Belief-Aware Decision Transformers for Offline-to-Online Decision-Making under Partial Observability: A Geosteering Case Study

Hibat Errahmen Djecta [*1,2] **Sergey Alyaev** [1] **Kristian Fossum** [1] **Reidar B. Bratvold** [2]

Abstract

Sequence-modeling approaches such as Decision Transformers learn offline policies directly from trajectories, but their internal representation of hidden state remains implicit and difficult to inspect or calibrate. We introduce a belief-aware Decision Transformer that makes hidden state explicit within a single offline sequence model: a shared transformer encoder feeds a structured belief head predicting physically meaningful hidden variables, and an action head conditioned on this belief, supervised jointly against simulator ground truth and offline actions. We exemplify the framework in geosteering, where the agent must steer a drilling trajectory through a thin reservoir layer using only noisy indirect measurements, with each decision irreversibly committing part of the well path. Comparing four architectural variants, we find that conditioning the action head on the predicted belief yields the strongest decision quality, while a strict belief bottleneck improves belief calibration at the cost of decisions. These findings suggest that belief structure should inform but not entirely constrain offline policies under structured partial observability, and that physically meaningful beliefs offer a natural interface for future offline-to-online adaptation.

1. Introduction

Many real-world decision-making problems require acting from incomplete and noisy information. In such settings, the agent does not directly observe the true state of the environment, but must infer decision-relevant hidden variables from a history of observations and previous actions. This is common in scientific, engineering, robotic, and medical domains, where online experimentation can be costly, unsafe, or practically infeasible. These problems are naturally described as partially observable sequential decision-making tasks, where the quality of a decision depends not only on the current observation, but also on the agent's internal belief about the hidden state.

Offline reinforcement learning provides an attractive framework for such domains because it allows policies to be trained from previously collected trajectories rather than through online trial-and-error. Decision Transformers (Chen et al., 2021) are a representative example of this direction: they reformulate offline reinforcement learning as conditional sequence modeling by predicting actions from histories of returns, states or observations, and previous actions. This formulation avoids unstable online interaction during training and allows the model to exploit long temporal contexts. However, in partially observable settings, the hidden state is usually only represented implicitly inside the sequence model. As a result, the model may learn actions that perform well on offline data, while its internal representation of uncertainty and hidden state remains difficult to inspect, calibrate, or adapt online.

A related line of work addresses partial observability by explicitly maintaining a belief state. Classical filtering and data-assimilation methods, such as particle filters and ensemble Kalman filters, provide structured estimates of hidden variables and uncertainty. Belief-state reinforcement learning and Partially Observable Markov Decision Process (POMDP) methods also use histories of observations to infer decision-relevant hidden information. These approaches are valuable because the belief can remain interpretable and physically meaningful. However, in many pipelines, belief estimation and decision-making are still treated as separate stages: the belief estimator is optimized for state estimation, while the policy is optimized for action quality. This separation can create a mismatch between what is estimated and what is useful for downstream decisions.

This formulation is also related to recent work on generative models for decision-making, including diffusion-based approaches that frame offline control as conditional gener-

[1]NORCE Research AS, Bergen, Norway [2]University of Stavanger, Stavanger, Norway. Correspondence to: Hibat Errahmen Djecta <hidj@norceresearch.no>.

Accepted at the 43^{rd} International Conference on Machine Learning (ICML 2026), Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning (DEMO 2026), Seoul, South Korea. PMLR 306, 2026.

ation over actions or trajectories. Rather than generating complete future trajectories, our model takes a lighter intermediate step: it explicitly predicts structured belief variables and uses them to condition action prediction. In this sense, the proposed architecture sits between return-conditioned sequence modeling and full generative planning.

The present work builds on three converging lines of research. Sequence-modeling offline reinforcement learning, including the Decision Transformer (Chen et al., 2021) and Trajectory Transformer (Janner et al., 2021), treats offline policy learning as conditional sequence generation but typically operates on raw observations or unstructured latent histories. Recent auxiliary-loss approaches such as NextLat (Teoh et al., 2025) and TD-JEPA (Bagatella et al., 2025) shape transformer hidden states toward belief-state-like quantities, but the resulting belief remains an abstract latent rather than an interpretable physical variable. In parallel, belief-state learning for POMDPs has produced explicit Bayesian filters (Kaelbling et al., 1998), variational belief inference (Igl et al., 2018), and task-aware joint belief-policy training such as BMIL (Gangwani et al., 2019), the closest neighbor to our approach. BMIL learns abstract task-aware belief representations through imitation, whereas our framework commits to a physically meaningful belief signature with simulator supervision, explicitly contrasts auxiliary, coupled, and bottleneck variants of belief usage within an offline sequence model, and validates the structured belief as an interface for offline-to-online adaptation. In the geosteering domain specifically (see Appendix B for geosteering background), decision-theoretic and ensemble-based approaches (Kullawan et al., 2014; Alyaev et al., 2019) have been complemented by deep reinforcement learning with particle filters (Muhammad et al., 2025; 2026) and, in a prior work, by a dual-network Deep Reinforcement Learning (DRL) agent (Djecta et al., 2025a) and a Decision Transformer trained on its trajectories (Djecta et al., 2025b). The present paper builds directly on this dataset and architecture but moves beyond raw-observation sequence modeling by introducing an explicit, physically grounded belief representation jointly trained with the action head.

