Contrastive Nonmyopic Objective Cost-Tradeoff Acquisition for Longitudinal Data

Dzung Dinh*
Dept. of Computer Science
UNC Chapel Hill
ddinh@cs.unc.edu

Boqi Chen*
Dept. of Computer Science
UNC Chapel Hill
bqchen@cs.unc.edu

Yunni Qu
Dept. of Computer Science
UNC Chapel Hill
quyunni@cs.unc.edu

Marc Niethammer
Dept. of Computer Science and Engineering
UC San Diego
mnieth ammer@ucsd.edu

Junier Oliva
Dept. of Computer Science
UNC Chapel Hill
joliva@cs.unc.edu

Abstract

In many critical domains, features are not freely available at inference time: each measurement may come with a cost of time, money, and risk. Longitudinal prediction further complicates this setting because both features and labels evolve over time, and measurements missed at earlier timepoints may become permanently unavailable. We propose CAPE (**C**ontrastive **A**cquisition-**P**lan **E**mbeddings), a partitioning acquisition policy for longitudinal active feature acquisition. CAPE first defines NOCT, an oracle-based objective that scores a set of future feature-time acquisitions by its expected predictive loss together with its acquisition cost. Based on NOCT, CAPE learns a contrastive acquisition-utility manifold of masked partial observations, where proximity reflects the expected utility of future acquisitions. Offline, training states are embedded and partitioned at each rollout timepoint; candidate future plans are scored within each partition using plug-in NOCT, and the best partition-level acquisition plans are cached. During inference, CAPE retrieves the cached plan via nearest-centroid lookup, enabling low-latency adaptive acquisitions. Experiments on synthetic and real-world healthcare datasets demonstrate that CAPE outperforms nonparametric, greedy, and RL-based alternatives, achieving higher accuracy at lower acquisition costs. Code is publicly available at https://github.com/dzungdinhh/CAPE.

1 Introduction

In many real-world scenarios, *the inference-time acquisition of information is a critical decision*. For example, in clinical longitudinal monitoring (a motivating application for this work), diagnostic decisions often rely on sequentially gathering information such as lab tests or imaging before predicting a patient's status. Each type of data acquisition comes at a cost of time, financial expense, or risk to the patient. In contrast, standard supervised machine learning pipelines typically assume that *all* features are freely available at inference.

In this work, we focus on *longitudinal* Active Feature Acquisition (LAFA) [26, 13], where, during inference, a model is responsible for (i) sequentially deciding *which features* to acquire and at *what time*, balancing their utility against acquisition costs, and (ii) *making predictions* with the partially observed features it has chosen to acquire. In this setting, leveraging the temporal context is crucial

*Equal contribution.

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

for both selecting valuable feature acquisitions and making accurate predictions. Fig. 1 illustrates an example of LAFA by an autonomous agent in a clinical scenario. At the current visit, the agent (1) reviews previously collected data alongside any measurement acquired at the current visit, (2) predicts the patient's current status $\hat{y}_t$, and (3) decides when to schedule the next visit and which tests to acquire then (accounting for measurements' acquisition costs). Practical constraints such as time, cost, and resource limitations necessitate a judicious selection. Between visits, the agent continues to forecast the disease progression utilizing the previous acquisitions.

This longitudinal AFA setting presents several challenges: deciding which features to acquire at each visit, facilitating early predictions for timely interventions, and accounting for temporal settings where missed acquisitions at earlier time points are permanently inaccessible. Although prior work has explored longitudinal AFA approaches, some methods [13] make a single prediction across all timepoints, neglecting dynamic predictions that are critical for early detection and treatment-plan adjustment in clinical settings [23]. Furthermore, recent reinforcement learning (RL)-based methods [23, 19] are difficult to optimize in practice, while greedy strategies [5] make myopic decisions based on short-term gain and may skip early measurements that later prove critical but cannot be reacquired. This motivates nonmyopic methods that are practical to train and deploy.

We propose `CAPE` (**C**ontrastive **A**cquisition-**P**lan **E**mbeddings), a contrastive partitioning policy for longitudinal active feature acquisition. The key idea is to use oracle-based valuation to learn an AFA-based acquisition-utility manifold, where proximity between partial states reflects similarity in the expected utility of future feature-time acquisitions. States within the same local region, therefore, share acquisition-utility profiles, allowing candidate acquisition subsets to be reused with similar accuracy–cost effects. `CAPE` exploits this structure by partitioning the learned space, scoring future acquisition plans within each partition, and caching partition-level plans for fast deployment. This extends oracle-based AFA to longitudinal acquisition planning and turns expensive nonmyopic search into an offline policy-fitting procedure.

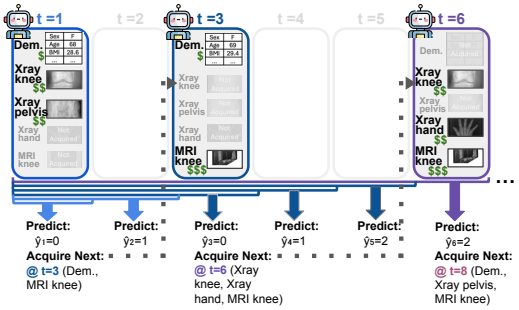


Figure 1: Illustration of clinical longitudinal AFA. At the current visit, an autonomous agent reviews previously collected and newly acquired data, predicts the patient's current status, and recommends a subset of examinations for a determined future time point. This process repeats until the agent recommends no further acquisition.

Contributions First, we use NOCT, an oracle-based plan objective that balances future predictive accuracy against feature-time acquisition cost, to supervise an acquisition-utility embedding space; to our knowledge, this is the first use of such a strategy in AFA. Second, we develop a novel partitioning acquisition policy that amortizes planning over regions of this embedding space, enabling `CAPE` to select and revise future acquisition plans at deployment without expensive online plan scoring. Third, across synthetic and real-world healthcare datasets, we show that `CAPE` outperforms a suite of nonparametric, greedy, and RL-based alternatives.

2 Related Work

Active Feature Acquisition Prior works in AFA [26, 27] study the trade-off between prediction and feature acquisition cost in a non-longitudinal context. Recent works [28, 30, 9] frame AFA as an RL problem; however, they are challenging to train in practice due to the complicated state and action space with dynamically evolving dimensions. Other works [14] use surrogate generative models, which model multidimensional distributions, or employ greedy policies that ignore joint informativeness (some features are only informative in combination) [3, 15, 7]. To overcome these issues, Ghosh and Lan [6] learn a differentiable acquisition policy by jointly optimizing selection and prediction models through end-to-end training.

In addition, Valancius et al. [29] propose a nonparametric, oracle-based AFA method that uses training instances to evaluate candidate acquisitions. Our method is closest to this oracle-based line, but differs in how this information is represented and reused. A direct extension of Valancius et al. [29] would retrieve a fresh neighborhood of similar training instances in the native feature

space and score many candidate plans online for each test state. This nearest-neighbor approach is computationally expensive, scaling linearly with the number of training instances, and may be more susceptible to the curse of dimensionality because neighborhoods are formed in the original high-dimensional feature space rather than in an acquisition-aware embedding. Instead, CAPE learns an acquisition-utility manifold in which local regions share similar future acquisition-utility profiles. It then partitions this space and fits a partitioning acquisition policy by scoring candidate future plans within each partition offline. At deployment, CAPE requires only a partition lookup before following the acquisition plan associated with the matched partition, enabling faster inference.

Longitudinal Active Feature Acquisition Longitudinal AFA extends the standard AFA to include the temporal dimension. In this setting, a policy must account for temporal constraints where past time points can no longer be accessed. As a result, a policy must decide which features to acquire and determine the optimal timing for each acquisition. For instance, Zois and Mitra [32] derive dynamic-programming and low-complexity myopic policies for finite-state Markov chains; ASAC [31] uses an actor-critic approach to jointly train a feature selection and a prediction network. Additionally, Nguyen et al. [19] use RL to optimize acquisition timing, but acquire all available features at the selected time point. Recently, Qin et al. [23] prioritize timely detection by allowing flexible follow-up intervals in continuous-time settings, ensuring that predictions are timely and accurate. However, RL-based approaches can be challenging to train due to issues such as sparse rewards, high-dimensional observation spaces, and complex temporal dependencies [14]. In contrast, CAPE uses the MDP formulation only as a formal lens for defining feasible future acquisition plans, while avoiding RL-based policy learning by learning an acquisition-utility manifold and fitting an offline partitioning acquisition policy that amortizes nonmyopic planning over local regions of the learned space.

3 Methodology

We propose CAPE, a method that learns a novel acquisition-utility manifold, an embedding space where proximity reflects the expected utility of future acquisitions. CAPE learns this manifold through two components: (1) a Nonmyopic Objective Cost-Tradeoff (NOCT) scores candidate future plans by their expected prediction loss and acquisition cost, producing acquisition-utility profiles that record which feature-time acquisitions consistently appear in the lowest-cost, highest-accuracy plans; (2) a contrastive objective supervised by these profiles pulls together instances with similar acquisition needs and pushes apart those with divergent ones. CAPE then utilizes the manifold through a partitioning acquisition policy strategy that exploits nearby regions that share similar acquisition needs. At deployment, a new instance is embedded and mapped to its nearest partition to retrieve the respective acquisition plan, resulting in simple and efficient inference.

3.1 Notation and Setup

We consider longitudinal AFA over a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ of N instances (e.g., patients). For each instance i, the covariates form a sequence of length L: $x_i = \{x_{i,t}\}_{t=1}^L$, where $t \in \{1, \dots, L\}$ indexes the time step. At each time step t, $x_{i,t} \in \mathbb{R}^M$ contains M (costly) feature measurements, but the policy may only acquire a subset at each visit [2]. We study a classification task with per-timepoint labels $y_{i,t} \in \{1, 2, \dots, C\}$, where C represents the number of classes. Following Qin et al. [23], Yoon et al. [31], each feature m has a time-invariant[3] cost c^m. Finally, we use colon notation for subsequences; for example, $x_{1:t} = (x_1, \dots, x_t)$ denotes measurements from timepoints 1 to t (inclusive).

At a timepoint t, let $o \subseteq \{(m, t') \mid m \in \{1, \dots, M\}, t' \leq t\}$ denote the *observed feature-time pairs*, and let $x_{i,o}$ be the corresponding partial observation. We assume a pre-trained conditioning classifier $\hat{y}$ that predicts the label at any target time k from partial information, denoted as $\hat{y}_k(x_{i,o})$.

3.2 Nonmyopic Objective/Cost Tradeoff (NOCT)

We now introduce CAPE through a longitudinal oracle-based acquisition utility that provides a provable (see below) proxy for the MDP value of partially observed states. While prior oracle-

[2] CAPE also supports multi-modal acquisitions where each modality yields a vector; this can be modeled by replacing scalar features with modality-specific vectors.

[3] CAPE easily extends to time-varying costs by replacing c^m with $c^{m,t}$ (cost of modality m at time t).

based AFA objectives [29] score candidates in the standard AFA setting, the introduced objective NOCT evaluates candidate feature-time plans in the longitudinal setting. Let $\mathcal{V}_{>t} = \{(m, t') \mid m \in \{1, \ldots, M\},\ t' \in \{t+1, \ldots, L\}\}$ represent the set of candidate feature-time pairs that are still available to acquire after time t. At a partially observed state (x_o, o, t), we have two options: (1) *choose an acquisition plan* $v \subseteq \mathcal{V}_{>t}$ (which specifies what to acquire in the future), or (2) *terminate acquisition* (i.e., $v = \varnothing$) and make predictions for the remaining timepoints using only the currently observed information x_o. The **n**onmyopic **o**bjective **c**ost-**t**radeoff (NOCT) determines the tradeoff of an acquisition plan of additional features $v \subseteq \mathcal{V}_{>t}$ for a current partially known instance of x_o, where o contains features from the past, up to time $\leq t$:

$$\mathrm{NOCT}\big(x_o, o, v\big) \equiv \mathbb{E}_{y_{t+1:L}, x_v | x_o}[\ell(x_o \cup x_v, y, t)] + \alpha \sum_{(m,t') \in v} c^m, \tag{1}$$

where $\ell(x_o \cup x_v, y, t) \equiv \sum_{k=t+1}^{L} \mathrm{CE}\big(\hat{y}_k(x_o \cup x_{v_{t+1:k}}),\ y_k\big)$ is the accumulated cross-entropy loss from $t+1$ to L when acquiring the future feature-time pairs specified by v. Here, $x_{v_{t+1:k}}$ denotes the subset of acquired features in v whose time indices lie in $\{t+1, \ldots, k\}$, i.e., the features that would be available by time k under plan v, given the currently observed information x_o. Using training trajectories $(x_i, y_i) \in \mathcal{D}$, the expectation in Eq. (1) can be estimated by evaluating candidate plans on observed future outcomes during training. Intuitively, Eq. (1) looks ahead at how acquiring a new set of feature-time pairs in $\mathcal{V}_{>t}$ is anticipated to reduce the classification loss from $t+1$ to L, while penalizing the acquisition cost. For simplicity, we refer to feature-time pairs as "features".

