DiffTORI with Adaptive Horizon and Hybrid Dynamics

Qiushi Zhang
Department of Mechanical Engineering
Carnegie Mellon University United States
georgez2@andrew.cmu.edu

Xilin Zhang
Department of Mechanical Engineering
Carnegie Mellon University United States
xilinzha@andrew.cmu.edu

Shixin Zhou
Department of Mechanical Engineering
Carnegie Mellon University United States
shixinz@andrew.cmu.edu

Mingyang Yu
Department of Mechanical Engineering
Carnegie Mellon University United States
myu3@andrew.cmu.edu

Qiyao Lin
Department of Mechanical Engineering
Carnegie Mellon University United States
qiyaolin@andrew.cmu.edu

Abstract: DiffTORI integrates model-based reinforcement learning with differentiable trajectory optimization but struggles with robustness due to fixed horizons, shallow latent models, and simple feed-forward networks. We propose dynamic horizon selection methods, a deeper residual MLP architecture, and variational latent-space regularization to enhance convergence and gradient propagation. Experiments reveal improved sample efficiency and stability from adaptive horizons and residual networks, while variational regularization exposes trade-offs between precision and latent structure.

Keywords: MBRL, DiffTORI, Adaptive Planning Horizon, Residual MLP Networks, Variational Latent-Space Regularization

1 Introduction

Model-based reinforcement learning(MBRL) promises sample-efficient, high-precision robot control [1], yet most current pipelines still train dynamics and reward models on surrogate losses—leading to the well-known objective–mismatch problem, in which accurate prediction does not guarantee good task performance [2]. DiffTORI closes this gap by embedding a differentiable trajectory optimizer inside the policy [3], allowing gradients from the true task loss to flow directly through cost and dynamics parameters and achieving state-of-the-art returns across vision-based RL and imitation-learning benchmarks. Nevertheless, DiffTORI exhibits several practical bottlenecks: its fixed planning horizon may be sub-optimal as task difficulty evolves; its feed-forward latent models can overfit or stall; and its deterministic encoder can learn entangled features that hamper generalization [4].

2 Problem statement & Prior work

Despite recent progress in differentiable planning, choosing an appropriate planning horizon H remains a fundamental challenge. A horizon that is too short fails to capture long-term consequences, whereas an excessively long horizon degrades gradient accuracy because of compounding model errors. Prior adaptive-horizon MPC work [5] alleviates this tension but has not been integrated into

end-to-end differentiable reinforcement-learning pipelines. Additionally, latent-space planners such as DiffTORI [3] rely on deterministic encoders and shallow networks, making them vulnerable to representation collapse and over-fitting.

3 Ideation and Methods

3.1 Adaptive horizon (Gradient & loss driven)

Inspired by ADAM optimizer [6], we propose the following horizon adaptation method We denote an encoder h_θ, latent dynamics d_θ, reward model R_θ and value predictor Q_θ. All trainable parameters are aggregated in θ.

Trajectory-optimization objective

$$a_{t:t+H} \;=\; \arg\max_{a_t,\dots,a_{t+H}} \sum_{l=t}^{t+H-1} \gamma^{l-t}\, R_\theta(z_l, a_l) + \gamma^H\, Q_\theta(z_{t+H}, a_{t+H}), \quad \text{s.t. } z_{l+1} = d_\theta(z_l, a_l). \tag{1}$$

The inner optimisation is solved with a differentiable nonlinear-least-squares layer (e.g. Theseus), enabling gradients $\partial a / \partial \theta$.

Policy-gradient update

$$\mathcal{L}_{\mathrm{PG}}(\theta) \;=\; -Q_{\tilde{\phi}}\big(s_t, a_t(\theta)\big), \qquad \theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L}_{\mathrm{PG}}. \tag{2}$$

$Q_{\tilde{\phi}}$ is a slowly-updated target critic; $a_t(\theta)$ is the first action of the optimised trajectory.

