM1 under a soft budget. At each recorded epoch, we fix the current policy and predictor and estimate the task-return gradient using ST, REINFORCE, and REINFORCE with a constant baseline estimated from 16 independent pilot minibatches. Each estimator uses the same $R = 64$ validation minibatches of size $N = 256$, sampled with replacement, with independent acquisition randomness across repetitions. We report $\widehat{V} = \sum_{r=1}^{R} \|g_r - \bar{g}\|_2^2/(R-1)$, where $\bar{g} = R^{-1}\sum_{r=1}^{R} g_r$, and the norm $\|\bar{g}\|_2$. Measurements exclude dropout, entropy regularization, and gradient clipping. Only the ST estimator updates the policy during training.

Table 4: Normalized cost–performance AUC for all datasets and methods in the soft- and hard-budget settings.

Soft-budget AFA

Dataset	NM-PPG (ours)	GDFS	DiFA	DIME	GSMRL	OL	AACO	SEFA
Cube-NM1	0.995±0.001	0.448±0.011	0.430±0.010	0.472±0.000	**0.997±0.001**	0.986±0.007	0.549±0.068	0.988±0.008
Cube-NM2	**0.964±0.005**	0.673±0.097	0.459±0.016	0.435±0.271	0.962±0.002	0.915±0.012	0.598±0.023	0.934±0.005
Syn1	**0.808±0.005**	0.740±0.018	0.736±0.023	0.659±0.156	0.807±0.006	0.798±0.007	0.730±0.023	0.772±0.012
Syn3	0.808±0.008	0.794±0.015	0.760±0.014	0.767±0.000	**0.814±0.006**	0.805±0.004	0.753±0.006	0.797±0.009
Connect4	**0.780±0.002**	0.774±0.001	0.766±0.002	0.729±0.003	0.743±0.001	0.710±0.002	0.744±0.001	0.740±0.001
Splice	0.834±0.025	0.819±0.017	0.798±0.011	0.823±0.021	0.826±0.018	**0.843±0.009**	0.716±0.030	0.826±0.008
EngineFaultDB	**0.728±0.004**	0.708±0.004	0.703±0.014	0.704±0.008	0.717±0.006	0.727±0.005	0.707±0.008	0.672±0.011
Metabric	**0.701±0.027**	0.638±0.029	0.658±0.018	0.693±0.010	0.699±0.030	0.624±0.024	0.603±0.016	0.658±0.022
Mortality	0.656±0.003	0.607±0.017	0.639±0.010	0.638±0.012	**0.657±0.009**	0.646±0.005	0.604±0.006	0.394±0.106
Diabetes	0.545±0.029	**0.557±0.002**	0.479±0.029	0.512±0.031	0.553±0.026	0.552±0.028	0.403±0.104	0.523±0.044
MNIST	**0.707±0.007**	0.698±0.008	0.702±0.011	0.642±0.010	0.652±0.014	0.389±0.016	0.641±0.002	0.591±0.008
Fashion-MNIST	**0.691±0.009**	0.653±0.003	0.658±0.005	0.662±0.017	0.650±0.013	0.495±0.011	0.604±0.003	0.580±0.030
Synthetic	0.894±0.003	0.664±0.021	0.596±0.010	0.583±0.063	**0.895±0.001**	0.876±0.006	0.657±0.012	0.873±0.006
Real-world	**0.705±0.007**	0.682±0.002	0.675±0.001	0.675±0.006	0.687±0.009	0.623±0.009	0.628±0.012	0.623±0.018
Overall	**0.768±0.005**	0.676±0.007	0.649±0.004	0.645±0.023	0.756±0.006	0.707±0.008	0.638±0.011	0.706±0.011

Hard-budget AFA

Dataset	NM-PPG (ours)	GDFS	DiFA	DIME	GSMRL	OL	AACO	SEFA
Cube-NM1	0.569±0.015	0.302±0.001	0.280±0.014	0.253±0.011	**0.570±0.008**	0.491±0.036	0.276±0.013	0.543±0.022
Cube-NM2	0.762±0.050	0.365±0.048	0.274±0.039	0.306±0.042	0.758±0.050	0.670±0.030	0.289±0.046	**0.767±0.050**
Syn1	**0.649±0.003**	0.593±0.007	0.591±0.009	0.592±0.018	0.637±0.007	0.589±0.020	0.586±0.018	0.612±0.005
Syn3	0.686±0.006	0.659±0.009	0.601±0.013	0.663±0.013	**0.691±0.014**	0.639±0.014	0.619±0.006	0.667±0.006
Connect4	0.773±0.004	**0.784±0.002**	0.771±0.005	0.741±0.007	0.739±0.005	0.729±0.006	0.739±0.006	0.755±0.002
Splice	0.852±0.006	0.838±0.007	0.835±0.006	0.846±0.007	**0.855±0.004**	0.848±0.003	0.806±0.014	0.847±0.005
EngineFaultDB	**0.638±0.000**	0.628±0.003	0.630±0.008	0.630±0.008	**0.638±0.008**	0.524±0.015	0.610±0.006	0.612±0.007
Metabric	**0.659±0.022**	0.641±0.023	0.643±0.020	0.643±0.003	0.636±0.030	0.613±0.013	0.585±0.030	0.606±0.006
Mortality	0.648±0.010	0.643±0.012	0.648±0.006	0.639±0.022	**0.650±0.010**	0.649±0.008	0.605±0.017	0.629±0.022
Diabetes	0.507±0.002	0.509±0.008	0.492±0.020	0.504±0.003	0.501±0.003	0.407±0.019	0.363±0.063	**0.514±0.014**
MNIST	**0.726±0.002**	0.723±0.010	0.720±0.009	0.675±0.009	0.692±0.013	0.479±0.011	0.681±0.005	0.628±0.003
Fashion-MNIST	0.711±0.002	**0.718±0.001**	0.708±0.002	0.689±0.011	0.682±0.001	0.507±0.002	0.640±0.003	0.637±0.018
Synthetic	**0.667±0.011**	0.480±0.010	0.436±0.011	0.453±0.017	0.664±0.015	0.597±0.021	0.443±0.019	0.647±0.007
Real-world	**0.689±0.003**	0.686±0.002	0.681±0.006	0.671±0.003	0.674±0.003	0.594±0.005	0.629±0.002	0.653±0.004
Overall	**0.682±0.005**	0.617±0.004	0.599±0.007	0.598±0.007	0.671±0.006	0.595±0.008	0.567±0.006	0.651±0.004

E.4.5 Acquisition Paths

Figures 7–11 compare representative soft-budget policies learned by NM-PPG, GDFS, GSMRL, and SEFA. For each dataset, we select operating points with similar average acquisition costs. Each panel is a feature-by-step heatmap: columns are acquisition steps, rows are acquisition groups or STOP, and color gives the percentage of test instances selecting that action. Orange lines show the three most common exact trajectories. Synthetic rows condition on the context value, while the clinical panels aggregate all test instances. For readability, we display the eight most frequently acquired groups and the first eight steps.

