Geosteering Through the Lens of Decision Transformers: Toward Embodied Sequence Decision-Making

Hibat Errahmen Djecta*
NORCE Research AS
Bergen, Norway
hidj@norceresearch.no

Sergey Alyaev
NORCE Research AS
Bergen, Norway
saly@norceresearch.no

Kristian Fossum
NORCE Research AS
Bergen, Norway
krfo@norceresearch.no

Reidar B. Bratvold
University of Stavanger
Stavanger, Norway
reidar.bratvold@uis.no

Dan Sui
University of Stavanger
Stavanger, Norway
dan.sui@uis.no

Abstract

Intelligent agents in embodied environments face the challenge of acting under uncertainty, where every decision both responds to incomplete information and reshapes the observations that follow. Building reliable world models is therefore central to long-horizon decision-making. While Reinforcement Learning (RL) has been a dominant approach, it often suffers from instability, limited interpretability, and reliance on costly online interaction. In this work, we explore Decision Transformers (DT) as a sequence modeling framework for uncertainty-aware control. As a testbed, we consider geosteering, where drilling trajectories must be continuously adjusted in real time based on indirect and noisy subsurface measurements. Our training data is generated by a dual-network deep RL (DRL) agent coupled with a particle filter (PF), embedding geological variability through probabilistic boundary estimates and noisy logs. Experiments demonstrate that longer temporal contexts allow the DT to capture delayed structural signals, leading to more consistent long-horizon trajectories. These findings position sequence modeling as a promising foundation for embodied world models in complex, uncertainty-laden decision-making domains.

1 Introduction

Trajectory optimization under uncertainty remains a central challenge in many real-time control tasks, ranging from autonomous navigation to robotic manipulation. In the geosteering domain, the problem is further complicated by limited visibility into the subsurface and the criticality of decisions that impact reservoir productivity and well integrity. The task requires continuously steering the drilling trajectory to maximize reservoir exposure, maintain trajectory smoothness, and respect operational constraints, all while interpreting indirect and often noisy measurements in real time. Geosteering resembles walking at night with only a short-range torch: the driller perceives just a narrow slice of the subsurface ahead, and each steering adjustment is irreversible. Unlike passive forecasting, this is an active, embodied decision-making problem where actions both depend on and reshape future observations.

*University of Stavanger, Stavanger, Norway

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Embodied World Models for Decision Making.

Historically, geosteering workflows have depended heavily on human expertise and manual interpretation of measurements such as gamma ray (GR) logs and inclination data. While effective in many settings, such manual processes are time-intensive and subject to inconsistency, especially as the complexity of reservoirs and data streams grows. Early automated approaches introduced decision-theoretic frameworks and dynamic programming tools to formalize the trade-offs involved in well placement Kullawan et al. [13, 14]. Building on this line of work, ensemble-based methods like Ensemble Kalman Filtering (EnKF) Evensen [6] have been integrated with decision support systems to improve robustness under geological uncertainty Alyaev et al. [2]. Later enhancements incorporated machine learning techniques, such as Generative Adversarial Networks (GANs) Goodfellow et al. [8], to generate diverse geological realizations for use in decision support frameworks Alyaev et al. [1], advancing toward the notion of learned world models Ha and Schmidhuber [9], Hafner et al. [10] for subsurface dynamics.

In parallel, RL has emerged as a powerful alternative, offering the ability to learn drilling strategies directly from data without relying on hand-crafted rules or priors. Muhammad et al. [18] introduced Deep Q-Networks (DQN) for geosteering, demonstrating that RL can outperform traditional model-based methods like Decision-supportive Dynamic Programming (DSDP) . This was extended by coupling DQN with particle filters (PF) to model boundary uncertainty more accurately Muhammad et al. [17]. The resulting hybrid system, known as the *Pluralistic Robot* Muhammad et al. [16], was benchmarked in the Geosteering World Cup (GWC) simulator and achieved expert-level performance in idealized test cases. However, the original formulation struggled with stability and generalization, especially when deployed on unseen geological structures.

A prior work Djecta et al. [5] has introduced a simulator-verified dual DRL agent combined with PFs, specifically designed to enhance policy stability and uncertainty-aware steering. This architecture outperformed traditional single-network designs in environments with noisy feedback and sparse rewards, moving closer to a practical embodied world model Fung et al. [7] for geosteering.

In this paper, we propose a new direction that reformulates geosteering as a sequence modeling problem and adopts the Decision Transformer architecture Chen et al. [4] as the core policy engine. Rather than relying on traditional value functions or Q-networks, the DT treats decision-making as conditional sequence generation, modeling the joint distribution of returns, states, and actions using an autoregressive Transformer. This enables stable training from offline datasets, eliminating the need for unstable online interaction with the environment, while leveraging the scalability and expressiveness of Transformer models.