Adaptive-horizon rule

We monitor the improvement in expected return $\Delta J = J(H) - J(H{-}1)$ and gradient magnitude $G = \|\nabla_\theta \mathcal{L}_{\mathrm{PG}}\|$. Horizon is adapted by

$$H_{t+1} = \begin{cases} H_t + 1, & G > \delta_G \;\wedge\; \Delta J > \delta_J, \\ H_t - 1, & G < \varepsilon_G \;\vee\; \Delta J < \varepsilon_J, \\ H_t, & \text{otherwise}, \end{cases} \tag{3}$$

where $\delta_{(\cdot)}$, $\varepsilon_{(\cdot)}$ are positive thresholds.

Algorithm 1 Gradient / Loss-driven Adaptive Horizon

Initialize horizon $H \leftarrow 1$, replay buffer $\mathcal{B}$, parameters θ
for each training step **do**
  Sample mini-batch $\{s_t\} \sim \mathcal{B}$
  $a_{t:t+H} \leftarrow \textsc{TheseusSolve}(\theta, H)$ ▷ differentiable TO
  Compute loss $\mathcal{L}_{\mathrm{PG}}$ and update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{PG}}$
  Estimate returns $J(H)$ and $J(H-1)$ via online rollouts
  Update H using rule(3)
end for

3.2 Adaptive horizon (Cost-to-go)

We extend the DiffTORI framework by introducing an adaptive horizon mechanism that selects the best planning horizon $H \in \{1, 3, 5\}$ at each timestep based on cost-to-go evaluation. For each candidate horizon, a full Theseus-based trajectory optimization is run, and the one with the lowest estimated return (immediate rewards plus terminal Q-value) is chosen to execute the first action.

This dynamic selection enables the planner to adjust its planning depth based on the environment: shorter horizons help in uncertain or volatile settings, while longer horizons are favored in stable,

predictable situations. By balancing short-term caution with long-term foresight, the method avoids the limitations of a fixed-horizon strategy.

Inspired by adaptive MPC [5], our contribution lies in integrating adaptive horizon selection into differentiable trajectory optimization via Theseus, extending DiffTORI beyond its fixed-horizon design.

3.3 Complex MLP with Residual Skips

We combine the physics model and residual model (block) together to form a new forward function. The physics model f_{physics} here in our case represents robot joint states, point clouds, or images, and the residual model f_{residual} is what we want to implement.

$$\hat{s}_{t+1} = f_{\text{physics}}(s_t, a_t) + f_{\text{residual}}(s_t, a_t) \tag{4}$$

This idea is inspired by a previous work that is related to hybrid models [7], where the work provides two approaches for designing the residual blocks: Replacing the entire MLPs with a residual block and adding a residual block to the original MLPs. The first approach turns out to have even worse performance, and the latter one improves the efficiency and performance of the program over tasks. Thus, our implementation is developed based on the latter one.

The key idea to modify the MLP and forward function is to reduce the potential degradation problem in MBRL, speed up the convergence and learning process, and improve the generalization across different tasks. While keeping the original MLP unchanged, we add a residual block which contains the following layers in Figure 1:

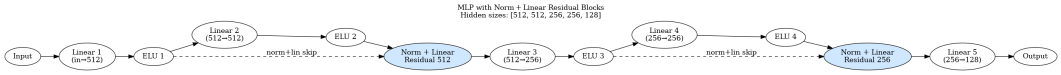


Figure 1: MLP with Norm + Residual Block

3.4 Variational latent-space regularization

We propose enhancing DiffTORI by introducing variational latent space regularization inspired by VAE techniques [8]. The core hypothesis is that a well-structured latent space could lead to more effective trajectory optimization by providing smoother state transitions and better generalization capabilities [9].