Connection to the optimal LAFA value Although `CAPE` does not train an RL policy or use Bellman recursion, we relate NOCT to the optimal value of the sequential LAFA MDP. Let $\mathbf{s} = (x_o, o, t)$ denote a partially observed state, and let $V^\star(\mathbf{s})$ be the optimal value of the corresponding finite-horizon LAFA Markov Decision Process (MDP), where rewards are negative prediction loss and acquisition cost. The full MDP formulation and proof are given in Appx. A.6.

Theorem 3.1 (NOCT lower bound)**.** *Let $\mathbf{s} = (x_o, o, t)$ be a partially observed state in the finite-horizon LAFA MDP with fixed predictor $\hat{y}$ and nonnegative acquisition costs.*

$$\textit{If } J_{\mathrm{NOCT}}(\mathbf{s}) = -\min_{v \subseteq \mathcal{V}_{>t}} \mathrm{NOCT}(x_o, o, v),\ \textit{then } J_{\mathrm{NOCT}}(\mathbf{s}) \leq V^\star(\mathbf{s}). \tag{2}$$

Thus, $-\mathrm{NOCT}$ lower bounds the optimal longitudinal AFA value and provides a principled surrogate for evaluating future acquisition plans. This result motivates NOCT as an objective for comparing future acquisition plans, but does not provide approximation or regret guarantees for the practical `CAPE` policy. In practice, `CAPE` does not solve Eq. (2) exactly at deployment. Instead, it estimates NOCT from training trajectories to learn an acquisition-utility manifold and fits a partitioning acquisition policy. The following subsections describe how this approximation yields an efficient deployment procedure by amortizing nonmyopic planning over partitioned regions of the learned space.

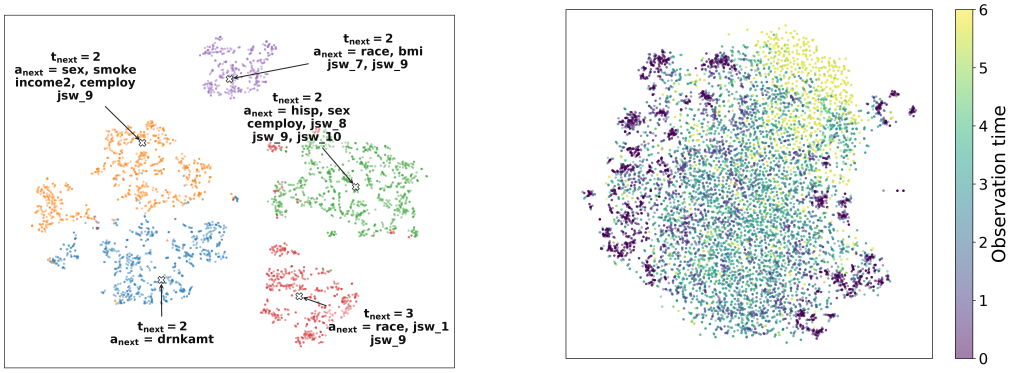
3.3 Acquisition-Aware Contrastive Representations

We learn an embedding g_ϕ that maps partial states according to their NOCT-guided future acquisition-utility profiles, using a contrastive objective. Intuitively, g_ϕ is trained so that nearby states benefit similarly from future feature acquisitions. This structure underlies the efficacy of the partitioning policy in Sec. 3.4, which tailors acquisition strategies to concentrated local regions of the learned manifold that, by construction, contain partial states whose candidate acquisition subsets induce similar accuracy–cost tradeoffs. Thus, the embedding defines an AFA-based acquisition-utility manifold, rather than a generic similarity space.

Plug-in NOCT For *training instances*, $(x_i, y_i) \in \mathcal{D}$, the direct estimate of $\mathrm{NOCT}\big(x_{i,o}, o, v\big)$ (Eq. (1)) is a look-ahead plug-in sample approach, which we use to learn our manifold:

$$\widehat{\mathrm{NOCT}}_{pl}\big(x_{i,o}, o, v\big) = \ell(x_{i,o} \cup x_{i,v}, y_i, t) + \alpha \sum_{(m,t') \in v} c^m. \tag{3}$$

Acquisition-Utility Profiles For each partial state $(x_{i,o}, o, t)$, we want a summary of which future measurements are expected to be useful. Here, our key idea is that if certain feature-time acquisitions repeatedly appear in low-NOCT future plans, then those acquisitions characterize the utility profile of the current partial state. We represent this profile by a distribution $\zeta_{i,o}$ over future acquisitions $(m, t') \in \mathcal{V}_{>t}$ and termination $\varnothing$.



(a) Acquisition regions at time $t = 1$.

(b) t-SNE colored by current timepoint.

Figure 2: Visualizations of the acquisition-aware embedding on the KLG task. **(a)** A t-SNE visualization of KLG patient embeddings at rollout time $t = 1$. Points denote training states, crosses mark region centers, and annotations show the region-specific next acquisition, where t_{next} denotes the next visit and a_{next} denotes the selected measurements. Each color corresponds to a local region of the learned space that shares a cached acquisition decision. **(b)** A t-SNE visualization of rollout-state embeddings colored by current observation timepoint. The learned representation exhibits temporal structure, suggesting that g_ϕ captures differences in acquisition context across rollout stages.

To construct $\zeta_{i,o}$ for a training instance, we rank candidate future plans $v \subseteq \mathcal{V}_{>t}$ using the plug-in objective $\widehat{\text{NOCT}}_{\text{pl}}(x_{i,o}, o, v)$ and take the top-r plans $\hat{v}_{i,o}^{(1)}, \ldots, \hat{v}_{i,o}^{(r)}$ with the smallest values. The profile $\zeta_{i,o}(m, t')$ is then given by a weighted frequency of how often (m, t') appears in these top plans, with weights based on their plug-in NOCT scores. Thus, $\zeta_{i,o}$ records which future measurements are consistently suggested by the best plans, and we use similarities between these profiles to supervise the contrastive embedding. We provide the formal $\zeta_{i,o}$ construction in Appx. A.5.

Contrastive Learning We learn a manifold that maps partially observed features into an embedding space where distance reflects similarity of acquisition-utility profiles. Formally, we learn an embedding network $g_\phi(x_{i,o}, t) \in \mathbb{R}^E$ such that two partial states $(x_{i,o}, o, t)$ and $(x_{j,o}, o, t)$ are mapped nearby when they share similar top future acquisitions, meaning that the same acquisition decisions are expected to produce similar accuracy gains under comparable cost budgets. Conversely, instances with divergent acquisition-utility profiles are pushed away in $\mathbb{R}^E$. Fig. 2a illustrates this structure, where the local regions in the learned space correspond to different reusable next-acquisition decisions.

To achieve this, we define a similarity score between instances based on their future-candidate distributions $\zeta_{i,o}$ and $\zeta_{j,o}$, using the exponential of their negative *Jensen-Shannon (*JS*) divergence*:

$$\texttt{sim}_{ij} = \exp\big(-\beta \, \text{JS}(\zeta_{i,o}, \zeta_{j,o})\big), \tag{4}$$

$\beta > 0$ controls how strongly divergence affects similarity. Letting $\delta_{ij} = \big\|g_\phi\big(x_{i,o}\big) - g_\phi\big(x_{j,o}\big)\big\|^2$ be the embedding distance, we minimize a contrastive loss [2]:

$$\mathcal{L}_{\text{emb}} = \frac{1}{2}\Big(\texttt{sim}_{ij} \cdot \delta_{ij} + (1 - \texttt{sim}_{ij}) \cdot \max\big(0, \, \gamma - \delta_{ij}\big)^2\Big), \tag{5}$$

where $\gamma > 0$ is the margin parameter. Minimizing $\mathcal{L}_{\text{emb}}$ encourages g_ϕ to create an embedding space where distance reflects the similarity of future-candidate distributions.

Fig. 2b shows that rollout states exhibit clear temporal structure in the embedding space. This motivates using the embedding not merely for retrieval, but to define local policy regions within the learned acquisition-utility manifold, where a temporal partition-level acquisition plan can be reused across states with similar expected utility for future acquisitions.

3.4 Partitioning Acquisition Policy Construction

In the acquisition-utility manifold, nearby states are expected to exhibit similar useful future-acquisition behavior. We use this structure to make nonmyopic planning efficient at inference

time: we partition concentrated regions of the learned embedding space at each time step t, fit a representative acquisition plan within each partition, and reuse that plan for nearby states.

Embedding-based Partitioning At time t, let $x_{i,o}$ denote the partially observed training instance for sample i. Using the embedding network g_ϕ, we compute acquisition-aware embeddings $e^t_{i,o} = g_\phi(x_{i,o}, t)$. We then cluster, e.g., using K-Means, $\{e^t_{i,o}\}_{i=1}^n$ *to partition the learned acquisition-utility manifold into K concentrated regions* $C_{1,t}, \ldots, C_{K,t}$ with centroids $\mu_{1,t}, \ldots, \mu_{K,t}$. Because g_ϕ is trained so that distances reflect similarity in acquisition-utility profiles, each region serves as a policy partition whose members are expected to share similar future acquisition behavior.

Partitioning Acquisition Policy For each partition $C_{k,t}$, we sample a partition-level candidate set $\mathcal{P}_{k,t} \subseteq 2^{\mathcal{V}_{>t}}$, including future acquisition plans and termination plan $\varnothing$. For each candidate plan $v \in \mathcal{P}_{k,t}$, we compute its average plug-in NOCT score over the partition:

$$\widehat{\mathrm{NOCT}}_{k,t}(v) = \frac{1}{|C_{k,t}|} \sum_{i \in C_{k,t}} \widehat{\mathrm{NOCT}}_{\mathrm{pl}}\big(x_{i,o}, o, v\big). \tag{6}$$

Intuitively, lower values of $\widehat{\mathrm{NOCT}}_{k,t}(v)$ indicate better accuracy-cost tradeoffs for the partition. We therefore cache the top acquisition plan for partition k as:

$$\hat{v}^*_{k,t} = \arg\min_{v \in \mathcal{P}_{k,t}} \widehat{\mathrm{NOCT}}_{k,t}(v) \tag{7}$$

where $\hat{v}^*_{k,t}$ is the candidate plan with the smallest partition-level plug-in NOCT scores. The cache stores the embeddings $\{e^t_i\}_{i=1}^n$, centroids $\{\mu_{k,t}\}_{k=1}^K$, cached plans $\{\hat{v}^*_{k,t}\}_{k=1}^K$, and their scores.

Policy-cache construction proceeds as a rollout over the training set. After the partition plans are cached at time t, each training instance is assigned to its respective policy partition, the corresponding cached plan is selected, and only the next immediate (temporally) acquisitions from that plan are executed (e.g., see Fig. 2a). Each instance's observed set o is then updated, and the instance advances to the corresponding next timepoint. Policy fitting then proceeds to time $t+1$.

3.5 Policy Deployment

At partitioning policy deployment, `CAPE` uses the learned acquisition-utility manifold to identify which policy partition region best matches the current partial observation (Alg. 1). Given a test state (x_o, o, t), we compute the associated embedding $e^t_o = g_\phi(x_o, t)$ and map it to the nearest cached centroid $\mu_{k,t}$ with $k = \arg\min_{k \in \{1,\ldots,K\}} \|e^t_o - \mu_{k,t}\|_2^2$. Intuitively, the cached plan $\hat{v}^*_{k,t}$ is the plan that achieved the best average NOCT tradeoff for similar training states at the same local acquisition-utility region.

The cached plan $\hat{v}^*_{k,t}$ is a future acquisition schedule with potential rather than a commitment. This keeps the policy adaptive: each new observation can change the utility of remaining acquisitions, so rather than following the full plan rigidly, `CAPE` reassesses at each step. If $\hat{v}^*_{k,t} = \varnothing$, then the cached region suggests that further acquisition is not worth its cost and `CAPE` terminates, predicting remaining timepoints from current information. Otherwise, `CAPE` executes only the earliest acquisitions in $\hat{v}^*_{k,t}$, which acquire features at the next earliest occurring timepoint t_{next} in $\hat{v}^*_{k,t}$. Rather than following the rest of the previously selected plan, `CAPE` updates o, advances to the next decision timepoint, t_{next}, and repeats the cached lookup. This mirrors the clinical intuition of ordering the next informative test, observing the result, and then reassessing.

Algorithm 1 `CAPE` Deployment

Require: Predictor $\hat{y}_k$, embedding g_ϕ, cached embeddings $\{e^t_i\}$, cached centroids $\{\mu_{k,t}\}$, cached plans $\{\hat{v}^*_{k,t}\}$.

