UTILITY-AUGMENTED TRANSFORMERS: FEEDBACK-CONDITIONED RETRIEVAL FOR SEQUENTIAL DECISION MAKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning has advanced sequential decision-making. Within this framework, conditional sequence models such as Decision Transformer achieve competitive performance by predicting actions from trajectory histories and target returns-to-go. These models typically incorporate actions and rewards as input features without directly conditioning attention projections on feedback, which may limit their effectiveness on feedback-informative tasks, where identical observations can require different actions depending on what previous actions achieved. This paper examines when input-level feedback conditioning is sufficient and when it is not. We introduce the Utility-Augmented Transformer (UAT), in which shifted action–reward feedback modulates queries, keys, and values, and an exponentially smoothed utility state supplies persistent conditioning, with zero initialization reducing UAT exactly to its vanilla Transformer backbone. The additive pathway provably contains the linear input-conditioning subclass, while the multiplicative pathway realizes a third-order query-key-feedback logit that input-level injection cannot represent, yielding a persistent approximation floor for the latter under bounded stylized designs. The utility state also admits a compressed belief-state reading under explicit sufficiency conditions. Extensive experiments show substantial gains of UAT on controlled tasks with informative, timely feedback and competitive performance across standard offline RL domains, under comparable parameter budgets and without increasing asymptotic attention complexity.

1 INTRODUCTION

Offline reinforcement learning (RL) enables agents to learn effective policies from fixed datasets and has been widely applied to sequential decision-making. Within this framework, sequence modeling approaches such as Decision Transformer (DT) predict actions conditioned on trajectory history and desired returns (Chen et al., 2021; Bhargava et al., 2024). Although these models receive reward information, the central question remains whether this information changes how they use their past. In a non-stationary environment, two histories can look identical while carrying different evidence about the hidden task state: the same chart can follow treatment success or failure, while the same room may conceal a shifted goal. An observation-only policy cannot distinguish these histories. Adding rewards to the token stream helps, but leaves a specific question: should feedback merely be another token to retrieve, or should it determine which past observations and actions are retrieved?

This distinction matters since sequence-model decision systems have largely followed the first route to incorporate feedback at the input level. Decision Transformer (DT) conditions action generation on returns, states, and actions through linear token embeddings (Chen et al., 2021). RvS shows that carefully conditioned supervised learning is a strong offline-RL baseline (Emmons et al., 2022). Subsequent work addresses stitching by changing the effective history length (Wu et al., 2023) or by relabeling targets with value estimates (Yamagata et al., 2023). These methods are effective when a return token is a sufficient condition for action selection. They do not, however, directly modify the attention kernel that decides which historical token should influence the current decision.

We study a complementary mechanism: feedback-conditioned retrieval, and propose the Utility-Augmented Transformer (UAT). UAT computes a utility feature from the previous action and its

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

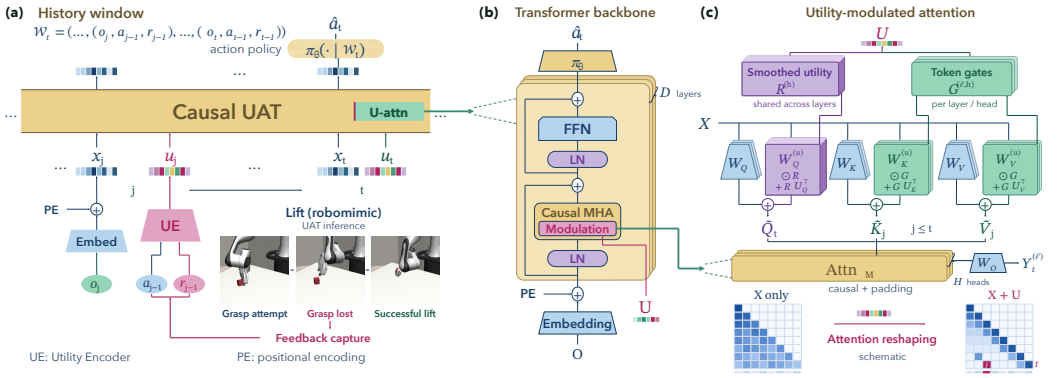


Figure 1: UAT overview. **(a)** Action prediction from observation histories and shifted action–reward embeddings (robomimic Lift). **(b)** Causal Transformer with utility-modulated attention. **(c)** Smoothed utility and token gates modulate queries and keys/values, respectively

resulting reward, converts it into token gates and a persistent utility state, and uses these signals to modulate the projections that define attention queries, keys, and values. It allows feedback to alter the similarity computation itself: two keys with the same observation embedding can receive different utility-dependent keys, and the current query can be shifted by accumulated feedback evidence.

We evaluate UAT against diverse baselines on controlled tasks and standard offline RL benchmarks. Interaction-order and feedback-corruption analyses examine its mechanism and boundaries, while ablation, efficiency, and sensitivity studies assess component contributions, computational cost, and robustness. Code is available online.[1] Overall, our contributions are threefold:

(1) **UAT architecture for feedback-blindness.** We formalize *feedback-informative* sequential decision tasks, and prove observation-only Transformers are *feedback-blind*. We propose Utility-Augmented Transformer (UAT), whose query, key, value projections are utility-affine functions of shifted action–reward states, allowing feedback to configure which past tokens are retrieved. Zero initialization makes UAT a parameter-clean extension of its vanilla Transformer backbone.

(2) **Expressiveness and approximation theory.** We prove an exact containment that the additive pathway of UAT contains the whole class of linear input-level conditioning, while its multiplicative pathway realizes a third-order query–key–feedback logit that no fixed linear embedding can represent. Under a bounded stylized design, depth-one input-level injection has an exact L_2 approximation floor, the trilinear class enjoys vanishing estimation error, and the utility state is a compressed belief state under an explicit sufficiency condition.

(3) **Empirical validation and boundary characterization.** Multi-seed experiments show substantial gains on controlled feedback-informative tasks and competitive performance on real benchmarks D4RL and robomimic, under comparable parameter budgets and without increasing asymptotic attention complexity. Mechanism probes confirm the predicted third-order interaction, and controlled corruption of reward feedback locates the advantage boundary of UAT.

2 Related Work

Offline Reinforcement Learning Methods. Offline RL learns policies from fixed datasets through approaches such as value learning and supervised trajectory modeling. Q-learning (Watkins & Dayan, 1992) and its deep variant DQN (Mnih et al., 2015) provide the basis for value learning, while CQL (Kumar et al., 2020) and IQL (Kostrikov et al., 2022) address the offline setting. Sequence modeling frames offline RL as conditional trajectory prediction (David et al., 2023). DT conditions a causal Transformer on return-to-go (Chen et al., 2021), while Trajectory Transformer combines prediction with planning (Janner et al., 2021). RvS achieves competitive results with MLPs given suitable conditioning, capacity, and regularization (Emmons et al., 2022). Elastic DT and Q-learning DT address trajectory stitching through adaptive context selection (Wu et al., 2023) and value-based

[1] Anonymous code: https://anonymous.4open.science/r/utility_augmented_transformer-1566.

return relabeling (Yamagata et al., 2023), respectively. UAT directly conditions query/key/value projections on feedback, complementing context selection and return relabeling.

Memory, adaptation, and partial observability. Perceptual aliasing (Whitehead & Ballard, 1991) motivates the use of history and belief states in RL (Kaelbling et al., 1998). Relevant approaches include recurrent memory in GTrXL (Parisotto et al., 2020), belief inference in VariBAD (Zintgraf et al., 2021), context-aware dynamics in CaDM (Lee et al., 2020), and in-context learning in AD (Laskin et al., 2023). Beyond RL, TENT (Wang et al., 2021) and CoTTA (Wang et al., 2022) adapt predictors to distribution shifts at test time. UAT uses a feedback-derived utility state to modulate retrieval. Section 5 specifies when this state admits a compressed belief-state interpretation.

Generative conditioning and multiplicative interactions. Feature-wise modulation (Perez et al., 2018), adaptive normalization (Huang & Belongie, 2017), spatially adaptive conditioning (Park et al., 2019), and hypernetworks (Ha et al., 2017) show that external signals can configure computation. Multiplicative interactions can be more expressive than additive feature concatenation in generic networks (Jayakumar et al., 2020), and depth can provide large efficiency advantages for compositional functions (Eldan & Shamir, 2016). Our analysis specializes this idea to attention logits: input-level injection shares a bilinear query–key core, whereas utility-gated projections induce a third-order interaction. Sequence models' universality does not by itself explain this difference, since it hides width, depth, sample, and optimization costs (Hornik et al., 1989; Yun et al., 2020).

3 Problem formulation

Consider a finite-horizon non-stationary POMDP $(\mathcal{S}, \mathcal{O}, \mathcal{A}, P, T, \mathcal{R}, \gamma, H)$ (Kaelbling et al., 1998; Cheung et al., 2020). At step t, the agent observes $o_t \in \mathcal{O}$, takes $a_t \in \mathcal{A}$, and receives $r_t \in \mathbb{R}$. Its visible history before choosing a_t is

$$H_t = (o_1, a_1, r_1, \ldots, o_{t-1}, a_{t-1}, r_{t-1}, o_t). \tag{1}$$

The latent state s_t is not observed. Let $b_t(s) = P(s_t = s \mid H_t)$ be the posterior belief. A score-based policy maps a history to a vector over actions, $F(H_t) \in \mathbb{R}^{|\mathcal{A}|}$.

Definition 1 (Perceptually aliased histories)**.** *Two histories H_t and H'_t are perceptually aliased if they contain the same observation sequence but different action–reward feedback sequences. The task has aliasing if such a pair occurs with positive probability.*

Definition 2 (Feedback-informative task)**.** *A task is feedback-informative if there is an aliased pair (H_t, H'_t) with positive probability and a belief-optimal score map Φ such that*

$$\Phi(b_t) \neq \Phi(b'_t), \qquad b_t \neq b'_t. \tag{2}$$

This definition captures cases where observations alone are insufficient and feedback changes the optimal decision, exemplified by the hidden-goal DarkRoom and reward-responsive treatment tasks in our experimental evaluation (Section 6). Fully observed locomotion may exhibit less aliasing.

An input-level sequence model embeds each history element as a token containing an observation component and a feedback component. In linear case, using null symbols (a_0, r_0) at episode starts,

$$x_j^{\text{in}} = W_o o_j + W_a a_{j-1} + W_r r_{j-1} + p_j. \tag{3}$$

A standard causal attention layer with head dimension d_m then computes

$$\alpha_{tj} = \frac{\exp((W_Q x_t^{\text{in}})^\top (W_K x_j^{\text{in}})/\sqrt{d_m})}{\sum_{\ell \le t} \exp((W_Q x_t^{\text{in}})^\top (W_K x_\ell^{\text{in}})/\sqrt{d_m})}, \qquad h_t^{\text{attn}} = \sum_{j \le t} \alpha_{tj} W_V x_j^{\text{in}}. \tag{4}$$

4 Utility-Augmented Transformer

4.1 Architecture

UAT retains a causal Transformer backbone (Vaswani et al., 2017), preserving its standard embeddings, layer normalization, feed-forward networks, and residual connections (He et al., 2016), while

adding a compact feedback pathway. Let $x_j \in \mathbb{R}^d$ be the observation-token embedding and let a_0 and r_0 denote null symbols. Shifted feedback is encoded by a small MLP *utility encoder* E:

$$u_j = E\big(f_a(a_{j-1}), r_{j-1}\big) \in \mathbb{R}^{d_u}. \tag{5}$$

Here f_a is a linear embedding function. The shift ensures causality: token j cannot observe the reward produced by the action selected at step j. For layer l and head h, define a token gate

$$g_j^{(l,h)} = \sigma\big(W_s^{(l,h)} u_j\big), \qquad \sigma = \tanh \text{ in the deployed model.} \tag{6}$$

The exact representation results use an affine-gate core with $\sigma(z) = z$ and linear query readouts. A persistent utility state is maintained by exponential smoothing and reset at each window boundary; the recurrence uses window-local indices:

$$\bar{u}_0 = 0, \qquad \bar{u}_t = \lambda \bar{u}_{t-1} + (1-\lambda) u_t, \quad 0 \le \lambda < 1, \qquad \rho_t^{(h)} = f_u^{(h)}(\bar{u}_t) \in [-1,1]^{d_m}. \tag{7}$$

In the deployed model, $f_u^{(h)}$ applies a linear map followed by $\tanh$. Token gates and $W_{Q,K,V}$ vary by layer and head; $\rho_t^{(h)}$ is shared across layers. Omitting these indices, the *utility-affine projections* are

$$\tilde{q}_t = \big(W_Q + \mathrm{diag}(\rho_t) W_Q^{(u)}\big) x_t + U_Q \rho_t, \tag{8}$$

$$\tilde{k}_j = \big(W_K + \mathrm{diag}(g_j) W_K^{(u)}\big) x_j + U_K g_j, \tag{9}$$

$$\tilde{v}_j = \big(W_V + \mathrm{diag}(g_j) W_V^{(u)}\big) x_j + U_V g_j. \tag{10}$$

Attention uses $\tilde{q}_t$, $\tilde{k}_j$, and $\tilde{v}_j$ in Eq. (4). Zero-initialized *modulation terms* exactly recover the vanilla Transformer. In Eqs. (8)–(10), $U_Q \rho_t$ and $U_K g_j$ inject utility features, while diagonal products make projections utility-dependent. Unchanged backbone components map attention-head outputs h_j^{attn} to layer states $h_j^{(l)} \in \mathbb{R}^d$, with $h_j^{(0)} = x_j$. After D layers, from $h_t^{(D)}$ an action head predicts:

$$z_t = W_\pi h_t^{(D)} + b_\pi, \qquad \begin{cases} \pi_\theta(\cdot \mid \mathcal{W}_t) = \mathrm{softmax}(z_t), & \text{discrete actions,} \\ \hat{a}_t = \tanh(z_t), & \text{continuous actions.} \end{cases} \tag{11}$$

4.2 Optimization Objective

Following DT (Chen et al., 2021), we train on B fixed offline trajectories of lengths $T_b \le T$, padded to T, supervising all valid positions in trailing windows $\mathcal{W}_{b,t}$ of at most L tokens. Local index i maps to trajectory time $j(t,i) = t - L + i$, with causal prefix $\mathcal{W}_{b,t}^{\le i}$. Define $m_{b,t,i} = \mathbf{I}[1 \le j(t,i) \le t \le T_b]$ and $N = \sum_{b,t,i} m_{b,t,i}$. The supervised loss is

$$\mathcal{L}_{\mathrm{sup}} = -\frac{1}{N} \sum_{b=1}^{B} \sum_{t=1}^{T} \sum_{\substack{1 \le i \le L \\ m_{b,t,i}=1}} \log \pi_\theta(a_{b,j(t,i)} \mid \mathcal{W}_{b,t}^{\le i}). \tag{12}$$

We additionally penalize utility-gate magnitudes. Let $i \in \{1, \dots, L\}$ index positions within a window, $m_{b,t,i}$ indicate a position, $N_{\mathrm{gate}} = \sum_{b,t,i} m_{b,t,i} = N$. Averaging over D layers and H heads gives

$$\mathcal{L}_{\mathrm{gate}} = \frac{1}{N_{\mathrm{gate}} D H} \sum_{b,t,i} m_{b,t,i} \sum_{l=1}^{D} \sum_{h=1}^{H} \left(\|\rho_{b,t,i}^{(h)}\|_2^2 + 2 \|g_{b,t,i}^{(l,h)}\|_2^2 \right). \tag{13}$$

Here $\rho_{b,t,i}$ is computed within the window ending at t. The factor of two accounts for the token gate shared by the key and value pathways. The complete loss function is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{sup}} + \beta \, \mathcal{L}_{\mathrm{gate}}, \qquad \beta \ge 0. \tag{14}$$

4.3 Implementation and Complexity

Parallel training. Following standard offline RL practice, we sample windows from logged trajectories, left-pad shorter windows, and zero padded utility features. For each layer and head, $\mathbf{X}$, $\mathbf{U}$, $\mathbf{G}$, and $\mathbf{R}$ stack token representations, shifted utilities, token gates, and query gates. Algorithm 1

Algorithm 1 UAT: one training step on a batch of windows.