In this paper, we propose a belief-aware sequence decision model for offline decision-making under structured partial observability. The model uses a shared sequence encoder with two coupled heads: a belief head that predicts interpretable hidden variables, and an action head that predicts the next decision conditioned on the learned belief. The belief variables are domain-specific but physically or semantically meaningful, such as distances to hidden boundaries, object locations, anatomical landmarks, or other quantities that influence decisions. This design makes the belief not only an auxiliary prediction, but part of the decision pathway.

A key motivation for this architecture is the offline-to-online transition. During offline training, structured beliefs can be supervised using simulation, retrospective labels, or domain-specific inference outputs. During deployment, the same belief variables can provide an interface for online correction using new observations, measurement consistency, or existing filtering tools, without retraining the full policy. This makes the architecture especially relevant for domains where policies must be trained offline but adapted cautiously as new evidence becomes available.

We instantiate the proposed framework in geosteering, an engineering decision-making problem where the agent must steer a drilling trajectory using noisy and indirect measurements while the true geological structure remains hidden. Geosteering provides a representative testbed for structured partial observability: the hidden state has physically meaningful components, online interaction is costly and irreversible, and simulator ground truth can be used to supervise belief variables during offline training. Building on an existing Decision Transformer setup trained from uncertainty-aware trajectories, we study whether adding an explicit structured belief branch can make the decision model more interpretable and better aligned with the hidden physical state.

Our contributions are summarized as follows:

- We formulate a generic belief-aware sequence modeling framework for offline decision-making under structured partial observability.
- We propose a two-head architecture that jointly predicts a structured belief and an action, coupling belief estimation with decision prediction.
- We empirically validate the structured belief as an interface for offline-to-online adaptation under controlled correction.
- We instantiate the framework in geosteering, using physically meaningful reservoir-boundary beliefs as the structured representation.

2. Methodology

This section presents the proposed belief-aware sequence decision model. We first formulate the generic offline decision-making setting under partial observability, then describe the two-head architecture that jointly predicts a structured belief and an action. The method is formulated independently of a specific application domain; the domain-specific instantiation is introduced later through the choice of belief variables and evaluation metrics.

We consider offline sequential decision-making under partial observability. At each timestep t, an agent receives an

observation o_t, selects an action a_t, and receives a reward r_t. The true environment state s_t is not directly observed. Instead, the agent must rely on a history of observations and previous actions to infer the hidden information that is relevant for decision-making.

The offline dataset consists of T tuples collected from previous policies, simulations, or historical records:

$$\mathcal{D} = \{(o_t, a_t, r_t, R_t)\}_{t=1}^{T}, \tag{1}$$

where $o_t \in \mathcal{O}$ is the observation, $a_t \in \mathcal{A}$ the action, $r_t \in \mathbb{R}$ the reward, and $R_t = \sum_{i \geq t} r_i$ the return-to-go from timestep t. The goal is to learn a policy from $\mathcal{D}$ in (1) without requiring costly or unsafe online exploration during training.

We focus on a class of partially observable problems where the hidden state admits a low-dimensional, physically meaningful belief representation. We denote this belief by b_t. Unlike an arbitrary latent embedding, b_t is defined through interpretable variables that summarize decision-relevant hidden information. The exact belief variables are domain-specific, but they should correspond to quantities that are meaningful for the decision problem and, when possible, can be supervised in simulation or estimated from retrospective data.

A common pipeline in such settings separates belief estimation and decision-making:

$$\begin{aligned} &\text{observations} \rightarrow \text{belief estimation} \\ &\qquad \rightarrow \text{decision model} \rightarrow \text{action}. \end{aligned} \tag{2}$$

The separation in (2) is useful because the belief can remain interpretable and compatible with domain-specific inference tools. However, it can also create a mismatch between the belief-estimation objective and the downstream decision objective. A belief may be well calibrated as a state estimate, but not necessarily optimized for the decisions that follow.

In this work, we explore whether belief estimation and decision-making can instead be learned within the same offline sequence model. The central hypothesis is that coupling belief prediction and action prediction can produce internal beliefs that remain interpretable while also being useful for downstream decision-making.

2.1. Belief-Aware Sequence Decision Model

Our architecture builds on the Decision Transformer formulation, where offline reinforcement learning is treated as conditional sequence modeling. A standard Decision Transformer predicts actions from a sequence of returns, observations, and previous actions. This provides a supervised learning framework for offline decision-making, while avoiding direct online interaction during training.