Initialize: `predictions` $\leftarrow$ [ ]; `terminate` $\leftarrow$ `false`; $t \leftarrow 0$; $o \leftarrow \varnothing$
while not `terminate` **and** $t < L$ **do**
  $e^t_o \leftarrow g_\phi(x_o, t)$
  Find nearest cached centroid $\mu_{k,t}$ to e^t_o
  Select cached plan $\hat{v}^*_{k,t}$ for centroid $\mu_{k,t}$
  if $\hat{v}^* = \varnothing$ **then**
    $t_{\text{next}} \leftarrow L$; `terminate` $\leftarrow$ `true`
  else
    $t_{\text{next}} \leftarrow \min\{\, t' \mid (m, t') \in \hat{v}^*_{k,t} \,\}$
    $o \leftarrow o \cup \{(m, t_{\text{next}}) \mid (m, t_{\text{next}}) \in \hat{v}^*_{k,t}\}$
  end if
  for $t' = t+1, \ldots, t_{\text{next}}$ **do**
    append $\hat{y}_{t'}(x_{o_{1:t'}})$ to `predictions`
  end for
  $t \leftarrow t_{\text{next}}$
end while
return `predictions`

4 Experiments

4.1 Datasets

Synthetic Inspired by Kossen et al. [13], we construct a synthetic dataset of $N = 8{,}000$ samples with $L = 10$ non-i.i.d. timepoints. Each timepoint contains two features: *digit* and *counter*. Labels are the cumulative sum of the digits when the counter values are zero. Specifically, the *digit* feature is a sequence of uniformly random L integers from $\{0, 1, 2\}$, e.g., 1010220110. The *counter* feature is created by concatenating countdown sequences that begin with randomly chosen starting values from $\{0, 1, 2\}$ and truncating to a fixed length of L. For example, starting values 2, 1, 2, and 1 yield 210, 10, 210, 10, and the final counter feature is their concatenation 2101021010 (of length L). Unlike Kossen et al. [13], we assign labels at every timestep: $y_t = \sum_{t'=1}^{t} \text{digit}_{t'} \mathbb{I}\{\text{counter}_{t'} = 0\}$; for example, label 0011333444 in the case of $\begin{smallmatrix}\text{counter:}\ 2\underline{1}0\underline{1}0\underline{2}1\underline{0}1\underline{0} \\ \text{digit:}\ \ 10\mathbf{1}0\mathbf{2}20\mathbf{1}1\mathbf{0}\end{smallmatrix}$. The policy should begin with acquiring the digit and counter at $t = 1$, then acquiring the digit at the next anticipated timestep where $\text{counter} = 0$, then acquiring the *counter* again to locate the next $\text{counter} = 0$ timestep, and repeat. Following Kossen et al. [13], we split data into train/val/test ($70\%/15\%/15\%$) .

OAI The Osteoarthritis Initiative (OAI; `https://nda.nih.gov/oai/`) contains $N = 4{,}796$ patients with left and right knee records followed for up to 96 months. We use 27 visit-level features, including tabular features from Chen et al. [1] and joint space width, across $L = 7$ visits. We predict KLG ($0 \sim 4$) [10] and WOMAC pain ($0 \sim 20$) [16]; following prior work [1, 19], we merge KLG $0/1$, binarize WOMAC pain at 5, and use a $50\%/12.5\%/37.5\%$ train/val/test split.

ILIADD The Intensive Longitudinal Investigation of Alternative Diagnostic Dimensions (ILIADD) dataset [25] is a remote mental-health study with 544 participants, 34 demographic/HiTOP-SR-derived features, and up to eight EMA surveys per day. We use length-$L = 10$ windows to predict negative affect, yielding $N = 40{,}279$ train/val/test windows ($69.5\%/15.25\%/15.25\%$).

Supplemental Materials Please see Appx. B for additional results and datasets.

4.2 Baselines

We expand the comparisons in recent work of Qin et al. [23] in longitudinal AFA by including *three additional nonuniform-feature-cost-capable state-of-the-art AFA methods*, DIME [5], DiFA [6], ACO [29], together with the RL method ASAC [31] from Qin et al. [23]. We consider 6 baselines in total: 1) **ASAC** [31]: an actor-critic method to jointly train a feature selector and a predictor network; 2) **RAS** [23]: an RL method for continuous-time longitudinal acquisition; 3) **AS** [23]: RAS with a constant acquisition interval; 4) **DIME** [5]: sequential selection by marginal utility; 5) **DiFA** [6]: a Gumbel-softmax-based differentiable policy; 6) **ACO** [29]: an oracle-based acquisition policy. For each baseline, *we perform a comparable hyperparameter search on the validation set using the authors' original code*; full details are in Appx. C.2.

We adapt DIME and DiFA for longitudinal data by restricting acquisitions to either the current or future time points. Note that DIME and DiFA acquire features one at a time within a timestep, while `CAPE` and other baselines select all desired features at t simultaneously before advancing. This gives DIME and DiFA additional within-timepoint adaptivity: after observing each acquired feature value, they can decide which feature to acquire next at the current timepoint or whether to advance to a later timepoint.

4.3 Implementation

Data Availability and Missingness Unlike standard AFA methods that assume fully observed data [24, 28, 18], our work is general and handles missingness. This is done by excluding missing features from candidate sets and pretraining the prediction network $\hat{y}_k$ and embedding network g_ϕ with random input-feature dropout.

Prediction Network Implementation At each time t, an MLP predicts outcomes $\hat{y}_k(x_{o_{1:t}})$ for $k \geq t$ from the observed features $x_{o_{1:t}}$[4] and the target time indicator k, using only observations available through t. We train the network with random input dropout to improve robustness to varying missingness patterns.

[4] Inherently missing and unacquired features are masked out.

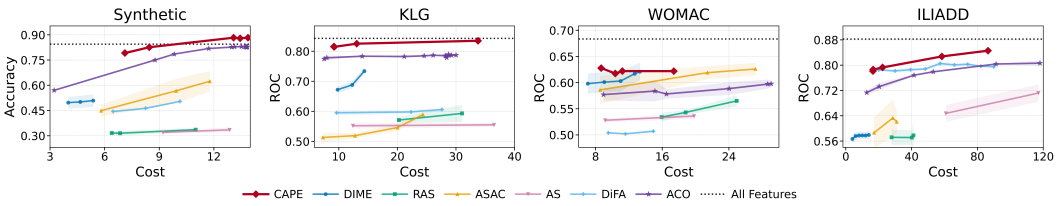


Figure 3: Performance/cost of models across various average acquisition costs (budgets), with standard error across 5 runs. Following Kossen et al. [13], we report accuracy instead of AUROC for synthetic data. The dotted horizontal line shows performance using all features. CAPE consistently yields higher predictive performance at comparable or lower acquisition costs.

CAPE Implementation. We train the MLP predictor $\hat{y}_k$ and then train the MLP embedding network g_ϕ. We set $\beta = \gamma = 1$, $K \in \{5, 10, 20, 50, 100\}$, and embedding dimension $E = 32$ (Synthetic) and $E = 64$ (OAI/ILIADD). We utilize Adam [12] with a learning rate of 10^{-3}. During the partitioning acquisition policy construction, we sample $|\mathcal{P}| = 5{,}000$ acquisition plans.

4.4 Performance-Cost Results

Fig. 3 shows performance against acquisition cost. For each panel, the x-axis represents different average acquisition costs per instance, and the y-axis represents the predictive performance. Higher budgets allow more acquisitions and typically improve performance. Performance-cost points are obtained by sweeping the cost-control parameter α (in Eq. (1)) for CAPE and the corresponding cost-control hyperparameters for each baseline. See Appx. C for details on optimal model selection for target budgets and dataset-specific feature costs. Following [13, 23], we report the synthetic dataset accuracy and real-world healthcare datasets AUROC.

Results CAPE achieves the strongest overall performance–cost tradeoff across all datasets, with the best or near-best accuracy/AUROC across the available acquisition budgets. CAPE remains above competing acquisition policies over most budgets on Synthetic, WOMAC, KLG, and ILIADD. These results suggest that the learned acquisition-utility manifold provides reusable acquisition decisions that generalize across similar partial states. DIME and ACO perform well in some regimes but are less consistent across datasets and budgets. RAS, ASAC, AS, and DiFA often improve as acquisition cost increases, but generally remain below CAPE and sometimes exhibit notably weaker performance in the low-cost regime. This regime is especially important, as a useful policy should identify high-value measurements before the budget becomes large enough to approximate the all-features setting. Additional results on Average Precision are in Appx. B.4.

Inference Time CAPE provides a favorable tradeoff between predictive performance and deployment efficiency. Across datasets, it is the most consistent leader in accuracy/AUROC as acquisition cost increases, as shown in Fig. 3. Importantly, this performance does not come at the expense of inference-time overhead (see Fig. 4): CAPE is also among the fastest methods per acquisition, often requiring only a lightweight partition lookup at test time. In contrast, ACO, the nearest-neighbor oracle-based baseline, is the next strongest competitor on the cost-benefit tradeoff, but it is also among the slowest methods, often orders of magnitude slower than CAPE. ASAC sits at the opposite extreme: fast but underperforming on the cost-benefit tradeoff. Thus, the figures show that CAPE does not sacrifice speed/accuracy for the other–**it occupies the strongest region of the joint accuracy–latency frontier**.

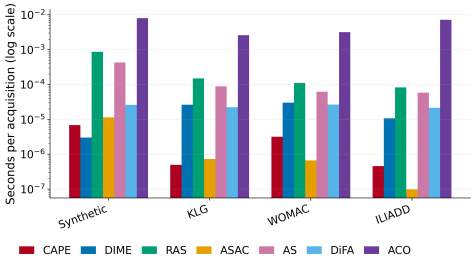


Figure 4: Inference latency per acquisition.

4.5 Ablation of the Acquisition-Aware Embedding

We evaluate the effectiveness of the learned acquisition-aware representation $g_\phi(\cdot)$ for constructing acquisition regions. To isolate this effect, we keep the same partitioning policy construction and replace only the representation space used to partition observation states. Specifically, the raw feature-

Table 1: Ablation comparing acquisition-aware embeddings $g_\phi(\cdot)$ used by CAPE against partitioning directly in the raw feature-value space. We report the mean $\pm$ standard errors across five runs.

Method	Synthetic		KLG		WOMAC		ILIADD	
	Acc.	cost	AUROC	cost	AUROC	cost	AUROC	cost
Feature Partitioning	0.774 ± 0.000 0.866 ± 0.000	7.178 ± 0.005 13.932 ± 0.004	0.804 ± 0.002 0.823 ± 0.001	9.912 ± 0.046 34.931 ± 0.053	0.618 ± 0.005 0.610 ± 0.004	8.461 ± 0.055 17.073 ± 0.204	0.762 ± 0.009 0.822 ± 0.004	17.283 ± 0.985 81.634 ± 1.818
CAPE	$\mathbf{0.792 \pm 0.000}$ $\mathbf{0.883 \pm 0.000}$	7.114 ± 0.005 13.877 ± 0.008	$\mathbf{0.815 \pm 0.004}$ $\mathbf{0.835 \pm 0.003}$	9.147 ± 0.082 33.758 ± 1.271	$\mathbf{0.628 \pm 0.004}$ $\mathbf{0.622 \pm 0.004}$	8.673 ± 0.275 17.395 ± 0.204	$\mathbf{0.786 \pm 0.001}$ $\mathbf{0.846 \pm 0.005}$	16.672 ± 1.717 86.450 ± 1.818

value baseline partitions states using the observed feature values x_o, comparing two observations $x_{i,o}$ and $x_{j,o}$ by $d(x_{i,o}, x_{j,o}) = \|x_{i,o} - x_{j,o}\|_2^2$. The result in Table 1 shows that this learned representation is crucial for effective partitioning. Across all datasets, partitioning in the learned embedding space improves predictive performance over partitioning directly in the raw feature-value space, while incurring comparable acquisition cost, which suggests that the improvement is not due to acquiring more information, but rather to selecting more useful acquisitions within the same policy-construction framework. Finally, the fact that raw feature-space partitioning often remains competitive with ACO provides further evidence for the underlying partition-based policy principle, while the additional gains from $g_\phi(\cdot)$ demonstrate the value of learning an acquisition-aware geometry.