Input: B length-L windows: observations $\mathbf{O}$, shifted feedback $(\mathbf{A}, \mathbf{r})$, action targets $\{a_{b,i}\}$, padding mask $\mathbf{m}$; parameters θ, weight β
Output: Updated parameters θ

$\mathbf{X} \leftarrow \mathrm{Embed}(\mathbf{O}, \text{timesteps})$
$\mathbf{U} \leftarrow \mathbf{m} \odot E(f_a(\mathbf{A}), \mathbf{r})$
$\overline{\mathbf{U}} \leftarrow \mathbf{C}_\lambda \mathbf{U}$; $\mathbf{R}^{(h)} \leftarrow f_u^{(h)}(\overline{\mathbf{U}})$ for all heads h ▷ Eq. (15)
for $l = 1, \ldots, D$ **do**
  For all heads h in parallel: $\mathbf{G}^{(l,h)} \leftarrow \tanh\big(\mathbf{U}(W_s^{(l,h)})^\top\big)$
  Compute $(\widetilde{\mathbf{Q}}^{(h)}, \widetilde{\mathbf{K}}^{(h)}, \widetilde{\mathbf{V}}^{(h)})$ for all heads from $\mathbf{X}, \mathbf{G}^{(l,h)}, \mathbf{R}^{(h)}$ via Eqs. (16)–(18)
  $\mathbf{Y} \leftarrow \mathrm{MHA}_{\mathbf{M}}\Big(\{(\widetilde{\mathbf{Q}}^{(h)}, \widetilde{\mathbf{K}}^{(h)}, \widetilde{\mathbf{V}}^{(h)})\}_{h=1}^{H}\Big)$ ▷ Eq. (19) ▷ Causal/padding masks
  Update $\mathbf{X}$ from $\mathbf{Y}$ using residual connections, normalization, and FFN.
end for
$N \leftarrow \sum_{b,i} m_{b,i}$; $\pi_{b,i} \leftarrow \mathrm{ActionHead}(\mathbf{X}_{b,i,:})$ for $m_{b,i} = 1$
$\mathcal{L}_{\mathrm{sup}} \leftarrow -\frac{1}{N} \sum_{\substack{b,i \\ m_{b,i}=1}} \log \pi_{b,i}(a_{b,i})$
$\mathcal{L}_{\mathrm{gate}} \leftarrow \frac{1}{NDH} \sum_{b,i} m_{b,i} \sum_{l=1}^{D} \sum_{h=1}^{H} \Big(\|\mathbf{R}_{b,i,:}^{(h)}\|_2^2 + 2\|\mathbf{G}_{b,i,:}^{(l,h)}\|_2^2 \Big)$
$\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{sup}} + \beta \mathcal{L}_{\mathrm{gate}}$; $\theta \leftarrow \mathrm{AdamW}(\theta, \nabla_\theta \mathcal{L})$ (Loshchilov & Hutter, 2019)
return θ

initializes $\mathbf{X}$ with observation embeddings and updates it in place to $h_j^{(l)}$. Zero initialization at each window boundary gives

$$\overline{\mathbf{U}} = \mathbf{C}_\lambda \mathbf{U}, \qquad (\mathbf{C}_\lambda)_{ij} = \begin{cases} (1-\lambda)\lambda^{i-j}, & j \le i, \\ 0, & j > i, \end{cases} \tag{15}$$

with $\lambda^0 = 1$, including at $\lambda = 0$. Suppressing layer and head indices, the parallel projections are

$$\widetilde{\mathbf{Q}} = \mathbf{X} W_Q^\top + \mathbf{R} \odot \big(\mathbf{X}(W_Q^{(u)})^\top\big) + \mathbf{R} U_Q^\top, \tag{16}$$

$$\widetilde{\mathbf{K}} = \mathbf{X} W_K^\top + \mathbf{G} \odot \big(\mathbf{X}(W_K^{(u)})^\top\big) + \mathbf{G} U_K^\top, \tag{17}$$

$$\widetilde{\mathbf{V}} = \mathbf{X} W_V^\top + \mathbf{G} \odot \big(\mathbf{X}(W_V^{(u)})^\top\big) + \mathbf{G} U_V^\top. \tag{18}$$

Restoring head indices, the multi-head attention output is

$$\mathbf{Y} = \mathrm{Concat}_{h=1}^{H} \left[\mathrm{softmax}_{\mathbf{M}} \left(\frac{\widetilde{\mathbf{Q}}^{(h)} (\widetilde{\mathbf{K}}^{(h)})^\top}{\sqrt{d_m}} \right) \widetilde{\mathbf{V}}^{(h)} \right] W_O. \tag{19}$$

$W_O \in \mathbb{R}^{d \times d}$ is the output projection. Algorithm 1 summarizes the training process; detailed definitions appear in Appendix B.

Autoregressive inference. UAT predicts actions step by step from the latest L-token window, incorporating observed action–reward feedback while keeping model parameters fixed.

Complexity and parameters. Under parallel training, UAT retains the Transformer's asymptotic complexity $O(BD(Ld^2 + L^2 d))$. Its utility pathway adds $P_{\mathrm{enc}} + (D+1)dd_u + 3Dd^2(1 + 1/H)$ parameters, independent of B and L; derivations and width assumptions appear in Appendix B.2.

5 Theory

The theory examines when feedback matters and the expressive gains of utility-conditioned retrieval through three analyses: (1) separating observation-only from feedback-aware policies; (2) comparing where feedback enters feedback-aware architectures; and (3) relating utility states to belief compression without assuming that arbitrary learned vectors represent posterior beliefs.

5.1 Feedback-blindness and exact containment

Proposition 1 (Observation-only Transformers are feedback-blind)**.** *Let $\mathcal{F}_{\mathrm{TF}}^{\mathrm{obs}}$ be the class of causal Transformers whose tokens and all intermediate features are functions only of $(o_{1:t})$ and position indices. For every aliased pair (H_t, H_t') and every $F \in \mathcal{F}_{\mathrm{TF}}^{\mathrm{obs}}$,*

$$F(H_t) = F(H_t'). \tag{20}$$

It separates observation-only from feedback-aware models, but does not divide UAT from input-level injection. Increasing depth and width cannot recover feedback variables that never enter the function.

Proposition 2 (Exact zero-gate degradation)**.** *If the utility pathways in Eqs.* (8)–(10) *are set to zero, then UAT is exactly the corresponding vanilla Transformer. Zero initialization makes this equality hold at initialization without changing the backbone parameterization.*

Proposition 2 gives UAT a safe identity direction: utility conditioning is an extension rather than a replacement for attention. It also makes the comparison with a vanilla backbone parameter-clean, because the backbone weights have the same meaning in both models.

Lemma 1 (Bias pathway as augmented-token attention)**.** *Set* $W_Q^{(u)} = W_K^{(u)} = W_V^{(u)} = 0$ *and define three-part tokens*

$$y_j = [x_j;\, g_j;\, \rho_j] \in \mathbb{R}^{d+2d_m}. \tag{21}$$

Then the remaining UAT projections are exactly

$$\tilde{q}_t = [W_Q, 0, U_Q]y_t, \qquad \tilde{k}_j = [W_K, U_K, 0]y_j, \qquad \tilde{v}_j = [W_V, U_V, 0]y_j. \tag{22}$$

Thus the bias-only subclass is a standard Transformer on the augmented token sequence $(y_1, \ldots, y_T)$.

The query side must use ρ_t while the key and value sides use g_j. Equation (21) therefore includes both utility vectors. Calling the token $[x_j; g_j]$ alone would not represent the query pathway.

Assumption 1 (Sufficient utility readout)**.** *The encoder and readouts can realize the feedback feature* ϕ_j *required by the target for the representation. In the simple case,* g_j *and* ρ_j *contain the raw shifted feedback* (a_{j-1}, r_{j-1}) *after padding. More generally, a statistic* $Z_t = (g_{1:t}, \rho_{1:t})$ *is* ϵ*-sufficient for the belief if there is a decoder* ψ *with* $\|\psi(Z_t) - b_t\|_1 \le \epsilon$ *on the histories of interest.*

Theorem 1 (Exact containment and strict extension of linear input conditioning)**.** *Consider one causal attention layer. Let* $\mathcal{F}_{\text{in}}^{(1)}$ *be the class obtained from linear token embeddings* $x_j^{\text{in}} = A_o o_j + A_f \phi_j + p_j$*, followed by fixed linear Q/K/V projections, with* ϕ_j *in the realizable gate/readout class. Under Assumption 1, the bias-only UAT subclass contains this matched class exactly when* $\lambda = 0$*. For the affine-gate core, every linear* ϕ_j *is realizable. In addition, the multiplicative pathway realizes logits*

$$\ell_{\text{UAT}}(x_t, x_j, u_j) = x_t^\top W_Q^\top \operatorname{diag}(g_j) W_K^{(u)} x_j = \sum_{m=1}^{d_m} g_{j,m} (w_{Q,m}^\top x_t)(w_{K,m}^{(u)\top} x_j), \tag{23}$$

which are third-order in (x_t, x_j, u_j) *when* g_j *is affine in* u_j*. Such a logit is not representable as any bilinear function of fixed linear embeddings of* (o_t, ϕ_t) *and* (o_j, ϕ_j)*.*

With exactly sufficient feedback, UAT's bias pathway inherits causal Transformer universality for continuous targets on compact, finite-horizon histories (Theorem 2, Appendix A.2). Universality guarantees neither greater parameter efficiency nor easier learning.

5.2 Interaction order, width, and depth

Definition 3 (K-channel gated-retrieval family)**.** *For linear gates* $s_k : \mathbb{R}^{d_u} \to \mathbb{R}$ *and bounded vectors* $q_k, w_k \in \mathbb{R}^{d_x}$*, define*

$$\mathcal{L}_K^{(3)} = \left\{ \ell(x, y, u) = \sum_{k=1}^{K} s_k(u)(q_k^\top x)(w_k^\top y) : \|q_k\|, \|w_k\| \le B \right\}. \tag{24}$$

Assumption 2 (Centered symmetric design)**.** *For the single-channel target* $\ell^*(x, y, u) = u(q^\top x)(w^\top y)$*, the variables* x, y, u *are mutually independent, centered, symmetric, and have finite fourth moments. Write* $\Sigma_x = \mathbb{E}[xx^\top]$*,* $\Sigma_y = \mathbb{E}[yy^\top]$*, and* $\sigma_u^2 = \mathbb{E}[u^2]$*.*

Theorem 3 (Interaction-order separation)**.** *Under Eq.* (24) *and Assumption 2:*

(a) ***Exact depth-one input-level floor.*** *Causal attention logit from fixed linear embeddings of* $[x; u]$ *and* $[y; u]$ *is a polynomial of total degree at most two. Its best* L_2 *approximation of* ℓ^* *has error*

$$\inf_{\ell : \deg(\ell) \le 2} \mathbb{E}[(\ell - \ell^*)^2] = \mathbb{E}[(\ell^*)^2] = \sigma_u^2 \, (q^\top \Sigma_x q) \, (w^\top \Sigma_y w). \tag{25}$$

Table 1: Controlled-task results (mean±std; 5 seeds). Bold marks the best mean.

Family	Model	DarkRoom						Sepsis		
		Gradual		Cyclical		Abrupt		Treatment		
		Acc.↑	Return↑	Acc.↑	Return↑	Acc.↑	Return↑	Acc.↑	Post-Acc.↑	Return↑
Behavioral cloning	TF	0.265 ±0.013	5.812 ±2.521	0.268 ±0.008	3.597 ±1.073	0.269 ±0.009	2.874 ±0.639	0.139 ±0.035	0.130 ±0.087	-0.797 ±0.103
Belief based	VariBAD	0.307 ±0.013	15.906 ±2.713	0.314 ±0.021	5.194 ±1.865	0.281 ±0.015	3.952 ±1.454	0.052 ±0.009	0.058 ±0.020	-1.157 ±0.134
Classical RL	CQL	0.608 ±0.032	83.472 ±5.395	0.489 ±0.029	57.340 ±6.484	0.491 ±0.023	59.070 ±4.315	0.264 ±0.037	0.204 ±0.026	-0.190 ±0.520
	IQL	0.310 ±0.118	33.902 ±12.928	0.247 ±0.102	18.694 ±11.833	0.257 ±0.150	21.628 ±20.702	0.083 ±0.017	0.096 ±0.022	-0.859 ±0.112
	DQN	0.297 ±0.067	25.294 ±7.932	0.278 ±0.044	19.954 ±5.888	0.300 ±0.034	24.034 ±6.143	0.119 ±0.012	0.041 ±0.027	-6.269 ±1.414
Memory and adaptation	GTrXL	0.339 ±0.015	8.932 ±2.551	0.344 ±0.014	5.364 ±1.346	0.348 ±0.011	5.902 ±1.418	0.243 ±0.019	0.222 ±0.030	-1.701 ±0.362
	TENT	0.264 ±0.006	5.878 ±1.470	0.267 ±0.009	3.132 ±1.377	0.277 ±0.003	3.928 ±0.620	0.182 ±0.011	0.148 ±0.015	-1.454 ±0.091
	CoTTA	0.196 ±0.031	1.938 ±1.165	0.150 ±0.045	-0.178 ±0.818	0.164 ±0.047	0.202 ±1.357	0.087 ±0.023	0.092 ±0.028	-2.124 ±0.472
Sequence modeling	DT	0.351 ±0.014	10.338 ±2.342	0.359 ±0.003	6.938 ±0.896	0.354 ±0.005	8.116 ±0.873	0.230 ±0.023	0.206 ±0.011	-1.777 ±0.245
	EDT	0.332 ±0.032	8.038 ±3.578	0.379 ±0.020	2.766 ±1.137	0.333 ±0.014	1.864 ±1.239	0.220 ±0.013	0.187 ±0.017	-2.175 ±0.260
	QDT	0.271 ±0.008	6.446 ±1.798	0.286 ±0.009	5.026 ±0.745	0.267 ±0.015	5.216 ±0.709	0.216 ±0.033	0.192 ±0.021	-1.979 ±0.214
Ours	**UAT**	**0.652** ±0.014	**88.182** ±3.146	**0.536** ±0.021	**59.536** ±5.295	**0.533** ±0.005	**59.794** ±1.740	**0.401** ±0.031	**0.290** ±0.025	**2.027** ±0.599

(b) ***Width lower bound with nonlinear key features.*** *If the channel vectors $\{q_k\}_{k=1}^{K}$ are linearly independent and the vectors $\{w_k \otimes s_k\}_{k=1}^{K} \subset \mathbb{R}^{d_x \times d_u}$ are linearly independent, then any finite factorization $\ell(x, y, u) = \sum_{m=1}^{M} \psi_m(x)\, \phi_m(y, u)$ of the K-channel target requires $M \geq K$.*

(c) ***Depth condition.*** *If an intermediate map can form exact polynomial products, two composed bilinear layers suffice. With standard ReLU or GELU feed-forward maps, the corresponding statement is approximation on compact sets, not automatic exact representation.*

Parts (a)–(c) of Theorem 3 show the real trade-off: input-level injection must pay token width or nonlinear depth to synthesize the interaction that a gated key supplies directly.