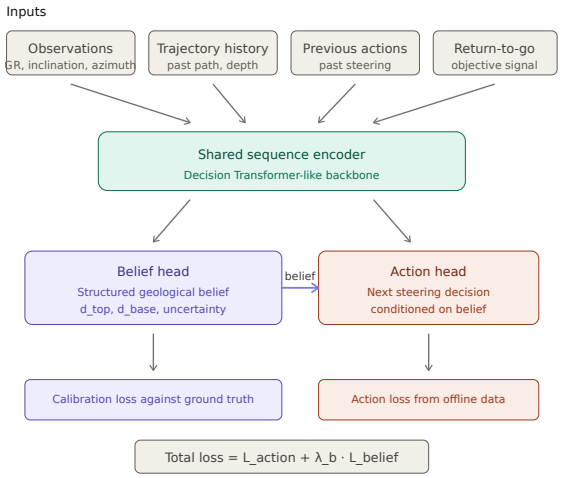

Figure 1. Proposed belief-aware sequence decision model. A shared sequence encoder processes the trajectory context τ_t defined in (3) (illustrated as separate input streams) and produces the hidden representation h_t. Two coupled heads operate on h_t: a structured belief head g_ϕ predicting $\hat{b}_t$ (here, reservoir-boundary distances d^{top}, d^{base}, with uncertainty as an optional extension) and an action head π_ψ predicting the next action $\hat{a}_t$ conditioned on $\hat{b}_t$. The total loss combines action and belief terms as in (11).

Given a context window of length $K \in \mathbb{N}$, we define the trajectory context at timestep t as

$$\tau_t = \{(R_i, o_i, a_i)\}_{i=t-K+1}^{t}, \tag{3}$$

where R_i, o_i, and a_i denote the return-to-go, observation, and previous action at timestep i. A shared sequence encoder f_θ with parameters θ maps the trajectory context τ_t in (3) into a hidden representation

$$h_t = f_\theta(\tau_t), \quad h_t \in \mathbb{R}^{d_h}, \tag{4}$$

where d_h is the hidden dimension and f_θ is a transformer-based temporal encoder. The hidden representation h_t in (4) is the substrate on which the two heads operate.

Figure 1 illustrates the proposed two-head architecture. Instead of predicting only the next action from h_t, the model produces two coupled outputs through specialized heads. A belief head g_ϕ with parameters ϕ predicts a structured belief $\hat{b}_t \in \mathbb{R}^d$:

$$\hat{b}_t = g_\phi(h_t). \tag{5}$$

An action head π_ψ with parameters ψ predicts the next action conditioned on both the hidden representation and the predicted belief:

$$\hat{a}_t = \pi_\psi(h_t, \hat{b}_t). \tag{6}$$

The action head in (6) is conditioned on both h_t and $\hat{b}_t$ from (5). This coupling makes the belief part of the decision pathway, rather than treating it only as an auxiliary prediction.

The model therefore learns not only what action to take, but also an explicit representation of the hidden quantities that support the decision. A stricter variant of the architecture removes the direct connection from h_t to the action head, so that

$$\hat{a}_t = \pi_\psi(\hat{b}_t), \tag{7}$$

forcing the action to be selected entirely through the structured belief bottleneck. The coupled version in (6) is more flexible, while the strict bottleneck in (7) is more interpretable.

2.2. Domain-Specific Belief Signature

The structured belief $\hat{b}_t \in \mathbb{R}^d$ is defined by a domain-specific signature

$$\hat{b}_t = [\hat{z}_{1,t}, \hat{z}_{2,t}, \ldots, \hat{z}_{d,t}], \tag{8}$$

where each component $\hat{z}_{j,t}$ in the signature (8) corresponds to a hidden variable relevant for decision-making and d is the belief dimension. The architecture does not prescribe what these variables must be; the belief signature is chosen based on the structure of the domain. When uncertainty is informative, each component can be augmented with a predicted standard deviation $\hat{\sigma}_{j,t}$, so that the belief carries both per-dimension means and uncertainties. The key requirement is that belief variables be meaningful for the decision problem, rather than arbitrary latent dimensions.

2.3. Offline Training Objective

The model is trained from offline trajectories with a joint objective that combines action prediction and belief calibration. We assume access to ground-truth structured beliefs $b_t^\star \in \mathbb{R}^d$, obtained from simulation or retrospective annotation, that share the dimensionality of the predicted belief $\hat{b}_t$. The action loss is the standard Decision Transformer behavior-cloning term, which uses the squared ℓ_2 norm to penalize action prediction error:

$$\mathcal{L}_{action} = \|a_t - \hat{a}_t\|_2^2, \tag{9}$$

and can be replaced by cross-entropy for discrete actions. The belief head is supervised against the ground-truth signature through a supervised regression loss

$$\mathcal{L}_{belief} = \left\| b_t^\star - \hat{b}_t \right\|_2^2. \tag{10}$$

When the belief signature includes predicted uncertainties, this term is naturally replaced by a Gaussian negative log-likelihood that simultaneously trains predicted means and variances. The full training objective combines action and belief losses with a non-negative scalar weight $\lambda_b \geq 0$:

$$\mathcal{L} = \mathcal{L}_{action} + \lambda_b \mathcal{L}_{belief}. \tag{11}$$

The coefficient λ_b balances action prediction and belief regression. Because the action head is conditioned on $\hat{b}_t$, gradients from $\mathcal{L}_{action}$ flow back through the belief, encouraging it to retain information useful for downstream decisions. At the same time, $\mathcal{L}_{belief}$ anchors the representation to physically meaningful quantities, keeping the belief interpretable.