Cube-NM1. NM-PPG first acquires the five cheap context bits and then selects features from a context-dependent block: approximately x_6–x_{15}, x_{26}–x_{35}, or x_{46}–x_{55} in the three displayed contexts. GSMRL also learns context-dependent routing and often uses fewer context bits. The myopic GDFS policy largely ignores the context bits because they provide little immediate predictive information. It therefore follows nearly the same sequence of non-context features in all three rows instead of routing each context to its informative block. SEFA is approximately non-myopic and sometimes acquires context bits, but its acquisitions are spread across many paths and are less consistently routed to the corresponding informative block.

Cube-NM2. Cube-NM2 has the same context-dependent structure as Cube-NM1 but with noisier informative features. NM-PPG again acquires the context bits first and routes the three displayed contexts toward their corresponding feature blocks, approximately x_6–x_{15}, x_{26}–x_{35}, and x_{46}–x_{55}. GSMRL learns similar routing while often using fewer context bits. GDFS does not select a context bit first and instead begins with nearly the same locally predictive features across contexts. SEFA uses context features and sometimes reaches the appropriate block, but its follow-up paths are more diffuse than those of NM-PPG and GSMRL.

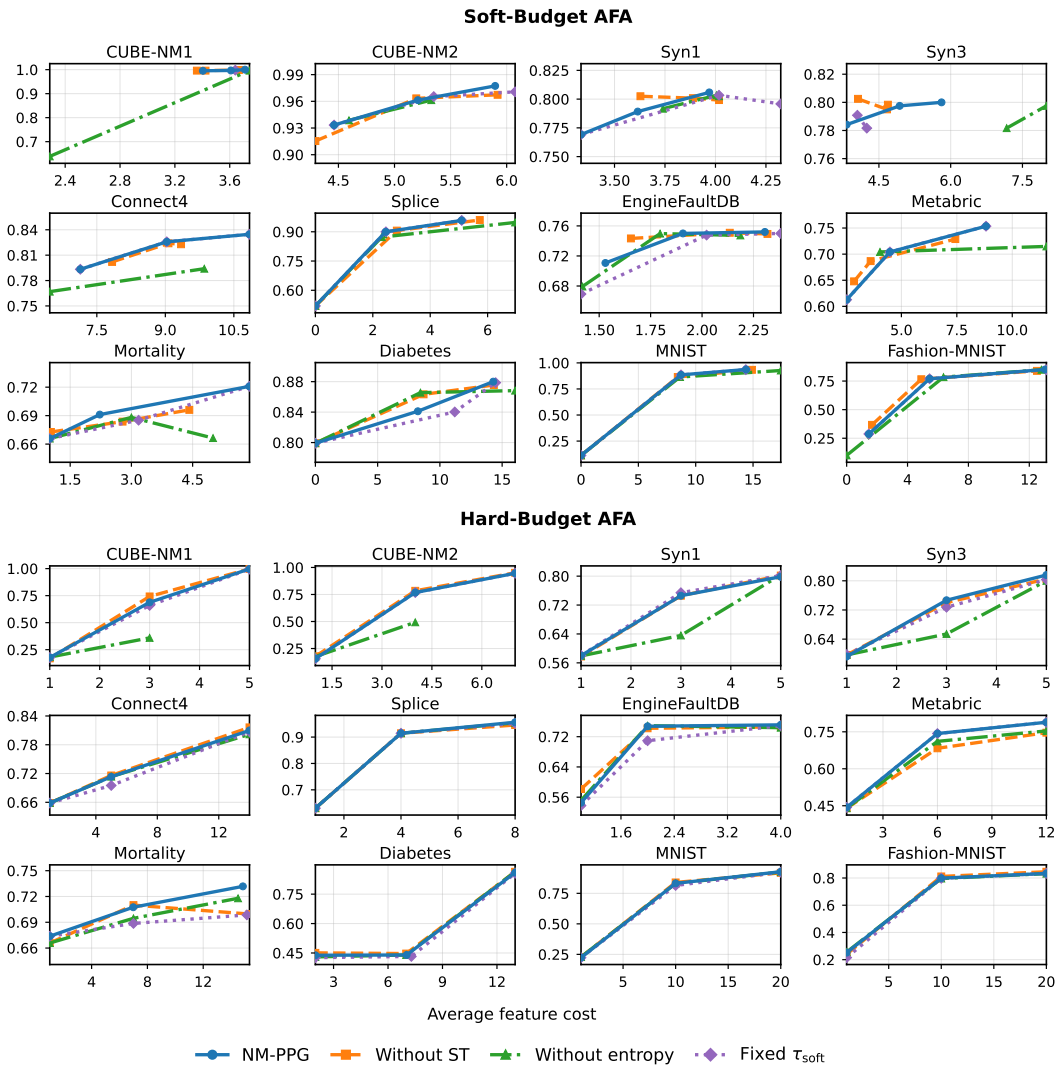

Figure 5: Cost–performance curves for NM-PPG and its component ablations in the soft- and hard-budget settings.

Syn1 and Syn3. On both datasets, NM-PPG acquires the context feature x_{11} first and changes its subsequent acquisitions according to its sign. On Syn1, it follows x_{11} with (x_2, x_1) when $x_{11} < 0$, but uses the x_3–x_6 group when $x_{11} \geq 0$. Syn3 exhibits the same context-first behavior with different follow-up groups. GSMRL similarly selects x_{11} first and routes the two subsets toward different follow-up groups. The myopic GDFS policy instead begins with x_1 on Syn1 and x_7 on Syn3, acquiring x_{11} only afterward in its most common paths. Its one-step score cannot value x_{11} through the later context-dependent decision it enables. SEFA often selects x_{11} early, but its approximate non-myopic scores produce less concentrated follow-up paths than NM-PPG.