In the original DiffTORI, the encoder h_θ maps high-dimensional observations to deterministic latent vectors $z_t = h_\theta(s_t)$. Our approach modifies this architecture to output probabilistic distributions rather than point estimates:

$$t\mu_t, \log(\sigma_t^2) = h_\theta(s_t) \tag{5}$$

During training, we sample from this distribution using the reparameterization trick:

$$z_t = \mu_t + \sigma_t \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{6}$$

The key modification to the loss function involves adding a KL divergence term between the encoded distribution and a prior (standard normal distribution):

$$\mathcal{L}_{\text{modified}} = \mathcal{L}_{\text{DiffTORI}} + \beta \cdot \text{KL}(q(z|s) \| p(z)) \tag{7}$$

where the KL divergence for Gaussian distributions has the closed form:

$$\text{KL}(q(z|s) \| p(z)) = \frac{1}{2} \sum \left(1 + \log \sigma^2 - \mu^2 - \sigma^2\right) \tag{8}$$

We implement a beta scheduler to gradually increase the weight of the KL term during training:

Algorithm 2 Beta Scheduling for KL Weight

procedure TRAINWITHSCHEDULING
  Initialize: β_{start}, β_{end}, total_steps
  current_step $\leftarrow 0$
  while training **do**
    progress $\leftarrow \min\left(1.0, \frac{\text{current\_step}}{\text{total\_steps}}\right)$
    $\beta \leftarrow \beta_{\text{start}} + \text{progress} \left(\beta_{\text{end}} - \beta_{\text{start}}\right)$
    $\mathcal{L}_{\text{modified}} \leftarrow \mathcal{L}_{\text{DiffTORI}} + \beta \, \text{KL}\left(q(z \mid s) \,\|\, p(z)\right)$
    Update parameters via gradient descent
    current_step $\leftarrow$ current_step $+ 1$
  end while
end procedure

During inference and planning, we use the mean of the latent distribution for deterministic behavior:

$$z_t = \mu_t = h_\theta(s_t) \tag{9}$$

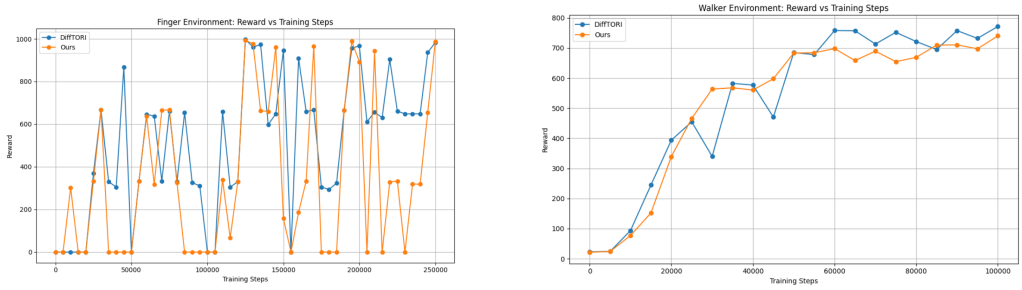
4 Experimental Results

To evaluate the effectiveness and generality of our proposed modifications, we selected two representative environments from the Meta-World benchmark:

- **finger-turn-easy** is a precision-based manipulation task with binary rewards, where success is only recognized when strict goal conditions are met. This environment is highly sensitive to planning accuracy and policy stability, making it suitable for testing fine-grained control and robustness under binary reward feedback.
- **walker-run** is a continuous locomotion task with dense reward signals, where smooth and progressive improvement is possible. This environment is ideal for evaluating thelearning stability, convergence rate, and long-horizon planning capability of different methods.

4.1 Adaptive horizon (Gradient & loss driven)

Our proposed ADAM liked adaptive horizon method achieves an improvement in stability and convergence rate for the walker-run task. As shown in Figure 2(b), the reward signal produced by the gradient and loss driven adaptive horizon method has fewer oscillations and a slightly higher convergence rate. However, for finger-turn-easy task, the reward signal oscillates significantly between 0 and 1000, although the pattern resembles the original DiffTORI method. Also, compared to the original DiffTORI method, reward signal generated by gradient and loss driven adaptive horizon method contain more zero points indicating an instantaneous failure of training.