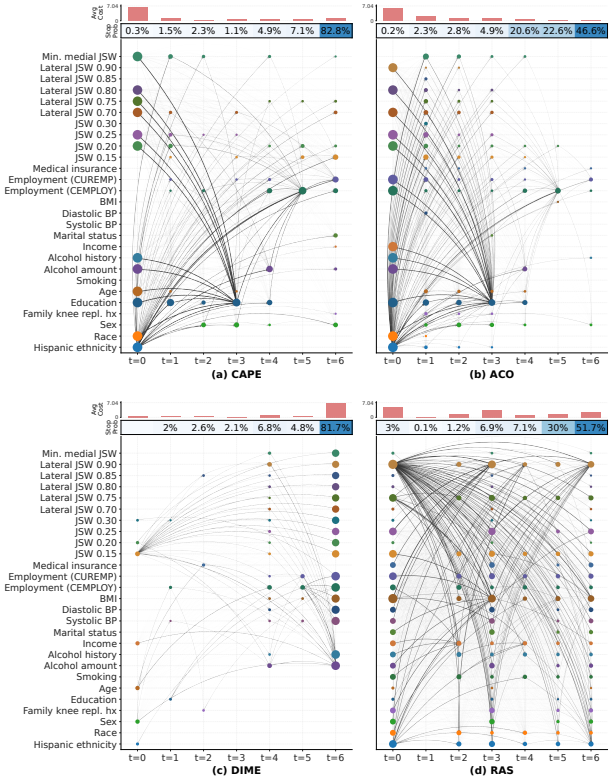


Figure 5: Acquisition dynamics on the WOMAC test set. For each method, **top** shows the mean acquisition cost per sample at each timepoint, and **middle** shows the proportion of samples terminating at each timepoint. The **bottom** graph shows feature–time acquisitions, with node size indicating acquisition frequency. A directed edge from (m, t) to (m', t'), where $t' > t$, indicates cross-time co-acquisition: samples acquiring (m, t) also acquire (m', t') later. Mean total acquisition costs per sample: 11.22 (CAPE), 11.89 (ACO), 9.95 (DIME), and 15.93 (RAS).

4.6 Acquisition Trajectory Analysis

Fig. 5 illustrates distinct acquisition dynamics on the WOMAC prediction task. CAPE combines early acquisition of demographic and clinical information with sustained follow-up through sparse, dynamic, feature-specific updates. Several joint-space-width measurements are selected for more patients at baseline than at later visits, where acquisition costs remain low. ACO similarly emphasizes baseline acquisition, while CAPE achieves this pattern with faster inference using NOCT-guided cached plans. In contrast, RAS repeatedly acquires a broad, relatively fixed set of features across timepoints, exhibiting less temporal variation in feature selection. DIME concentrates acquisitions at later visits, consistent with its greedy selection criterion, which may undervalue early measurements.

4.7 Robustness to Acquisition-Decision Perturbations and Distance from Learned Regions

We evaluate CAPE's robustness through two complementary analyses: (i) recovery from a deliberately perturbed acquisition decision and (ii) performance on test instances farther from the learned centroids.

4.7.1 Robustness to Acquisition-Decision Perturbations

CAPE revisits its acquisition recommendations as new observations become available. We assess whether this process remains effective after an early acquisition deviates from the policy's recommendation. At $t = 0$, CAPE follows its standard policy. At $t = 1$, we replace the recommended plan with a randomly selected cached plan from a different partition. From $t \geq 2$, CAPE resumes standard embedding-based planning using the resulting partial observations.

Table 2 shows small changes in both AUROC and average acquisition cost across WOMAC, KLG, and ILIADD, suggesting that the overall performance–cost tradeoff is relatively insensitive to this single early perturbation. This stability further illustrates CAPE's ability to adapt subsequent acquisition plans to the observations actually obtained.

Table 2: WOMAC/KLG/ILIADD AUROC with average cost, without and with random-plan perturbation.

Dataset	W/o perturb. AUROC	W/ perturb. AUROC	Δ AUROC	W/o perturb. cost	W/ perturb. cost	Δ Cost
KLG	0.8164	0.8101	-0.0063	9.390	9.060	-0.330
WOMAC	0.6207	0.6246	0.0039	11.447	11.251	-0.196
ILIADD	0.8269	0.8258	-0.0011	65.126	65.010	-0.116

4.7.2 Robustness Across Nearest-Centroid Distances

CAPE retrieves cached acquisition plans by mapping each test state (x_o, o, t) to its nearest centroid $\mu_{k,t}$ in the learned embedding space. We assess whether its performance remains stable for test instances farther from these centroids. For each instance, we compute the average nearest-centroid distance across all acquisition steps and compare the closest 80% (*inliers*) with the farthest 20% (*outliers*).

Table 3 shows small AUROC differences between the nearer and farther groups across WOMAC, KLG, and ILIADD, with relatively stable average acquisition costs. This stability suggests that cached acquisition plans remain useful even for test instances less closely matched to the learned centroids. Together with the plan-perturbation results, these findings support CAPE's use of a shared plan cache that accommodates variation in both acquisition histories and the partial observations encountered during inference.

Table 3: WOMAC/KLG/ILIADD AUROC with average cost for the nearest and farthest nearest-centroid-distance groups.

Dataset	Inliers AUROC	Outliers AUROC	Δ AUROC	Inliers cost	Outliers cost	Δ Cost
KLG	0.8152	0.8188	0.0037	9.522	8.861	-0.661
WOMAC	0.6151	0.6059	-0.0091	11.578	10.922	-0.656
ILIADD	0.8309	0.8145	-0.0164	65.617	63.160	-2.456

5 Conclusion

We introduced CAPE, a longitudinal AFA framework built on two complementary innovations: contrastive embeddings supervised by future acquisition-utility profiles, and a partitioning acquisition policy that caches plans over the learned space. To our knowledge, this is the first AFA framework to combine these mechanisms. NOCT guides both representation learning and offline plan selection, enabling states with similar acquisition needs to share planning effort. At deployment, nearest-centroid lookup supports fast acquisition decisions that adapt as new observations arrive. Across synthetic and real-world healthcare datasets, CAPE improves performance–cost tradeoffs over diverse LAFA baselines while enabling low-latency inference. These results highlight acquisition utility as a useful organizing principle for representation learning, connecting reusable nonmyopic planning with adaptive decisions about what to measure and when.

Acknowledgments

This research was partly funded by National Science Foundation (NSF) grant DMS2324394 and by National Institutes of Health (NIH) grants 1R01AA02687901A1, 1OT2OD032581-02-321, and 1OT2OD038045-01. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the NIH and NSF.

References

[1] Boqi Chen, Junier Oliva, and Marc Niethammer. A unified model for longitudinal multi-modal multi-view prediction with missingness. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 410–420. Springer, 2024.

[2] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[3] Ian Connick Covert, Wei Qiu, Mingyu Lu, Na Yoon Kim, Nathan J White, and Su-In Lee. Learning to maximize mutual information for dynamic feature selection. In *International Conference on Machine Learning*, pages 6424–6447. PMLR, 2023.

[4] Adam Dunkels, Joakim Eriksson, Niclas Finne, and Nicolas Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks, 2011.

[5] 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.

[6] Aritra Ghosh and Andrew Lan. Difa: Differentiable feature acquisition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7705–7713, 2023.

[7] Wenbo Gong, Sebastian Tschiatschek, Sebastian Nowozin, Richard E Turner, José Miguel Hernández-Lobato, and Cheng Zhang. Icebreaker: Element-wise efficient information acquisition with a bayesian deep latent gaussian model. *Advances in neural information processing systems*, 32, 2019.

[8] Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SyxtJh0qYm`.

[9] Jaromír Janisch, Tomáš Pevnỳ, and Viliam Lisỳ. Classification with costly features as a sequential decision-making problem. *Machine Learning*, 109(8):1587–1615, 2020.

[10] Jonas H Kellgren, JS Lawrence, et al. Radiological assessment of osteo-arthrosis. *Ann Rheum Dis*, 16(4): 494–502, 1957.

[11] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707, 2020.

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

[13] Jannik Kossen, Cătălina Cangea, Eszter Vértes, Andrew Jaegle, Viorica Patraucean, Ira Ktena, Nenad Tomasev, and Danielle Belgrave. Active acquisition for multimodal temporal data: A challenging decision-making task. *Transactions on Machine Learning Research*, 2023.

[14] Yang Li and Junier Oliva. Active feature acquisition with generative surrogate models. In *International conference on machine learning*, pages 6450–6459. PMLR, 2021.

[15] 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*, pages 4234–4243. PMLR, 2019.

[16] Sara McConnell, Pamela Kolopack, and Aileen M Davis. The western ontario and mcmaster universities osteoarthritis index (womac): a review of its utility and measurement properties. *Arthritis Care & Research: Official Journal of the American College of Rheumatology*, 45(5):453–461, 2001.

[17] Mahathir Monjur, Yubo Luo, Zhenyu Wang, and Shahriar Nirjon. Soundsieve: Seconds-long audio event recognition on intermittently-powered systems. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 28–41, 2023.

[18] Feng Nan, Joseph Wang, and Venkatesh Saligrama. Feature-budgeted random forest. In *International conference on machine learning*, pages 1983–1991. PMLR, 2015.

[19] Khanh Nguyen, Huy Hoang Nguyen, Egor Panfilov, and Aleksei Tiulpin. Active sensing of knee osteoarthritis progression with reinforcement learning. *arXiv preprint arXiv:2408.02349*, 2024.

[20] Sid E O'Bryant, Stephen C Waring, C Munro Cullum, James Hall, Laura Lacritz, Paul J Massman, Philip J Lupo, Joan S Reisch, Rachelle Doody, Texas Alzheimer's Research Consortium, et al. Staging dementia using clinical dementia rating scale sum of boxes scores: a texas alzheimer's research consortium study. *Archives of neurology*, 65(8):1091–1095, 2008.

[21] Sid E O'Bryant, Laura H Lacritz, James Hall, Stephen C Waring, Wenyaw Chan, Zeina G Khodr, Paul J Massman, Valerie Hobson, and C Munro Cullum. Validation of the new interpretive guidelines for the clinical dementia rating scale sum of boxes score in the national alzheimer's coordinating center database. *Archives of neurology*, 67(6):746–749, 2010.

[22] Ronald Carl Petersen, Paul S Aisen, Laurel A Beckett, Michael C Donohue, Anthony Collins Gamst, Danielle J Harvey, CR Jack Jr, William J Jagust, Leslie M Shaw, Arthur W Toga, et al. Alzheimer's disease neuroimaging initiative (adni) clinical characterization. *Neurology*, 74(3):201–209, 2010.

[23] Yuchao Qin, Mihaela van der Schaar, and Changhee Lee. Risk-averse active sensing for timely outcome prediction under cost pressure. *Advances in Neural Information Processing Systems*, 36, 2023.

[24] Arman Rahbar, Linus Aronsson, and Morteza Haghir Chehreghani. A survey on active feature acquisition strategies. *arXiv preprint arXiv:2502.11067*, 2025.

[25] Whitney R. Ringwald, Colin E. Vize, and Aidan G. C. Wright. Do you feel what i feel? the relation between congruence of perceived affect and self-reported empathy in daily life social situations. *Emotion*, 2025. URL `https://api.semanticscholar.org/CorpusID:278234840`.

[26] Maytal Saar-Tsechansky, Prem Melville, and Foster Provost. Active feature-value acquisition. *Management Science*, 55(4):664–684, 2009.

[27] Victor S Sheng and Charles X Ling. Feature value acquisition in testing: a sequential batch test algorithm. In *Proceedings of the 23rd international conference on Machine learning*, pages 809–816, 2006.

[28] Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. *Advances in neural information processing systems*, 31, 2018.

[29] Michael Valancius, Maxwell Lennon, and Junier Oliva. Acquisition conditioned oracle for nongreedy active feature acquisition. In *International Conference on Machine Learning*, pages 48957–48975. PMLR, 2024.

[30] Haiyan Yin, Yingzhen Li, Sinno Jialin Pan, Cheng Zhang, and Sebastian Tschiatschek. Reinforcement learning with efficient active feature acquisition. *arXiv preprint arXiv:2011.00825*, 2020.

[31] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Asac: Active sensing using actor-critic models. In *Machine Learning for Healthcare Conference*, pages 451–473. PMLR, 2019.

[32] Daphney-Stavroula Zois and Urbashi Mitra. Active state tracking with sensing costs: Analysis of two-states and methods for n-states. *IEEE Transactions on Signal Processing*, 65(11):2828–2843, 2017.

Appendices and Supplemental Materials

A Additional Discussion

A.1 Broader impacts

In this paper, we introduce a nonmyopic active feature acquisition framework, CAPE, which could aid the decision-making process in real-world applications such as healthcare. Specifically, CAPE could provide a more efficient and cost-effective diagnosis by selectively acquiring essential medical examinations, thus minimizing patient burden (e.g., time and costs) through personalized and timely intervention. However, as in any ML system, using CAPE should involve thorough/robust evaluations and supervision by medical professionals to prevent any potential negative impact.

Moreover, we acknowledge that our experiments focus on the healthcare domain, an important application area; however, our method is not specific to any domain. Similar resource-constrained scenarios also exist in other domains, such as in energy-aware audio recognition [17] or IoT devices [4]. In fact, our AFA formulation has applications across a myriad of domains, including robotics (acquisition of sensors with resource constraints), education (acquisition of different exam types without overburdening students), and beyond.