5.3 Finite-sample implication

Under bounded i.i.d. sampling and the centered symmetric design, the bilinear input-level class retains the approximation floor in Eq. (25), while empirical risk minimization over a bounded trilinear class containing the target has vanishing excess risk. Theorem 4 gives the finite-sample comparison, and its full assumptions, statement, and proof appear in Appendix A.4.

5.4 Utility state as compressed belief evidence

Assumption 4 (Lipschitz belief-optimal scores)**.** *The belief-optimal score map $\Phi : \Delta(\mathcal{S}) \to \mathbb{R}^{|\mathcal{A}|}$ is L_Φ-Lipschitz with respect to total variation on beliefs and Euclidean distance on scores.*

Proposition 3 (Sufficient utility statistics are compressed beliefs)**.** *Let Z_t be an ϵ-sufficient utility statistic for b_t as in Assumption 1, and let Assumption 4 hold. Then $\hat{F}(H_t) = \Phi(\psi(Z_t))$ satisfies*

$$\|\hat{F}(H_t) - \Phi(b_t)\| \leq L_\Phi \epsilon. \tag{26}$$

Conversely, if Z_t depends only on observations and (H_t, H'_t) is an aliased pair with score separation $\delta = \|\Phi(b_t) - \Phi(b'_t)\|$, then every observation-only predictor assigns the same score to both histories, so at least one history has error at least $\delta/2$.

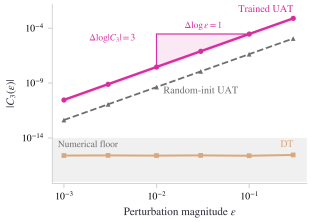
Proposition 3 connects the architecture to classical POMDP theory while preserving a falsifiable empirical question: reward corruption, delay, and shuffling should degrade performance precisely when that information is decision-relevant.

6 Experiments

We evaluate UAT with all baselines on controlled feedback-informative tasks DarkRoom and sepsis (Komorowski et al., 2018; Johnson et al., 2016), and with DT, CQL, and Vanilla Transformer (hereafter "TF") on standard offline RL benchmarks D4RL (Fu et al., 2020) and robomimic (Mandlekar et al., 2022). We isolate utility-conditioned retrieval by comparing architectures with matched

Table 2: Cross-domain results (mean±std; 5 seeds). Bold marks the best mean.

Model	D4RL (↑)			robomimic (↑)			
	Locomotion			Can		Lift	
	M	M-R	M-E	Dense	Sparse	Dense	Sparse
TF	50.399 ±2.001	25.613 ±1.698	65.496 ±7.312	19.8 ±4.1	19.8 ±4.1	69.2 ±5.6	69.2 ±5.6
CQL	47.868 ±8.985	9.300 ±1.197	65.262 ±8.264	0.0 ±0.0	0.0 ±0.0	42.2 ±4.9	50.0 ±4.6
DT	56.395 ±1.100	54.318 ±2.481	86.833 ±6.886	14.0 ±3.0	21.6 ±5.1	88.6 ±5.0	87.0 ±2.1
UAT	**56.975** ±1.899	**54.619** ±3.289	**86.923** ±4.576	**23.2** ±6.1	**35.8** ±9.0	**91.4** ±1.1	**88.0** ±4.7



Interaction order

Model	Slope	c	$\lvert C_3\rvert/\lvert F\rvert$
UAT-1L	3.00 ± 0.00	2.9e-02	5e-06
UAT-1L rand. init	3.00 ± 0.00	4.2e-04	2e-06
DT-1L	-0.01 ± 0.01	2.4e-16	7e-17

Decision quality

Model	Return	Plan-acc	NLL
UAT-1L	1.53 ± 0.40	0.38 ± 0.02	1.16 ± 0.02
DT-1L	-1.83 ± 0.71	0.18 ± 0.05	1.55 ± 0.01

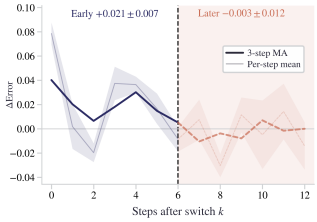


Figure 2: Depth-one sepsis controls. **Left:** Third-order retrieval responses. **Middle:** Interaction statistics (top) and decision metrics (bottom). **Right:** Post-switch oracle-error reduction on training logs, with early/later summaries. Bands and ± denote one standard deviation over five seeds.

feedback information and similar parameter counts. Appendix C provides all details about benchmarks, baselines, metrics, hyperparameters, and parameter counts.

6.1 Main Experimental Results

Main experimental results in tables 1 and 2 highlight several points. **(1)** In general, UAT outperforms all baselines, with substantial gains on controlled DarkRoom and sepsis tasks and smaller but consistent improvements over DT on D4RL and robomimic, supporting its utility encoding and attention modulation. **(2)** UAT's gains over input-conditioned DT and belief-based VariBAD support the value of feedback-conditioned retrieval, especially on controlled tasks where hidden regime changes make past outcomes essential for decision-making. **(3)** Further, DT and UAT generally outperform Vanilla-TF and remain competitive across benchmarks, supporting reward conditioning alongside action history. CQL excels on controlled tasks but varies across domains, possibly reflecting interactions between sparse rewards and conservative value estimation (Bhargava et al., 2024). Additional D4RL results averaged across dataset levels show that UAT clearly leads on HalfCheetah and Hopper but slightly trails DT on Walker2d (Table 7, Appendix E.1); comparisons at larger model scales appear in Appendix E.2, where UAT continues to outperform DT.

6.2 Mechanism Analysis

Guided by third-order interaction theory, we examine UAT's utility pathways, their advantages over DT's input-level conditioning, and the boundaries of these gains. The setup details of the experiments in this section appear in Appendix D.2 and D.3. **(1) How UAT uses feedback information.** With matched feedback and capacity, the depth-one sepsis probe $|C_3(\varepsilon)|$ shows cubic scaling for trained and randomly initialized UAT but only round-off for DT, identifying an architectural interaction (Figure 2). Oracle-error reductions on unedited training logs concentrate just after switches; separate rollouts also favor UAT. **(2) Boundary Analysis.** On cyclical DarkRoom, UAT leads DT with clean feedback, but increasing noise or delay gradually shrinks this advantage; both approach Vanilla-TF under strong reward noise or full reward dropout (Figure 3). These findings reinforce that feedback and its integration matter, suggesting UAT benefits from moderately frequent switches with informative, timely feedback. Additional mechanistic visualizations appear in Appendix E.3.

6.3 Ablation Experiments

Our ablations show that full UAT achieves the best mean on every metric across both benchmarks (Table 9). On sepsis, every ablation sharply reduces accuracy and turns positive returns negative,

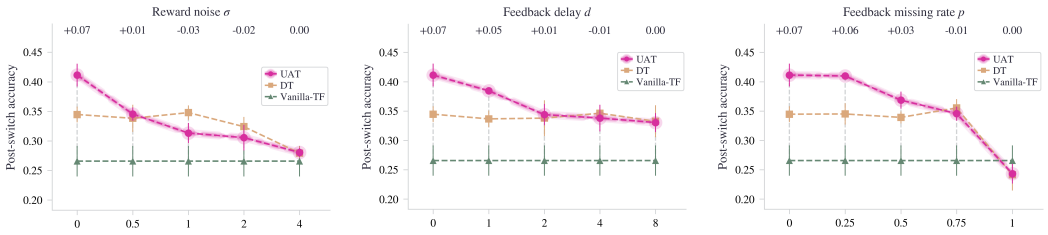


Figure 3: Cyclical DarkRoom accuracy over ≤ 5 decisions starting at each switch-marked transition (mean ± 1 std; five seeds). **Left:** Gaussian reward noise σ in training-log std units; **middle:** (a_s, r_s) delayed to $s + d$; **right:** independent dropout p, with missing rewards represented as zeros.

Table 3: Throughput with default settings. Training: valid targets/s over epochs 3–5 (200 updates/epoch). Inference: actions/s from one full-context, batch-one measurement.

Model	DarkRoom		Sepsis		D4RL		robomimic	
	Train↑	Infer↑	Train↑	Infer↑	Train↑	Infer↑	Train↑	Infer↑
Vanilla-TF	388.7k	486	236.1k	483	87.7k	501	147.4k	490
DT	388.4k	487	235.8k	480	87.6k	503	149.4k	504
UAT	298.7k	243	183.0k	238	78.1k	297	121.0k	269

supporting all components' contributions. On the more complex D4RL benchmark, effects are more varied: removing rewards, query modulation, or persistent aggregation causes larger drops, while input-only feedback nearly matches UAT; nevertheless, every ablation lowers mean return. Settings and detailed results appear in Appendices D.5 and E.4, respectively.

6.4 Efficiency and Sensitivity

We also assess UAT's efficiency and sensitivity. **(1) Efficiency.** With matched hardware, batch size, and decision-context length within each benchmark, UAT has 11–23% lower training throughput and 41–51% lower inference throughput than TF and DT, while sustaining 238–297 actions per second at batch size one (Table 3). **(2) Sensitivity.** On cyclical DarkRoom, overall and post-shift accuracy remain broadly stable across tested λ and β ranges. Context length has larger effects: moderate windows perform consistently, whereas longer window ($L = 160$) reduces performances (Figure 4). These results favor task-specific context selection, emphasized by Decision S4 (David et al., 2023); additional D4RL sensitivity results appear in Appendix E.5, where effects are more nuanced.

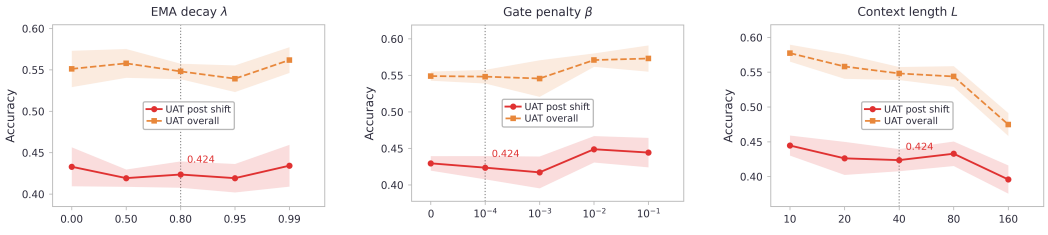


Figure 4: Sensitivity to λ, β, and L on cyclical DarkRoom.

7 Discussion and Conclusion

This paper distinguishes two uses of feedback in sequence decision models: encoding it as input or using it to configure retrieval. UAT implements the latter through compact utility-conditioned query, key and value modulation. Our theory identifies a third-order interaction, establishes controlled one-layer and finite-sample separations, and provides a conditional compressed-belief interpretation. Experiments on DarkRoom, sepsis, D4RL, and robomimic support UAT's effectiveness, while ablations and feedback corruption tests clarify component contributions and operating boundaries. The theory remains stylized, and broader evaluation is needed to assess applicability across tasks, feedback conditions, and larger model scales. Future work will combine utility-conditioned retrieval with adaptive context selection or value-based return relabeling.

Ethics Statement

This work uses existing datasets and simulated environments. The sepsis benchmark combines de-identified MIMIC-III-derived initial states with synthetic dynamics and rewards; no patient interventions are performed. Data access and sharing remain subject to the source data-use agreement. Results do not establish clinical efficacy or deployment safety. Dataset biases and unreliable feedback remain risks, and real-world clinical or robotic deployment requires independent validation, fairness and privacy safeguards, and human oversight.

Reproducibility Statement

The main text and appendices detail the architecture, training procedure, theoretical assumptions and proofs, benchmark construction, evaluation metrics, ablations, and efficiency measurements. Code and experimental configurations are available in our anonymous repository: `https://anonymous.4open.science/r/utility_augmented_transformer-1566`.

Use of AI assistance

Generative AI tools were used for language editing and code debugging. The authors are responsible for the research content, derivations, experiments, and claims.

References

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, March 2003. ISSN 1532-4435.

Prajjwal Bhargava, Rohan Chitnis, Alborz Geramifard, Shagun Sodhani, and Amy Zhang. When should we prefer decision transformers for offline reinforcement learning? In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=vpV7fOFQy4`.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: reinforcement learning via sequence modeling. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Reinforcement learning for non-stationary Markov decision processes: the blessing of (more) optimism. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Shmuel Bar David, Itamar Zimerman, Eliya Nachmani, and Lior Wolf. Decision S4: Efficient sequence-based RL via state spaces layers. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=kqHkCVS7wbj`.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir (eds.), *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pp. 907–940, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v49/eldan16.html`.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. RvS: What is essential for offline RL via supervised learning? In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=S874XAIpkR-`.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning, 2020.

David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=rkpACe1lx`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Los Alamitos, CA, USA, June 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.90. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.90.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, July 1989. ISSN 0893-6080.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Siddhant M. Jayakumar, Wojciech M. Czarnecki, Jacob Menick, Jonathan Schwarz, Jack Rae, Simon Osindero, Yee Whye Teh, Tim Harley, and Razvan Pascanu. Multiplicative interactions and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rylnK6VtDH.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016. doi: 10.1038/sdata.2016.35. URL https://doi.org/10.1038/sdata.2016.35.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(98)00023-X. URL https://www.sciencedirect.com/science/article/pii/S000437029800023X.

Matthieu Komorowski, Leo A. Celi, Omar Badawi, Anthony C. Gordon, and A. Aldo Faisal. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716–1720, 2018. doi: 10.1038/s41591-018-0213-5. URL https://doi.org/10.1038/s41591-018-0213-5.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hy0a5MMPUv.

Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993. ISSN 0893-6080. doi: https://doi.org/10.

1016/S0893-6080(05)80131-5. URL https://www.sciencedirect.com/science/article/pii/S0893608005801315.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

Emilio Parisotto, H. Francis Song, Jack W. Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant M. Jayakumar, Max Jaderberg, Raphaël Lopez Kaufman, Aidan Clark, Seb Noury, Matthew M. Botvinick, Nicolas Heess, and Raia Hadsell. Stabilizing transformers for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2332–2341, 2019. doi: 10.1109/CVPR.2019.00244.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7191–7201, 2022. doi: 10.1109/CVPR52688.2022.00706.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992. doi: 10.1007/BF00992698. URL https://doi.org/10.1007/BF00992698.

Steven D. Whitehead and Dana H. Ballard. Learning to perceive and act by trial and error. *Mach. Learn.*, 7(1):45–83, July 1991. ISSN 0885-6125. doi: 10.1023/A:1022619109594. URL https://doi.org/10.1023/A:1022619109594.

Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. Elastic decision transformer. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Taku Yamagata, Ahmed Khalil, and Raúl Santos-Rodríguez. Q-learning decision transformer: leveraging dynamic programming for conditional sequence modelling in offline RL. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: variational Bayes-adaptive deep RL via meta-learning. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

A Details of Theorems and Proofs

A.1 Feedback-blindness and exact containment

Proof of Proposition 1.

Let H_t and H'_t be perceptually aliased. By definition they contain the same observation sequence and the same time indices. A model in $\mathcal{F}_{\mathrm{TF}}^{\mathrm{obs}}$ constructs every token and intermediate feature as a deterministic or learned function of those observations and positions only. Its output has the form

$$F(H_t) = f_\theta(o_1, \ldots, o_t, 1, \ldots, t). \tag{27}$$

Since the two histories agree on every argument of f_θ, $F(H_t) = F(H'_t)$. The result holds for any depth, width, positional encoding, or parameter count, because the missing variables are absent from the input domain.

Proof of Proposition 2.

Set

$$W_Q^{(u)} = W_K^{(u)} = W_V^{(u)} = 0, \qquad U_Q = U_K = U_V = 0. \tag{28}$$

Equations (8)–(10) reduce to $\tilde{q}_t = W_Q x_t$, $\tilde{k}_j = W_K x_j$, and $\tilde{v}_j = W_V x_j$ for every utility value, including the zero initialization. The subsequent attention, feed-forward, residual, normalization, and action-head operations are unchanged. Hence the function is exactly the corresponding vanilla Transformer.

Proof of Lemma 1.

With the multiplicative matrices set to zero, Eqs. (8)–(10) become

$$\tilde{q}_t = W_Q x_t + U_Q \rho_t, \qquad \tilde{k}_j = W_K x_j + U_K g_j, \qquad \tilde{v}_j = W_V x_j + U_V g_j. \tag{29}$$

Concatenating block matrices gives

$$[W_Q, 0, U_Q] \begin{bmatrix} x_t \\ g_t \\ \rho_t \end{bmatrix} = W_Q x_t + U_Q \rho_t, \tag{30}$$

and similarly for keys and values. Thus the projections are ordinary linear projections of $y_j = [x_j; g_j; \rho_j]$. A standard Transformer on this augmented sequence with the displayed block matrices has identical Q/K/V, attention logits, values, and outputs.

Proof of Theorem 1.

Containment. Consider an input-level layer with linear embeddings

$$x_j^{\mathrm{in}} = A_o o_j + A_f \phi_j + p_j \tag{31}$$

and projections W_Q, W_K, W_V. Under Assumption 1 and the theorem's gate-representability condition, choose UAT's observation embedding so that $x_j = A_o o_j + p_j$, and choose the encoder/readouts so that $g_j = \rho_j = \phi_j$; $\lambda = 0$ makes ρ_j depend on the current shifted utility feature. For the affine-gate core, this choice includes every linear ϕ_j. In the bias-only subclass choose

$$U_Q = W_Q A_f, \qquad U_K = W_K A_f, \qquad U_V = W_V A_f. \tag{32}$$

Then

$$\tilde{q}_t = W_Q x_t + U_Q \phi_t = W_Q x_t^{\mathrm{in}}, \tag{33}$$

with the same identity for keys and values. Therefore every function in the stated linear input-conditioning class is realized exactly.

Strict extension. Choose a scalar utility feature and an affine gate $g_j = u_j$. Equation (23) gives

$$\ell_{\mathrm{UAT}}(x_t, x_j, u_j) = \sum_m u_j (w_{Q,m}^\top x_t)(w_{K,m}^{(u)\top} x_j), \tag{34}$$

a homogeneous degree-three polynomial in (x_t, x_j, u_j). By contrast, fixed linear embeddings of $[o_t; \phi_t]$ and $[o_j; \phi_j]$ followed by $W_Q^\top W_K$ yield a bilinear form in the two augmented token vectors and hence total degree at most two in the primitive variables. Polynomial equality would require every degree-three coefficient to vanish, so nonzero targets of the displayed form are outside the input-level class.

A.2 Conditional universality

Theorem 2 (Conditional universality)**.** *Fix horizon T, let the augmented history domain be compact, and let the target history-to-score map be continuous. Under Assumption 1 with $\epsilon = 0$, if standard causal Transformers are universal approximators on the augmented token domain in the sense of Yun et al. (2020), then for every $\eta > 0$ there is a finite UAT, using only the bias pathway, whose uniform approximation error is at most η.*

Proof of Theorem 2.

Fix the finite horizon T. By Lemma 1, the bias-only UAT subclass is a standard causal Transformer on tokens $y_j = [x_j; g_j; \rho_j]$. Under the $\epsilon = 0$ case of Assumption 1, the utility components preserve the feedback information in H_t; concatenating them with observation embeddings therefore yields a token sequence that is information-equivalent to the raw history. The history domain is compact by assumption and the target is continuous, so the assumed universality of causal Transformers on the augmented token domain gives, for every $\eta > 0$, a finite-depth, finite-width network with uniform error at most η. The UAT realization is obtained by the block matrices in Lemma 1. If sufficiency holds only up to ϵ, the same construction and an L_F-Lipschitz target give total error at most $L_F\epsilon + \eta$.

A.3 Interaction order, width, and depth

Proof of Theorem 3.

Affine-gate realization. For $\ell(x, y, u) = \sum_{k=1}^K s_k(u)(q_k^\top x)(w_k^\top y)$, choose

$$g(u) = (s_1(u), \ldots, s_K(u)), \tag{35}$$

set row k of W_Q to $q_k^\top$, and set row k of $W_K^{(u)}$ to $w_k^\top$. Equation (23) then equals ℓ exactly. The stated parameter count follows from the K rows of the gate map and the K rows of each relevant projection.

Part (a). A linear embedding of $[x; u]$ has the form $A_x x + A_u u$. The scalar logit

$$(A_{q,x} x + A_{q,u} u)^\top (A_{k,y} y + A_{k,u} u) \tag{36}$$

contains only monomials of total degree zero, one, or two in (x, y, u). For any such monomial m, independence implies $\mathbb{E}[m\ell^*] = \mathbb{E}[u(q^\top x)(w^\top y)m]$. If m contains no factor u, centering of u makes the expectation zero. If m contains a factor u, degree at most two leaves at most one factor from x or y; the missing variable then appears only in ℓ^* and its centered independent factor makes the expectation zero. Thus ℓ^* is orthogonal to every degree-at-most-two logit. The projection identity gives Eq. (25):

$$\mathbb{E}[(\ell - \ell^*)^2] = \mathbb{E}[\ell^2] + \mathbb{E}[(\ell^*)^2] \geq \mathbb{E}[(\ell^*)^2], \tag{37}$$

with equality at $\ell = 0$. Independence yields

$$\mathbb{E}[(\ell^*)^2] = \mathbb{E}[u^2]\,\mathbb{E}[(q^\top x)^2]\,\mathbb{E}[(w^\top y)^2] = \sigma_u^2 (q^\top \Sigma_x q)(w^\top \Sigma_y w). \tag{38}$$

Part (b). Expand the target as a third-order tensor,

$$T_{i,j,r} = \sum_{k=1}^K q_{k,i} w_{k,j} s_{k,r}. \tag{39}$$

A factorization $\ell(x, y, u) = \sum_{m=1}^{M} \psi_m(x)\phi_m(y, u)$ has separation rank at most M across the split $x \mid (y, u)$, so the unfolding $T_{(1)} \in \mathbb{R}^{d_x \times (d_x d_u)}$ satisfies $\mathrm{rank}(T_{(1)}) \le M$. Flattening the target tensor along the query index gives $T_{(1)} = Q^\top K(W, S)$, where Q, W, and S have rows $q_k^\top$, $w_k^\top$, and $s_k^\top$, respectively, and $K(W, S)_{k,:} = w_k \otimes s_k$ defines the row-wise Kronecker product. By hypothesis, $\{q_k\}$ is linearly independent, so Q has full row rank K; and $\{w_k \otimes s_k\}$ is linearly independent, so $K(W, S)$ has full row rank K. Sylvester's rank inequality now gives

$$\mathrm{rank}\big(Q^\top K(W, S)\big) \;\ge\; \mathrm{rank}(Q^\top) + \mathrm{rank}\big(K(W, S)\big) - K \;=\; K,$$

while $\mathrm{rank}(Q^\top K(W, S)) \le \min\{K, K\} = K$. Hence $\mathrm{rank}(T_{(1)}) = K$, and any factorization requires $M \ge K$.

Part (c). Let the first layer compute features $q_k^\top x$ on the query token and $(s_k(u), w_k^\top y)$ on the key token. If an intermediate map forms exact polynomial products, choose a second bilinear map between the degree-two feature expansions of these two token vectors. It contains the coefficient of

$$(q_k^\top x)\, s_k(u)\, (w_k^\top y), \tag{40}$$

and summing over k realizes the target. Standard ReLU or GELU feed-forward maps are universal approximators on compact sets (Leshno et al., 1993), so they approximate such polynomial feature maps to arbitrary accuracy; exactness is not implied by their usual polynomial-degree counting.

A.4 Finite-sample implication

Assumption 3 (Bounded learning design)**.** *Samples (x_i, y_i, u_i, Y_i), $i = 1, \dots, n$, are i.i.d., Assumption 2 holds, and*

$$\|x_i\| \le B_x, \qquad \|y_i\| \le B_y, \qquad |u_i| \le B_u, \qquad Y_i = \ell^*(x_i, y_i, u_i) + \varepsilon_i, \quad |\varepsilon_i| \le B_\varepsilon. \tag{41}$$

The noise ε_i is centered and independent of (x_i, y_i, u_i); write $\sigma_\varepsilon^2 = \mathbb{E}[\varepsilon_i^2]$.

Let $\mathcal{G}_A$ be a bounded-norm bilinear input-level class and

$$\mathcal{G}_B = \big\{\beta u (a^\top x)(b^\top y) : \|a\|, \|b\| \le 1,\ |\beta| \le B_\beta\big\} \tag{42}$$

with $B_\beta \ge \max\{1, \|q\|_2 \|w\|_2\}$, so that $\ell^* \in \mathcal{G}_B$. Denote squared risk by $R(g) = \mathbb{E}[(g - Y)^2]$ and let $\hat{g}_A, \hat{g}_B$ be empirical risk minimizers over the two classes.

Theorem 4 (Approximation floor versus vanishing estimation error)**.** *Under Assumption 3, set*

$$G = B_\beta B_u B_x B_y, \qquad Y_{\max} = G + B_\varepsilon, \qquad L = 2(G + Y_{\max}), \qquad M = (G + Y_{\max})^2. \tag{43}$$

(a) for every sample size,

$$\mathbb{E}[R(\hat{g}_A)] \ge \sigma_\varepsilon^2 + \kappa^2, \qquad \kappa^2 = \sigma_u^2 (q^\top \Sigma_x q)(w^\top \Sigma_y w); \tag{44}$$

(b) with probability at least $1 - \delta$ over the sample,

$$R(\hat{g}_B) \le \sigma_\varepsilon^2 + C\, \frac{LG}{\sqrt{n}} + C\, M \sqrt{\frac{\log(1/\delta)}{n}} \tag{45}$$

for a universal constant C. Consequently, with probability at least $1 - \delta$,

$$\mathbb{E}[R(\hat{g}_A)] - R(\hat{g}_B) \ge \kappa^2 - C\, \frac{LG}{\sqrt{n}} - C\, M \sqrt{\frac{\log(1/\delta)}{n}}, \tag{46}$$

which is positive once n is large relative to the bounded complexity factors.

Proof of Theorem 4.

Part (a). By Theorem 3(a), every $g \in \mathcal{G}_A$ has

$$\mathbb{E}[(g - \ell^*)^2] \ge \kappa^2. \tag{47}$$

Because ε is centered and independent,

$$R(g) = \mathbb{E}[(g - \ell^* - \varepsilon)^2] = \mathbb{E}[(g - \ell^*)^2] + \sigma_\varepsilon^2 \ge \sigma_\varepsilon^2 + \kappa^2. \tag{48}$$

The ERM cannot have risk below the infimum over the class.

Part (b). Since $\ell^* \in \mathcal{G}_B$, optimality of $\hat{g}_B$ and a standard uniform-convergence decomposition give

$$R(\hat{g}_B) - R(\ell^*) \le 2 \sup_{g \in \mathcal{G}_B} \left| \frac{1}{n} \sum_{i=1}^{n} (g(x_i, y_i, u_i) - Y_i)^2 - \mathbb{E}(g - Y)^2 \right|. \tag{49}$$

All predictions satisfy $|g| \le G$ and $|Y| \le Y_{\max}$. The squared loss is therefore L-Lipschitz in the prediction on this range and bounded by $M = (G + Y_{\max})^2$. Symmetrization, concentration for bounded random variables, and the contraction principle yield

$$R(\hat{g}_B) - R(\ell^*) \le CL\,\mathcal{R}_n(\mathcal{G}_B) + CM\sqrt{\frac{\log(1/\delta)}{n}} \tag{50}$$

with probability at least $1 - \delta$ for a universal C (Bartlett & Mendelson, 2003). For the class in Eq. (42),

$$\mathcal{R}_n(\mathcal{G}_B) = \frac{1}{n}\mathbb{E} \sup_{\|a\|,\|b\| \le 1, |\beta| \le B_\beta} \left| \sum_{i=1}^{n} \sigma_i \beta u_i (a^\top x_i)(b^\top y_i) \right| \tag{51}$$

$$\le \frac{B_\beta}{n}\,\mathbb{E}\left[\left\| \sum_{i=1}^{n} \sigma_i u_i x_i y_i^\top \right\|_{\mathrm{op}} \right] \tag{52}$$

$$\le \frac{B_\beta}{n} \sqrt{\mathbb{E}\left[\left\| \sum_{i=1}^{n} \sigma_i u_i x_i y_i^\top \right\|_F^2 \right]} \le \frac{B_\beta B_u B_x B_y}{\sqrt{n}} = \frac{G}{\sqrt{n}}. \tag{53}$$

Here σ_i are Rademacher variables, and the last displayed inequality uses $\|\cdot\|_{\mathrm{op}} \le \|\cdot\|_F$ and Jensen's inequality. Since $R(\ell^*) = \sigma_\varepsilon^2$, Eq. (45) follows.