2.4. Offline-to-Online Perspective

Although the model is trained offline, the structured belief $\hat{b}_t$ provides a natural interface for online adaptation. At deployment, the true hidden state is unavailable, but new observations arrive sequentially. Because the belief components have physical or semantic meaning, they can be updated from new evidence using measurement-consistency criteria or existing filtering and data-assimilation tools, without retraining the full policy. Adaptation therefore focuses on correcting the structured belief that conditions the action head, rather than re-fitting the encoder f_θ or the action head π_ψ:

$$\begin{aligned} &\text{offline trajectories} \rightarrow \text{structured belief learning} \\ &\rightarrow \text{policy} \rightarrow \text{online belief update on } \hat{b}_t \\ &\rightarrow \text{adapted action}. \end{aligned} \tag{12}$$

The pipeline in (12) is particularly useful in domains where online interaction is costly, unsafe, or limited. The full offline training procedure is summarized in Algorithm 1 (Appendix A). We empirically evaluate this interface in Section 3.4 under a controlled correction protocol.

2.5. Evaluation Criteria

The method is evaluated along two complementary dimensions.

First, decision quality measures whether the learned policy produces good actions under partial observability. Depending on the domain, this may include cumulative return, task success rate, constraint violations, safety metrics, or other application-specific objectives.

Second, belief quality measures whether the structured belief captures meaningful hidden variables. This can be evaluated using prediction error against ground-truth structured variables when available, uncertainty calibration, or consistency with new observations.

The central hypothesis is that coupling belief prediction and action prediction within the same offline sequence model can produce beliefs that are both interpretable and useful for decision-making.

3. Experiments

This section describes the experimental setup used to instantiate the proposed belief-aware sequence decision model in geosteering. The experiments are designed to evaluate whether the proposed architecture can preserve decision quality while making the hidden geological belief explicit. We focus on three questions: (i) whether the model can learn effective steering decisions from offline trajectories, (ii) whether the belief head predicts meaningful reservoir-boundary quantities, and (iii) whether coupling the belief and action heads affects decision quality.

3.1. Setup and Dataset

We evaluate the framework in a synthetic geosteering environment (see Appendix B for geosteering background). At each decision step, the agent observes trajectory and logging information and selects a steering action that affects the future well path. The true geological structure is hidden from the policy during decision-making, but simulator ground truth is available for constructing belief targets during training. The observation vector includes directional and geological measurements such as measured depth, true vertical depth, inclination, azimuth, spatial coordinates, and gamma-ray measurements. The action corresponds to a steering decision, represented as changes in inclination and azimuth, $a_t = (\Delta\mathrm{INCL}_t, \Delta\mathrm{AZIM}_t)$. The decision objective is to keep the trajectory inside the target reservoir interval while maintaining a smooth and feasible well path.

The offline dataset is generated from trajectories produced by an uncertainty-aware geosteering agent. Each recorded transition is a tuple $(o_t, a_t, r_t, o_{t+1}, R_t)$, where o_t is the agent's observation at timestep t, o_{t+1} the observation after acting, and R_t the return-to-go. The dataset contains approximately 348k records from around 20k simulated trajectories, with each trajectory divided into overlapping windows τ_t as in (3) of fixed context length K, following the Decision Transformer sequence format. We use the same train/validation/test split (80/10/10) across all model variants.

For belief-aware models, target beliefs are computed from the ground-truth reservoir boundaries available in simulation. The structured belief target is $b_t^\star = [d_t^{top}, d_t^{base}] \in \mathbb{R}^2$, where d_t^{top} and d_t^{base} denote the signed distances from the current well position to the top and base reservoir boundaries, respectively. These quantities provide a compact and physically interpretable representation of the hidden geological state.

3.2. Models and Training

We compare four model variants that differ in how the belief is used by the action head. The standard Decision Transformer serves as the reference. The Parallel Two-Head model adds the belief head (5) as an auxiliary task but does not use $\hat{b}_t$ in the action head. The Coupled Two-Head model uses (6), conditioning the action head on the predicted belief. The Strict Bottleneck variant uses (7), predicting actions only through the structured belief. Table 1 summarizes these variants. This comparison isolates three effects: the value of adding belief supervision, the effect of explicitly conditioning actions on the learned belief, and the performance trade-off introduced by a hard structured bottleneck.

Table 1. Compared model variants.

