HYBRID SE-PG: RELIABLE GRADIENT ESTIMATION IN PRE-CONVERGENCE DIFFERENTIABLE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Differentiable optimization enables end-to-end learning through iterative solvers, yet obtaining accurate gradients remains challenging when solvers operate in the pre-convergence regime. Existing approaches face a fundamental trade-off: Implicit Differentiation (IFT) provides accurate gradients only near solver fixed points, while truncated methods improve efficiency at the cost of missing future gradient propagation. We introduce Hybrid Spectral Extrapolation Phantom Gradient (Hybrid SE-PG), a gradient estimation framework for pre-convergence solver states. SE-PG exploits locally predictable gradient evolution to approximate missing gradient propagation, while an adaptive hybrid mechanism integrates IFT as solvers approach convergence. This enables efficient gradient estimation across different solver convergence states without requiring full convergence before gradient computation. We evaluate Hybrid SE-PG on differentiable optimization benchmarks, including classical control tasks and high-dimensional manipulation problems. Gradient fidelity experiments show that SE-PG provides more accurate approximations to converged IFT reference under pre-convergence solver states. Training experiments further demonstrate improved optimization efficiency compared with implicit differentiation baselines while maintaining accurate gradient estimation.

1 INTRODUCTION

Differentiable optimization has emerged as a powerful paradigm in robotic policy learning, enabling the seamless integration of model-based priors with data-driven optimization Amos & Kolter (2017); Amos et al. (2018); Pineda et al. (2022); Agrawal et al. (2019); Karkus et al. (2017); Okada et al. (2017); Srinivas et al. (2018). By treating the solver as a differentiable layer, these methods allow for the end-to-end learning of complex cost functions and dynamics.

During differentiable optimization, iterative solvers frequently operate before convergence due to limited computational budgets. Accurate gradients require information from future solver iterations, while fully unrolling the solver is computationally expensive. Truncated differentiation improves efficiency by limiting the propagation horizon but inevitably discards future gradient information. This creates a precision-efficiency challenge: how to obtain informative long-horizon gradients in a low-cost manner.

Implicit Differentiation (IFT) Bai et al. (2019); Blondel et al. (2022); Gould et al. (2016); Lorraine et al. (2020) provides exact infinite-horizon gradients by solving for the inverse Jacobian at the solver's fixed point h^* (e.g., iLQR Li & Todorov (2004); Tassa et al. (2014)). However, its accuracy relies on the validity of the fixed-point assumption and may degrade when applied to pre-convergence solver states. At the other extreme, approximation-based methods such as **Truncated Backpropagation Through Time (TBPTT)** Williams & Peng (1990); Tallec & Ollivier (2017); Shaban et al. (2019) enable efficient gradient computation by unrolling the solver for a finite horizon, but inevitably discard future gradient propagation beyond the truncation horizon. Recent extensions, such as **Phantom Gradient (PG)** methods Geng et al. (2021), extend truncated differentiation by constructing phantom solver iterations from converged states, but they typically assume that the starting point is close to the fixed point.

To address this challenge, we propose **Hybrid Spectral Extrapolation Phantom Gradient (Hybrid SE-PG)** (see Figure 1). The key idea is to exploit the locally predictable gradient evolution of transient solver trajectories. By modeling local gradient evolution, SE-PG approximates missing

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

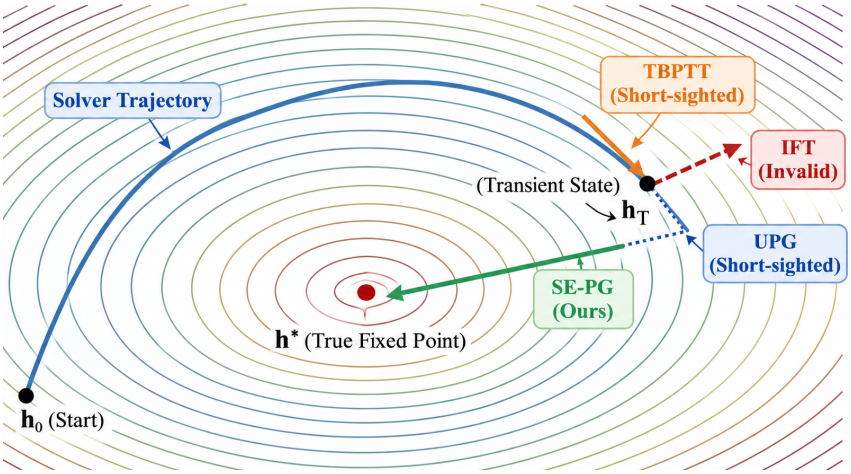


Figure 1: Geometric Intuition. When the solver terminates at a transient state $h_T \neq h^*$: IFT relies on the fixed-point assumption and may produce inaccurate gradients. TBPTT and PG truncate future gradient propagation beyond the rollout horizon. In contrast, SE-PG estimates the missing gradient propagation from a short phantom trajectory through local spectral extrapolation.

gradient propagation beyond the truncation horizon, while Hybrid SE-PG adaptively integrates IFT near convergence.