Consequent risk gap. Combine part (a) with part (b). The lower bound on the input-level ERM holds in expectation, while the upper bound for the trilinear ERM holds on the stated probability event. Subtracting the two displays gives the claimed gap. The gap is positive when

$$C\left(\frac{LG}{\sqrt{n}} + M\sqrt{\frac{\log(1/\delta)}{n}} \right) < \kappa^2. \tag{54}$$

A.5 Extended remarks on the theoretical assumptions

Sufficient utility readout. Assumption 1 is the informational bridge between feedback and belief compression. It can be satisfied trivially by retaining raw shifted action–reward features, but that choice may increase width. A learned encoder is useful when it compresses feedback without discarding decision-relevant information. The sufficiency statement is task-relative: a statistic can be sufficient for optimal action scores without being sufficient for reconstructing the full latent state.

Centered symmetric design. Assumption 2 is deliberately stronger than the benchmark distribution because it produces an exact orthogonality constant. Its role is to show that the depth-one input-level deficiency is not removed by larger embedding width. Relaxing symmetry replaces the exact floor by a weaker approximate bound, but does not change the degree argument for exact representation.

Bounded learning design. Assumption 3 is used to avoid unbounded-loss tail issues. The resulting theorem is a high-probability statement with bounded predictions, bounded targets, and bounded loss. It should be read as a clean comparison of approximation and estimation errors, not as a sample-complexity guarantee for reinforcement learning.

Lipschitz belief-optimal scores. Assumption 4 is standard for score-vector policies and avoids treating a discontinuous argmax as a continuous function. At exact action ties, one can either smooth the policy or state the result for any continuous approximation.

Proof of Proposition 3.

By ϵ-sufficiency,

$$\|\psi(Z_t) - b_t\|_{\mathrm{TV}} \le \|\psi(Z_t) - b_t\|_1 \le \epsilon. \tag{55}$$

Assumption 4 therefore gives

$$\|\Phi(\psi(Z_t)) - \Phi(b_t)\| \le L_\Phi \epsilon. \tag{56}$$

For the converse, an observation-only predictor assigns the same vector p to the aliased histories H_t and H'_t. By the triangle inequality,

$$\|p - \Phi(b_t)\| + \|p - \Phi(b'_t)\| \ge \|\Phi(b_t) - \Phi(b'_t)\| = \delta. \tag{57}$$

Hence at least one summand is at least $\delta/2$. This statement is uniform over all observation-only predictors.

B Details of Parallel Training, Complexity and Parameters

B.1 Parallel Training

Throughout this appendix, B counts windows per computational batch, as in Algorithm 1.

Window sampling and alignment. We sample windows from fixed offline trajectories. For valid decision time t, let $s_t^{\mathrm{win}} = \max(1, t - L + 1)$ and define

$$\mathcal{W}_t = \big((o_j, a_{j-1}, r_{j-1})\big)_{j=s_t^{\mathrm{win}}}^{t}. \tag{58}$$

Short windows are left-padded to end at local position $i = L$. Position i maps to trajectory time $j(t,i) = t - L + i$; the validity mask excludes $j(t,i) < 1$. Each valid position predicts $a_{j(t,i)}$ with feedback input $(a_{j(t,i)-1}, r_{j(t,i)-1})$. Null symbols (a_0, r_0) apply only at episode starts; later windows retain logged feedback $(a_{s_t^{\mathrm{win}}-1}, r_{s_t^{\mathrm{win}}-1})$ at their first valid position.

Batched representations. Stack sampled observation embeddings as $\mathbf{X} \in \mathbb{R}^{B\times L\times d}$ and utility features as $\mathbf{U} \in \mathbb{R}^{B\times L\times d_u}$. For shifted logged feedback $(\mathbf{A}, \mathbf{r})$ and validity mask $\mathbf{m} \in \{0,1\}^{B\times L}$,

$$\mathbf{U} = \mathbf{m} \odot E(f_a(\mathbf{A}), \mathbf{r}), \tag{59}$$

where broadcasting the mask over features zeros padded utilities.

Window-local utility aggregation. Each window independently initializes its persistent state:

$$\overline{\mathbf{U}}_{b,0,:} = \mathbf{0}, \qquad \overline{\mathbf{U}}_{b,i,:} = \lambda \overline{\mathbf{U}}_{b,i-1,:} + (1-\lambda)\mathbf{U}_{b,i,:}. \tag{60}$$

Unrolling yields Eq. (15) independently per window. Entries of $\mathbf{C}_\lambda$ above the diagonal are zero; at $\lambda = 0$, $\mathbf{C}_0 = \mathbf{I}_L$. For a window ending at trajectory time t, the endpoint state is

$$\bar{u}_t = (1-\lambda) \sum_{j=s_t^{\mathrm{win}}}^{t} \lambda^{t-j} u_j. \tag{61}$$

No aggregated state is inherited from before s_t^{win}, even for overlapping windows.

Gate construction and attention. Layer l, head h uses gate tensors

$$\mathbf{G}^{(l,h)} = \tanh\Big(\mathbf{U}(W_s^{(l,h)})^\top\Big), \qquad \mathbf{R}^{(h)} = f_u^{(h)}(\overline{\mathbf{U}}), \tag{62}$$

with $\mathbf{G}^{(l,h)}, \mathbf{R}^{(h)} \in \mathbb{R}^{B\times L\times d_m}$. Token readouts are layer-specific; query readouts are computed once and shared across layers. Keys and values share token gates.

Equations (16)–(18) project the final tensor dimension and broadcast elementwise modulation across batch and window positions. Equation (19) computes multi-head attention independently per window, including head concatenation and output projection; $\top$ transposes the projected keys' last two dimensions. The mask restricts valid queries to valid current or earlier keys and excludes padded queries from softmax, zeroing their outputs. The resulting $\mathbf{Y}$ passes through the remaining residual, normalization, and feed-forward operations to produce the layer outputs.

Parallel supervision. The action head processes every valid final-layer position. Supervised and gate losses share the validity mask, with $N = \sum_{b,i} m_{b,i}$ in the flattened window batch. Repeated trajectory positions contribute once per window occurrence, as in Eqs. (12) and (13). Logged preceding actions and rewards allow independent parallel evaluation under teacher forcing.

B.2 Complexity and Parameters

We count arithmetic work for B length-L windows, D layers, H heads, and $d = Hd_m$, assuming linear gate readouts with optional elementwise activations and fixed λ. Parameter counts use the bias-free projections in Eqs. (8)–(10); overhead is relative to an otherwise identical Transformer backbone.

(1) Transformer backbone. Per-layer query, key, value, and output projections cost $O(BLd^2)$, as does a feed-forward network of width $O(d)$. Attention scores and value aggregation cost $O(BHL^2d_m) = O(BL^2d)$. Across D layers,

$$C_{\mathrm{TF}} = O\big(BD(Ld^2 + L^2d)\big)\,. \tag{63}$$

(2) Utility encoding and aggregation. Let c_{enc} and P_{enc} denote the utility encoder's per-token cost and parameter count, including any additional action-embedding parameters. Encoding costs $O(BLc_{\mathrm{enc}})$ and adds P_{enc} parameters, without layer or head factors because the encoder is shared.

Direct multiplication by $\mathbf{C}_\lambda$ costs $O(BL^2d_u)$ with no trainable parameters. A work-efficient causal prefix scan reduces this to $O(BLd_u)$ work at $O(\log L)$ parallel depth.

(3) Gate readouts. Token-gate readouts map $\mathbb{R}^{d_u}$ to $\mathbb{R}^{d_m}$ per layer and head, adding $DHd_md_u = Ddd_u$ parameters. Query readouts have the same dimensions but share weights across layers, adding $Hd_md_u = dd_u$ parameters. Together,

$$C_{\mathrm{read}} = O(BL(D+1)dd_u)\,, \qquad P_{\mathrm{read}} = (D+1)dd_u. \tag{64}$$

Elementwise gate activations add only $O(BLDd)$ work.

(4) Modulated projections. Per layer and head, the multiplicative matrices $W_Q^{(u)}, W_K^{(u)}, W_V^{(u)} \in \mathbb{R}^{d_m \times d}$ contribute

$$P_{\mathrm{mult}} = 3DHd_md = 3Dd^2. \tag{65}$$

Applying them costs $O(BLDd^2)$; subsequent elementwise multiplication implements diagonal modulation at $O(BLDd)$ cost.

The three additive maps $U_Q, U_K, U_V \in \mathbb{R}^{d_m \times d_m}$ contribute

$$P_{\mathrm{add}} = 3DHd_m^2 = \frac{3Dd^2}{H}, \tag{66}$$

with computational cost $O(BLDd^2/H)$. Shared key/value gates do not reduce these counts: their projection matrices remain distinct.

(5) Total cost and parameters. With direct EMA multiplication, the additional cost is

$$\Delta C = O\left(BL\left[c_{\mathrm{enc}} + Ld_u + (D+1)dd_u + Dd^2\left(1 + \frac{1}{H}\right)\right]\right). \tag{67}$$

The additional parameter count is

$$\Delta P = P_{\mathrm{enc}} + (D+1)dd_u + 3Dd^2\left(1 + \frac{1}{H}\right). \tag{68}$$

A prefix scan replaces Ld_u by d_u in the computational bound, leaving ΔP unchanged.

For $d_u = O(d)$ and $c_{\mathrm{enc}} = O(d^2)$, these terms are bounded by the backbone cost:

$$C_{\mathrm{UAT}} = O\big(BD(Ld^2 + L^2d)\big)\,. \tag{69}$$

Backpropagation preserves this order, and gate regularization adds only $O(BLDd)$ work. Parallel training thus retains the Transformer's asymptotic arithmetic complexity with constant-factor utility overhead; additional parameters are independent of B and L.

C Main Experimental Details

This section describes the setup details of main experiments; Appendix D covers the settings of additional experiments. The additional empirical results not represented above are described in Appendix E. Settings follow the supplied code and configurations.

C.1 BENCHMARKS

Controlled tasks compare UAT with baselines under feedback-informative conditions; classical offline RL benchmarks assess cross-domain performance against traditional RL methods.

C.1.1 CONTROLLED BENCHMARKS

DarkRoom and sepsis involve hidden, changing goals or treatment efficacy, with action outcomes informing the latent state. Models share a frozen offline log and choose their own actions during closed-loop evaluation. The generator defaults to data seed 0 and 600 episodes per task or DarkRoom schedule. Exploration cycles by episode through $\epsilon \in \{0.1, 0.3, 0.5, 1.0\}$, choosing a uniform random action with probability ϵ and the privileged reference action otherwise. Logged actions provide supervision; privileged goals, regimes, switch flags, and reference-action annotations serve only diagnostics, excluded from policy inputs and supervised targets.

DarkRoom. DarkRoom is a 10×10 navigation task with a hidden goal moving on gradual, cyclical, or abrupt schedules.

Data and inputs. Observations contain nine local 3×3 boundary-wall indicators and a proximity cue, positive with probability 0.5 within Manhattan distance 2 of the goal and zero otherwise. Coordinates are hidden. Goals lie in $\{1, \dots, 8\}^2$; initial agent positions are uniform excluding the initial goal. Logging reference greedily approaches pre-action goal, favoring the first coordinate on distance ties.

Environment and rewards. Episodes last 200 decisions with four cardinal actions and boundary-clipped movement. Actions are replaced by uniform random cardinal actions with probability 0.1; reaching the goal does not terminate episodes. Goal-schedule segments have uniform integer lengths 30–60, truncated at the horizon. Under *gradual* changes, each segment samples one of eight cardinal/diagonal directions; the goal moves every third within-segment step, clipped to the interior. Under *cyclical* changes, four distinct interior goals and a random initial phase are sampled per episode; goals cycle at segment boundaries. Under *abrupt* changes, each boundary relocates the goal by Manhattan distance at least 4. Reward is

$$r_t = 2\,\mathbf{I}[d_t^{\text{new}} = 0] + 0.5\,\text{sign}(d_t^{\text{prev}} - d_t^{\text{new}}) - 0.01, \tag{70}$$

where d_t^{prev} is the previous transition's stored distance and d_t^{new} the distance after movement and schedule update; these may reference different goals at a goal change.

Evaluation and metrics. Each seed uses 50 fresh procedural episodes. Accuracy is agreement with the privileged greedy rule along the policy's own trajectory; return sums actual rewards. Both are averaged across episodes within each seed.

Sepsis. This semi-synthetic treatment task combines clinical initial states with synthetic transitions, hidden efficacy regimes, resistance, termination hazards, and rewards.

Data and inputs. The source documents $39{,}232$ initial vectors of 46 features from MIMIC-III through the sepsis preprocessing pipeline (Johnson et al., 2016; Komorowski et al., 2018). The default generator samples $4{,}000$ indices without replacement: $3{,}200$ training and 800 held-out. These sets are disjoint, but patient-level uniqueness cannot be independently checked without the missing identity mapping. Standardization and PCA-20 use only training initial states and are reused throughout; episodes initialize from the corresponding pool. Nine actions combine vasopressor doses $v \in \{0, 1, 3\}$ with fluid doses $f \in \{0, 1, 4\}$.

The main log replaces the older regime-blind severity heuristic with a privileged deterministic six-step planner (discount 0.99). It ranks first actions using the full simulator state, resistance history, and true regime schedule, followed by a fixed regime-aware heuristic. Planner clones suppress noise, fix progression jitter at 1, and use expected hazard rewards. This reference supplies evaluation labels but is not an exact optimal POMDP policy.

Environment and rewards. Episodes last at most 90 steps. At reset, 7–11 regime segments receive raw integer lengths 6–13, rescaled to the horizon, rounded down with minimum one, and adjusted in the final segment to fill the horizon. The initial regime is uniform over three possibilities; subsequent regimes differ from predecessors. Regimes favor fluids, vasopressors while penalizing fluids, or conservative treatment. Let S, ℓ, P, H denote SOFA, lactate, mean blood pressure, and heart rate. Table 4 lists treatment increments before progression and noise. For consecutive identical actions,

Table 4: Synthetic sepsis treatment effects and regime constants.