Model	Belief Prediction	Action Input
Decision Transformer	No	h_t
Parallel Two-Head	Yes	h_t
Coupled Two-Head	Yes	$(h_t, \hat{b}_t)$
Strict Bottleneck	Yes	$\hat{b}_t$

Table 2. Experimental configuration.

Category	Value
Dataset size	∼348k records
Episodes	∼20k simulated trajectories
Train/Val/Test split	80/10/10
Random seeds	4
Test trajectories	∼2,000
Context length K	20
Hidden size	128
Feed-forward size	512
Attention heads	2
Dropout	0.1
Optimizer	Adam
Learning rate	1×10^{-4}
Batch size	64
Epochs	200
Action loss	MSE
Belief loss	MSE

All models are trained offline by minimizing the joint objective in (11), with the same training budget and architecture hyperparameters to ensure a fair comparison. The default configuration follows the previous Decision Transformer setup (Djecta et al., 2025b): context length $K = 20$, hidden size 128, feed-forward size 512, two attention heads, Adam optimizer, Mean Squared Error (MSE) losses for both action and belief heads. Each model variant is trained with 4 random seeds, and we report mean and standard deviation of all metrics across seeds. Evaluation is performed on the held-out test set, corresponding to approximately 2,000 trajectories. Full hyperparameter details are summarized in Table 2.

Table 3. Decision-making performance on held-out trajectories.

Model	Val. MSE ↓	RCR ↑	Violations ↓
Decision Transformer	0.021 ± 0.006	0.72 ± 0.11	8.4 ± 2.7
Parallel Two-Head	0.020 ± 0.005	0.73 ± 0.10	8.0 ± 2.5
Coupled Two-Head	0.017 ± 0.004	0.78 ± 0.09	6.3 ± 2.1
Strict Bottleneck	0.024 ± 0.007	0.69 ± 0.12	9.1 ± 3.0

Table 4. Belief prediction quality for structured belief models. Errors are reported as mean ± standard deviation across held-out test trajectories.

Model	Top Error ↓	Base Error ↓	Mean Error ↓
Parallel Two-Head	0.18 ± 0.06	0.21 ± 0.07	0.20 ± 0.06
Coupled Two-Head	0.15 ± 0.05	0.18 ± 0.06	0.17 ± 0.05
Strict Bottleneck	0.14 ± 0.05	0.17 ± 0.06	0.16 ± 0.05

3.3. Evaluation and Results

We evaluate the models using both decision-quality and belief-quality metrics. Decision quality is measured using validation action MSE, Reservoir Contact Ratio (RCR), and boundary violations. RCR measures the proportion of the predicted trajectory that remains inside the target reservoir interval, and is the main domain-specific metric for long-horizon steering quality. Belief quality is evaluated only for belief-aware models through the prediction error of the top and base reservoir-boundary distances:

$$\mathrm{BE}_k = \frac{1}{N}\sum_{t=1}^{N}\left|d_t^k - \hat{d}_t^k\right|, \quad k \in \{\mathrm{top}, \mathrm{base}\}, \qquad (13)$$

where N is the number of test timesteps. We additionally report the mean error $\mathrm{BE}_{\mathrm{mean}} = \frac{1}{2}(\mathrm{BE}_{\mathrm{top}} + \mathrm{BE}_{\mathrm{base}})$. These metrics test whether the learned belief corresponds to meaningful hidden geological quantities rather than an arbitrary auxiliary representation.

Table 3 reports decision quality across all four model variants, and Table 4 reports belief prediction quality for the three belief-aware models.

3.4. Offline-to-Online Belief Correction

A central claim of the proposed architecture is that the structured belief $\hat{b}_t$ provides an interface for online correction without retraining. We test this on a shifted test set (∼500 trajectories) where the gamma-ray noise variance is doubled relative to training, simulating sensor degradation.

At a fraction $p \in \{0, 0.05, 0.10, 0.20\}$ of randomly selected timesteps, the predicted belief $\hat{b}_t$ is replaced by the simulator ground truth $b_t^\star$ before being passed to the action head; the encoder f_θ and action head π_ψ remain frozen. This ground-truth replacement is an idealized upper bound on what online updates of $\hat{b}_t$ alone can achieve, and isolates the value of the belief interface from the choice of update mechanism.

Table 5 shows two patterns. The Coupled variant retains its offline advantage at $p = 0$, since access to h_t provides robustness when the belief is unreliable. The Strict Bottleneck variant improves more steeply with p and catches up by $p \geq 0.20$, because its action depends entirely on $\hat{b}_t$ and therefore translates corrections directly into improved decisions. The coupled design is preferable without an online interface; the bottleneck design becomes preferable as belief updates become available. This provides a first empirical validation of the structured belief as an offline-to-online interface.