Clinical Datasets. On Mortality, x_{17} is serum protein, x_{25} is a grouped urinalysis measurement containing urine pH and hematest, x_{12} is potassium, and x_{19} is sodium. NM-PPG begins with serum protein and then follows several instance-dependent paths through urinalysis, potassium, sodium, hemoglobin and hematocrit (x_9), and creatinine (x_7). GSMRL also learns a multi-step policy, commonly beginning with creatinine before acquiring hemoglobin and hematocrit, urinalysis, sodium, and serum protein. In contrast, the myopic GDFS policy usually acquires potassium and stops, while SEFA often acquires only alkaline phosphatase (x_3) or stops immediately. The latter behavior is consistent with its lower F1-score at a similar average cost.

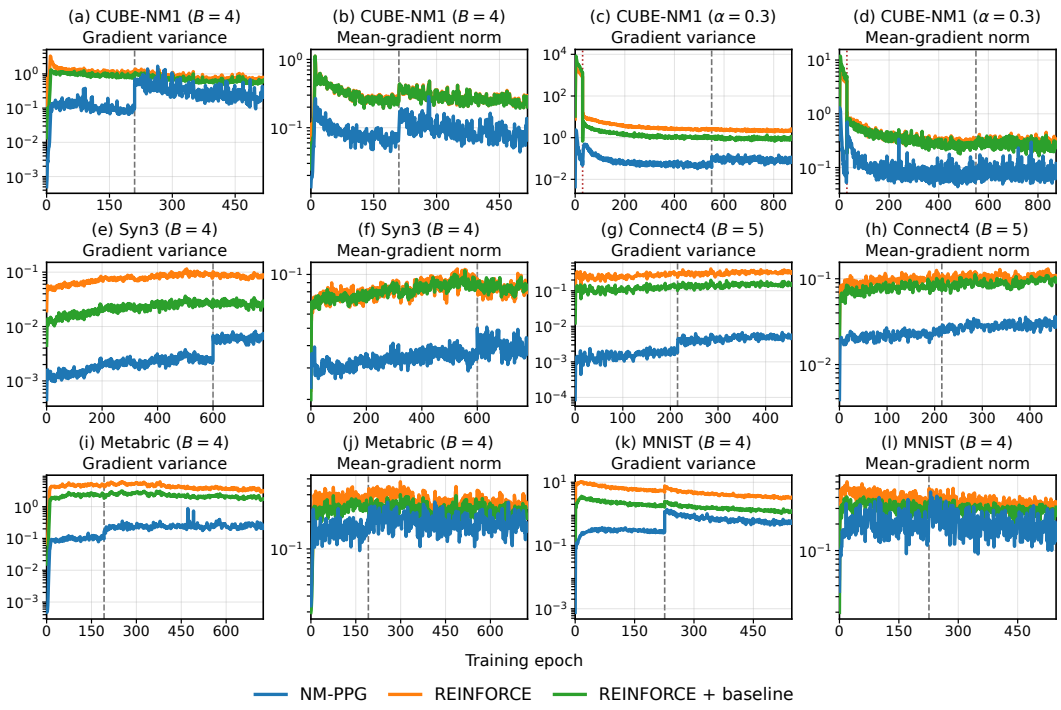

Figure 6: Empirical gradient variance and mean-gradient norm during NM-PPG training. Panel titles specify the dataset and objective setting. Within each run, all three estimators are evaluated at the same current policy and predictor. Dashed lines mark temperature-stage restarts. Dotted lines mark the end of stopping warmup.

On Diabetes, x_2 is age (RIDAGEYR), x_{21} is triglycerides (LBXTR), and x_{23} is LDL cholesterol (LBDLDL). NM-PPG and GSMRL first acquire age and then acquire triglycerides only for a subset of instances. GDFS instead pays for triglycerides immediately and usually stops, while SEFA primarily begins with LDL cholesterol. Thus, NM-PPG and GSMRL learn a context-first screening policy that uses inexpensive demographic information to decide whether a costly laboratory measurement is needed.

On METABRIC, NM-PPG first acquires GATA3 expression (x_{281}) for every instance and then branches to CDK1 (x_{26}), TGFBR2 (x_{269}), EGFR (x_{153}), or KLRG1 (x_{357}) according to its observed value. GSMRL learns a similar GATA3-first policy. GDFS instead begins with EGFR and follows a largely fixed sequence headed by MAPT, while SEFA primarily starts from CDK1. At comparable costs, this adaptive routing yields about 0.68 accuracy for NM-PPG and GSMRL, compared with 0.57 for GDFS and SEFA.

Together, these results show clear non-myopic structure in all three medical datasets: an early measurement determines which follow-up measurements are useful. Myopic methods such as GDFS, DiFA, and DIME score acquisitions by their immediate predictive effect and therefore cannot explicitly value this delayed, context-dependent routing, whereas NM-PPG can optimize it through the full acquisition trajectory. This distinction is particularly important in medical screening, where unnecessary tests increase financial cost, patient burden, diagnostic delay, and potentially clinical risk. A context-first policy can reserve costly or invasive measurements for patients whose earlier observations indicate that the expected diagnostic benefit justifies them.