In summary, this paper makes the following contributions:

- We propose Hybrid SE-PG, a differentiable optimization framework that combines SE-PG and IFT to provide efficient gradient estimation across different solver convergence states.
- We develop SE-PG, which exploits local gradient evolution in pre-convergence solver states to approximate the missing gradient propagation beyond the truncation horizon.
- We evaluate Hybrid SE-PG on differentiable optimization benchmarks, including classical control and high-dimensional manipulation tasks. Experiments demonstrate improved gradient fidelity and training efficiency compared with implicit differentiation baselines.

2 Related Work

Differentiable Optimization for Learning and Control. Differentiable optimization layers integrate numerical solvers into neural networks, enabling end-to-end learning through optimization procedures Amos & Kolter (2017); Agrawal et al. (2019); de Avila Belbute-Peres et al. (2018); Chen et al. (2018); Baydin et al. (2018). In robotics, differentiable trajectory optimization has been applied to learn costs and dynamics through control solvers Jin et al. (2020); Okada et al. (2017); Pereira et al. (2018), with recent methods such as DiffMPC Amos et al. (2018), DiLQR Wang et al. (2025), and DiffTORI Wan et al. (2024) demonstrating their effectiveness in control and manipulation.

Implicit and Truncated Differentiation. Implicit Differentiation (IFT) computes gradients through the fixed-point condition of iterative solvers and has been widely used in bilevel optimization and deep equilibrium models Gould et al. (2016); Bai et al. (2019); Franceschi et al. (2018); Lorraine et al. (2020). However, its accuracy depends on the validity of the fixed-point assumption and may degrade for pre-convergence solver states. Alternatively, unrolling- based methods such as BPTT and TBPTT compute gradients through finite solver iterations Domke (2012); Maclaurin et al. (2015); Finn et al. (2017); Williams & Peng (1990); Shaban et al. (2019). While computationally efficient, truncated methods discard future gradient propagation beyond the truncation horizon, introducing a bias-accuracy trade-off.

Phantom Gradients and Gradient Extrapolation. Phantom Gradient (PG) methods Geng et al. (2021) estimate gradients from transient solver states using phantom updates, providing gradient estimation without requiring full solver convergence. However, they still rely on finite-horizon

propagation and retain truncation bias. Related approaches, including Neumann series approximations Lorraine et al. (2020); Liao et al. (2018) and Jacobian-Free Backpropagation (JFB) Fung et al. (2022), provide efficient approximations of implicit gradients by avoiding explicit Jacobian inversion, but generally rely on converged or near-converged solver states. Beyond gradient estimation, spectral properties of iterative optimization dynamics have been studied in numerical optimization Scieur et al. (2016); Walker & Ni (2011). Inspired by these observations, we introduce spectral extrapolation to exploit the locally predictable gradient evolution of transient solver trajectories and approximate missing gradient propagation beyond the truncation horizon.

3 PRELIMINARIES

Control tasks are often formulated as nonlinear optimal control problems with the following structure:

$$\begin{aligned} \min_{\{x_{1:T},\, u_{1:T}\}} \quad & \sum_{t=1}^{T} g_\theta(x_t, u_t) \\ \text{s.t.} \quad & x_{t+1} = f_\theta(x_t, u_t), \quad t = 1, \dots, T-1, \\ & x_1 = x_{\text{init}}, \\ & \underline{u} \le u_t \le \overline{u}, \quad t = 1, \dots, T, \end{aligned} \tag{1}$$

where $g_\theta(x_t, u_t)$ and $f_\theta(x_t, u_t)$ denote the cost function and system dynamics, respectively, parameterized by neural networks with parameters θ.