The potential of Transformer-based architectures in RL has been widely demonstrated beyond our context. DT has shown competitive performance with offline RL algorithms across diverse domains such as Atari, Gym, and D4RL Chen et al. [4]. The model treats trajectories as sequences of triplets and uses causal attention to predict actions conditioned on goals. Additionally, surveys such as Yuan et al. [20] have highlighted the taxonomy of Transformer-RL architectures and their growing impact across hierarchical RL, meta-RL, and multi-agent systems. Specialized designs like SPformer Han et al. [11] have also demonstrated the efficacy of Transformers in multi-agent autonomous driving, using physical positional encoding to boost convergence and safety. Likewise, SceneRep Transformers Liu et al. [15] leverage predictive latent distillation to improve sample efficiency in dense urban environments.

Inspired by these developments, we adapt the DT paradigm to the geosteering domain using a dataset derived from 20,000 episodes of our previously trained dual DRL agent. By reframing geosteering as long-horizon planning under uncertainty, we evaluate the model under different input sequence lengths and architectural variations, showing that it can produce geologically consistent and feasible trajectories. This work demonstrates how sequence modeling can serve as a robust alternative to traditional RL in subsurface decision-making, aligning with embodied world models that move beyond passive prediction toward goal-directed interaction with uncertain environments.

The remainder of the paper is organized as follows. Section 2 presents the geosteering problem formulation, the DT architecture, and the data generation process. Section 3 outlines the experimental setup, hyperparameter studies, and comparative results across model configurations. Section 4 discusses the limitations of the study and avenues for improvement. Finally, Section 5 concludes the paper with a summary of findings and directions for future research. Additional experimental details are provided in Appendix A.1 to support reproducibility.

2 Methodology

In this section, we present the modeling framework used to apply the DT to geosteering. Our formulation can be seen as building an uncertainty-aware embodied world model for decision-making, where the agent learns to predict and act over extended horizons under partial observability. We begin by formulating geosteering as a constrained sequential decision-making problem under uncertainty, then describe how this is operationalized through a transformer-based sequence prediction architecture.

2.1 Problem Formulation

The geosteering task can be viewed as a sequential decision-making process under uncertainty Bratvold and Begg [3], which can be formally expressed as a Partially Observable Markov Decision Process (POMDP). The true state of the system at time t, denoted $s_t \in \mathbb{R}^n$, represents the hidden position of the drill bit and surrounding geological structure. Since this state cannot be observed directly, the agent instead receives partial and noisy observations, such as GR measurements and directional data, which provide incomplete information about the environment. Based on the observation history, the agent must choose a steering action $a_t = (\Delta\text{INCL}_t, \Delta\text{AZIM}_t)$ that determines the well's future trajectory.

The primary objective is to minimize cumulative trajectory error over a finite planning horizon H. If $\hat{s}_t$ denotes the state predicted from the executed actions, the optimization target can be written as

$$\min_{\pi} \mathbf{E}\left[\sum_{t=1}^{H} \|s_t - \hat{s}_t\|^2\right], \tag{1}$$

where π is the policy mapping observation histories to actions. In practice, this means producing decisions that keep the trajectory close to the target reservoir structure, while remaining robust to uncertainty.

The action space is bounded by both physical and operational constraints. Each action must satisfy mechanical limitations:

$$\|a_t\| \leq \delta_{\text{max}}, \quad a_t \in \mathcal{A}, \tag{2}$$

where δ_{max} denotes the maximum allowed change in inclination and azimuth, and $\mathcal{A}$ the feasible action set. These limits ensure safety, just as in robotics where actuators must respect torque or velocity bounds.

At each decision point, the agent faces multiple alternatives in $\mathcal{A}$. Each choice affects not only the immediate trajectory segment but also propagates downstream, meaning that local decisions compound into long-horizon outcomes. This dependence makes the task particularly sensitive to sequential reasoning.

Finally, uncertainty permeates the process. Observations are noisy functions of the hidden geological state:

$$s_t = f(x_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{3}$$

where x_t is the true spatial position, $f(x_t)$ the expected measurement at that location, and ϵ_t Gaussian noise. This formulation captures both aleatoric uncertainty from imperfect sensors and epistemic uncertainty due to incomplete subsurface models.

Overall, geosteering can be cast as a POMDP-driven sequential decision-making process: the agent must act under partial observability, respect strict constraints, and plan over long horizons to maximize reservoir exposure and maintain safe, feasible well trajectories.

2.2 Decision Transformer for Geosteering

Having framed geosteering as a POMDP-driven sequential decision-making process, we now describe how the DT operationalizes this formulation. Instead of directly minimizing state error over a horizon, the DT learns a conditional sequence model that generates actions consistent with desired outcomes (Figure 1). Each drilling episode is expressed as a sequence of triplets