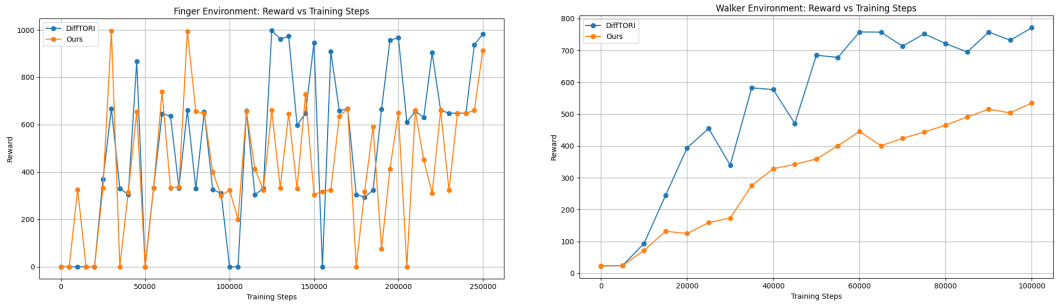


(a) finger-turn-easy (b) walker-run

Figure 2: Result of gradient & loss driven aptive horizon method

4.2 Adaptive horizon (Cost-to-go)

In Figure 3, our adaptive horizon method and DiffTORI achieve similar final performance in both environments, but with distinct learning dynamics. In the finger-turn-easy task (a), our method exhibits higher variance and sharp fluctuations, likely due to the task's sensitivity to frequent horizon changes and chaotic dynamics. DiffTORI maintains a tighter reward band, suggesting that a fixed horizon better suits tasks with low planning ambiguity. In the walker-run task (b), although DiffTORI achieves slightly higher final rewards, our method shows smoother and more stable improvement, particularly during early training (0–60k steps), indicating better robustness and consistent policy refinement.



(a) finger-turn-easy (b) walker-run

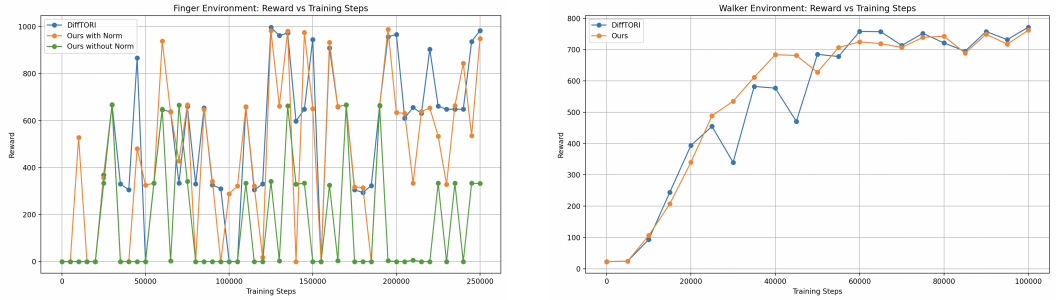
Figure 3: Result of Cost-to-go adaptive horizon method

4.3 Complex MLP with residual skips

In Figure 4, the baseline DiffTORI displays sharp oscillations yet eventually attains the maximum return of 1000. In contrast, the plain five-layer residual MLP (*without* normalization) frequently collapses to zero-reward episodes

In the continuous-reward locomotion task the residual MLP matches—and after 40 k steps modestly surpasses—the baseline in both convergence rate and final return, with visibly smoother learning curves.

Residual depth expands modeling capacity, but its utility depends on the reward structure: it can hinder learning in binary-reward settings where identity bias is heavily punished, but offers measurable gains under dense rewards.