E.4.6 Runtime

Tables 5 and 6 report mean wall-clock training and test-set evaluation times, respectively, for both objective settings. Training times exclude shared predictor pretraining, while evaluation times cover inference over the complete test split. NM-PPG has evaluation times comparable to the other learned

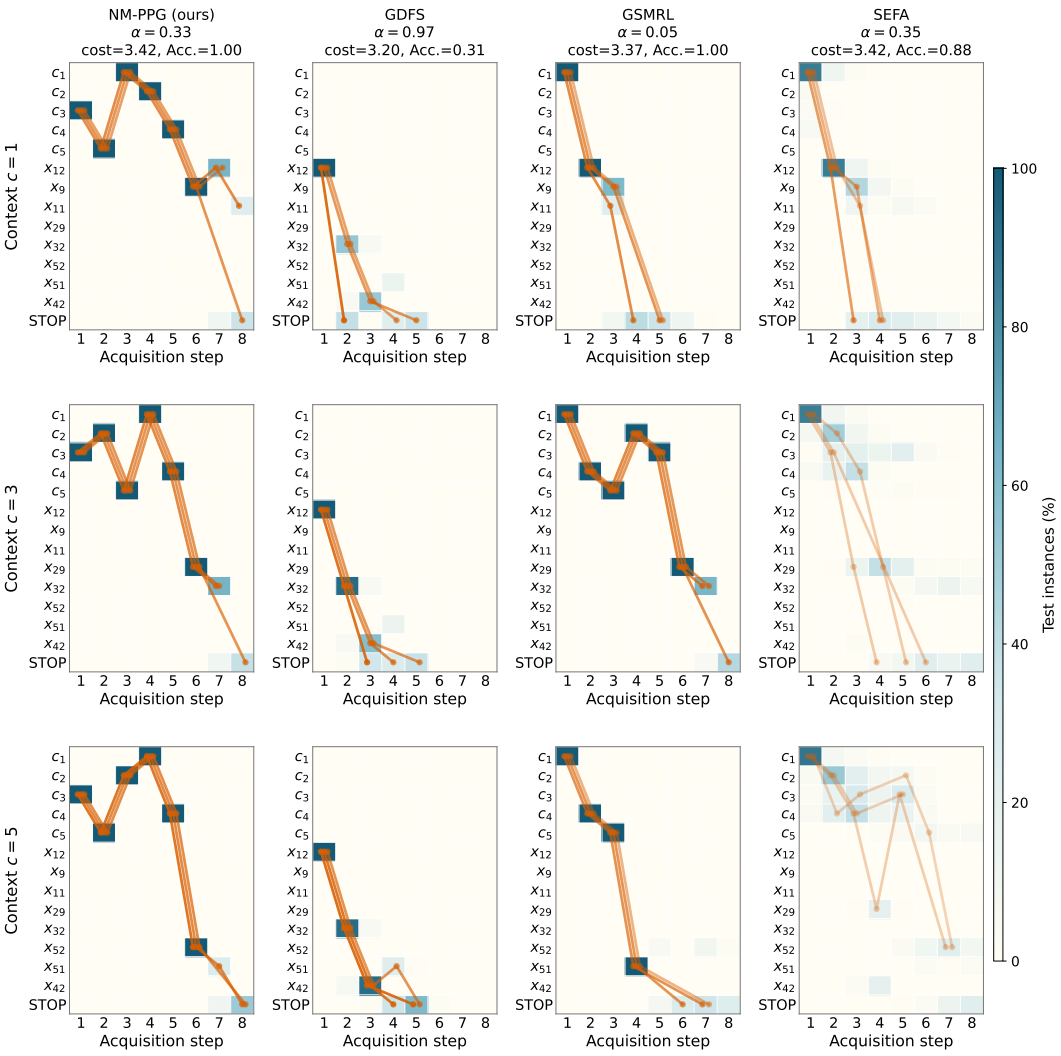

Figure 7: Acquisition paths on Cube-NM1 for three context values. Policies are selected at comparable average costs; panel titles report the corresponding α, test cost, and accuracy.

policies. AACO has negligible policy-training time because its acquisition rule is non-parametric, but its candidate-subset computations make evaluation substantially slower.

E.4.7 Training Dynamics

Figures 12 and 13 show representative NM-PPG training dynamics for every dataset. Rows correspond to the operating points indicated beside each dataset, while columns show prediction loss, acquisition cost, and the complete objective. Blue and orange curves are deterministic evaluations on the training and validation splits, respectively, smoothed with a 15-epoch moving average within each temperature stage. Dashed lines mark transitions in τ_{soft}, and the solid black line marks the globally best validation checkpoint selected for evaluation.

In the soft-budget setting, prediction loss and acquisition cost evolve jointly as the policy learns both which features to acquire and when to stop. In the hard-budget setting, the budget constrains cost directly, and improvements in the objective are consequently driven mainly by lower prediction loss. The close train–validation agreement across most runs indicates limited overfitting, while changes at stage boundaries reflect both the sharper relaxation and the restart from the best checkpoint found so far. The selected epoch need not occur in the final or sharpest stage, which motivates validation-based model selection across all stages rather than retaining only the final iterate.