$$(s_t, a_t, R_t), \tag{4}$$

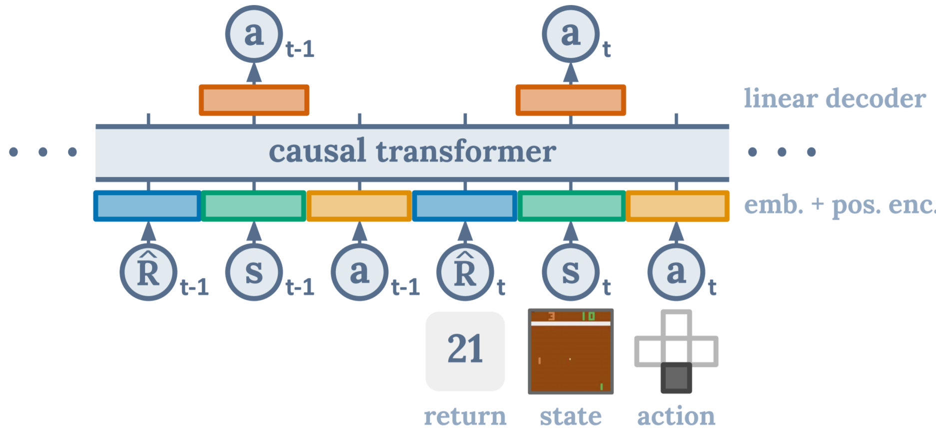

Figure 1: Overview of the Decision Transformer architecture, which conditions on sequences of returns, states, and actions to predict future actions [4].

where s_t is the observed state (including GR measurements, inclination, azimuth, and spatial coordinates), a_t is the chosen steering action, and R_t the return-to-go from step t. The learning problem can be expressed as aligning the DT's autoregressive policy $\pi_\theta(a_t \mid s_{\leq t}, a_{<t}, R_{\leq t})$ with expert demonstrations. Formally, the model predicts an action $\hat{a}_t^\theta$ at each timestep, and training minimizes the deviation from the demonstrated expert action a_t:

$$\min_\theta \mathbf{E}\left[\sum_{t=1}^{H} \|a_t - \hat{a}_t^\theta\|^2\right], \tag{5}$$

which corresponds to a supervised version of the earlier trajectory optimization, now framed directly at the action level.

At each timestep, the model receives the triplet (s_t, a_{t-1}, R_t), which are projected into a shared latent representation. This embedding can be written as

$$x_t = W_s s_t + W_a a_{t-1} + W_R R_t + p_t, \tag{6}$$

where W_s, W_a, and W_R are learnable projections and p_t encodes temporal ordering. The sequence $\{x_t\}_{t=1}^{\text{SEQ\_LEN}}$ is processed by stacked transformer blocks, with multi-head self-attention enabling the model to capture dependencies between past and future states across long horizons. The contextualized representation is mapped through a linear output layer to generate the predicted action $\hat{a}_t$.

Training proceeds entirely offline on expert trajectories generated by a dual-network DRL agent with a PF, ensuring that geological uncertainty is embedded in the data. By treating the task as supervised sequence modeling, the DT bypasses unstable trial-and-error exploration while still capturing long-term dependencies. The context length SEQ_LEN determines the effective planning horizon: short contexts focus on local accuracy, while longer contexts condition decisions on broader structural signals. This directly ties back to the problem formulation, where the trade-off between immediate error minimization and long-horizon consistency governs trajectory quality.

2.3 Data Acquisition and Preparation

Unlike standard RL approaches that learn through online interaction, the DT framework requires a pre-collected offline dataset of full trajectories. These trajectories must be structured to reflect complete episodes of decision-making, with clearly defined returns-to-go and action histories. This shift imposes specific requirements on the dataset: sequences must be complete, consistent in length, and diverse enough to represent a meaningful return distribution. Additionally, because the DT model is trained in a supervised fashion, the data must be split into training, validation, and test sets to evaluate generalization.

This is in contrast to typical DRL training, where new trajectories are continuously generated online and stored in a replay buffer for sampling without concern for strict train/test separation.

To meet the DT's requirements, we re-trained a previously developed dual DRL agent Djecta et al. [5] in a synthetic geosteering environment, explicitly for the purpose of data collection. This environment simulates horizontal drilling through geological formations and produces GR logs along with true structural boundaries. At each timestep, the agent selects a steering action based on the current state, receives a scalar reward, and transitions to a new state.

Because the dataset was generated by a dual-network DRL agent combined with a PF, uncertainties from noisy logs and probabilistic boundary estimates are already incorporated. Thus, each trajectory encodes realistic decision-making under uncertainty, rather than deterministic or noise-free drilling paths.

During retraining, we logged all environment-agent interactions across multiple episodes. Each episode corresponds to one full simulated well trajectory, with data recorded at every decision step in the form:

$$(s_t, a_t, r_t, s_{t+1}, \texttt{return\_to\_go}_t), \tag{7}$$

where:

- s_t: Current environment state, including GR, inclination, azimuth, and spatial features.
- a_t: Agent's action at step t, defined as changes in inclination and azimuth.
- r_t: Reward based on trajectory alignment with the reservoir zone.
- s_{t+1}: Next state after applying a_t.
- $\texttt{return\_to\_go}_t$: Discounted sum of future rewards from step t onward.

Each record was tagged with an `episode_id`, allowing us to organize and split the data on a per-trajectory basis. The complete dataset was saved as a DataFrame and serialized for training use.