A.2 Assumptions & Limitations

Like traditional AFA methods and RL-based baselines, CAPE assumes that the problem has a finite horizon with a limited number of time steps. Additionally, we acknowledge that, similar to other baselines presented in this paper, the performance of CAPE is sensitive to the data quality. In particular, poor label annotation in real-world clinical datasets can degrade the accuracy. However, extending to an infinite-horizon setting and mitigating the sensitivity to the data quality are beyond the scope of this paper. We will leave these challenges for future work.

A.3 Time Complexity

With a total of L time steps and a total number of features M, the time complexity is bounded by $O(ML^2)$ for RL baselines (RAS, AS, ASAC), and $O(M^2L^2)$ for DiFA and DIME. For ACO, the nearest-neighbor oracle baseline, deployment requires retrieving similar training instances and scoring candidate plans. A naive nearest-neighbor search over n training instances costs $O(NML)$ per replanning step. After retrieval, scoring $|\mathcal{P}|$ candidate plans over the retrieved neighborhood $\mathcal{I}(x_o)$ gives a total trajectory complexity of $O(NML^2 + |\mathcal{I}(x_o)|ML^2|\mathcal{P}|)$ for the ACO baseline, assuming $O(ML)$ computation per candidate plan per neighbor. During deployment, CAPE computes the current embedding, finds the nearest cached region, and retrieves the corresponding cached plan. With K cached regions per timepoint and embedding dimension E, each nearest-centroid lookup costs $O(KE)$, and the full lookup overhead is at most $O(LKE)$ for CAPE .

A.4 Candidate-plan Construction $|\mathcal{P}|$

CAPE does not enumerate the combinatorial plan space; instead, it evaluates a fixed sampled candidate set offline. For each partition $C_{k,t}$, we construct a candidate-plan set $\mathcal{P}_{k,t}$ whose elements are plans $v \subseteq \mathcal{V}_{>t}$, where $\mathcal{V}_{>t}$ contains all feature-time pairs after time t and $|\mathcal{V}_{>t}| = M(L-t)$. For each candidate plan, we first sample its cardinality q uniformly from the integers between 0 and $M(L-t)$, and then uniformly sample q distinct feature-time pairs from $\mathcal{V}_{>t}$. Repeating this procedure yields $|\mathcal{P}|$ binary acquisition masks $v \in \{0,1\}^{M\times(L-t)}$.

A.5 Details on Acquisition-Utility Profile Construction $\zeta_{i,o}$

In this section, we formally introduce the acquisition-utility profiles $\zeta_{i,o}$. For any partial state $(x_{i,o}, o, t)$ and corresponding labels y_i from the training dataset $\mathcal{D}$, we identify the top-r candidate plans $\hat{v}_{i,o}^{(1)}, \ldots, \hat{v}_{i,o}^{(r)} \subseteq \mathcal{V}_{>t} = \{(m, t') \mid m \in \{1, \ldots, M\},\ t' \in \{t+1, \ldots, L\}\}$, in ascending order of the plug-in objective $\widehat{\mathrm{NOCT}}_{\mathrm{pl}}(x_{i,o}, o, \hat{v}_{i,o}^{(l)})$ (Eq. (3)). $\widehat{\mathrm{NOCT}}_{\mathrm{pl}}(x_{i,o}, o, \hat{v}_{i,o}^{(l)})$ quantifies the accuracy–cost tradeoff for candidate plan $\hat{v}_{i,o}^{(l)}$, and we have access to $x_{i,\hat{v}_{i,o}^{(l)}}$ during training since

it is from the training set. Each candidate plan $\hat{v}_{i,o}^{(l)}$ induces a uniform distribution $\nu_{\hat{v}_{i,o}^{(l)}}$ over its selected feature-time pairs (and termination), defined as $\nu_{\hat{v}_{i,o}^{(l)}}(m,t') = \frac{1}{|\hat{v}_{i,o}^{(l)}|}$ if $(m,t') \in \hat{v}_{i,o}^{(l)}$ and 0 otherwise. Intuitively, $\nu_{\hat{v}_{i,o}^{(l)}}$ spreads equal probability mass over all acquisitions in the future candidate plan $\hat{v}_{i,o}^{(l)}$. When $\hat{v}_{i,o}^{(l)}$ contains no further acquisition, $\nu_{\hat{v}_{i,o}^{(l)}}(\varnothing) = 1$; otherwise, $\nu_{\hat{v}_{i,o}^{(l)}}(\varnothing) = 0$. The acquisition-utility profile $\zeta_{i,o}$ is the weighted sum of these distributions:

$$\zeta_{i,o} = \sum_{l=1}^{r} w_{i,o}^{(l)} \, \nu_{\hat{v}_{i,o}^{(l)}} \tag{8}$$

where

$$w_{i,o}^{(l)} = \frac{\exp\big(-\widehat{\mathrm{NOCT}}_{\mathrm{pl}}(x_{i,o}, o, \hat{v}_{i,o}^{(l)})\big)}{\sum_{j=1}^{r} \exp\big(-\widehat{\mathrm{NOCT}}_{\mathrm{pl}}(x_{i,o}, o, \hat{v}_{i,o}^{(j)})\big)}.$$

Intuitively, given a partial state $(x_{i,o}, o, t)$, $\zeta_{i,o}$ indicates how likely each feature at each future timepoint (and termination) is to be selected by weighting its accuracy–cost tradeoff.

A.6 Proof of Theorem 3.1: $-\mathrm{NOCT}$ Lower Bounds the Optimal Longitudinal AFA MDP Value

Longitudinal AFA modeled as an MDP Longitudinal AFA can be formulated as a Markov Decision Process (MDP). The state $\mathbf{s}_t = \big(x_o,\, o,\, t\big)$, where x_o denotes the (partial) observations through time t and $o \subseteq \{(m,t') \mid m \in \{1,\dots,M\},\, t' \in \{1,\dots,t\}\}$ records which features were acquired. We suppose access to a pre-trained arbitrary conditioning classifier $\hat{y}$ [28, 14] that can predict the label at any target timepoint k with the current information, i.e., $\hat{y}_k(x_o)$. Note that the same input information can be used to predict at other time points, $\hat{y}_{k'}(x_o)$ (where forward prediction corresponds to when feature observations in o occur before k'). The action space is $a \in \{a_{t'}\}_{t'=t+1}^{L} \cup \{\varnothing\}$, where $a_{t'} \in \{0,1\}^M$ indicates which of the M features to acquire at time t', or, when $a = \varnothing$, to terminate and predict all remaining timepoints. For a non-terminal action, the transition is $\big(x_o,\, o,\, t\big) \to \big(x_{o \cup a_{t'}},\, o \cup a_{t'},\, t'\big)$, where (for notational simplicity) $o \cup a_{t'}$ denotes the union of the previous acquisitions and tuples of features at time t' in $a_{t'}$. Since this implies that we are not acquiring any information until time t', the reward is the negative cross-entropy (CE) loss of making predictions up to $t'-1$ (i.e., $\hat{y}_{t+1}, \dots, \hat{y}_{t'-1}$) with previous information (x_o) and the respective negative CE loss and cost of acquiring new features at time t': $-\sum_{k=t+1}^{t'-1} \mathrm{CE}(\hat{y}_k(x_o),\, y_k\big) - \mathrm{CE}(\hat{y}_{t'}(x_{o \cup a_{t'}}),\, y_{t'}\big) - \alpha \sum_{m=1}^{M} a_{t',m} c^m$, where α is an *application-specific* accuracy-cost tradeoff parameter; certain applications allow a higher cost (i.e., lower α) and some are more cost-restricted (i.e., high α). When terminated ($a = \varnothing$) at timepoint t, we are not acquiring any additional information for the remaining timepoints; thus, the reward is $-\sum_{k=t+1}^{L} \mathrm{CE}(\hat{y}_k(x_o),\, y_k\big)$, which represents the negative cross-entropy loss accumulated from $t+1$ to L using the current information at hand. We denote our policy as $\pi\big(x_o,\, o,\, t\big)$, which selects actions to maximize the expected cumulative rewards through L.

Theorem 3.1. Let $V^\star(\mathbf{s})$ be the optimal value of the corresponding finite-horizon LAFA. Consider a partially observed state $\mathbf{s} = (x_o, o, t)$ given a fixed predictor $\hat{y}$ and nonnegative acquisition costs. We have:

$$\text{If } J_{\mathrm{NOCT}}(\mathbf{s}) = -\min_{v \subseteq \mathcal{V}_{>t}} \mathrm{NOCT}(x_o, o, v), \text{ then } J_{\mathrm{NOCT}}(\mathbf{s}) \le V^\star(\mathbf{s}). \tag{9}$$

Problem Setup. We extend the argument of Valancius et al. [29, Theorem C.1] to the longitudinal setting. For any observations (x_o, o, t), let the set of remaining feature-time pairs that are still available after time t be:

$$\mathcal{V}_{>t} = \{(m,t') \mid m \in \{1,\dots,M\},\, t' \in \{t+1,\dots,L\}\}.$$

To follow the convention of reinforcement learning literature, we redefine the objective NOCT (Eq. (1)) as maximization instead of minimization without loss of generality. For notational convenience, we write $\texttt{NOCT-MAX}_t^s(x_o) \equiv \texttt{NOCT-MAX}_t^s(x_o, o)$, since x_o is indexed by the observation set o. That is, for any partial observation (x_o, o, t), we define $\texttt{NOCT-MAX}_t^s(x_o)$ as:

$$\texttt{NOCT-MAX}_t^s(x_o) := \max_{\substack{v \subseteq \mathcal{V}_{>t} \\ \text{s.t } |\text{rounds}(v)| \leq s}} \Big[-\mathbb{E}_{y_{t+1:L}, x_v | x_o}[\ell(x_o \cup x_v, y, t)] - \alpha \sum_{(m,t') \in v} c^m \Big],$$

where $s \in \mathbb{N}$ represents the maximum number of possible future acquisition rounds, and let $\text{rounds}(v) := \{t' \mid \exists m \in \{1, \ldots, M\} : (m, t') \in v\}$. We then define

$$V_t^0(x_o) := -\mathbb{E}_{y_{t+1:L}|x_o}[\ell(x_o, y, t)],$$

$$\Delta_{t,t'}(x_o, a_{t'}) := \mathbb{E}_{y_{t+1:t'}, x_{a_{t'}}|x_o} \left[\sum_{k=t+1}^{t'-1} \text{CE}\big(\hat{y}_k(x_o), y_k\big) + \text{CE}\big(\hat{y}_{t'}(x_{o \cup a_{t'}}), y_{t'}\big) \right],$$

and

$$V_t^s(x_o) := \max\Big\{ V_t^0(x_o), \max_{\substack{t' \in \{t+1,\ldots,L\} \\ a_{t'} \in \{0,1\}^M}} \Big[-\Delta_{t,t'}(x_o, a_{t'}) - \alpha \sum_m a_{t',m} c^m + \mathbb{E}_{x_{a_{t'}}|x_o}[V_{t'}^{s-1}(x_{o \cup a_{t'}})] \Big] \Big\}. \tag{10}$$

Here, $V_t^s(x_o)$ denotes the optimal expected cumulative reward from time $t+1$ through L, given the current partial observations x_o, with at most s further acquisition rounds. We see that V_t^0 is the expected negative cumulative loss when we immediately terminate at t, and we predict the future labels using only what we have acquired so far, x_o. Thus, V_t^s is the value for the optimal longitudinal AFA policy, such that it either (i) terminates, returning V_t^0 or (ii) executes another acquisition at the future time step $t' \in (t+1, ..., L)$ and acquires the feature $a_{t'} \in \{0,1\}^M$.

Claim. For every (x_o, o, t) with $s \in \mathbb{N}_{\geq 0}$, we have $V_t^s(x_o) \geq \texttt{NOCT-MAX}_t^s(x_o)$.

Proof.