Regime z	**Treatment increments**	p_z	b_z	δ_z
0	$\Delta P = e(0.20f + 0.08v)$; $\Delta\ell = -e(0.15f + 0.04v)$; $\Delta S = -e(0.06f + 0.02v)$	0.03	0.008	0.95
1	$\Delta P = 0.25ev$; $\Delta\ell = -0.18ev + 0.05f$; $\Delta S = -0.08ev + 0.10f$; $\Delta H = 0.05\mathbf{I}[v \geq 3]$	0.10	0.020	0.85
2	**If** $v, f \leq 1$: $m = 0.5e\max(v, f)$, $\Delta S = -0.03m$, $\Delta\ell = -0.02m$. **Otherwise**: $A = 0.08(v + f)$, $\Delta S = A$, $\Delta\ell = 0.4A$, $\Delta H = 0.3A$.	0.05	0.032	0.75

efficacy is $e = 1$ for repeat counter $c \leq 2$ and $e = \max(0.4, \delta_z^{c-2})$ thereafter; changing actions resets c to zero.

Progression adds $p_z J + \eta$ to S, with $J = \text{clip}(1 + 0.3\xi, 0.5, 2)$, $\xi \sim \mathcal{N}(0, 1)$, and $\eta \sim \mathcal{N}(0, b^2)$. Normally $b = b_z$; with probability 0.1, it is multiplied by $1.5 + 0.5U$, $U \sim \text{Uniform}(0, 1)$. Independent noise of standard deviation $0.05b$ enters all 46 components. Lactate also increases by 0.12 in regime 1 and by $0.15U$ with probability 0.2 in regime 2. Nonterminal reward is $0.6(S_t - S_{t+1}) + 0.3(\ell_t - \ell_{t+1})$. With $[x]_+ = \max(x, 0)$, termination hazards are

$$h_{\text{death}} = \min\{0.0004 + 0.0035[S_{t+1}]_+ + 0.0022[\ell_{t+1}]_+,\ 0.010\}, \tag{71}$$

$$h_{\text{recovery}} = \min\{0.0008 + 0.0060[-S_{t+1}]_+ + 0.0045[-\ell_{t+1}]_+,\ 0.009\}. \tag{72}$$

One uniform draw determines mutually exclusive death or recovery. Terminal reward replaces shaping: -1 for death, $+1$ for recovery; at the horizon, $+1$ iff $S < 0.3$ and $\ell < 0.3$, otherwise -1.

Evaluation and metrics. Each seed uses 50 fresh closed-loop episodes from held-out indices, scoring each decision against the planner. Accuracy averages agreement within and across episodes. With zero-based simulator indices and episode length T_{ep}, Post-Acc averages agreement over each realized switch window $\{\tau, \ldots, \min(\tau + 4, T_{\text{ep}} - 1)\}$, then over windows within episodes and across eligible episodes. Decision τ precedes its new-regime reward. Return sums undiscounted simulator rewards; separately reported recovery rate counts recovery terminations, excluding stable timeouts.

C.1.2 Classical RL Benchmarks

These tasks assess performance beyond constructed hidden regimes without assuming feedback informativeness.

D4RL. We evaluate nine standard offline RL tasks: HalfCheetah, Hopper, and Walker2d at medium, medium-replay, and medium-expert levels (Fu et al., 2020; Chen et al., 2021; Bhargava et al., 2024).

Data and inputs. We retain all D4RL locomotion trajectories, sampling proportionally to length with uniform window starts. Observations use offline standardization; undiscounted RTG is divided by 1000. Actions remain unstandardized, with tanh sequence-model outputs.

Environment and rewards. Evaluation retains original rewards and a 1000-step horizon without imposed regime switches. The factory instantiates Gymnasium `v4` (`HalfCheetah-v4`, `Hopper-v4`, `Walker2d-v4`), falling back to Gym `v3` if the import fails; evaluation environments are independent of the offline dataset. Recorded runs use PyTorch 2.4.1 (CUDA 12.1), Gymnasium 1.2.3, and MuJoCo 3.6.0. The factory passes no seed to `gymnasium.make` or `reset`; only the optional action-noise wrapper uses per-episode seeds computed by `evaluate_episodes_parallel`, leaving unperturbed rollouts with the backend's unseeded `reset`.

Evaluation and metrics. We report final-checkpoint unclipped D4RL normalized return over 100 episodes at predefined higher targets: 12000, 3600, and 5000 for HalfCheetah, Hopper, and Walker2d, respectively. Neither checkpoints nor targets are selected by evaluation scores. At each dataset level, scores are averaged across environments within each seed, then summarized by mean and standard deviation over seeds 0–4.

Robomimic. We evaluate Can and Lift under dense and sparse rewards (Mandlekar et al., 2022). UAT maintains approximately 960k parameters per setting, allocated differently across utility and backbone components. Tables 5 and 6 list hyperparameters and model sizes.

Data and inputs. We retain all machine-generated (MG) low-dimensional trajectories. Observations use offline standardization; logged actions remain unstandardized, with tanh outputs. Evaluation uses closed-loop rollouts, not a held-out trajectory split.

Environment and rewards. Rollouts use robosuite (version 1.5.1), a Panda arm with operational-space pose control at 20 Hz, and a 400-step horizon. Dense and sparse settings provide shaped and binary success rewards, respectively; both are relabelled as

$$r'_t = 10(r_t - r_{t-1}) + 10\,\mathbf{I}[\text{first success}],$$

conditioning on reward increments and a first-success bonus. Initial target return is 20.

Evaluation and metrics. We report final-checkpoint success rates over 100 episodes per seed, averaged over seeds 0–4. An episode succeeds if the environment's success flag is reached at any step. Final evaluations use model seed plus 60,000, separately from intermediate evaluations.

C.2 Baselines

Controlled-task baselines adapt the cited mechanisms to a common training interface without reproducing every original paper's full pipeline.

C.2.1 Behavior Cloning.

Vanilla-TF. This observation–action Transformer predicts logged actions using DT's causal layout with zero scalar RTG, retaining observation/action histories, timestep embeddings, and the backbone. Biases and timestep embeddings can yield nonzero return-token embeddings despite removing reward conditioning.

C.2.2 Classical RL.

CQL. On discrete controlled tasks, CQL adds its conservative penalty (Kumar et al., 2020) to the window-based TD loss below:

$$\mathcal{L}_{\mathrm{CQL}} = \mathcal{L}_{\mathrm{TD}} + \alpha_{\mathrm{CQL}}\,\mathbb{E}_t\left[\log \sum_a \exp Q_t(a) - Q_t(a_t)\right]. \tag{73}$$

Discrete actions use $\alpha_{\mathrm{CQL}} = 5$. Continuous-control D4RL uses the same coefficient with a branching Q-network: 11 bins per action dimension on $[-1, 1]$, mean per-dimension Q as joint value, and the dimension-averaged logsumexp-minus-taken gap as conservative penalty. TD uses one-step backups with $\gamma = 0.99$ and smooth-L_1 loss, without a separately evaluated target network.

IQL. IQL (Kostrikov et al., 2022) reuses CQL's window encoder and Q head at the same width, adding linear value and actor heads. It avoids querying unseen actions by fitting V_t with expectile regression on the logged-action value and backing up Q through V:

$$\mathcal{L}_{\mathrm{IQL}} = \mathbb{E}_t[L_2^\tau(Q_t(a_t) - V_t)] + \mathbb{E}_t[\ell(Q_t(a_t),\, r_t + \gamma V_{t+1})] - \mathbb{E}_t\left[e^{\beta(Q_t(a_t)-V_t)} \log \pi(a_t \mid h_t)\right], \tag{74}$$

where $L_2^\tau(x) = |\tau - \mathbb{1}[x < 0]|\, x^2$ with $\tau = 0.7$, ℓ is the smooth-L_1 loss with $\gamma = 0.99$, and the advantage-weighted actor uses $\beta = 3$ with weights clipped at 100. Value targets and advantage weights are detached, next positions are masked as in DQN, and no separately evaluated target network is used. Evaluation acts greedily under the actor head.

DQN. Offline DQN (Mnih et al., 2015) uses smooth-L_1 loss between $Q_t(a_t)$ and detached target $r_t + 0.99 \max_a Q_{t+1}(a)$ at adjacent window positions, masking unavailable next positions and omitting the final position. Both Q values come from one forward pass, without a separately evaluated target network. DQN and CQL optimize TD objectives rather than supervised action prediction; IQL pairs TD value learning with advantage-weighted action prediction.

C.2.3 MEMORY AND ADAPTATION.

GTrXL. Our adaptation uses GRU-gated residuals and identity-map reordering (Parisotto et al., 2020): layer-normalized attention and MLP outputs pass through ReLU and mix with the pre-block state through a GRU gate initialized with update-gate bias 2. Windows are encoded independently without Transformer-XL segment memory or hidden-state carryover across windows or evaluation episodes.

TENT. At evaluation, TENT minimizes discrete-action prediction entropy by updating a Vanilla-TF backbone's LayerNorm affine parameters (Wang et al., 2021); UAT's policy stays fixed. Each action takes one SGD step (learning rate 10^{-3}, momentum 0.9). Continuous actions replace entropy with MSE between two dropout-perturbed predictions. Adaptation persists across steps and episodes without source-checkpoint resets.

CoTTA. CoTTA adapts the same Vanilla-TF backbone at evaluation with an exponential-moving-average teacher and stochastic weight restoration (Wang et al., 2022). Each action takes one AdamW step (learning rate 10^{-3}) on all trainable student weights, matching an EMA teacher (coefficient 0.999) through discrete-logit KL divergence or continuous-action MSE. Each student parameter is then independently restored to the frozen source checkpoint with probability 0.01. Teacher and student persist across steps and episodes; restoration is the only return toward the source.

C.2.4 BELIEF-BASED.

VariBAD. VariBAD (Zintgraf et al., 2021) infers a posterior over a latent task variable from the interaction history and conditions the policy on it. A GRU encoder reads shifted inputs (o_t, a_{t-1}, r_{t-1}) and outputs a diagonal Gaussian $q(m \mid h_t) = \mathcal{N}(\mu_t, \operatorname{diag} \sigma_t^2)$ with 16 latent dimensions at every window position. The encoder and a reward decoder $p(r_j \mid o_j, a_j, m)$ are trained with the evidence lower bound

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_t \left[\sum_j \big(\hat{r}_j(m_t) - r_j\big)^2 + \kappa \, \mathrm{KL}(q(m \mid h_t) \,\|\, q(m \mid h_{t-1})) \right], \quad m_t \sim q(m \mid h_t), \tag{75}$$

where each position's belief reconstructs every reward in the window, the first valid position uses a standard-normal prior, and $\kappa = 0.1$. The policy reads $(o_t, \mu_t, \log \sigma_t^2)$ and is trained with the same action cross-entropy as the sequence models; following VariBAD, policy gradients do not reach the encoder. Offline training replaces the original on-policy PPO with this supervised objective. Encoder, decoder, and policy share one hidden width, chosen to match UAT's trainable count. Evaluation re-encodes the trailing K-step window at each step and acts greedily.

C.2.5 SEQUENCE MODELING.

DT. DT (Chen et al., 2021) interleaves RTG, observation, and action tokens, predicting actions from observation tokens and decrementing desired remaining return by observed rewards at evaluation. The feedback-concatenating depth-one DT in Appendix D.2 is a separate mechanism control.

EDT. EDT adds an expectile return head with auxiliary loss and selects history length at inference (Wu et al., 2023), using expectile 0.99. It scores four candidate lengths, the unique integers $\mathrm{round}((i+1)K/4)$ for $i = 0, 1, 2, 3$ clipped to $\{1, \ldots, K\}$, and acts from the highest-scoring prefix.

QDT. Before policy training, QDT fits a high-expectile value estimator to relabel RTG (Yamagata et al., 2023) with documented local update $r_t + \max(G_{t+1}, V(o_{t+1}))$. The observation-to-scalar value network has two width-$4d$ ReLU hidden layers, fitted by AdamW (learning rate 10^{-3}, batch 256) for $2{,}000$ DarkRoom and $8{,}000$ sepsis updates with expectile 0.99. DT is then trained on relabelled targets as the acting policy.

C.2.6 RELATION TO GENERIC CONDITIONING MECHANISMS

LoRA (Hu et al., 2022), FiLM (Perez et al., 2018), and hypernetworks (Ha et al., 2017) motivate UAT's utility-affine design through general adaptation and conditioning. They can be used in decision models, but comparable baselines require specified conditioning signals, modulation locations, and

Table 5: Main experimental configurations. K counts environment decisions; B counts sampled windows. Robomimic configurations follow the order Can-D / Can-S / Lift-D / Lift-S.

Group	Setting	DarkRoom	Sepsis (planner-guided trajectories)	D4RL (locomotion, 100k)	robomimic
Training	Model seeds	0–4	0–4	0–4	0–4
	Updates	20,000	80,000	100,000 ($10 \times 10{,}000$)	30,000 (300×100)
	Batch size B / context K	256 / 40	256 / 40	256 / 20	256 / 20; Can-S 512 / 20
Architecture	UAT layers / heads / width	3 / 4 / 64	3 / 4 / 64	3 / 1 / 128	3 / 1 / 128; Lift-D 4 heads
	UAT feed-forward width	256	256	512	512 / 499 / 644 / 449
	Dropout	0.1	0.1	0.1	0.1; Can-S 0.05
Optimization	Backbone learning rate	10^{-3}	3×10^{-4}	10^{-4}	10^{-4} / 1.5×10^{-4} / 3×10^{-5} / 3×10^{-5}
	Utility learning rate	10^{-3}	10^{-3}	10^{-3}	Equal to backbone
	Backbone / utility weight decay	10^{-5} / 10^{-5}	10^{-4} / 0	10^{-4} / 0	0.1 / 0.1; Can-S 0.01 / 0.01
	Warmup updates	500	4,000	10,000	3,000; Can-S 1,000
	After warmup	Constant	Cosine to 10%	Constant	Cosine to 10%; Can-S to 15%
Feedback	Gate coefficient β	10^{-4}	10^{-4}	10^{-2}	10^{-4}
	EMA decay λ	0.8	0.8	0.8	0.8
	RTG / utility reward divisor	1 / 1	1 / 1	1000 / 1000	1 / 1
	Initial target return	120	10	Task-specific	20
Baseline	QDT value-fit updates	2,000	8,000	Not in this sweep	Not in this sweep
Evaluation	Evaluation episodes	50	50	100 per target	50; final epoch 100
	Evaluation batch size	50	50	48	Serial rollout
	Evaluation schedule	Final	Final	Final iteration	Every 50 epochs
	Reported checkpoint rule	Final	Final	Final, higher target	Final

training protocols. Our decision-model comparisons and controlled feedback-injection and Q/K/V-modulation ablations (Appendix D.5) assess UAT's design without claiming superiority over all variants of these mechanisms.