Table 1: Structure of a single row in the dataset

episode_id	step_t	state_t	action_t	return_to_go_t	reward_t	next_state	done

For model input preparation, each episode was divided into overlapping sequences of fixed length SEQ_LEN using a sliding window. Each training example takes the form:

$$\{(s_t, a_t, R_t)\}_{t=1}^{\text{SEQ\_LEN}}, \tag{8}$$

which conforms to the input specification of the DT: a sequence of return-to-go, state, and action triplets used to condition the model's predictions autoregressively.

This pipeline enables offline supervised training of a sequence model that mimics expert-like drilling behavior based on learned correlations between past state-action-return sequences and future decisions.

3 Experimental setup and results

This section outlines the experimental environment, dataset construction, and the influence of key hyperparameters on the performance of the Decision Transformer in a geosteering context. All model variants were trained for 200 epochs with a batch size of 64 using the Adam optimizer (learning rate 1×10^{-4}). Experiments were executed on a local workstation with a 13th Gen Intel® Core™ i7-13800H × 20 processor, 32GB VRAM, and Ubuntu 22.04. The transformer architecture consisted of 2 layers with hidden dimension 128, feed-forward dimension 512, and 2 attention heads, with a maximum sequence length of 20 tokens. For reproducibility, all details are summarized in Appendix A.1 .

3.1 Dataset Creation

As we mentioned before, to build the dataset for supervised training, we first re-trained a dual DRL agent Djecta et al. [5] on the geosteering environment described previously. The agent was trained using a reward function designed to maximize reservoir contact and maintain trajectory smoothness.

After approximately 20,000 full episodes of training, the agent was used to generate a dataset of trajectories, which were recorded by logging each step's state, action, and estimated return-to-go.

The final dataset consists of approximately 348,000 rows, with each episode containing around 29 steps on average. This volume ensures sufficient coverage of geological variability and behavioral diversity. Each training batch is composed of a collection of sequences that are sampled uniformly from the dataset. The dataset was split into 80% training, 10% validation, and 10% test sets. All splits were sampled uniformly from the full collection of overlapping sequences, ensuring consistent distribution across geological patterns and trajectory shapes.

3.2 Model Evaluation and Hyperparameter Exploration

This section explores how model structure and key hyperparameters impact performance. Two main directions were investigated: the influence of input sequence length and the role of multi-head attention. Each configuration was trained using the same dataset and evaluated both in terms of validation loss and final trajectory accuracy.

3.2.1 Attention Head Ablation

To evaluate the impact of attention heads in the multi-head self-attention (MHSA) block Vaswani et al. [19], we conducted an ablation study by training models with different head configurations. The baseline model used 4 attention heads, while alternative configurations used 2.

Results show that reducing the number of attention heads to 2 significantly improved both training stability and generalization. As shown in Figure 2, the model with 4 heads exhibits signs of overfitting: it converges quickly on the training set but fails to maintain performance on the validation set, with validation loss high along the epochs. In contrast, the 2-head configuration maintains a better balance between training and validation loss, converging more slowly but generalizing more effectively.

This behavior suggests that for relatively short sequences (`SEQ_LEN` = 20) and structured geological environments, a smaller attention capacity is sufficient. The use of fewer heads also reduces computational complexity and the risk of overfitting.

Based on these observations, the 2-head configuration was selected as the default for the remaining experiments in this study.

3.2.2 Effect of Sequence Length

We experimented with sequence lengths (`SEQ_LEN`) ranging from 1 to 20 to evaluate how temporal context influences model performance. As shown in Figure 3, shorter sequences converged quickly and achieved lower training loss, and this trend was mirrored in the validation loss curves (Figure 4). However, these short contexts lacked the capacity to capture long-term dependencies essential for geologically consistent steering.

In contrast, longer sequences such as `SEQ_LEN` = 20 converged more slowly and exhibited higher per-step loss, but consistently produced trajectories that aligned better with reservoir structures. This reveals a trade-off between local prediction accuracy and long-horizon planning quality—a point that will become more evident in the following subsections, where we compare trajectory outcomes and Reservoir Contact Ratio (RCR).

3.2.3 Discrepancy Between Loss and Trajectory Quality

While per-step validation loss is commonly used to evaluate predictive models, our experiments with the DT in geosteering reveal a deeper insight: minimizing step-wise error does not always correlate with high-quality long-term decisions.

This becomes particularly evident when considering a real geosteering scenario during the steering phase. In practice, the expectation is that the well should remain within the reservoir boundaries to maximize contact.