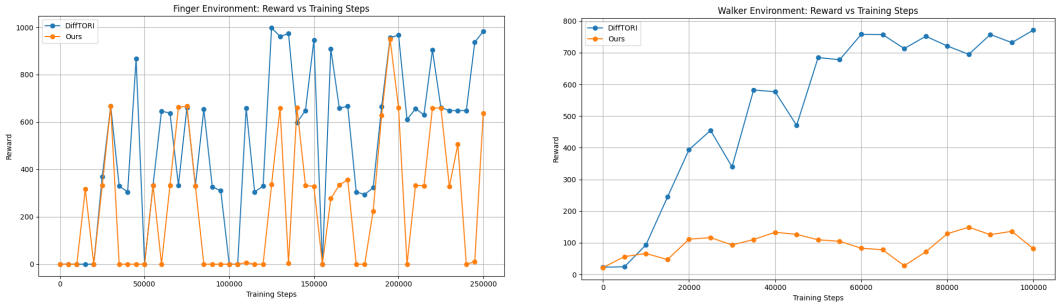


(a) finger-turn-easy (b) walker-run

Figure 4: Result of Complex MLP with residual

4.4 Variational latent-space regularization

Despite extensive parameter tuning, including adjustments to the β scheduler, Theseus optimizer damping parameters, learning rate, and weight decay, our results consistently showed performance degradation compared to the original DiffTORI in Figure 5.



(a) finger-turn-easy (b) walker-run

Figure 5: Result of latent-space regularization

5 Analysis

5.1 Adaptive horizon (Gradient & loss driven)

The difference of performance achieved by Gradient & loss driven adaptive horizon method between walker-run and finger-turn task can be explained by the sparsity of reward. For finger-turn task, the reward is sparse, resulting in more oscillations in gradient and reward signal. Since these information are relied on to compute new horizon in each training iteration, the updated horizon can be problematic resulting in unstable training.

5.2 Adaptive horizon (Cost-to-go)

Our adaptive horizon method struggles due to key design issues. Selecting the planning horizon at each timestep causes frequent switching, leading to inconsistent rollouts and unstable policy updates. Additionally, training the Q-function on mixed-length trajectories (H=1,3,5) introduces conflicting value targets that impede convergence. Early-stage noise in both dynamics and value estimates further misguides horizon selection. Unlike methods such as Optimal-Horizon MPC with DDP [10], which use gradual scheduling and analytic terminal costs for stability, our approach incurs higher variance by performing full optimization across multiple horizons at each step. Without warm-starting or smoothing, it becomes overly sensitive to transient cost fluctuations, reducing both training stability and final performance.

5.3 Complex MLP with residual skips

Residual MLP frequently collapses to zero-reward episodes, which is caused by an *identity-bias* failure in which the dominant skip path simply echoes the input state. Adding LayerNorm and a scaling linear layer inside each block alleviates catastrophic collapses: the return climbs rapidly to 1000 at 17.5 k steps but then oscillates between 300 and roughly 800 rather than stabilizing. Because *finger_turn_easy* supplies a binary reward, any excursion into identity bias yields an immediate zero, and the penalty outweighs the expressive benefit of deeper residual modeling. We therefore conclude that this architecture is ill-suited to sparse, binary-reward tasks where transient mis-predictions carry disproportionate cost.

For the residual MLP, dense feedback buffers the impact of temporary identity bias, allowing the deeper network's enlarged function class to dominate. Thus, it has a smoother learning curve and a faster convergence rate.

5.4 Variational latent-space regularization

Our experiments reveal several factors likely contributing to the performance degradation. The KL penalty creates a fundamental trade-off by forcing conformity to a prior distribution at the expense of task-specific features critical for control precision. While this promotes latent space smoothness, it appears to sacrifice discriminative information necessary for optimal action selection. In trajectory optimization, probabilistic latent representations introduce compounding uncertainties—small variations in early timesteps amplify through the planning horizon, significantly degrading long-horizon planning. Additionally, an architectural mismatch exists as downstream components designed for deterministic inputs now process representations from altered training dynamics, creating inconsistencies between training and deployment conditions [11]. The modified optimization landscape with competing objectives likely creates convergence points suboptimal for task performance.