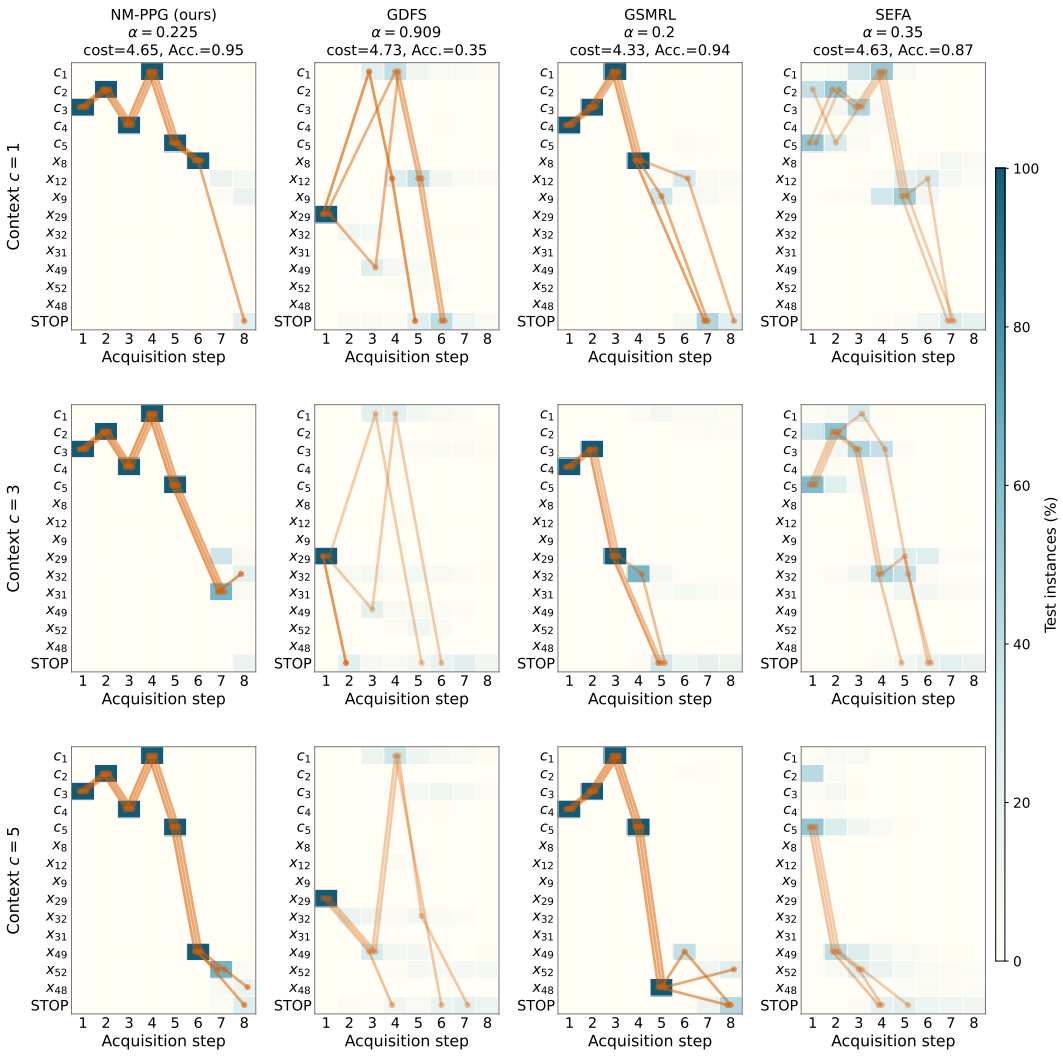

Figure 8: Acquisition paths on Cube-NM2 for three context values. Policies are selected at comparable average costs; panel titles report the corresponding α, test cost, and accuracy.

F Myopic vs. Non-Myopic Policies in AFA

Definition 1 formalizes myopic and non-myopic acquisition through the planning depth h. Specializing (4) to $h = 1$, the state-action value is

$$Q_1^*(x(m_t), a) = \begin{cases} V_0^*(x(m_t)), & a = d + 1, \\ \alpha c(a) + \mathbb{E}_{\mathbf{x}_a | x(m_t)}[V_0^*(x(m_{t+1}))], & a \in \mathcal{F}_B(m_t; k). \end{cases} \tag{77}$$

The myopic policy selects an action in $\arg\min_{a \in \mathcal{A}_B(m_t;k)} Q_1^*(x(m_t), a)$. It therefore asks which is cheaper in expectation: stopping now, or acquiring one feature and then predicting. Applying this rule repeatedly may still acquire many features; *myopic* refers to the lookahead used for each decision, not the total number of acquisitions. A non-myopic policy uses $h > 1$, allowing a current feature to be valued through the later acquisitions and stopping decisions it enables. The maximally non-myopic case $h = k$ accounts for every remaining decision permitted by the deployment horizon.

The benefit of non-myopic planning is dataset dependent. If the predictive utility of the remaining features is largely additive, the best next acquisition depends weakly on future observations, or feature costs are nearly uniform, repeated myopic selection can be close to or exactly optimal. This helps explain why strong myopic baselines perform well on many AFA datasets, whereas datasets

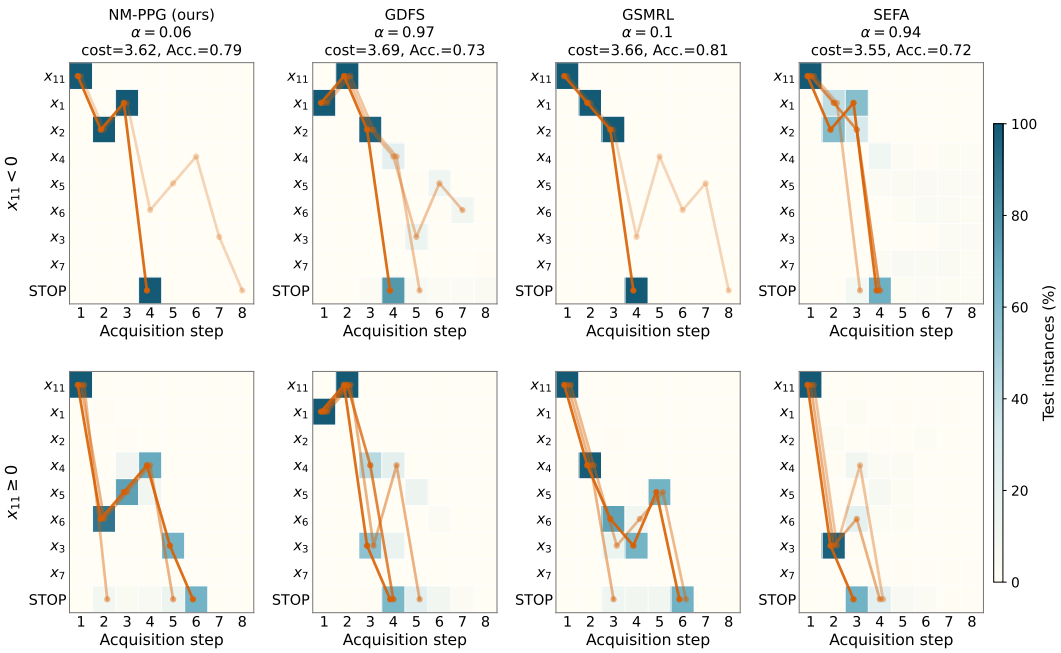

Figure 9: Acquisition paths on Syn1, conditioned on the sign of the context feature x_{11}. Policies are selected at comparable average costs.