As shown in Figure 5, models trained with short context windows, specifically `SEQ_LEN` = 1, tend to diverge from the intended geological boundaries (Figure 5, Black lines, Real Top and Real Bottom), despite achieving lower validation loss. This discrepancy arises because the DT, when limited to

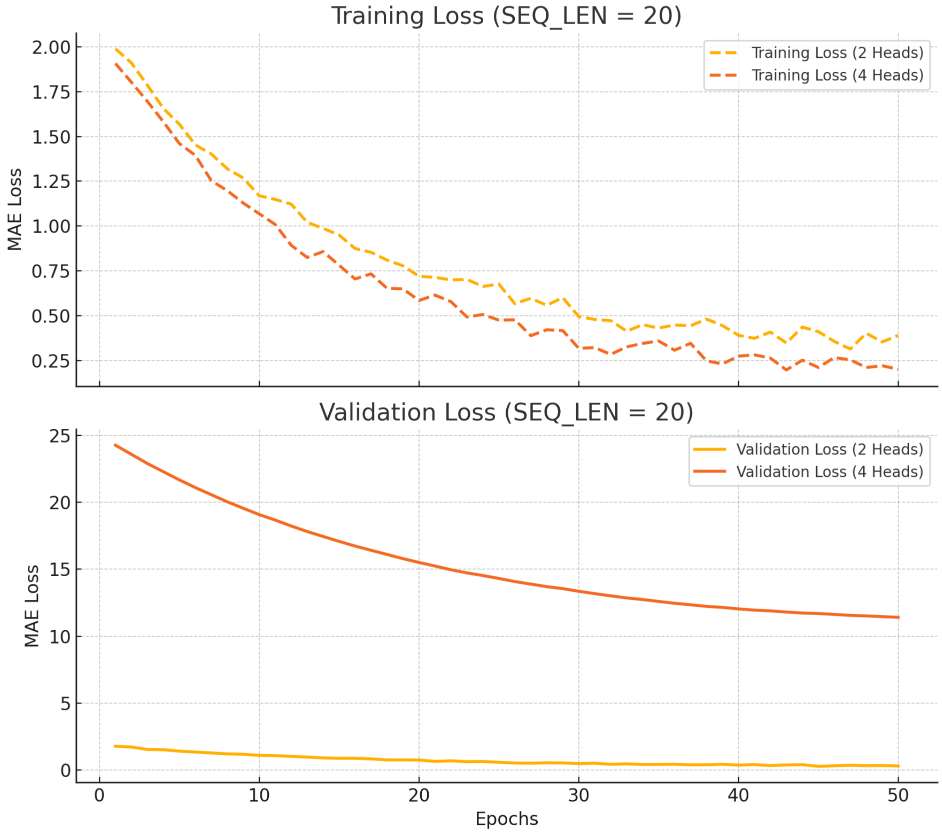

Figure 2: Training loss for models with 2 vs. 4 attention heads. The 2-head variant converges more steadily and generalizes better in structured geological settings.

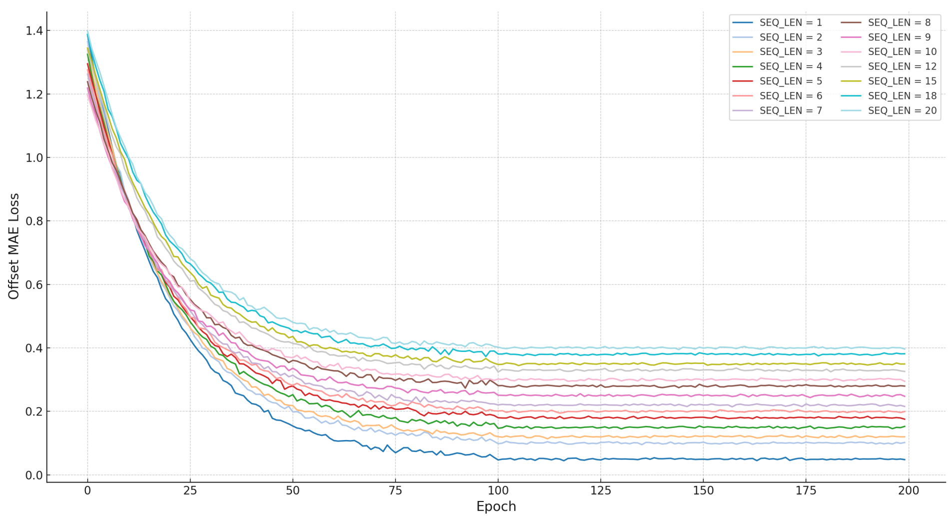

Figure 3: Training loss across context lengths. Short sequences converge faster but underfit, while longer ones converge slower yet capture richer dependencies.