Base case $s = 0$. Since there is no further action, both the MDP agent and NOCT-MAX terminate:

$$V_t^0(x_o) = -\mathbb{E}_{y_{t+1:L}|x_o}[\ell(x_o, y, t)] = \texttt{NOCT-MAX}_t^0(x_o).$$

Inductive hypothesis. Assume for $s - 1 \geq 0$, we have:

$$V_{t'}^{s-1}(x) \geq \texttt{NOCT-MAX}_{t'}^{s-1}(x) \quad \forall (x, t'). \tag{IH}$$

Inductive step. We fix (x_o, t) and see that:

$$\begin{aligned}
&V_t^s(x_o) \\
&= \max\Big\{ V_t^0(x_o), \max_{\substack{t' \in \{t+1,\ldots,L\} \\ a_{t'} \in \{0,1\}^M}} \Big[-\Delta_{t,t'}(x_o, a_{t'}) - \alpha \sum_m a_{t',m} c^m \\
&\qquad + \mathbb{E}_{x_{a_{t'}}|x_o}[V_{t'}^{s-1}(x_{o \cup a_{t'}})] \Big] \Big\} \\
&\overset{\text{(IH)}}{\geq} \max\Big\{ V_t^0(x_o), \max_{\substack{t' \in \{t+1,\ldots,L\} \\ a_{t'} \in \{0,1\}^M}} \Big[-\Delta_{t,t'}(x_o, a_{t'}) - \alpha \sum_m a_{t',m} c^m \\
&\qquad + \mathbb{E}_{x_{a_{t'}}|x_o}[\texttt{NOCT-MAX}_{t'}^{s-1}(x_{o \cup a_{t'}})] \Big] \Big\} \\
&= \max\Big\{ V_t^0(x_o), \\
&\max_{\substack{t' \in \{t+1,\ldots,L\} \\ a_{t'} \in \{0,1\}^M}} \Big[-\Delta_{t,t'}(x_o, a_{t'}) - \alpha \sum_m a_{t',m} c^m + \mathbb{E}_{x_{a_{t'}}|x_o} \Big[\max_{\substack{v' \subseteq \mathcal{V}_{>t'} \\ \text{s.t } |\text{rounds}(v')| \leq s-1}} \Big(\\
&\qquad - \mathbb{E}_{y_{t'+1:L}, x_{v'}|x_o, x_{a_{t'}}}[\ell(x_o \cup x_{a_{t'}} \cup x_{v'}, y, t')] - \alpha \sum_{(m,t'') \in v'} c^m \Big) \Big] \Big] \Big\} \\
&\overset{*}{\geq} \max\Big\{ V_t^0(x_o),
\end{aligned}$$

$$
\begin{aligned}
&\max_{\substack{t' \in \{t+1,\ldots,L\} \\ a_{t'} \in \{0,1\}^M}} \max_{\substack{v' \subseteq \mathcal{V}_{>t'} \\ \text{s.t } |\text{rounds}(v')| \leq s-1}} \Big[\\
&\quad - \mathbb{E}_{y_{t'+1:L}, x_{v'}, x_{a_{t'}} | x_o} [\ell(x_o \cup x_{a_{t'}} \cup x_{v'}, y, t')] \\
&\quad - \Delta_{t,t'}(x_o, a_{t'}) - \alpha \Big(\sum_{(m,t'') \in v'} c^m + \sum_m a_{t',m} c^m \Big) \Big] \Big\} \\
&= \max \Big\{ V_t^0(x_o), \max_{\substack{v \subseteq \mathcal{V}_{>t} \\ \text{s.t } |\text{rounds}(v)| \leq s}} [-\mathbb{E}_{y_{t+1:L}, x_v | x_o} [\ell(x_o \cup x_v, y, t)] - \alpha \sum_{(m,t'') \in v} c^m] \Big\} \\
&= \texttt{NOCT-MAX}_t^s(x_o).
\end{aligned}
$$

where * follows from:

$$
\begin{aligned}
\forall v' \in \Omega, x_{a_{t'}}, \max_{v \in \Omega} \mathcal{L}\left(x_{a_{t'}}, v\right) \geq \mathcal{L}\left(x_{a_{t'}}, v'\right) \quad &\implies \\
\forall v' \in \Omega, \mathbb{E}_{x_{a_{t'}} | x_o} \left[\max_{v \in \Omega} \mathcal{L}\left(x_{a_{t'}}, v\right) \right] \geq \mathbb{E}_{x_{a_{t'}} | x_o} \left[\mathcal{L}\left(x_{a_{t'}}, v'\right) \right] &\implies \\
\mathbb{E}_{x_{a_{t'}} | x_o} \left[\max_{v \in \Omega} \mathcal{L}\left(x_{a_{t'}}, v\right) \right] \geq \max_{v \in \Omega} \mathbb{E}_{x_{a_{t'}} | x_o} \left[\mathcal{L}\left(x_{a_{t'}}, v\right) \right] &
\end{aligned}
$$

for $\Omega := \left\{ v \subseteq \mathcal{V}_{>t'} \;\middle|\; |\text{rounds}(v)| \leq s-1 \right\}$ and

$$
\mathcal{L}(x_{a_{t'}}, v) := -\mathbb{E}_{y_{t'+1:L}, x_v | x_o, x_{a_{t'}}} [\ell(x_o \cup x_{a_{t'}} \cup x_v, y, t')] - \alpha \left(\sum_{(m,t'') \in v} c^m + \sum_m a_{t',m} c^m \right).
$$

Note that one may recover the non-acquisition-round constrained problem by considering large enough s, yielding a lower bound on the longitudinal AFA MDP.

Theorem 3.1 establishes why NOCT is an appropriate objective for evaluating future acquisition plans. Under a fixed predictor and nonnegative acquisition costs, the negative of the minimum ideal expected NOCT objective provides a lower bound on the optimal finite-horizon LAFA value. Thus, NOCT is not an ad hoc acquisition score: it is a principled plan-level surrogate tied to the same horizon-wide prediction-loss and acquisition-cost tradeoff as the sequential LAFA problem.

A.7 Longitudinal AFA MDP Challenges

Modeling longitudinal AFA as an MDP introduces several difficulties: (1) *a large and complicated action space*, since one requires decisions on when to acquire and which features to acquire; (2) *a complicated state space*, as it evolves dynamically over the combinatorial acquisitions as new features are captured over time; and (3) *a credit assignment problem*, since acquisitions are rewarded at an aggregate level, making it hard to disentangle exactly which of the acquired features were most responsible for prediction rewards [14]. Despite these challenges, prior work [23, 31, 13, 19] often relies on RL for longitudinal AFA. In contrast, `CAPE` is grounded in an MDP formulation but does not learn a Bellman value function, use temporal-difference learning, or optimize an actor-critic policy. Instead, we define NOCT as an explicit plan-level objective over future acquisition trajectories. The ideal NOCT objective induces a lower bound on the optimal longitudinal AFA MDP value, and `CAPE` uses plug-in NOCT scores from training trajectories to learn an acquisition-utility manifold and fit a partitioning acquisition policy.

B Additional results

Please find additional results below.

ADNI We additionally consider the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset (`https://adni.loni.usc.edu`) [22], a longitudinal, multi-center observational study of cognitive decline. Following Qin et al. [23], we use $N = 1{,}002$ patients diagnosed as cognitively unimpaired, mild cognitive impairment, or Alzheimer's disease [20, 21], with four PET/MRI biomarkers (FDG, AV45, Hippocampus, and Entorhinal) across $L = 12$ visits. We split the dataset into train/val/test

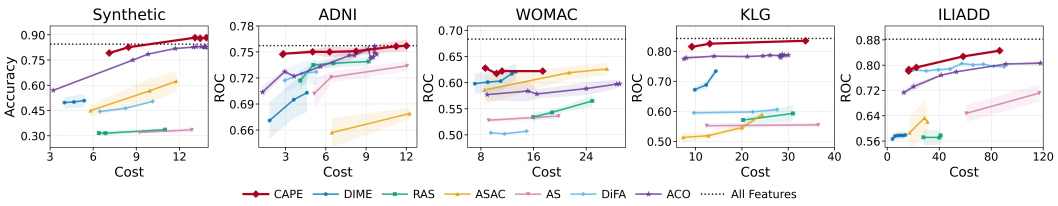


Figure 6: Performance/cost of models across various average acquisition costs (budgets), with standard error across 5 runs. Following Kossen et al. [13], we report accuracy instead of AUROC for synthetic data. The dotted horizontal line shows performance using all features. CAPE consistently yields higher predictive performance at comparable or lower acquisition costs.

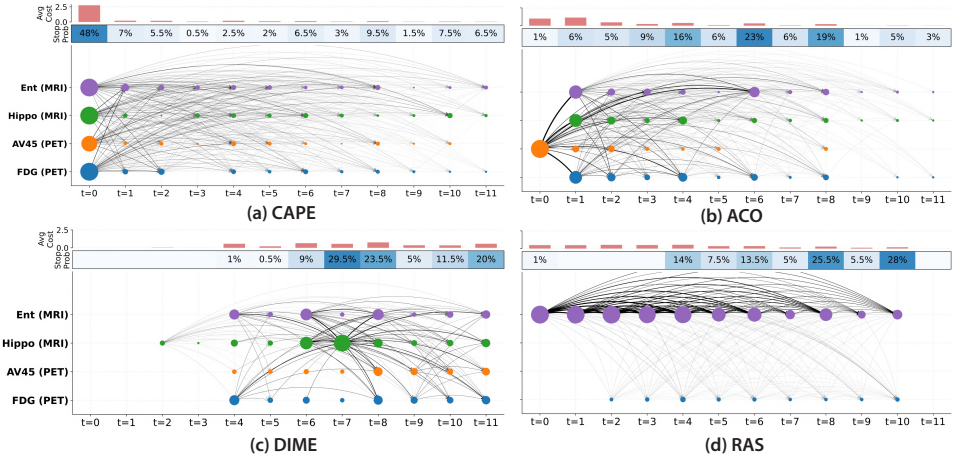


Figure 7: Acquisition dynamics on the test set of the ADNI dataset. For each method, **top** shows the average acquisition cost at each timepoint (mean per sample), and **middle** shows the termination distribution that occurred at each timepoint. The **bottom** panel visualizes feature-time acquisition as a directed graph: each node corresponds to a feature acquired at a specific timepoint, and node size shows how often that feature is acquired at that time. A directed arrow from (m, t) to (m', t') (with $t' > t$) indicates the frequent co-acquisition across time, i.e., samples that acquire (m, t) also acquire (m', t') later. Average acquisition cost per sample: 4.0375 (CAPE), 3.985 (ACO), 4.0475 (DIME), and 3.9750 (RAS).

$(64\%/16\%/20\%)$. Each patient may have a different pattern of missing features or labels, and we only report the results based on those that have a diagnosis label. We observe similar patterns, e.g., Figs. 6–8.

B.1 Hyperparameter Sensitivity

We evaluate sensitivity to the contrastive margin γ, divergence weight β, embedding dimension E, and the number of top plans r, while holding all other settings fixed. As shown in Tables 5–8, CAPE remains highly robust across a broad range of configurations, indicating that our method can be readily applied without excessive hyperparameter tuning. Furthermore, across these varied settings, CAPE consistently maintains its performance advantage over the baselines reported in Figure 6.

B.2 Candidate-Plan Set Size and Offline Construction

We first evaluate how the candidate-set size $|\mathcal{P}|$ affects the performance–cost tradeoff. We varied the number of candidate plans $|\mathcal{P}| \in \{100, 500, 1000, 5000, 10000\}$ across datasets with different trajectory lengths (L) and feature dimensions (M). As shown in Table 9, a small candidate set (e.g., $|\mathcal{P}| = 100$) can yield less optimal plans, resulting in higher acquisition cost on datasets like KLG or ILIADD. Expanding the set to $|\mathcal{P}| = 10000$ suggests diminishing returns, as additional computational expense does not translate into meaningful performance gains, supporting $|\mathcal{P}| = 5000$ as a strong practical default.

Figure 9 shows the offline construction time across datasets with varying trajectory lengths (L) and feature dimensions (M). While runtime scales with $|\mathcal{P}|$, it remains substantially faster than baselines

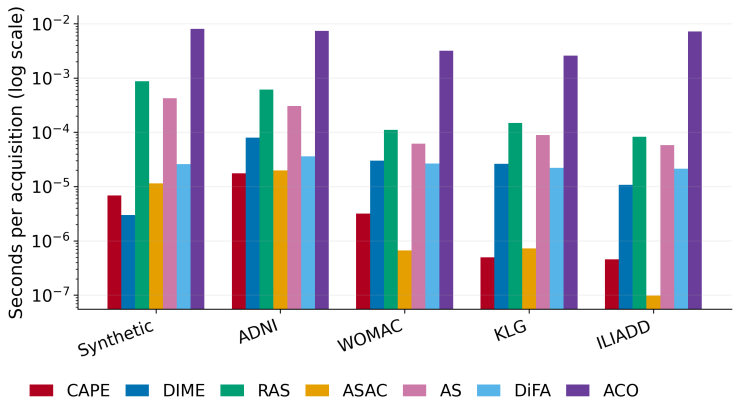


Figure 8: Inference latency per acquisition.

Table 4: Ablation comparing acquisition-aware embeddings $g_\phi(\cdot)$ used by CAPE against partitioning directly in the raw feature-value space. We report the mean $\pm$ standard errors across five runs.