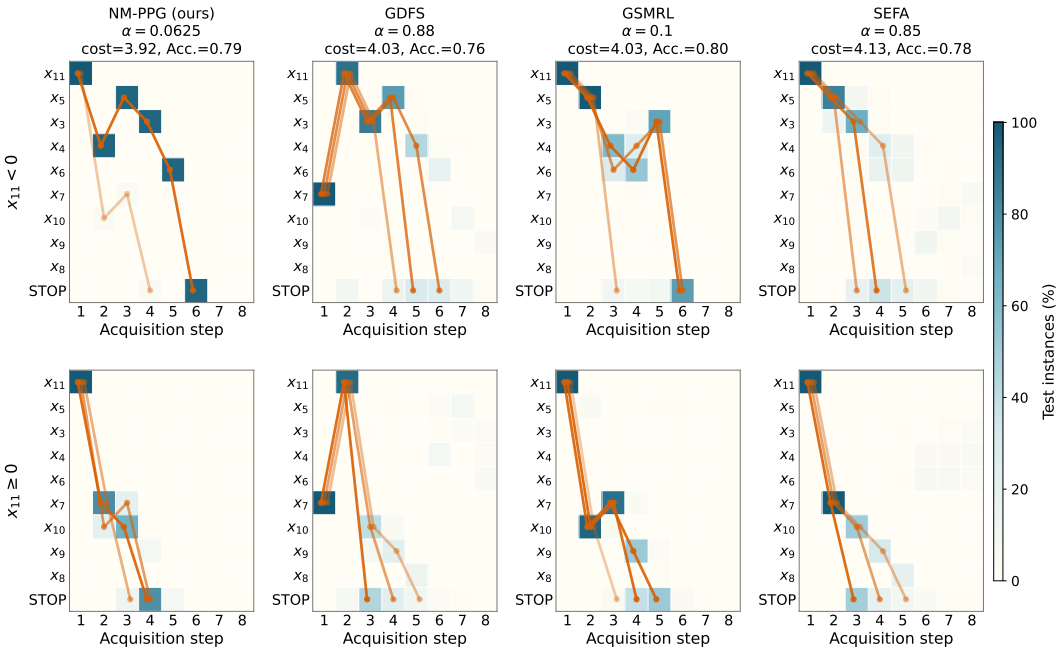

Figure 10: Acquisition paths on Syn3, conditioned on the sign of the context feature x_{11}. Policies are selected at comparable average costs.

constructed around delayed feature interactions favor non-myopic methods (Aronsson et al., 2026; Schütz et al., 2026). Non-myopic planning matters when an acquisition cannot be valued from its immediate one-step effect alone.

Three structures are particularly important. *(i) Joint information.* Features may be weakly informative individually but strongly informative together. The first feature can then appear unhelpful

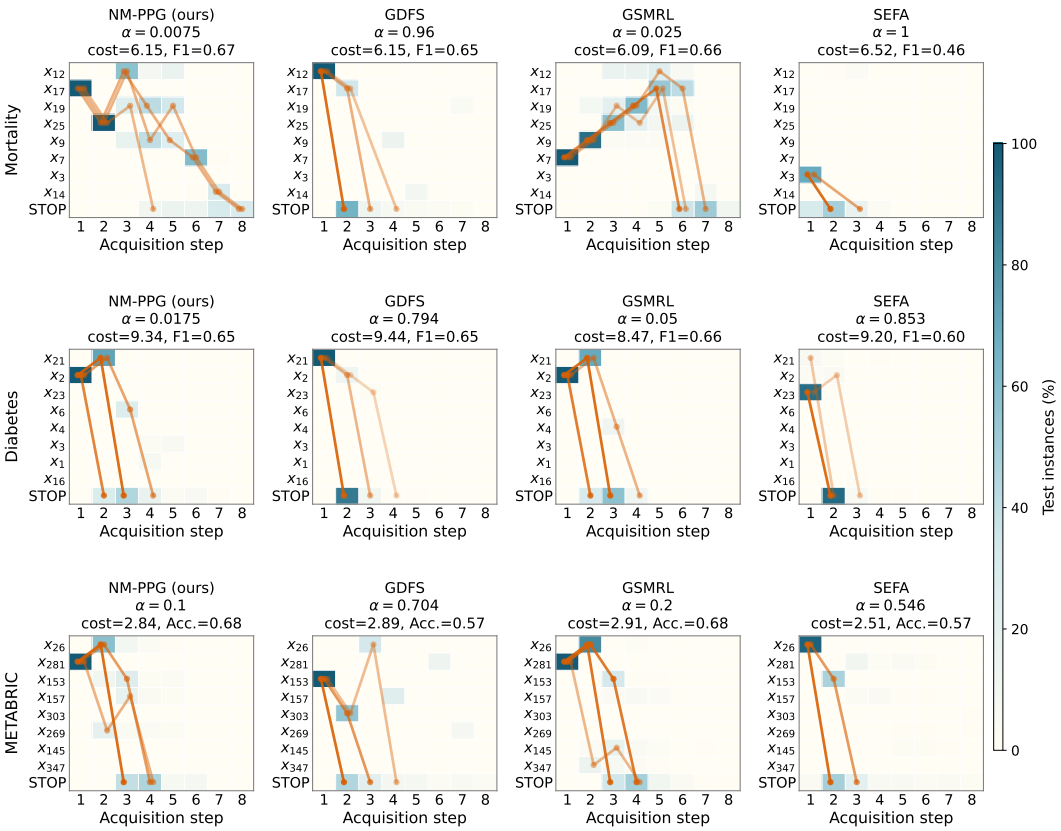

Figure 11: Acquisition paths on Mortality, Diabetes, and METABRIC. Rows denote datasets, and panel titles report the selected soft-budget operating point and its test performance.

Table 5: Training runtime in seconds for each dataset and method under both objective settings.

Dataset	NM-PPG (ours)	GDFS	DiFA	DIME	GSMRL	OL	AACO	SEFA
Soft-Budget AFA								
Cube-NM ($n_c = 5, \sigma = 0.1$)	4635	4187	3399	1795	1853	1506	5	1049
Cube-NM ($n_c = 5, \sigma = 0.2$)	4721	3805	3323	866	3203	2004	4	1350
Syn1	745	631	773	212	801	685	5	89
Syn3	821	684	812	238	658	663	5	100
Connect4	12327	11853	14532	3786	8648	4956	5	5511
Splice	652	602	700	161	385	562	5	202
EngineFaultDB	6523	10083	6232	3229	4102	3828	5	552
Metabric	522	458	492	306	672	725	5	387
Mortality	3218	2265	2865	1580	1565	1650	5	284
Diabetes	5214	5501	6671	1852	2646	4132	5	691
MNIST	11289	17886	18065	5822	16347	5031	5	16713
Fashion-MNIST	9259	18071	19772	8903	10920	5282	5	26766
Hard-Budget AFA								
Cube-NM ($n_c = 5, \sigma = 0.1$)	2927	4187	3399	1795	1771	708	5	1049
Cube-NM ($n_c = 5, \sigma = 0.2$)	2578	3805	3323	866	2159	733	4	1350
Syn1	721	631	773	212	499	509	5	89
Syn3	529	684	812	238	592	500	5	100
Connect4	7283	11853	14532	3786	5635	2793	5	5511
Splice	459	602	700	161	292	241	5	202
EngineFaultDB	5928	10083	6232	3229	4396	1647	5	552
Metabric	424	458	492	306	317	166	5	387
Mortality	2528	2265	2865	1580	2534	1719	5	284
Diabetes	3261	5501	6671	1852	2066	2000	5	691
MNIST	11236	17886	18065	5822	7759	3427	5	16713
Fashion-MNIST	9024	18071	19772	8903	7403	3600	5	26766