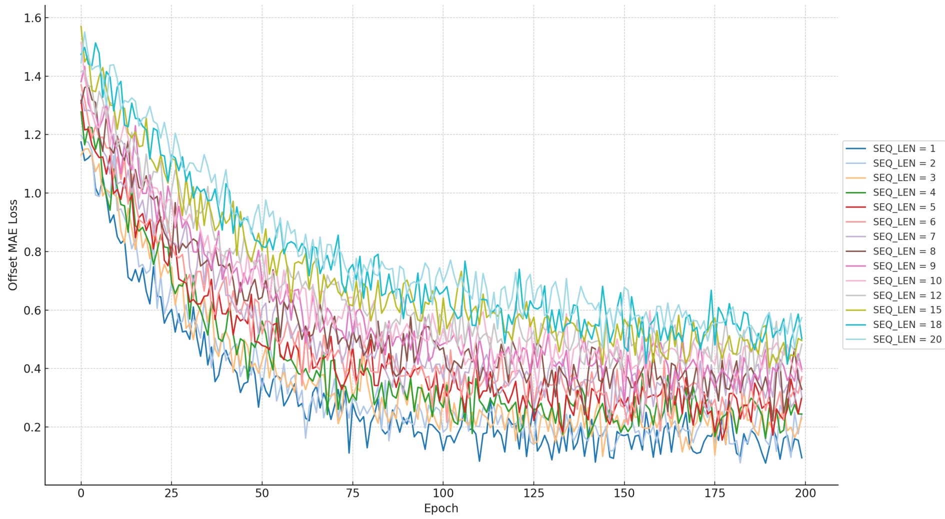

Figure 4: Validation loss for different context lengths. Longer sequences (`SEQ_LEN`=20) achieve lower error, reflecting improved long-horizon planning.

short sequences, lacks sufficient historical context to anticipate structural changes in the subsurface. With minimal hindsight, the model behaves greedily, focusing on immediate outcomes rather than planning across the trajectory.

In contrast, increasing the sequence length to `SEQ_LEN` = 20 allows the DT to attend to a richer history of prior states, actions, and return signals. This enables the self-attention layers to capture long-range dependencies, structural trends, and delayed geological feedback, all critical for effective decision-making in directional drilling. Although the longer-sequence model incurs slightly higher per-step loss (likely due to increased variance in the training signal), it produces trajectories that better track the target formation over time.

This observation is quantitatively supported by the RCR shown in Figure 6. RCR measures the proportion of trajectory points that remain inside the reservoir zone. As the figure indicates, models with longer context windows consistently achieve higher RCR values. This confirms that the DT benefits from longer sequences by planning more geologically consistent and operationally viable trajectories.

These results highlight a core principle in sequential modeling: context length governs the planning horizon. For geosteering, where decisions compound over time and geological feedback is delayed, longer sequences empower the model to reason holistically about future consequences, a hallmark capability of Transformer-based policies.

So, while validation loss reflects prediction accuracy at the token level, it may obscure deficiencies in high-level trajectory planning. Evaluating models like the DT in geosteering requires a broader set of metrics, including domain-specific indicators such as RCR and visual comparisons to structural ground truth.

4 Limitations

The Decision Transformer in this study was trained on synthetic trajectories generated from the behavior of a dual-network DRL agent, rather than expert human data. While this ensured consistency and uncertainty-aware signals, it may not fully capture the variability and subtleties of expert-driven decisions. The evaluation was conducted in a realistic geosteering scenario but limited in scope, focusing only on steering phases and a single reservoir setting. Broader validation on diverse geological environments, noisy field logs, and expert-labeled data will be necessary to confirm generalizability and practical applicability.

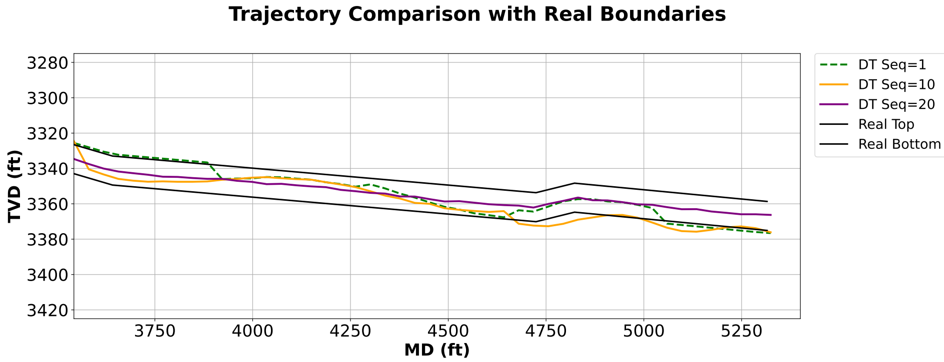

Figure 5: Trajectory predictions across context lengths where the black lines represent the reservoir boundaries. Short contexts (`SEQ_LEN=1`) diverge from target zones, while longer ones align with reservoir structure.

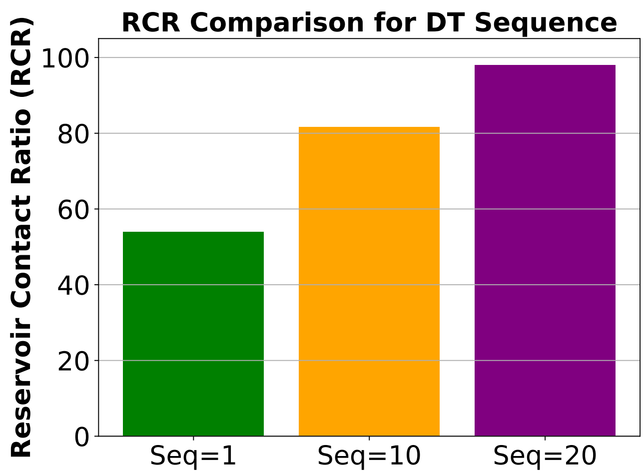

Figure 6: Reservoir Contact Ratio (RCR) across sequence lengths. Longer context windows consistently achieve higher reservoir exposure.