C.3 Implementation and Hyperparameters

Inputs and parallel optimization. UAT uses one observation token per decision; shifted action–reward feedback (a_{t-1}, r_{t-1}) modulates Q/K/V through the utility pathway. DT uses three tokens per decision, (G_t, o_t, a_t); Vanilla-TF keeps this layout with zero RTG. A K-decision context thus contains K attention tokens for UAT and $3K$ for DT and Vanilla-TF; K corresponds to decision-window length L in the method. DT and Vanilla-TF predict actions before the corresponding action tokens are visible. Windows inside episodes retain predecessor action/reward, but utility aggregation resets at each window boundary. All valid action-prediction positions receive parallel teacher-forced supervision: discrete-action cross-entropy or masked continuous-action MSE, equivalent to fixed-variance Gaussian NLL up to scale and a constant. UAT adds the gate penalty in Eq. (13); training targets require no online policy rollouts.

Architecture and optimization. Table 5 lists controlled-task and 100k-update locomotion settings. UAT's utility encoder has 8-dimensional action embeddings, a width-32 ReLU network, and 16-dimensional outputs. Tanh query readouts use a window-local EMA with fixed $\lambda = 0.8$; shared tanh key/value gates use unsmoothed shifted utility. Gate-readout weights and additive utility matrices start at zero; multiplicative utility projection weights follow $\mathcal{N}(0, 0.1^2)$, yielding zero initial modulation despite nonzero utility parameters. AdamW uses moments $(0.9, 0.999)$, $\epsilon = 10^{-8}$, and gradient-norm clipping at 0.25. Python, NumPy, and PyTorch are seeded; fresh controlled-task evaluation episodes use model seed plus $50{,}000$.

Controlled tasks use seeds 0–4; sepsis uses the 80k-update planner-guided trajectory dataset. All locomotion rows in Table 2, including Vanilla-TF, use Table 5's 100k setting: $10 \times 10{,}000$ updates, $\beta = 10^{-2}$, $\lambda = 0.8$, seeds 0–4, and $10{,}000$ warmup updates.

Parameter matching. Comparable Transformer controls search integer feed-forward widths 8–8192 at UAT's task-specific reference depth, then nearby depths if needed, to match its trainable parameter count within 2.5%. Recorded controlled-task matches retain depth, heads, and hidden width, widening feed-forward blocks from 256 to 388; Table 6 lists benchmark-specific counts. QDT

Table 6: Trainable parameter counts by benchmark and component. Dashes denote model–benchmark combinations not evaluated. D4RL (one attention head) pairs list Hopper / (HalfCheetah and Walker2d); robomimic pairs list Can / Lift.

Model/component	DarkRoom	Sepsis	D4RL (locomotion)	robomimic	Counted component
Vanilla-TF	216,729	211,815	1,031,338 / 1,034,035	959,927 / 958,644	Policy
DT	216,731	211,626	1,031,341 / 1,033,654	959,677 / 958,649	Policy
EDT	216,796	211,691	–	–	Policy and return head
QDT	216,731	211,626	–	–	Acting policy only
TENT / CoTTA	216,731	211,626	–	–	Adapted base / student
VariBAD	216,415	211,420	–	–	Encoder, reward decoder, and policy
UAT	216,703	211,638	$\approx 1{,}030{,}000$	$\approx 960{,}000$	Full model

counts only its acting policy, excluding the offline value relabeler; TENT/CoTTA count the adapted base, excluding CoTTA's frozen teacher. CQL/DQN retain natural Q-network sizes, and IQL keeps CQL's width with added value and actor heads; GTrXL retains its natural size because matching its extra gates would require an unusually narrow feed-forward block. These exceptions are not parameter-matched architectural controls. VariBAD is recurrent rather than Transformer-based and matches UAT's count within 0.2% by adjusting its shared hidden width. Depth-one probes disable feed-forward computation and instead adjust hidden width; their counts include inactive allocated parameters and do not ensure equal effective functional capacity.

D Additional Experiments

This appendix details the scaling, mechanism, boundary, ablation, efficiency, and sensitivity experiments and their interpretation limits.

D.1 Scaling Analysis

Motivated by DT scaling analyses on Atari (Bhargava et al., 2024), we compare DT and UAT on HalfCheetah medium-replay at approximately 1M and 10M trainable parameters as a preliminary test of UAT at larger capacities. We increase hidden and feed-forward widths, retaining three layers and one attention head. Counts include all utility components and are matched within 2.5% at each scale.

Both scales use seeds $\{0, 1, 2\}$, context $K = 20$, batch size $B = 256$, and Table 5's locomotion optimization settings. The $\approx$ 1M rows reuse the corresponding 100k locomotion runs: the 50k entries are iteration 4 of the $10 \times 10{,}000$ run, and the 100k entries are its final checkpoint. The $\approx$ 10M models are trained separately for $50{,}000$ updates ($10 \times 5{,}000$) and evaluated at the final checkpoint. Scores are unclipped normalized returns over 100 episodes at the predefined higher target, reported as mean $\pm$ sample standard deviation across three seeds.

D.2 Interaction-order measurement: experimental details

Controls and training. Figure 2's depth-one comparison isolates first-layer retrieval under Theorem 3's affine assumptions. Both controls use one layer and head, with embedding/block LayerNorm, feed-forward computation, and residual mixing disabled. UAT-1L uses width 64, a linear utility encoder with action/hidden/output dimensions $8/32/16$, affine gates and query readout, and $\lambda = 0$. Input-augmented DT-1L linearly incorporates the same shifted action–reward feedback, unlike the main experiments' standard RTG-based DT. Its width follows Appendix C.3's allocated-parameter matching routine.

Training uses seeds 0–4, PCA-20 observations, context 40, batch size 256, and $80{,}000$ updates on the common sepsis log. AdamW uses backbone/utility learning rates $3 \times 10^{-4}/10^{-3}$ and weight decay $10^{-4}/0$, with $4{,}000$ warmup steps and cosine decay to 10% of peak. Dropout is 0.1, gradient clipping 0.25, and the gate coefficient 10^{-4}; dropout is disabled during measurement. The untrained reference uses the same UAT architecture and seeds, with token-gate weights W_s and linear query-readout weights of f_u drawn from $\mathcal{N}(0, 0.3^2)$ to activate utility, rather than retaining zero-gate initialization.

Probe construction and measurement. The probe samples 256 full length-40 training-log windows shared across models and seeds. The final position is query t; privileged stored regime labels select

distinct history slots $1 \le j, k < t$ sharing and differing from its label, respectively. Labels only select diagnostic positions and construct perturbation directions. At switch transitions, the stored label precedes the simulator's regime update, affecting slot interpretation but not the polynomial-order calculation.

For probe p, the pre-softmax margin is $F_p = \ell_{tj} - \ell_{tk} = \log(\alpha_{tj}/\alpha_{tk})$. We perturb query observation x, observation y at j, and raw reward $u = r_{j-1}$ entering shifted utility at j. Observation directions are $v_x = v_y = (\mu_1 - \mu_0)/\|\mu_1 - \mu_0\|_2$, where μ_z is the mean training-log observation with stored regime label z; the reward direction is $v_u = 1$. Recorded observation/reward standard deviations are metadata, not perturbation scales. RTG, actions, other observations/rewards, and query-side utility remain fixed. With $\lambda = 0$, the reward edit cannot propagate through an EMA to later positions. The eight-point contrast is

$$C_{3,p}(\varepsilon) = \frac{1}{8} \sum_{s_x, s_y, s_u \in \{\pm 1\}} s_x s_y s_u \, F_p(x + \varepsilon s_x v_x,\ y + \varepsilon s_y v_y,\ u + \varepsilon s_u v_u)\,. \tag{76}$$

It cancels terms of degree at most two: under the restricted affine controls, UAT's response is cubic and DT's is zero in exact arithmetic. Measurement uses 64-bit floating-point arithmetic for checkpoints, inputs, directions, projections, and differences, capturing first-layer Q/K logits before softmax with the checkpoint fixed throughout the sweep.

For model seed s, the ordinate is $A_s(\varepsilon) = 256^{-1} \sum_p |C_{3,p,s}(\varepsilon)|$. We evaluate $\varepsilon \in \{0.3, 0.1, 0.03, 0.01, 0.003, 0.001\}$ and fit $\log A_s$ against $\log \varepsilon$ on $[0.001, 0.1]$. The coefficient is the exponentiated fitted intercept; relative response is $A_s(0.1)/(256^{-1} \sum_p |F_{p,s}(0)|)$. Both aggregate absolute magnitudes and are summarized by means and population standard deviations across five model seeds. DT's slope at floating-point cancellation is not interpreted as an interaction order.

Decision diagnostics and held-out evaluation. For model seed s, diagnostic seed $1000s + 17$ selects $4{,}096$ unedited full training-log windows paired across methods, with endpoints following at least one recorded switch. We score final actions against the planning reference and NLL against logged actions. Full-window eligibility excludes zero-based endpoint times before 39. At lag 0, shifted reward comes from the preceding transition, so gains cannot be attributed to newly switched-regime feedback.

The per-lag curve separately evaluates all eligible logged windows at lags 0–12. Its thick line is a centered three-point moving average of the seed mean, with two-point endpoint averages; faint points and shading show the unsmoothed mean and seed standard deviation. Thus, uncertainty describes raw estimates. The shaded split and Early/Later annotations use the same cut: early lag $k \le 6$ and later lag $k \ge 7$. Aggregates are 0.021 ± 0.007 and -0.003 ± 0.012; throughout the interaction-order analysis, uncertainty is the population standard deviation $[S^{-1} \sum_s (x_s - \bar{x})^2]^{1/2}$ over S model seeds.

Separately, 100 fresh closed-loop rollouts per seed, in batches of at most 25, start from held-out initial-state indices and use the same deterministic six-step planner for scoring. Episode return and planning accuracy are averaged within seeds before aggregation, on a population distinct from the training-log diagnostics. Held-out return is 1.533 ± 0.402 for UAT-1L versus -1.828 ± 0.705 for DT-1L. These observations establish neither uniformly positive gains at every early lag nor faster adaptation of either model.

Scope. Trained and activated random UAT yield slopes near 3, while DT remains at round-off, supporting an architectural interaction-order distinction within the restricted single-layer class. This does not establish that cubic interactions cause decision gains, that deployed multilayer models share this decomposition, or that random/trained coefficient magnitudes measure performance effects. Training-log and held-out measurements are therefore reported separately.

D.3 Boundary Analysis

Controlled interventions. The boundary experiment shares the main controlled task's cyclical DarkRoom environment, frozen behavior log, scoring reward, and model-capacity convention. At each corruption level, UAT and DT are separately trained and evaluated with matched seeds and a common corruption stream, rather than corrupting only a clean checkpoint at inference. Runs use seeds 0–4, 20,000 updates, batch size 256, context 40, and 50 evaluation episodes in batches of 25; other settings follow Table 5's DarkRoom column. Feedback perturbations are:

(1) **Noise:** replace visible reward by $\widetilde{r}_t = r_t + \sigma s_r \xi_t$, with independent $\xi_t \sim \mathcal{N}(0,1)$ and clean training-log reward standard deviation s_r. Noise is unclipped, with $\sigma \in \{0, 0.5, 1, 2, 4\}$.
(2) **Delay:** hold (a_t, r_t) for $d \in \{0, 1, 2, 4, 8\}$ additional transitions before revealing it, preserving action–reward pairing. Until arrival, reward and utility-action inputs are zero, while ordinary action-token history retains the policy's chosen actions. The usual one-step causal shift makes $t + d + 1$ the first action that can use the pair.
(3) **Dropout:** independently zero rewards with probability $p \in \{0, 0.25, 0.5, 0.75, 1\}$. The producing action remains available without an explicit missingness mask. Dropout remains statistically distinguishable from genuine rewards because zero lies outside the implemented clean reward's support.

DT rebuilds return-to-go from corrupted rewards and decrements its remaining-return input by the visible stream. UAT instead receives the same visible action–reward pair through utility, without a return-to-go token. Environment transitions and reported returns use true rewards. Training corruption is generated once per run with seed $s+7$ and fixed across updates; closed-loop evaluation generates its own corrupted feedback. All panels reuse reward-unconditioned Vanilla-TF's clean run as reference.

Scoring and interpretation. Each point averages agreement with the privileged action reference within recorded five-index switch windows, then within episodes and across seeds. Boundary and main-table results use across-seed sample standard deviation, $[((S-1)^{-1}\sum_s (x_s - \bar{x})^2)]^{1/2}$; interaction-order results in Appendix D.2 use population standard deviation, $[S^{-1}\sum_s (x_s - \bar{x})^2]^{1/2}$. DarkRoom flags the transition advancing the goal schedule but scores its action against the pre-action goal. Thus, recorded windows begin with the preceding-goal decision and include up to four subsequent decisions, not five under the new goal. Figure 3 follows this convention.

This boundary concerns switch-window action agreement: UAT retains only a small margin at $d = 2$ (approximately 0.01), with no consistent advantage at longer tested delays. It establishes neither a universal threshold nor architectural identity. At full dropout, DT's training return-to-go is zero, but evaluation starts at 120 and never decrements it, creating a train/evaluation mismatch. UAT receives no utility reward. Convergence toward Vanilla-TF reflects this intervention rather than isolating removal of all informative feedback without other distribution changes.

D.4 Mechanistic Interpretability Analysis

We inspect trained sepsis UAT on length-40 logged windows with fixed parameters and dropout disabled. Holding the layer input and head fixed, we visualize full attention, attention without additive or multiplicative Q/K modulation, and their difference, alongside gates g_j, ρ_t, and shifted rewards r_{t-1}. EMA resets per window; regime markers are diagnostic only.

Paired inference interventions disable Q/K or V modulation across all layers. We compare reference-action probabilities over $1{,}494$ windows and five seeds at post-switch lags 0–8, averaging eligible windows within each seed and reporting the across-seed mean $\pm$ sample standard deviation.

D.5 Ablation Study

Table 9 compares components on sepsis and the three medium locomotion tasks. Each variant is trained from scratch with the full model's data, seeds, training budget, and evaluation rule. Interventions are grouped as follows:

(1) **Feedback design.** Input-only feedback retains the shifted action–reward encoder and EMA but adds its projection to observation embeddings instead of modifying Q/K/V. UAT has no return-to-go token. The *w/o reward feedback* variant zeros the utility-encoder reward input, retaining action history and action-conditioned modulation.
(2) **Modulation.** Removing query modulation zeros its additive and multiplicative corrections while retaining key/value modulation; removing key/value modulation zeros both correction types for keys and values while retaining query modulation. Both apply at every active utility layer.
(3) **State and regularization.** Setting $\lambda = 0$ replaces persistent aggregation with current shifted utility; $\beta = 0$ removes only the training objective's gate penalty.

Capacity accounting excludes permanently inactive parameters: removed query or key/value maps receive no weights, and feed-forward width is increased to match full UAT's trainable count within

Table 7: D4RL locomotion by task (mean±std; 5 seeds).