Table 6: Evaluation runtime in seconds for each dataset and method under both objective settings.

Dataset	NM-PPG (ours)	GDFS	DiFA	DIME	GSMRL	OL	AACO	SEFA
Soft-Budget AFA								
Cube-NM ($n_c = 5, \sigma = 0.1$)	0.3	0.3	0.4	0.4	0.2	0.3	32.7	3.1
Cube-NM ($n_c = 5, \sigma = 0.2$)	0.3	0.3	0.3	0.4	0.3	0.2	55.9	4.7
Syn1	0.3	0.2	0.3	0.3	0.3	0.5	13.7	1.0
Syn3	0.2	0.6	0.3	0.3	0.2	0.2	14.1	0.7
Connect4	0.5	3.5	1.0	0.8	0.8	0.4	129.9	9.8
Splice	0.2	0.3	0.4	0.3	0.2	0.2	7.9	3.7
EngineFaultDB	0.4	0.3	0.6	0.5	0.4	0.3	27.9	1.4
Metabric	0.2	0.2	0.3	0.4	0.2	0.2	32.2	4.6
Mortality	0.2	0.3	0.4	0.7	0.2	0.5	25.9	1.4
Diabetes	0.5	0.5	0.6	0.5	0.4	0.4	31.6	2.8
MNIST	0.7	2.4	4.1	0.7	0.8	0.7	485.4	144.8
Fashion-MNIST	0.7	0.6	0.7	1.4	0.8	0.5	458.4	18.4
Hard-Budget AFA								
Cube-NM ($n_c = 5, \sigma = 0.1$)	0.3	0.7	0.2	0.2	0.3	0.2	31.5	7.5
Cube-NM ($n_c = 5, \sigma = 0.2$)	1.8	0.8	0.2	0.3	0.3	0.2	35.1	8.4
Syn1	0.2	0.2	0.4	0.2	0.3	0.2	13.4	0.9
Syn3	0.2	0.2	0.2	0.4	0.3	0.3	14.2	0.6
Connect4	0.5	0.6	0.7	1.0	0.5	0.5	151.2	13.2
Splice	0.2	0.2	0.2	0.2	0.2	0.1	9.9	0.7
EngineFaultDB	0.5	0.5	0.5	0.6	0.5	0.3	54.2	2.0
Metabric	0.2	0.1	0.2	0.2	0.3	0.1	25.2	3.4
Mortality	0.2	0.4	0.2	0.2	0.2	0.2	37.2	2.8
Diabetes	0.5	0.4	3.1	0.4	0.3	0.3	54.1	3.4
MNIST	0.5	0.4	0.4	0.8	0.6	0.4	765.7	188.0
Fashion-MNIST	0.4	0.4	0.4	0.6	0.6	0.4	809.3	15.4

under (77), despite enabling a valuable later acquisition. *(ii) Context-dependent routing.* A feature may carry little direct label information but reveal which other feature is informative for the current instance. AFA must then be both adaptive and non-myopic: the first acquisition routes different instances toward different follow-up features. *(iii) Non-uniform costs.* A cheap feature may reveal whether a costly measurement is needed at all. A myopic policy can favor an immediately informative but consistently expensive feature, whereas a non-myopic policy can exploit a cheaper multistep acquisition path.

Thus, no universal advantage of non-myopic AFA should be expected. When delayed interactions, context-dependent routing, and heterogeneous costs are weak, myopic policies can be highly competitive. When these effects are strong, non-myopic planning can be essential. The required planning depth is determined by the length of the relevant interaction rather than the total number of features, as illustrated by Example 1.