5 Conclusion

Geosteering exemplifies the challenge of making rapid, uncertainty-laden decisions with long-term consequences, and this work shows how a DT can help address that challenge. Using a dataset of trajectories generated from our dual-network DRL agent, we demonstrated that the DT can learn horizon-aware policies in an offline, supervised setting. Our experiments highlight the impact of sequence length: short contexts achieved lower per-step loss but produced geologically inconsistent trajectories, whereas longer contexts (`SEQ_LEN` = 20) captured delayed geological signals and achieved higher RCR. This decouples trajectory quality from local prediction accuracy and underscores the need for sequence-level evaluation in subsurface decision-making.

Beyond drilling automation, our approach connects to embodied world models and decision-making under uncertainty. By reframing geosteering as offline sequence modeling, the DT offers a stable, interpretable framework for capturing long-horizon dependencies without fragile online interaction.

Future directions include extending the model with Trajectory Transformers Janner et al. [12], integrating world-model approaches for real-time uncertainty-aware planning, and adopting foundation-model pretraining for transfer across geological settings. This research is ongoing, and the results presented here should be viewed as an intermediate step toward more comprehensive embodied world models for geosteering.

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

[1] S Alyaev, K Fossum, HE Djecta, J Tveranger, and A Elsheikh. Distinguish workflow: a new paradigm of dynamic well placement using generative machine learning. In *ECMOR 2024*, volume 2024, pages 1–16. European Association of Geoscientists & Engineers, 2024.

[2] Sergey Alyaev, Erich Suter, Reider Brumer Bratvold, Aojie Hong, Xiaodong Luo, and Kristian Fossum. A decision support system for multi-target geosteering. *Journal of Petroleum Science and Engineering*, 183:106381, December 2019. ISSN 0920-4105. doi: 10.1016/j.petrol.2019.106381. URL `http://dx.doi.org/10.1016/j.petrol.2019.106381`.

[3] Reidar B. Bratvold and Steve H. Begg. *Making Good Decisions: An Interdisciplinary Approach to Topics in Petroleum Engineering and Geosciences*. Society of Petroleum Engineers, 2010. ISBN 9781555632588. URL `https://www.reidar-bratvold.com/making-good-decisions`.

[4] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15084–15097. Curran Associates, Inc., 2021.

[5] Hibat Errahmen Djecta, Sergey Alyaev, Kristian Fossum, Reidar B. Bratvold, Ressi Bonti Muhammad, and Apoorv Srivastava. Uncertainty-aware well placement: Simulator-verified dual-network reinforcement learning approach meets particle filters. In Maciej Paszynski, Amanda S. Barnard, and Yongjie Jessica Zhang, editors, *Computational Science – ICCS 2025 Workshops*, pages 188–202, Cham, 2025. Springer Nature Switzerland.

[6] Geir Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003. doi: 10.1007/s10236-003-0036-9. URL `https://doi.org/10.1007/s10236-003-0036-9`.

[7] Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, Arjun Majumdar, Andrea Madotto, Franziska Meier, Florian Metze, Louis-Philippe Morency, Théo Moutakanni, Juan Pino, Basile Terver, Joseph Tighe, Paden Tomasello, and Jitendra Malik. Embodied ai agents: Modeling the world, 2025. URL `https://arxiv.org/abs/2506.22355`.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL `https://arxiv.org/abs/1406.2661`.

[9] David Ha and Jürgen Schmidhuber. World models. 2018. doi: 10.5281/ZENODO.1207631. URL `https://zenodo.org/record/1207631`.

[10] Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *CoRR*, abs/1912.01603, 2019. URL `http://arxiv.org/abs/1912.01603`.

[11] Ye Han, Lijun Zhang, Dejian Meng, Xingyu Hu, and Yixia Lu. Spformer: A transformer based drl decision making method for connected automated vehicles, 2024. URL `https://arxiv.org/abs/2409.15105`.

[12] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem, 2021. URL https://arxiv.org/abs/2106.02039.

[13] K. Kullawan, R. Bratvold, and J.E. Bickel. A decision analytic approach to geosteering operations. *SPE Drilling & Completion*, 29, 03 2014. ISSN 1064-6671. doi: 10.2118/167433-PA.

[14] K. Kullawan, R.B. Bratvold, and J.E. Bickel. Sequential geosteering decisions for optimization of real-time well placement. *Journal of Petroleum Science and Engineering*, 165:90–104, 2018. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2018.01.068. URL https://www.sciencedirect.com/science/article/pii/S0920410518300809.