Method	Synthetic		ADNI		WOMAC		KLG		ILIADD	
	Acc.	cost	AUROC	cost	AUROC	cost	AUROC	cost	AUROC	cost
Feature Partitioning	0.774 ± 0.000 0.866 ± 0.000	7.178 ± 0.005 13.932 ± 0.004	0.739 ± 0.000 0.746 ± 0.002	2.760 ± 0.011 12.328 ± 0.107	0.618 ± 0.005 0.610 ± 0.004	8.461 ± 0.055 17.073 ± 0.204	0.804 ± 0.002 0.823 ± 0.001	9.912 ± 0.046 34.931 ± 0.053	0.762 ± 0.009 0.822 ± 0.004	17.283 ± 0.985 81.634 ± 1.818
CAPE	$\mathbf{0.792 \pm 0.000}$ $\mathbf{0.883 \pm 0.000}$	7.114 ± 0.005 13.877 ± 0.008	$\mathbf{0.748 \pm 0.001}$ $\mathbf{0.757 \pm 0.007}$	2.821 ± 0.020 12.012 ± 0.208	$\mathbf{0.628 \pm 0.004}$ $\mathbf{0.622 \pm 0.004}$	8.673 ± 0.275 17.395 ± 0.204	$\mathbf{0.815 \pm 0.004}$ $\mathbf{0.835 \pm 0.003}$	9.147 ± 0.082 33.758 ± 1.271	$\mathbf{0.786 \pm 0.001}$ $\mathbf{0.846 \pm 0.005}$	16.672 ± 1.717 86.450 ± 1.818

Table 5: Test performance and average cost by contrastive-loss margin γ and cost regime.

γ	Cost regime	ADNI AUROC	ADNI Cost	WOMAC AUROC	WOMAC Cost	KLG AUROC	KLG Cost	ILIADD AUROC	ILIADD Cost
0.5	Low	0.7468	2.788	0.6236	8.981	0.8197	9.359	0.7926	19.543
1	Low	0.7488	2.768	0.6201	8.980	0.8168	9.638	0.7931	19.780
2	Low	0.7507	2.772	0.6224	8.822	0.8188	9.984	0.7956	19.338
0.5	High	0.7539	11.295	0.6256	18.191	0.8363	38.084	0.8453	89.853
1	High	0.7547	12.002	0.6241	17.245	0.8353	38.299	0.8454	90.817
2	High	0.7548	12.105	0.6254	17.457	0.8351	37.550	0.8459	89.042

Table 6: Test performance and average cost by divergence weight β and cost regime.

β	Cost regime	ADNI AUROC	ADNI Cost	WOMAC AUROC	WOMAC Cost	KLG AUROC	KLG Cost	ILIADD AUROC	ILIADD Cost
0.5	Low	0.7495	2.775	0.6217	8.966	0.8177	9.272	0.7941	19.425
1	Low	0.7487	2.760	0.6230	8.781	0.8164	9.390	0.7933	19.582
2	Low	0.7506	2.772	0.6228	9.055	0.8165	9.574	0.7965	19.868
0.5	High	0.7607	11.785	0.6250	18.074	0.8310	33.251	0.8448	90.956
1	High	0.7520	11.640	0.6248	17.484	0.8374	37.567	0.8460	88.172
2	High	0.7515	11.845	0.6231	17.391	0.8341	33.699	0.8459	88.761

Table 7: Test performance and average cost by embedding dimension E and cost regime.

E	Cost regime	ADNI AUROC	ADNI Cost	WOMAC AUROC	WOMAC Cost	KLG AUROC	KLG Cost	ILIADD AUROC	ILIADD Cost
16	Low	0.7487	2.763	0.6192	9.024	0.8195	9.766	0.7937	19.496
32	Low	0.7488	2.768	0.6235	8.965	0.8101	8.845	0.7956	19.441
64	Low	0.7487	2.760	0.6183	8.964	0.8178	9.582	0.7940	19.454
128	Low	0.7537	2.772	0.6234	8.902	0.8147	9.284	0.7980	19.328
16	High	0.7579	11.490	0.6232	17.288	0.8322	37.052	0.8460	88.529
32	High	0.7537	11.922	0.6268	16.859	0.8353	39.903	0.8460	89.421
64	High	0.7579	11.837	0.6236	17.378	0.8329	32.547	0.8481	90.211
128	High	0.7543	12.390	0.6235	17.547	0.8351	35.930	0.8463	89.195

like DIME and AS across all datasets at default $|\mathcal{P}| = 5000$. Importantly, this computation is strictly an offline policy-construction cost and does not impact CAPE's highly efficient nearest-centroid lookup during inference.

Table 8: Test performance and average cost by number of top plans r and cost regime.

r	Cost regime	ADNI AUROC	ADNI Cost	WOMAC AUROC	WOMAC Cost	KLG AUROC	KLG Cost	ILIADD AUROC	ILIADD Cost
3	Low	0.7493	2.808	0.6197	8.853	0.8134	9.190	0.7948	19.365
5	Low	0.7488	2.768	0.6254	9.104	0.8169	9.294	0.7918	19.248
10	Low	0.7468	2.788	0.6237	8.943	0.8189	9.719	0.7945	19.552
3	High	0.7545	11.700	0.6209	17.150	0.8359	36.646	0.8464	89.641
5	High	0.7551	11.935	0.6265	17.277	0.8374	38.735	0.8466	87.570
10	High	0.7545	11.787	0.6251	17.433	0.8354	35.513	0.8425	85.558

Table 9: CAPE performance varying the number of candidate plans $|\mathcal{P}|$ across datasets.

$\|\mathcal{P}\|$	Synthetic Acc.	Synthetic Cost	ADNI AUROC	ADNI Cost	KLG AUROC	KLG Cost	ILIADD AUROC	ILIADD Cost
100	0.8709	14.498	0.7485	11.772	0.8348	42.087	0.8367	105.824
500	0.8673	14.113	0.7505	11.312	0.8349	35.249	0.8411	95.380
1,000	0.8805	13.780	0.7498	11.677	0.8381	40.366	0.8401	94.587
5,000	0.8845	13.647	0.7520	11.640	0.8374	37.567	0.8460	88.172
10,000	0.8838	13.865	0.7503	11.738	0.8364	34.542	0.8481	86.389

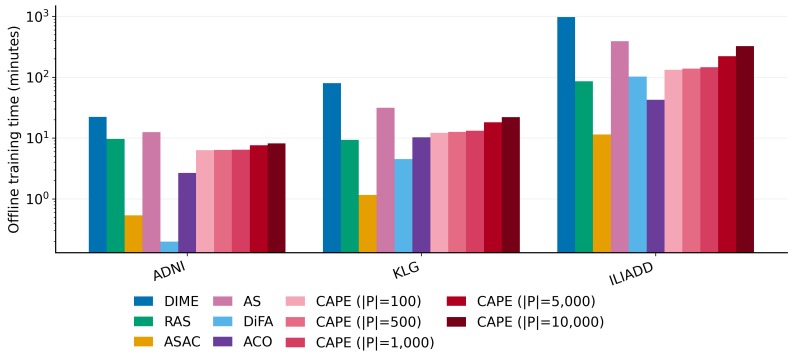


Figure 9: Offline training time in minutes (lower is better).

B.3 Offline Training Cost and Scalability

We evaluate the offline scalability of CAPE using a controlled setup on the Intensive Longitudinal Investigation of Alternative Diagnostic Dimensions (ILIADD) dataset. In the main experiments, we used a trajectory length of $L = 10$ with $M = 34$ features to predict negative affect. We evaluate scalability along two axes:

- **Scaling Time (L):** We fixed M and increased the time length to $L \in \{20, 30, 50\}$, scaling up to 1,700 (feature, time) pairs.
- **Scaling Features (M):** We fixed $L = 20$ and increased the feature dimensions to $M \in \{68, 102\}$, scaling up to 2,040 (feature, time) pairs. To do this, we retained the original 34 features and appended standard Gaussian noise features.

As shown in Table 10 and Figures 10–11, CAPE scales to much larger horizons and dimensionalities. Across every (L, M) combination, CAPE achieves the highest AUROC and the lowest acquisition cost. Furthermore, CAPE's offline construction time grows predictably and is comparable to other baselines (avoiding the substantial runtime growth that limits baselines like RAS at high L), all while maintaining sub-millisecond inference latency per acquisition.

Table 10: ILIADD test AUROC and average acquisition cost across methods by (L, M).

L	M	RAS AUROC	RAS Cost	ASAC AUROC	ASAC Cost	AS AUROC	AS Cost	ACO AUROC	ACO Cost	CAPE AUROC	CAPE Cost
10	34	0.5780	41.097	0.6328	28.600	0.7415	204.497	0.7789	53.147	**0.7862**	**16.672**
20	34	0.7507	283.176	0.6309	109.795	0.7308	109.639	0.7517	111.830	**0.8294**	**42.338**
30	34	0.7579	441.695	0.6710	110.000	0.7224	109.756	0.7468	110.168	**0.8376**	**56.791**
50	34	0.7492	1508.057	0.5641	40.645	0.6969	109.785	0.6872	33.475	**0.7621**	**25.844**
20	68	0.7146	442.117	0.6592	109.992	0.6567	187.338	0.6539	101.002	**0.7764**	**49.605**
20	102	0.6555	725.170	0.6372	109.930	0.6008	267.578	0.6618	54.408	**0.6912**	**52.451**

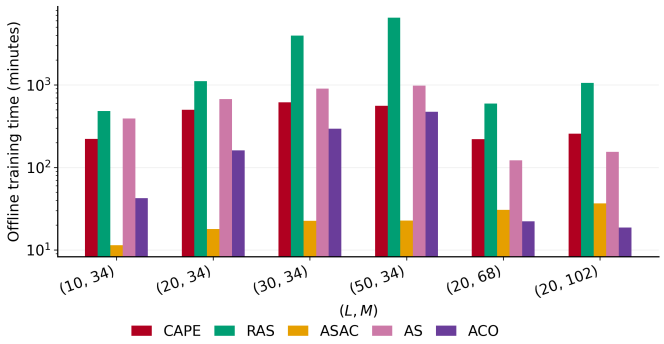


Figure 10: ILIADD offline training time across methods by (L, M), in minutes. Note that L is the number of timesteps and M is the number of features.

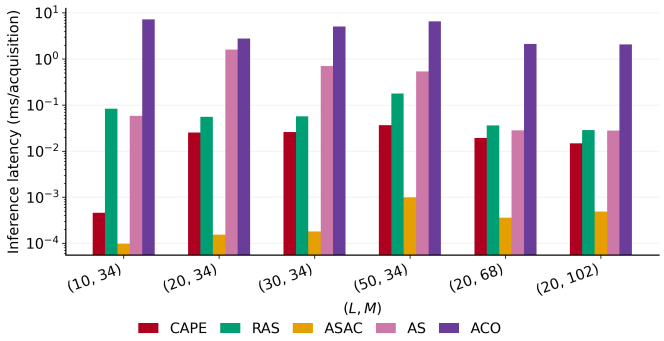


Figure 11: ILIADD inference latency across methods by (L, M), in milliseconds per acquisition. Note that L is the number of timesteps and M is the number of features.

B.4 Average Precision Evaluation

We evaluate average precision (AP), detailed in Figure 12. The results demonstrate that CAPE consistently achieves higher AP while incurring lower acquisition costs compared to the baselines. These results confirm that the strong Pareto-frontier performance observed in AUROC translates directly to AP.

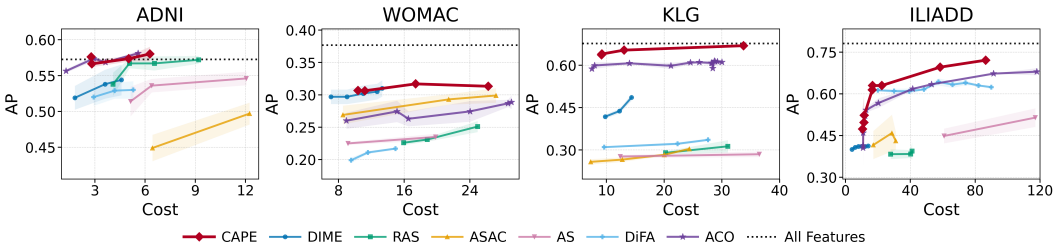
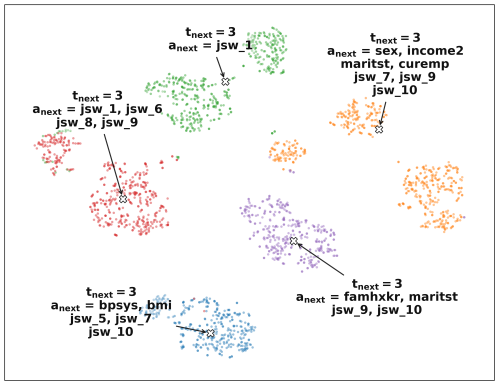
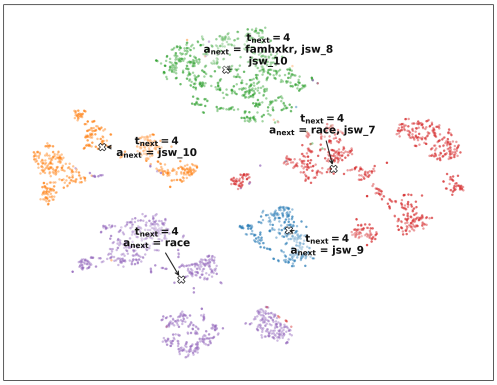


Figure 12: Average precision (AP) versus average acquisition cost (budget) of CAPE and the baselines. The dotted horizontal line shows performance using all features.