3.1 DIFFERENTIABLE OPTIMIZATION THROUGH ITERATIVE SOLVERS

The solution to the above problem is $h^*(\theta) = \{x^*(\theta), u^*(\theta)\}$. Many modern trajectory optimization solvers (e.g., iLQR, projected gradient descent) compute h^* via an iterative procedure. Starting from an initial guess $h^{(0)}$, the solver generates a sequence

$$h^{(k+1)} = \mathcal{F}\Big(h^{(k)}, \theta\Big), \tag{2}$$

where $\mathcal{F} : \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^N$ denotes a single iteration of the solver. For example, in the context of iLQR Wang et al. (2025), $\mathcal{F}$ represents a single iteration step involving system linearization, the backward Riccati pass, and the forward rollout. Under standard regularity conditions, the iteration converges to a unique fixed point h^* satisfying

$$h^* = \mathcal{F}(h^*, \theta)\,. \tag{3}$$

Our goal is to compute the gradient of a loss function $\mathcal{L}(h^*)$ with respect to the parameters θ, given by

$$\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial h^*} \frac{dh^*}{d\theta}. \tag{4}$$

Here, $\partial \mathcal{L}/\partial h^*$ can be computed with automatic differentiation, while the Jacobian $dh^*/d\theta$ is the quantity of interest.

3.2 IMPLICIT DIFFERENTIATION AND THE CONVERGENCE REQUIREMENT

The standard approach to compute the total derivative $dh^*/d\theta$ is Implicit Differentiation (IFT) Wang et al. (2025). By differentiating the fixed-point condition Eq. (3) at optimality, we obtain:

$$\frac{dh^*}{d\theta} = \left.\frac{\partial \mathcal{F}}{\partial h}\right|_{h^*} \frac{dh^*}{d\theta} + \left.\frac{\partial \mathcal{F}}{\partial \theta}\right|_{h^*}. \tag{5}$$

Rearranging terms yields the closed-form expression:

$$\frac{dh^*}{d\theta} = \left(I - \left.\frac{\partial \mathcal{F}}{\partial h}\right|_{h^*}\right)^{-1} \left.\frac{\partial \mathcal{F}}{\partial \theta}\right|_{h^*}. \tag{6}$$

While Eq. (6) theoretically yields the exact gradient, its validity strictly relies on the condition that h^* is a true fixed point.

Algorithm 1 Hybrid SE-PG (per-batch)

Require: Batch inputs θ, solver horizon T, phantom horizon k, tolerance δ_{conv}, damping λ
Ensure: Gradient $\nabla_\theta \mathcal{L}(h^*)$ for parameter update

1: Forward Pass: Run solver to obtain state $h_T = \mathcal{F}_T(\theta)$.
2: Stability Check: Compute convergence mask $m_i = \mathbb{I}(\|h_T^{(i)} - \mathcal{F}(h_T^{(i)})\| < \delta_{conv})$ and batch consensus $\alpha = \frac{1}{N}\sum_{i=1}^{N} m_i$.
3: Fusion Weights: Compute confidence weights $w_i = m_i \cdot \alpha^2$ for all samples.
4: for each sample $i \in \{1, \ldots, N\}$ do
5: IFT Calculation: Compute implicit gradient $G_{\text{IFT}}^{(i)}$ at $h_T^{(i)}$.
6: if $w_i < 1$ then
7: Phantom Rollout: Perform a k-step damped phantom rollout using Eq. 8.
8: Linearization: Compute J, g at anchor $h_{\text{anchor}} = h_{T+k-1}$ to define operator B and g_{anchor}.
9: Spectral Extrapolation: Estimate $\hat{\rho}$ from Eq. 10 and compute:

$$G_{\text{SE}}^{(i)} \leftarrow v_{T+k} + \frac{\hat{\rho}}{1-\hat{\rho}}\Delta v_{T+k}.$$

10: end if
11: Hybrid Fusion: Combine gradients based on confidence:

$$G_{\text{Hybrid}}^{(i)} \leftarrow (1 - w_i)G_{\text{SE}}^{(i)} + w_i G_{\text{IFT}}^{(i)}.$$

12: end for
13: return Batch Gradients $\{G_{\text{Hybrid}}^{(i)}\}_{i=1}^{N}$

3.3 Neumann Series and Truncated Gradient Propagation

For $J_* = \frac{\partial \mathcal{F}}{\partial h}|_{h^*}$, the implicit gradient can be interpreted through the Neumann series:

$$(I - J_*)^{-1} = \sum_{i=0}^{\infty} J_*^i. \tag{7}$$

This representation characterizes the infinite-horizon gradient propagation under the local fixed-point dynamics. In contrast, truncated methods only propagate gradients through a finite number of solver iterations and ignore the remaining contributions.