[15] Haochen Liu, Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Augmenting reinforcement learning with transformer-based scene representation learning for decision-making of autonomous driving, 2023. URL https://arxiv.org/abs/2208.12263.

[16] Ressi B. Muhammad, Yasaman Cheraghi, Sergey Alyaev, Apoorv Srivastava, and Reidar B. Bratvold. Geosteering robot powered by multiple probabilistic interpretation and artificial intelligence: Benchmarking against human experts. *SPE Journal*, pages 1–15, 01 2025. ISSN 1086-055X. doi: 10.2118/218444-PA. URL https://doi.org/10.2118/218444-PA.

[17] Ressi Bonti Muhammad, Apoorv Srivastava, Sergey Alyaev, Reidar Brumer Bratvold, and Daniel M. Tartakovsky. High-precision geosteering via reinforcement learning and particle filters, 2024. URL https://arxiv.org/abs/2402.06377.

[18] Ressi Bonti Muhammad, Sergey Alyaev, and Reidar Brumer Bratvold. Optimal sequential decision-making in geosteering: A reinforcement learning approach, 2025. URL https://arxiv.org/abs/2310.04772.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[20] Weilin Yuan, Jiaxing Chen, Shaofei Chen, Dawei Feng, Zhenzhen Hu, Peng Li, and Weiwei Zhao. Transformer in reinforcement learning for decision-making: a survey. *Frontiers of Information Technology Electronic Engineering*, 25:763–790, 07 2024. doi: 10.1631/FITEE.2300548.

A Appendix

A.1 Experimental Setup Details

Table 2: Experimental setup and training details for the Decision Transformer in geosteering.

Category	Details
Dataset size	$\sim$348k state–action–return records from $\sim$20k episodes
Data split	80% train, 10% val, 10% test
Input format	Sequences of triplets $\{(s_t, a_t, R_t)\}$
Features	GR, inclination, azimuth, spatial coordinates
Model	Decision Transformer
Transformer blocks	3
Hidden size (d_{model})	128
Feedforward size (d_{ff})	512
Attention heads	2 (ablation: 4)
Dropout	0.1
Activation	ReLU
Context length (`SEQ_LEN`)	1, 5, 10, 20 (default = 20)
Output head	Linear projection to action $\hat{a}_t$
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Learning rate	1×10^{-4}
Batch size	64
Epochs	200
Loss function	Mean Squared Error (MSE)
Evaluation metrics	Validation MSE, Reservoir Contact Ratio (RCR), trajectory consistency
Hardware	Intel i7-13800H (20 cores), NVIDIA RPL-P GPU (32GB VRAM), Ubuntu 22.04

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction explicitly state the main contributions: reframing geosteering as a sequential decision-making problem, applying Decision Transformers for offline embodied control, and showing how context length affects trajectory quality. These claims are supported by the experiments (Sections 3–5).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We acknowledge limitations such as training solely on synthetic trajectories generated by a dual-network DRL agent, and evaluation constrained to steering phases rather than full drilling operations. While a realistic geosteering scenario was considered, broader validation on expert-labeled data and diverse geological settings is left for future work (Section 4).

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: The paper does not include new theorems or proofs; it is primarily empirical and methodological.

   Guidelines:

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

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper

   Answer: [Yes]

   Justification: Section 2.3 and Section 3 describe the dataset generation, model architecture, hyperparameters, and evaluation metrics, with additional implementation details summarized in Appendix A.1. Together, these provide sufficient information to enable reproduction of the main results.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: Due to project and licensing restrictions, the dataset and code are not yet publicly available. However, we intend to release an anonymized dataset and training scripts upon acceptance, following NeurIPS guidelines.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 3 specifies training epochs, optimizer, batch size, sequence length variations, and compute environment. Dataset splits (train/val/test) are described in Section 2.3, and additional implementation details, including Transformer architecture and hyperparameters, are provided in Appendix A.1.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: The paper reports validation loss curves and Reservoir Contact Ratio (RCR) as primary evaluation metrics.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 3 and appendix A.1 report the compute environment. Each model was trained for 200 epochs, with training runs typically completing within 6 hours per configuration.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work adheres to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Paper discusses the broader implications: positive impacts include safer and more consistent geosteering decisions, while potential negative impacts could arise if models are misused for automated decision-making without human oversight.

Guidelines:

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

11. **Safeguards**

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

Answer: [NA]

Justification: No high-risk pretrained models or scraped datasets are released. The contribution is methodological and experimental on synthetic data.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All external datasets, simulators, and methods are cited properly. Licenses for public assets are respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [NA]

Justification: No new dataset or code is released as part of this submission, though we plan future release upon acceptance.

Guidelines:

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

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

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

Answer: [NA]

Justification: The work does not involve human subjects, crowdsourcing, or user studies.

Guidelines:

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

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

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

Answer: [NA]

Justification: Not applicable as no human subjects were involved.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [No]

Justification: LLMs (ChatGPT) were used only for writing assistance, editing, and restructuring text. They were not used for data generation, modeling, or experimental methodology, hence not affecting scientific rigor.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.