4. Discussion and Limitations

Across the four variants in Tables 3 and 4, three patterns emerge. First, adding belief supervision without using the belief for action prediction (Parallel Two-Head) provides only marginal improvement over the baseline Decision Transformer, suggesting the auxiliary loss mainly acts as regularization. Second, conditioning actions on the predicted belief (Coupled Two-Head) yields substantially better RCR and fewer boundary violations, indicating that performance gains come from integrating the belief into the decision pathway. Third, forcing actions to depend only on the structured belief (Strict Bottleneck) improves belief prediction accuracy but reduces decision quality, since the low-dimensional belief does not capture all decision-relevant information contained in h_t.

The offline-to-online belief correction experiment in Section 3.4 further clarifies this trade-off. Offline, the strict bottleneck performs worst because the belief signature is incomplete; online, it becomes the most responsive to belief correction since updates to $\hat{b}_t$ directly affect actions. The coupled variant is more robust offline, but its contin-

Table 5. RCR (↑) on the shifted test set as a function of the belief correction rate p. Mean ± std over 4 seeds. Here the Decision Transformer has no belief variable and is therefore evaluated only at p = 0.

Model	$p = 0$	$p = 0.05$	$p = 0.10$	$p = 0.20$
Decision Transformer	0.61 ± 0.10	—	—	—
Coupled Two-Head	0.66 ± 0.09	0.69 ± 0.08	0.72 ± 0.08	0.75 ± 0.07
Strict Bottleneck	0.58 ± 0.11	0.64 ± 0.10	0.70 ± 0.09	0.76 ± 0.08

ued reliance on h_t limits the impact of corrected beliefs. This suggests that offline and online settings favor different levels of belief–policy coupling. The online correction experiment uses ground-truth belief replacement as a controlled upper bound. Realistic online updates based on measurement-consistency losses or particle filters over the structured belief are left to future work.

Overall, the results indicate that structured beliefs should guide, but not fully constrain, offline decision policies. The coupled two-head architecture provides the best balance between interpretability and access to richer contextual information. In domains where the belief representation is closer to a sufficient statistic for action, stricter bottlenecking may become more effective.

The framework depends on the availability of meaningful belief targets during training. In the geosteering instantiation, simulator ground truth provides structured labels such as distances to reservoir boundaries; this may not be available in all domains. For applications such as surgery, healthcare, or robotics, defining the appropriate belief signature may require domain expertise, simulation environments, or retrospective annotations. The architecture also assumes that the decision-relevant hidden state can be summarized by a relatively low-dimensional structured belief, which may be restrictive in settings where the hidden state is weakly structured. A linear probe on the frozen Decision Transformer hidden state would help disentangle whether the belief head adds new representational capacity or merely makes already-encoded information explicit; we leave this analysis to future work. Finally, the geosteering experiments should be interpreted as a testbed demonstration rather than evidence of generality. Broader validation under stronger distribution shifts, real-world measurements, and additional structured POMDP domains is needed.

5. Conclusion

This paper presented a belief-aware Decision Transformer for offline decision-making under structured partial observability. The model uses a shared sequence encoder with two coupled heads, a structured belief head and an action head, jointly trained to make hidden decision-relevant variables explicit while preserving the supervised sequence-modeling formulation of Decision Transformers. The architecture sits between standard return-conditioned sequence models and generative trajectory methods, offering a lighter intermediate that keeps the belief interpretable and physically grounded.

We instantiated the framework in geosteering, where decisions must be made from noisy measurements while the true geological structure remains hidden, and structured beliefs naturally correspond to reservoir-boundary distances. Comparing four architectural variants showed that conditioning the action head on the predicted belief gives the strongest decision quality, while a strict belief bottleneck improves belief calibration at the cost of decisions, suggesting that belief structure should inform but not entirely constrain offline policies.

A controlled correction experiment further showed that updating only the structured belief at deployment recovers decision quality lost to distribution shift. Future work will focus on realistic online update mechanisms and on validating the approach on broader structured POMDP domains.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning, with a specific focus on offline decision-making under structured partial observability. The proposed framework is instantiated in geosteering, where improved decision quality could reduce energy waste during drilling and improve operational safety. The general framework may also apply to other domains involving online experimentation. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

H.E. Djecta, S. Alyaev, K. Fossum, and R.B. Bratvold acknowledge the support from the project DISTINGUISH (Decision support using neural networks to predict geological uncertainties when geosteering), funded by Aker BP, Equinor, and the Research Council of Norway (RCN PETROMAKS2 project no. 344236).

The authors thank ROGII Inc. for providing academic licenses for Solo Cloud, StarSteer, and related data used to train the DRL agent that generated the dataset for this study.

References

Alyaev, S., Suter, E., Bratvold, R. B., Hong, A., Luo, X., and Fossum, K. A decision support system for multi-target geosteering. *Journal of Petroleum Science and Engineering*, 183:106381, 2019. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2019.106381. URL https://www.sciencedirect.com/science/article/pii/S0920410519308022.

Bagatella, M., Pirotta, M., Touati, A., Lazaric, A., and Tirinzoni, A. Td-jepa: Latent-predictive representations for zero-shot reinforcement learning, 2025. URL https://arxiv.org/abs/2510.00739.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021. URL https://arxiv.org/abs/2106.01345.