Environment	Model	By dataset			Mean
		Medium	Medium-replay	Medium-expert	
HalfCheetah	Vanilla-TF	42.1 ±0.2	31.7 ±1.4	61.4 ±1.4	45.1 ±0.8
	CQL	**42.8** ±0.5	22.9 ±1.7	42.2 ±0.6	36.0 ±0.6
	DT	42.2 ±0.2	35.0 ±1.2	82.0 ±2.2	53.1 ±1.0
	UAT	42.4 ±0.2	**37.4** ±0.6	**87.1** ±2.3	**55.6** ±0.9
Hopper	Vanilla-TF	42.1 ±1.7	22.9 ±1.5	42.3 ±8.4	35.8 ±2.9
	CQL	47.0 ±2.2	1.8 ±0.0	61.1 ±21.5	36.6 ±7.2
	DT	54.0 ±3.4	66.3 ±6.2	**74.2** ±21.7	64.9 ±8.2
	UAT	**57.5** ±3.6	**75.2** ±13.8	69.2 ±15.7	**67.3** ±8.5
Walker2d	Vanilla-TF	66.9 ±7.1	22.3 ±3.0	92.8 ±19.7	60.7 ±5.1
	CQL	53.8 ±25.8	3.2 ±2.5	92.4 ±11.6	49.8 ±7.5
	DT	**72.9** ±2.1	**61.6** ±3.6	104.2 ±2.5	**79.6** ±2.0
	UAT	71.1 ±3.2	51.3 ±5.0	**104.4** ±2.7	75.6 ±2.7

2.5% (211,638 on sepsis; ≈ 1.03M on locomotion). Existing fixed-checkpoint counterfactual code that disables Q and/or KV paths only at inference is not used for Table 9. Sepsis uses the planner-guided trajectory dataset, seeds 0–4, 80,000 updates, context 40, cosine decay to 10%, $\beta = 10^{-4}$, and $\lambda = 0.8$, matching Table 5; accuracy is agreement with the planning oracle, and Full UAT copies the main-table sepsis cells. D4RL (M) averages HalfCheetah, Hopper, and Walker2d medium scores within each seed (0–4), then reports the mean and sample standard deviation. Table 9 reports aggregate accuracy and returns.

D.6 Efficiency and Sensitivity

Efficiency. Table 3 compares Vanilla-TF, DT, and UAT with matched hardware, precision, batch size, context length, and policy-capacity convention. Measurements use fp32 (no mixed precision) on one NVIDIA RTX A6000 with PyTorch 2.4.1 (CUDA 12.1). Inference uses full context, batch size one, evaluation mode, no gradients, five warmup calls, and one latency measurement averaging 50 timed calls with CUDA synchronization. It measures model-only action prediction on fixed synthetic inputs, excluding environment simulation; throughput is $1000/\text{latency(ms)}$ actions per second. Training timing includes window sampling, forward/backward passes, optimizer/scheduler updates, and logging. Throughput is $\sum_{\text{updates}} \sum_{b,i} m_{b,i}$ divided by elapsed training time, counting valid action targets rather than padding or all three internal token types. Each of five 200-update epochs records valid-target counts and CUDA-synchronized wall-clock time; Table 3 averages throughput over epochs 3–5.

Sensitivity. Figure 4 varies λ, β, and context length individually on cyclical DarkRoom, reporting overall and post-shift action accuracy. Grids are $\lambda \in \{0, 0.5, 0.8, 0.95, 0.99\}$, $\beta \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, and $L \in \{10, 20, 40, 80, 160\}$, with reference settings $(\lambda, \beta, L) = (0.8, 10^{-4}, 40)$. Points show mean ± sample standard deviation over seeds 0–4 at Table 5's DarkRoom budget of 20,000 updates. Small reference-setting differences in overall accuracy from the main table may reflect stochastic training and finite-sample rollout evaluation. Fixed update counts mean changing L changes supervised tokens per update. In the finite-window EMA, total weight after K valid steps is $1 - \lambda^K$; increasing λ changes both older-feedback weighting and aggregate magnitude, so this is not a pure memory-length sweep.

E Additional Detailed Results

E.1 D4RL Results across tasks in Main Experiment

For per-seed metric x_s, main results report $\bar{x} = S^{-1} \sum_s x_s$ and sample standard deviation $[((S-1)^{-1} \sum_s (x_s - \bar{x})^2)]^{1/2}$, not episode-level variability. Table 2 averages HalfCheetah, Hopper, and Walker2d within each seed at a fixed dataset level; Table 7 reports completed runs by task under the same protocol. Mechanism and boundary analyses also use five seeds, following separate protocols in Appendices D.2 and D.3.

Table 8: HalfCheetah medium-replay capacity comparison (mean±std; 3 seeds; target 12000). The 1M / 50k cell is iteration 4 of the $10 \times 10{,}000$ locomotion run, not a separately trained 50k job.

Scale	Updates	Params (DT / UAT)	DT	UAT	Δ
≈ 1M	50,000	1,033,654 / 1,033,880	32.7 ±4.0	37.2 ±0.2	+4.4
≈ 1M	100,000	1,033,654 / 1,033,880	35.0 ±1.7	37.4 ±0.7	+2.4
≈ 10M	50,000	9,828,886 / 9,827,672	32.2 ±0.5	33.7 ±1.9	+1.6

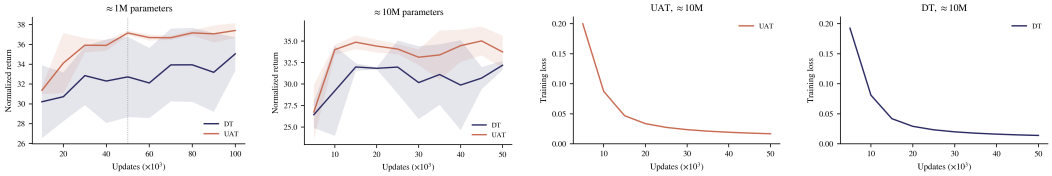


Figure 5: HalfCheetah medium-replay scaling results. **Left to right:** return curves for ≈ 1M models under the 100k locomotion protocol and ≈ 10M models trained for 50k updates, followed by training losses for UAT and DT. Return curves show mean ± sample std over seeds 0–2.

Averaged across the three dataset levels, UAT leads the evaluated methods on HalfCheetah and Hopper, scoring 55.6 and 67.3 versus DT's 53.1 and 64.9, respectively (Table 7). UAT achieves higher means than DT in six of nine environment–dataset combinations, with the largest gain on Hopper medium-replay (75.2 versus 66.3). On Walker2d, UAT modestly trails DT overall (75.6 versus 79.6), mainly due to medium-replay, while their medium-expert scores are nearly identical (104.4 versus 104.2). These results support competitive performance across locomotion tasks, with gains varying by environment and dataset level.

E.2 Scaling Analysis Results

On HalfCheetah medium-replay, UAT outperforms parameter-matched DT at both scales (Table 8). At approximately 1M parameters, its normalized-return gains are 4.4 and 2.4 after 50k and 100k updates, respectively; at 10M, the gain remains 1.6 after 50k updates. Figure 5 shows higher mean returns through most of training and declining training losses for both 10M models. DT's late upward return trend leaves open further improvement with longer training, but UAT maintains an advantage across the evaluated capacities and budgets.

E.3 Mechanistic Interpretability Analysis Results

Figure 6 illustrates how feedback modulation shapes retrieval and action prediction. **(1) Attention redistribution.** Compared with Q/K modulation disabled, full UAT assigns stronger attention to selected historical positions, particularly around $j = 19$ and $j = 29$ in the displayed window. The localized changes in $\Delta\alpha_{tj}$ show that modulation affects queries and historical positions differently. **(2) Gate dynamics.** Token gates change more abruptly with feedback, while query gates exhibit smoother transitions, consistent with window-local utility aggregation. Several gate dimensions return toward their earlier patterns when the initial regime recurs, illustrating feedback-responsive internal states. **(3) Decision dependence.** Disabling Q/K or V modulation reduces mean reference-action probability from approximately 0.6 to 0.2–0.3. This gap persists across the displayed post-switch lags with small across-seed variation, supporting the trained policy's dependence on both retrieval and value modulation.

E.4 Ablation Study Results

Table 9 shows lower mean performance under every ablation, with particularly large losses on sepsis. **(1) Feedback design.** Both reward removal and input-only conditioning sharply degrade sepsis performance, supporting the importance of feedback and its integration. **(2) Modulation.** Query and key/value modulation both contribute substantially on sepsis; query modulation matters more on D4RL. **(3) State and regularization.** Removing either component substantially degrades sepsis performance, while persistent aggregation contributes more than gate regularization on D4RL. Effects on D4RL are more heterogeneous and sometimes marginal, but all ablations still lower mean return.

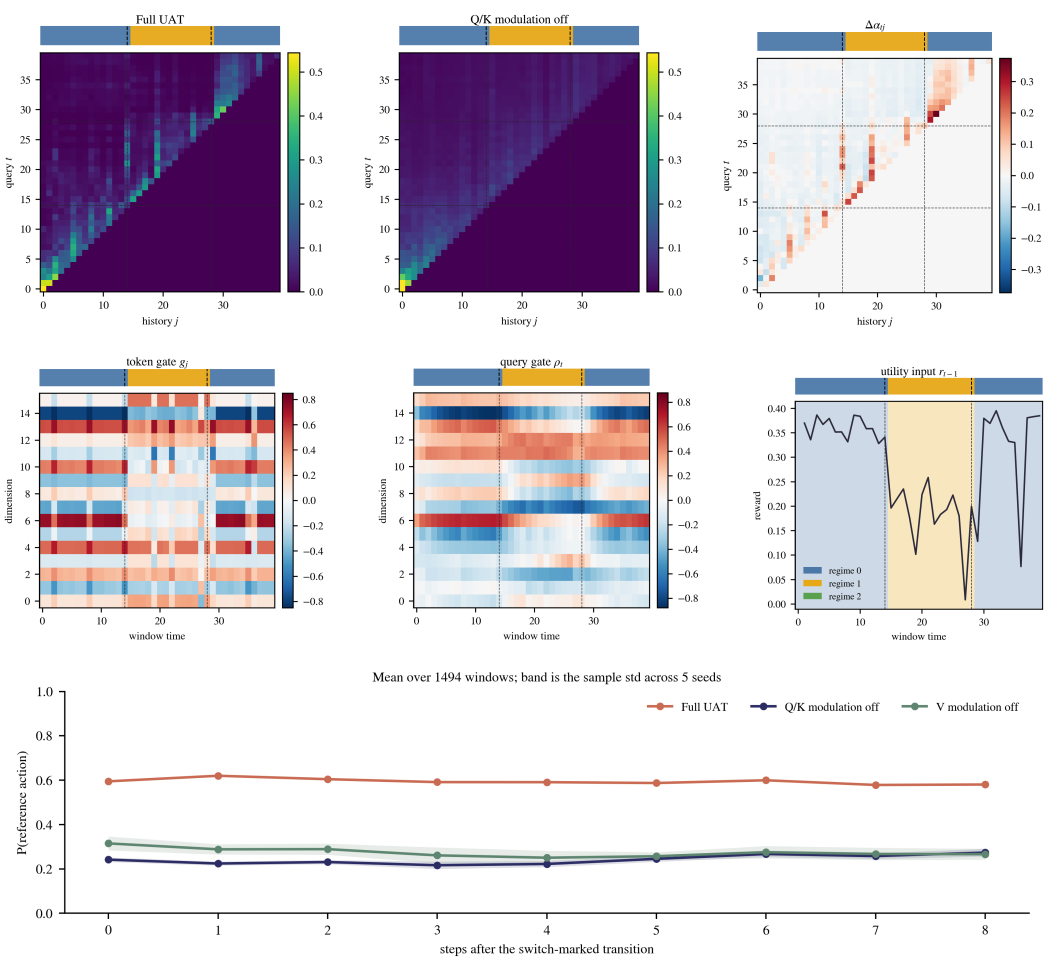


Figure 6: Feedback modulation in sepsis. **Top:** full attention, Q/K modulation off, and their difference. **Middle:** token/query gates and shifted rewards, with regime annotations. **Bottom:** reference-action probabilities under inference interventions (1,494 windows; mean ± 1 sample std across five seeds).

Table 9: Ablation results on sepsis and D4RL (mean$\pm$std; 5 seeds). D4RL (M) averages normalized returns across the three medium locomotion tasks within each seed. Bold marks the best mean.

Group	Variant	Sepsis		D4RL (M)
		Acc.↑	Return↑	Return↑
Feedback design	Input-only feedback	0.067 $\pm$0.010	-0.959 $\pm$0.159	56.905 $\pm$2.413
	w/o reward feedback	0.074 $\pm$0.007	-0.597 $\pm$0.182	49.307 $\pm$2.918
Modulation	w/o query modulation	0.077 $\pm$0.022	-0.810 $\pm$0.125	50.130 $\pm$1.815
	w/o key/value modulation	0.074 $\pm$0.011	-0.727 $\pm$0.086	54.543 $\pm$3.237
State and regularization	w/o persistent state ($\lambda = 0$)	0.077 $\pm$0.006	-0.793 $\pm$0.198	51.725 $\pm$3.516
	w/o gate regularization ($\beta = 0$)	0.074 $\pm$0.014	-0.702 $\pm$0.138	56.100 $\pm$1.608
Full	**UAT**	**0.401** $\pm$0.031	**2.027** $\pm$0.599	**56.975** $\pm$1.899

E.5 Sensitivity Check in D4RL

Figure 7 shows UAT's sensitivity to λ, β, and L on HalfCheetah medium-replay. Points report mean final-checkpoint normalized returns at target 12000 after 100,000 updates ($10 \times 10{,}000$) over seeds 0–4; shading denotes sample standard deviation. Context length causes the largest variation over the tested ranges, with longer contexts offering no consistent improvement, consistent with DarkRoom (Figure 4) and prior context-selection findings (Wu et al., 2023; Bhargava et al., 2024). Decision S4 reports lower D4RL performance with shorter backpropagation horizons (David et al., 2023).

These task-dependent variations support task-specific tuning without contradicting UAT's observed robustness in controlled feedback-informative settings.

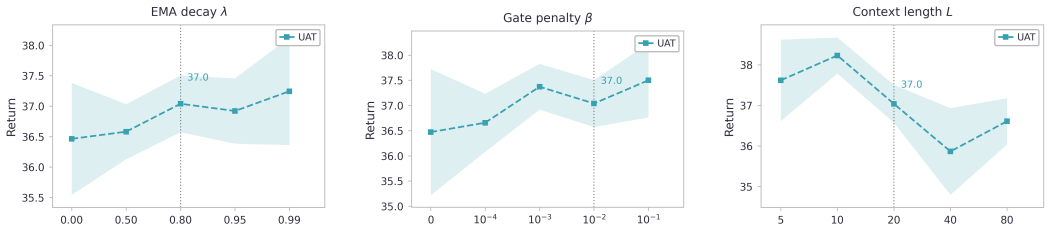


Figure 7: D4RL sensitivity to EMA decay λ, gate penalty β, and context length L on HalfCheetah medium-replay. Mean $\pm$ sample standard deviation over seeds 0–4 after 100,000 updates; the band is sample standard deviation. The metric is the final normalized return at target 12000.