6 Conclusions

In this project, we successfully implemented Adaptive Horizon, Latent Regularization, and Residual MLP in the previous work. We found that Horizon Adaptation and Residual MLP can provide faster convergence speed and more stable reward outcomes over complex tasks. For the simpler tasks, there wasn't much difference in the outcome. We also found a fundamental tension between general representation learning principles and the specific requirements of control tasks. While structured latent spaces offer theoretical advantages, they must be carefully balanced against the precision needed for effective trajectory optimization.

7 Limitations and future work

Limitations: (1) The total number of interaction steps was limited to approximately 100,000 due to computational constraints, preventing observation of late-stage fluctuations and asymptotic behavior. (2) Evaluation was confined to the DiffTORI benchmark suite; the adaptive horizon mechanism showed clear benefits only in environments with dense, continuous rewards. In sparse-reward settings, the impact of adaptive horizon selection was minimal, indicating limited generalization. As a result, the method's stability under prolonged training and its robustness across varying reward structures remain uncertain. (3) DDP-based approaches achieve better efficiency and stability by gradually increasing the planning horizon and reusing previous trajectories via warm-starts, while our Theseus-based optimization in cost-to-go method requires solving each horizon from scratch without reuse.

Future Work: (1) Allocate longer training budgets to observe full convergence dynamics and late-stage behavior. (2) Apply the method to additional DiffTORI tasks and external benchmarks such as DM-CONTROL and META-WORLD for broader empirical validation. (3) Investigate alternative residual network designs and richer latent-dynamics parameterizations to reduce model error in sparse-reward domains. (4) Incorporate adaptive scaling or shaping of the reward signal so that the pseudo-gradient remains informative even in sparse and discrete reward settings. (5)Stabilize horizon switching via smoothing or learned horizon policies, and to incorporate DDP-style warm-starting to reduce redundant optimization in Theseus. (6) Explore joint adaptation of the discount factor γ as a complementary or alternative mechanism to planning-horizon adjustment.

References

[1] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models, 2018. URL https://arxiv.org/abs/1805.12114.

[2] R. Wei, N. Lambert, A. McDonald, A. Garcia, and R. Calandra. A unified view on solving objective mismatch in model-based reinforcement learning, 2024. URL https://arxiv.org/abs/2310.06253.

[3] W. Wan, Z. Wang, Y. Wang, Z. Erickson, and D. Held. Difftori: Differentiable trajectory optimization for deep reinforcement and imitation learning, 2024. URL https://arxiv.org/abs/2402.05421.

[4] W. Röpke, R. Avalos, R. Rădulescu, A. Nowé, D. M. Roijers, and F. Delgrange. Integrating RL and planning through optimal transport world models. In *Proceedings of the 17th Workshop on Adaptive and Learning Agents (ALA)*, 2025. URL https://openreview.net/forum?id=DFQ8rwU9zZ. arXiv preprint arXiv:2403.12345 (if available); workshop paper.

[5] K. Stachowicz and E. A. Theodorou. Optimal-horizon model predictive control with differential dynamic programming. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1440–1446. IEEE, 2022.

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL https://arxiv.org/abs/1412.6980. Version 9, published 2023.

[7] S. Han, Z. Jin, Q. Liu, and C. Xu. Doumh: Improving douzero with multiple heads. In *Proceedings of the 2024 7th International Conference on Computer Information Science and Artificial Intelligence*, pages 297–302, 2024.

[8] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[9] P.-A. Andersen, M. Goodwin, and O.-C. Granmo. The dreaming variational autoencoder for reinforcement learning environments. *arXiv preprint arXiv:1810.01112*, 2018.

[10] D. Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95, 1966.

[11] Y. Chen, M. Gao, and Z. Li. Intelligent vehicle driving decision-making model based on variational autoencoder network and deep reinforcement learning. *Expert Systems with Applications*, 178:115080, 2019.