Djecta, H. E., Alyaev, S., Fossum, K., Bratvold, R. B., Muhammad, R. B., and Srivastava, A. Uncertainty-aware well placement: Simulator-verified dual-network reinforcement learning approach meets particle filters. In *Computational Science – ICCS 2025 Workshops*, pp. 188–202. Springer Nature Switzerland, 2025a. URL https://doi.org/10.1007/978-3-031-97554-7_14.

Djecta, H. E., Alyaev, S., Fossum, K., Bratvold, R. B., and Sui, D. Geosteering through the lens of decision transformers: Toward embodied sequence decision-making. In *NeurIPS 2025 Workshop on Embodied World Models for Decision Making*, 2025b. URL https://openreview.net/forum?id=QXLWeLJ0ub.

Gangwani, T., Lehman, J., Liu, Q., and Peng, J. Learning belief representations for imitation learning in pomdps, 2019. URL https://arxiv.org/abs/1906.09510.

Igl, M., Zintgraf, L., Le, T. A., Wood, F., and Whiteson, S. Deep variational reinforcement learning for pomdps, 2018. URL https://arxiv.org/abs/1806.02426.

Janner, M., Li, Q., and Levine, S. Reinforcement learning as one big sequence modeling problem. *CoRR*, abs/2106.02039, 2021. URL https://arxiv.org/abs/2106.02039.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. 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.

Kullawan, K., Bratvold, R., and Bickel, J. E. A decision analytic approach to geosteering operations. *SPE Drilling & Completion*, 29(01):36–46, 03 2014. URL https://doi.org/10.2118/167433-PA.

Muhammad, R. B., Srivastava, A., Alyaev, S., Bratvold, R. B., and Tartakovsky, D. M. High-precision geosteering via reinforcement learning and particle filters. *Computational Geosciences*, 29(2), 2025. ISSN 1573-1499. doi: 10.1007/s10596-025-10352-y. URL http://dx.doi.org/10.1007/s10596-025-10352-y.

Muhammad, R. B., Alyaev, S., and Bratvold, R. B. Optimal sequential decision-making in geosteering: A reinforcement learning approach. *Geoenergy Science and Engineering*, 258:214304, 2026. ISSN 2949-8910. doi: https://doi.org/10.1016/j.geoen.2025.214304. URL https://www.sciencedirect.com/science/article/pii/S2949891025006621.

Teoh, J., Tomar, M., Ahn, K., Hu, E. S., Sharma, P., Islam, R., Lamb, A., and Langford, J. Next-latent prediction transformers learn compact world models, 2025. URL https://arxiv.org/abs/2511.05963.

A. Training Procedure

This appendix details the offline training procedure for the belief-aware sequence decision model. At each iteration, a minibatch of trajectory windows is sampled from the offline dataset $\mathcal{D}$, the shared encoder produces hidden representations, both heads are evaluated, and parameters are updated by minimizing the joint objective in (11). Algorithm 1 summarizes the full procedure.

Algorithm 1 Offline Training of the Belief-Aware Sequence Decision Model

Input: Offline dataset $\mathcal{D}$, context length K, belief weight λ_b, model parameters θ, ϕ, ψ
Initialize: sequence encoder f_θ, belief head g_ϕ, action head π_ψ
for each training epoch **do**
  for each minibatch sampled from $\mathcal{D}$ **do**
    Sample trajectory windows $\tau_t = \{(R_i, o_i, a_i)\}_{i=t-K+1}^{t}$ with corresponding target action a_t and target belief $b_t^\star$.
    Encode the trajectory context: $h_t = f_\theta(\tau_t)$
    Predict the structured belief: $\hat{b}_t = g_\phi(h_t)$
    Predict the action conditioned on the hidden representation and belief: $\hat{a}_t = \pi_\psi(h_t, \hat{b}_t)$
    Compute the action loss:

$$\mathcal{L}_{action} = \|a_t - \hat{a}_t\|_2^2$$

    Compute the belief regression loss:

$$\mathcal{L}_{belief} = \left\| b_t^\star - \hat{b}_t \right\|_2^2$$

    Compute the joint objective:

$$\mathcal{L} = \mathcal{L}_{action} + \lambda_b \mathcal{L}_{belief}$$

    Update θ, ϕ, ψ by backpropagation through the full model.
  end for
end for
Output: Trained encoder f_θ, belief head g_ϕ, and action head π_ψ

B. Geosteering Background

This appendix provides additional context on geosteering for readers unfamiliar with the domain. Our goal is to motivate why geosteering is a representative testbed for structured partially observable decision-making, rather than to give a comprehensive overview of the field.