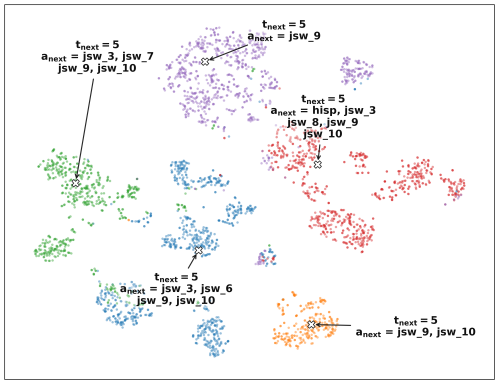
B.5 Additional Acquisition-Embedding Visualizations



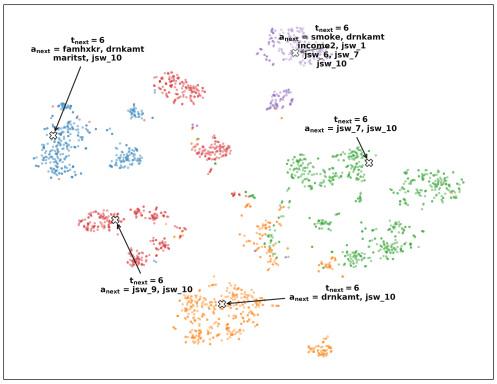
(a) Acquisition regions at time $t = 2$.



(b) Acquisition regions at time $t = 3$.



(c) Acquisition regions at time $t = 4$.



(d) Acquisition regions at time $t = 5$.

Figure 13: Extended visualizations of the acquisition-aware embeddings on the KLG task at later rollout stages. Building upon Figure 2, the t-SNE projections at rollout times $t \in \{2, 3, 4, 5\}$ demonstrate how the learned representation maintains distinct local regions for shared cached acquisition decisions as the trajectory progresses.

C Experimental Setup

C.1 Cost of Features

Synthetic data. Our synthetic dataset includes 2 features, one being the digit and the other being the counter. Since both are generated through random selection, we assign an equal cost, 1, to the two features. Upon publication, we will release the synthetic dataset.

ADNI. The ADNI dataset includes 4 features per time point, with FDG and AV45 extracted from PET scans and Hippocampus and Entorhinal extracted from MRI scans. Since PET is a more expensive diagnosis, we follow Qin et al. [23] and assign a cost of 1 to FDG and AV45, and a cost of 0.5 to Hippocampus and Entorhinal. Access to ADNI data may be requested via `https://adni.loni.usc.edu`, and the Data Use Agreement is available at `https://adni.loni.usc.edu/terms-of-use/`.

OAI. We use a total of 27 features for the OAI dataset, with 17 clinical measurements and 10 joint space width (JSW) measurements extracted from knee radiography. For the clinical measurements, we assign a low cost of 0.3 to those that require minimal effort, e.g., age, sex, and race, and a slightly higher cost of 0.5 to blood pressure and BMI calculation. For JSW, a cost of 1.0 is assigned for the

Table 11: RAS diagnostic-error coefficient grids and selected values.

Dataset	Search grid for γ^{RAS}	Selected γ^{RAS}
Synthetic	$\{200, 250, 280, 300, 310, 320, 350, 400, 5000, 7000, 9000\}$	$\{5000, 7000, 9000\}$
ADNI	$\{50, 75, 100, 175, 200, 250, 300, 350, 400, 450\}$	$\{50, 75, 100, 175\}$
WOMAC	$\{200, 250, 300, 350, 400, 450, 2000, 3000, 4500, 9000\}$	$\{2000, 4500, 9000\}$
KLG	$\{200, 250, 300, 350, 400, 450, 1200, 1700\}$	$\{1200, 1700\}$
ILIADD	$\{2, 5, 10, 15, 20, 30\}$	$\{5, 10, 15\}$

minimum JSW and 0.8 for those measured at different positions. Access to the OAI data may be requested via `https://nda.nih.gov/oai/`.

ILIADD. We use a total of 34 features for the ILIADD dataset [25], with features consisting of demographics and scales derived from the HiTOP-SR, factor scales, symptom scales, and alcohol use scales, and temporal features assessing momentary positive affect, energy, stress, and impulsivity. For all measurements, we assign an equal cost of 1.0.

C.2 Baseline Implementation

For ASAC, RAS, and AS, we use the implementation available at `https://github.com/yvchao/cvar_sensing`, which is under the BSD-3-Clause license. For DIME [5] and ACO [29], we extend the methods to the longitudinal setting by building on top of the official implementation available at `https://github.com/suinleelab/DIME` and `https://github.com/lupalab/aaco`, respectively. At the time of writing, the DiFA codebase is not publicly released, and researchers may request it directly from the authors.

Following Qin et al. [23], we share the same neural CDE predictor [11] for ASAC, RAS, and AS. Following the authors' released code, we treat the drop rate p of the auxiliary observation strategy π_0 (the same notation as in Qin et al. [23]) as a hyperparameter and sweep $p \in \{0.0, 0.3, 0.5, 0.7\}$. We select p based on the outcome predictor's average accuracy over five randomly masked validation sets generated by π_0. This matches the original evaluation protocol to ensure comparability and reproducibility. The selected drop rates are $p = 0.3$ (Synthetic), $p = 0.7$ (ADNI), $p = 0.5$ (WOMAC), $p = 0.3$ (KLG), and $p = 0.7$ (ILIADD).

ASAC [31]. We select the coefficient for the acquisition cost: 1) $\mu \in \{0.0015, 0.002, 0.003\}$ for the synthetic dataset, 2) $\mu \in \{0.005, 0.02\}$ for ADNI, 3) $\mu \in \{0.0012, 0.0016, 0.003\}$ for the WOMAC of OAI dataset, 4) $\mu \in \{0.00125, 0.0015, 0.00175, 0.005\}$ for the KLG of OAI dataset, and 5) $\mu \in \{0.02, 0.005, 0.002\}$ for ILIADD. Note that we report multiple final μ values because we evaluate ASAC across multiple acquisition costs.

RAS [23]. For the synthetic dataset, we set the allowed acquisition interval to $(\Delta_{\min} = 0.2, \Delta_{\max} = 1.0)$ for the synthetic dataset, $(\Delta_{\min} = 0.5, \Delta_{\max} = 1.5)$ for ADNI, and $(\Delta_{\min} = 0.5, \Delta_{\max} = 4.5)$ for WOMAC, KLG, and ILIADD.

Moreover, for RAS, we tune the diagnostic-error coefficient γ^{RAS} on the validation set using a dataset-specific grid search. Table 11 lists the search grids and the final selected values. We report multiple selected values because RAS is evaluated across multiple acquisition-cost settings. For all datasets, we use a tail-risk quantile of 0.1, an invalid-visit penalty of 10, and a discount factor of 0.99, following the authors' defaults.

AS [23]. We set the minimum and maximum allowed acquisition intervals as in the RAS setting. In addition, we select a fixed acquisition interval of: 1) $\tilde{\Delta} \in \{0.2, 0.4\}$ for the synthetic dataset, 2) $\tilde{\Delta} \in \{0.5, 1.0, 1.5\}$ for ADNI and ILIADD datasets, 3) $\tilde{\Delta} \in \{0.5, 1.5\}$ for WOMAC of OAI, and 4) $\tilde{\Delta} \in \{0.5, 1.5\}$ for KLG of OAI dataset. Note that we report multiple final $\tilde{\Delta}$ because we evaluate AS across multiple acquisition costs.

DIME [5]. For this baseline, we extend the original model to the longitudinal setting, while still keeping its greedy feature acquisition strategy. Both the prediction network and value network of DIME share the same architecture as our `CAPE` prediction model, and hence we use the same training strategies.

DiFA [6]. Following Ghosh and Lan [6], we use a variational autoencoder with arbitrary conditioning (VAEAC) probabilistic model [8] for imputation. When using the same architecture with CAPE, DiFA performs poorly. For a stronger DiFA baseline, we use the architecture recommended in the DiFA paper. That is, both the prediction network and the feature policy model have two fully connected layers (hidden size 128) with skip connections, dropout regularization, and LeakyReLU activation function.

ACO [29]. We use the same prediction network and acquisition costs as CAPE (see Appx. C.3 for details). For the nearest neighbor search, we use 10 neighbors across all datasets. To extend ACO to the longitudinal setting, the method selects the earliest available subset of features and then replans based on the acquired features.

C.3 CAPE Implementation

Prediction Network. We use an MLP architecture for CAPE's prediction network. Specifically, the network consists of two hidden layers with ReLU. Each hidden layer contains 10 units for the synthetic and ADNI datasets. For WOMAC, KLG, and ILIADD, we use a hidden unit of 32. Moreover, we apply random dropout to the input during training of the predictor.

For the embedding network, we use four hidden layers with ReLU. Each hidden layer contains 32 units for the synthetic and ADNI datasets. For WOMAC, KLG, and ILIADD, we use a hidden unit of 64.

We select the hyperparameter α in Eq. (1) to vary the performance-cost tradeoff. For the synthetic dataset, we use $\alpha \in \{0.0009, 0.002, 0.006, 0.008, 0.01, 0.02, 0.03\}$. For ADNI, we use $\alpha \in \{0.0007, 0.001, 0.003, 0.006, 0.01, 0.05, 0.07\}$. For WOMAC of OAI, we use $\alpha \in \{0.00075, 0.0025, 0.005, 0.006, 0.009\}$. For KLG of OAI, we use $\alpha \in \{0.00075, 0.001, 0.0025, 0.005, 0.006\}$. For ILIADD, we use $\alpha \in \{0.1, 0.07, 0.03, 0.01, 0.007, 0.003, 0.001, 0.0007, 0.0003, 0.0001, 7 \times 10^{-5}, 3 \times 10^{-5}, 10^{-5}, 3 \times 10^{-6}, 10^{-6}\}$.

Role of the budget parameter α. α specifies the desired cost–benefit operating point; it is not a test-time threshold. As in many prior cost-sensitive AFA methods—including RAS, AS, ASAC, and ACO—a separate policy is trained for each cost–benefit tradeoff. Since α affects the top-r acquisition-utility profiles $\zeta_{i,o}$, learned embedding g_ϕ, and cached plans $\hat{v}^*_{k,t}$, our implementation trains a separate embedding and constructs a separate cache for each α.

CAPE's "adaptability" refers to within-trajectory replanning, not adaptation to arbitrary cost preferences using a single cache. After each new acquisition step, CAPE re-embeds the updated partial state (x_o, o, t) using g_ϕ and may retrieve a different cached plan $\hat{v}^*_{k,t}$ based on the realized trajectory, rather than committing to the complete plan selected previously. Accordingly, each point on the reported frontier corresponds to an α-specific policy for both CAPE and the cost-sensitive baselines.

C.4 Hardware and Timing

All experiments were conducted using a single NVIDIA L40S GPU and an Intel Xeon Gold 6526Y CPU. To prevent inaccuracies from asynchronous CUDA execution, device synchronization was enforced before and after all timed regions, with durations recorded via `time.perf_counter()`.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction do include the claims made in the paper. Please refer to our Abstract and Introduction sections (Sec. 1).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Please refer to Appendix A.2 for the discussion on limitations.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Please refer to Appendix A.2 for assumptions of our methods and the Appendix A.6 for the assumptions/proof.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide detailed information needed to reproduce the experimental results in Sec. 4 and Appendix C, which covers details for the experimental setup. Moreover, we also provide a detailed pseudocode for the deployment (Algorithm 1).

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Code is publicly available at `https://github.com/dzungdinhh/CAPE`. The healthcare datasets used in our paper can be downloaded upon request from the author of the data.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We provide details of the experimental settings and how to (and what we selected) for the hyperparameter of our method and baselines in Sec. 4 and Appendix C.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Please refer to Fig.3 and Table 1 for the error bars. We report the standard error computed over 5 random seeds to reflect variability due to different random initializations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Please refer to Appendix C.4 for the hardwares used in the papers, and Appendix A.3 and Sec. 4 for the running time.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Please refer to Appendix A.1 where we mention the broader impacts of our work.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have cited the related works and provided details about the codebases used for the baselines and the datasets used in our work; please refer to Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Our synthetic dataset will be fully documented (with a README covering setup, dependencies, and usage) and released under the MIT license upon publication. Please see `https://github.com/dzungdinhh/CAPE` for the code to recreate the synthetic dataset.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: We only use LLM for grammar checking/editing

    Guidelines:

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