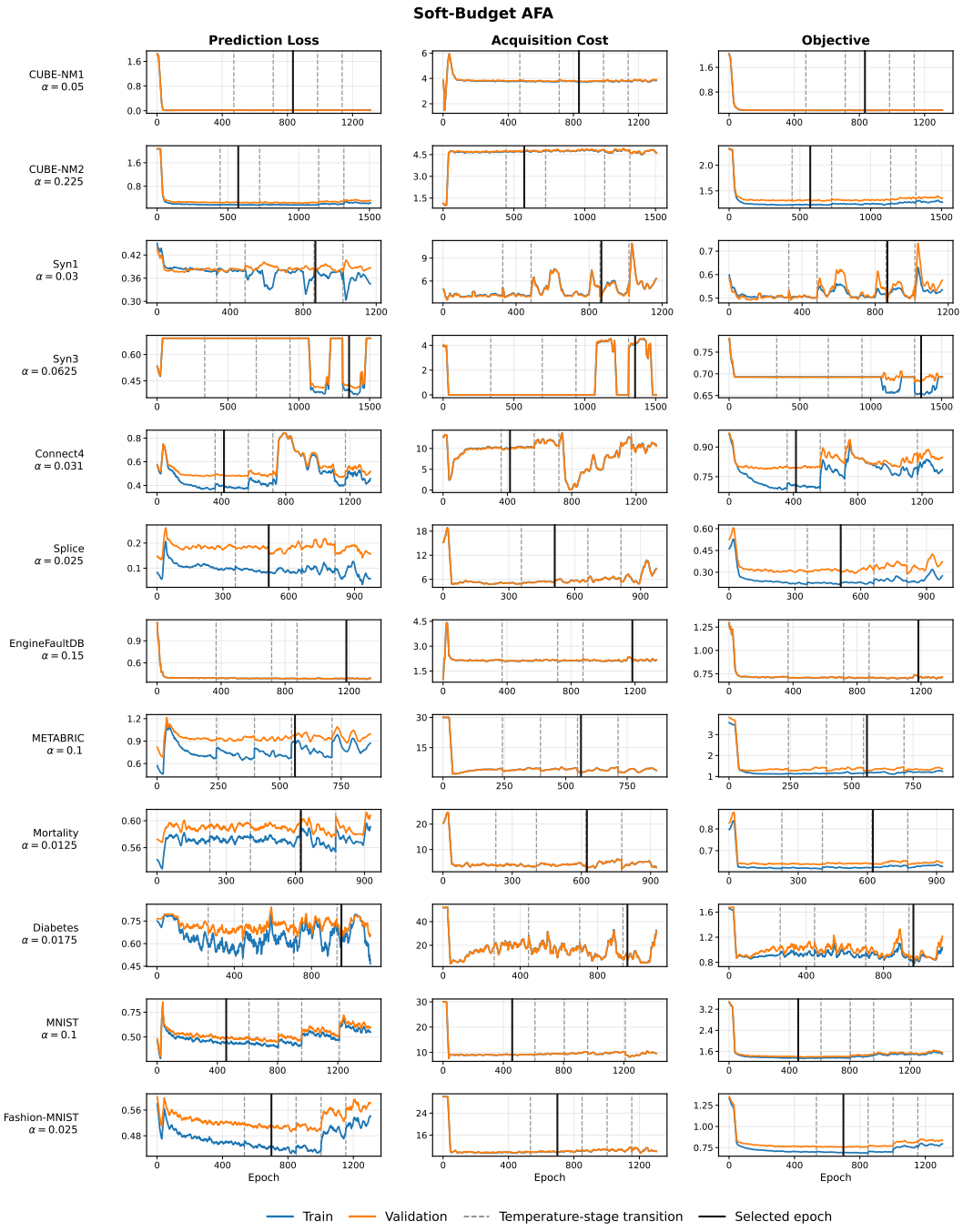

Figure 12: NM-PPG training dynamics under the soft-budget objective. Each row shows one representative operating point. Curves are 15-epoch moving averages computed separately within each temperature stage.

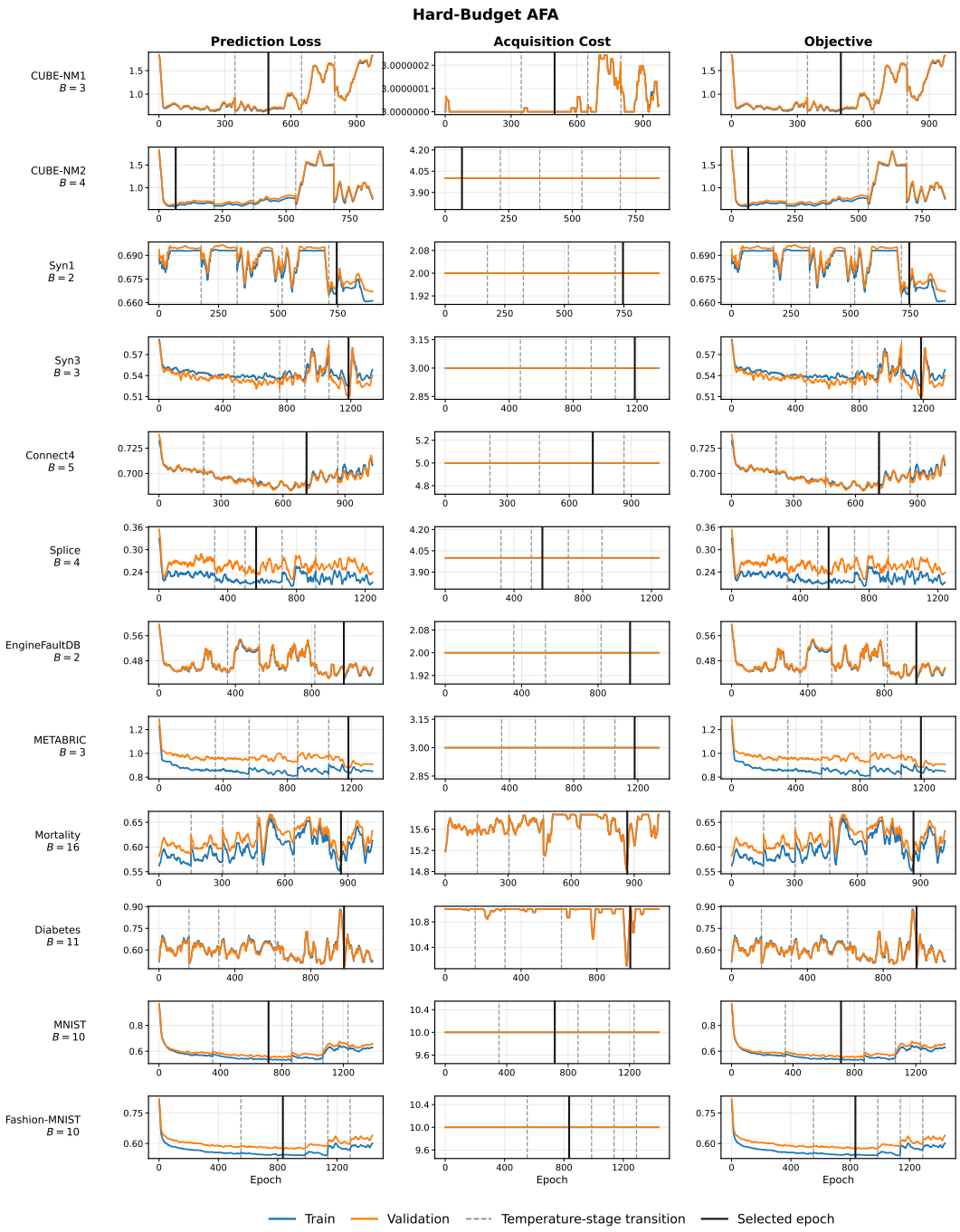

Figure 13: NM-PPG training dynamics under the hard-budget objective. Each row shows one representative operating point. Curves are 15-epoch moving averages computed separately within each temperature stage.