B.1. The geosteering problem

Geosteering is the real-time process of adjusting the trajectory of a horizontal well during drilling, with the goal of keeping the well inside a target reservoir layer. A horizontal well typically extends several kilometers laterally through a thin geological layer (often only a few meters thick) that contains hydrocarbons. Staying inside this layer maximizes reservoir contact and well productivity; exiting the layer reduces production and may render parts of the well unusable. Decisions are made every few meters of drilling progress, and each decision irreversibly commits part of the well path. Figure 2 illustrates this real-time decision loop: LWD sensors gather measurements, a geosteering team analyses the data to infer the surrounding stratigraphic structure, and a steering decision is made to keep the well within the target reservoir layer.

The decision-relevant state of the system is the local geological structure surrounding the drill bit: in particular, the position of the layer boundaries (top and base of the reservoir), the lithology, and any local geological features such as faults. This state is not directly observable. Instead, the operator receives indirect, noisy measurements through logging-while-drilling (LWD) sensors, including gamma-ray (GR) logs that capture the natural radioactivity of the surrounding rock and indirectly indicate layer transitions; inclination and azimuth, which describe the orientation of the drill bit; and measured depth (MD) and true vertical depth (TVD), which describe the position of the bit along the well path and its vertical position. Geosteering is therefore a partially observable problem: the agent perceives a noisy projection of the hidden subsurface state and must infer where the boundaries are in order to steer.

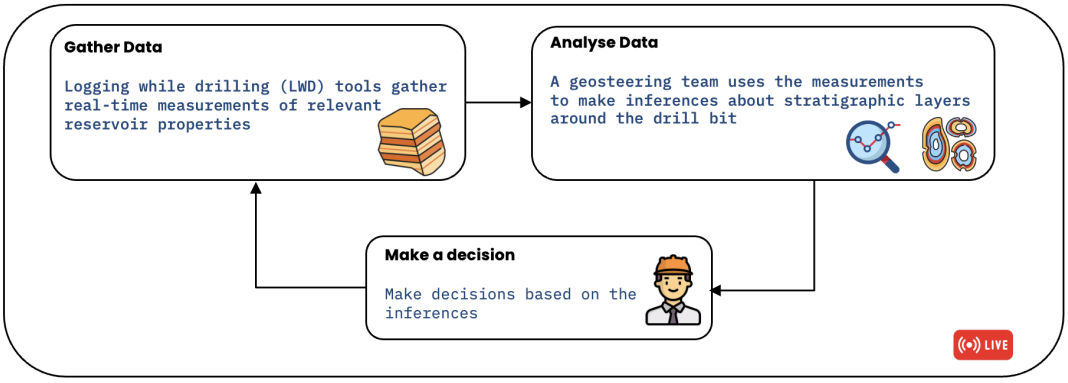

Figure 2. The geosteering decision loop. At each drilling step, logging-while-drilling (LWD) sensors provide indirect measurements of the surrounding geology. A geosteering team analyses these measurements to form inferences about the stratigraphic layer structure around the drill bit. A steering decision is then made to keep the well trajectory within the target reservoir layer (highlighted in amber), after which new measurements are gathered and the cycle repeats. The dashed orange line shows the well path threading through the reservoir between its top and base boundaries.

At each decision step, the operator selects a steering action that adjusts the drilling direction. In our formulation, this corresponds to a small change in inclination and azimuth, as defined in the main text. These actions are mechanically constrained: dogleg severity and trajectory smoothness limit how aggressively the well can turn. Crucially, decisions are irreversible: once a section of the well has been drilled, it cannot be undrilled. If the well exits the reservoir, recovery requires careful re-entry trajectories that further constrain the remaining well path.

B.2. Why geosteering is a structured POMDP

Geosteering exhibits all the structural properties that motivate this work. The system is partially observable through noisy, indirect sensors. The decision-relevant belief state is low-dimensional and physically meaningful, corresponding to the distances to layer boundaries and their uncertainty. Actions are irreversible, committing the agent to long-horizon consequences. Online experimentation is costly, since real wells cost on the order of millions of dollars and cannot be repeated. Finally, simulator ground truth is available, which enables offline training with supervised belief targets. These properties also appear in other domains such as surgical sub-task control, Intensive Care Unit (ICU) treatment planning, and autonomous navigation under occlusion, supporting the broader applicability of the framework introduced in this paper.

The primary domain-specific evaluation metric in our experiments is the Reservoir Contact Ratio (RCR): the proportion of the executed well trajectory that lies within the target reservoir layer. RCR captures the long-horizon decision objective directly. A high RCR indicates that the agent has successfully kept the well inside the productive zone for most of its length, while a low RCR indicates frequent or sustained excursions outside the target layer.

B.3. Simulation environment

The synthetic geosteering environment used in this paper, originally introduced in a prior work (Djecta et al., 2025a), generates 2D vertical cross-sections through layered subsurface formations with stochastic layer geometry, lithology, and noisy gamma-ray logs. At each timestep the simulator returns the current observation vector and, for training purposes only, the ground-truth distances to the top and base reservoir boundaries. Trajectories were collected by running a previously trained dual-network DRL agent coupled with a particle filter, yielding the offline dataset of approximately 348,000 transitions across 20,000 simulated wells used in this paper.