Unrolling-based Phantom Gradient (UPG) further improves truncated gradients by performing damped phantom rollouts from transient solver states Geng et al. (2021). However, finite rollout horizons still omit the remaining gradient tail.

4 Method: Hybrid SE-PG

We propose **Hybrid SE-PG**, a reliable gradient estimation framework designed to improve gradient estimation in pre-convergence solver regimes. Our approach combines two complementary strategies: (1) **Spectral Extrapolated Phantom Gradient (SE-PG)**, which estimates the missing gradient propagation from transient states via local spectral extrapolation (Sec. 4.2); and (2) an **Adaptive Hybrid Mechanism**, which seamlessly transitions to the IFT gradient as the solver stabilizes (Sec. 4.3). The complete procedure is summarized in Algorithm 1.

4.1 Problem Setup: Detached Tail and Transient States

We consider a parametric trajectory optimization problem solved by an iterative solver $\mathcal{F}$. Let $h^{(t+1)} = \mathcal{F}(h^{(t)}, \theta)$ denote the solver iteration. In practice, the solver runs for a finite number of iterations T, producing a state h_T. During gradient computation, we detach this state from previous

solver iterations:

$$h_T = \mathrm{detach}(h_T), \qquad \frac{\partial h_T}{\partial \theta} \equiv 0.$$

A fundamental challenge is that the available solver output h_T is often a transient state (pre-convergence) rather than a fixed point. Existing gradient estimation methods typically assume that the starting point is sufficiently close to the fixed point, which may not hold under limited solver iterations. Our goal is to estimate the missing gradient propagation beyond the transient state h_T by constructing a phantom trajectory and approximating its remaining gradient tail.

4.2 Spectral Extrapolated Phantom Gradients (SE-PG)

To bridge the gap between the finite phantom rollout from the transient state h_T and the infinite gradient propagation horizon, we propose the SE-PG estimator. This method constructs a phantom continuation of the solver dynamics, identifies a stable linearization anchor, and applies spectral extrapolation to estimate the remaining gradient propagation.

Following the phantom gradient formulation, we extend the solver dynamics beyond the transient state h_T using a damped phantom process:

$$h^{(j+1)} = (1-\lambda)h^{(j)} + \lambda\,\mathcal{F}(h^{(j)}, \theta), \quad \text{for } j \geq T. \tag{8}$$

Subsequently, we define the Linearization Anchor at the penultimate step of a k-step phantom horizon: $h_{\text{anchor}} = h_{T+k-1}, k \geq 2$. To analyze the local gradient dynamics, we linearize the solver map $\mathcal{F}$ at this anchor, defining the state Jacobian J_{anchor} and the parameter sensitivity g_{anchor} as:

$$J_{\text{anchor}} := \left.\frac{\partial \mathcal{F}}{\partial h}\right|_{h_{\text{anchor}}}, \quad g_{\text{anchor}} := \left.\frac{\partial \mathcal{F}}{\partial \theta}\right|_{h_{\text{anchor}}}.$$

Based on J_{anchor}, we construct the effective transition operator $B := (1-\lambda)I + \lambda J_{\text{anchor}}$. This operator governs the local contraction behavior of the gradient propagation. Under local stability, the dominant contraction factor is smaller than one.

We now analyze the evolution of the phantom gradient propagation v_t along the phantom trajectory. By differentiating the phantom dynamics (Eq. 8), the gradient follows an affine recurrence relation (see derivation in Appendix B):

$$v_{t+1} = Bv_t + \lambda g_{\text{anchor}}, \quad \text{for } t \geq T + k - 1. \tag{9}$$

This recurrence shows that the convergence behavior of the phantom gradient is governed by the transition operator B. Consequently, the increments of the gradient $\Delta v_t = v_t - v_{t-1}$ satisfy the homogeneous relation $\Delta v_t = B\Delta v_{t-1}$, which follows the same contraction behavior as a power iteration. We thus estimate the effective local contraction factor $\hat{\rho}$ using the ratio of successive increments at the anchor:

$$\hat{\rho} \;=\; \mathrm{clip}\Big(\frac{\|\Delta v_{T+k}\|_2}{\|\Delta v_{T+k-1}\|_2 + \epsilon}, 0, 1-\delta\Big), \tag{10}$$

where $\Delta v_{T+k} = v_{T+k} - v_{T+k-1}$. Relying on the local stability assumption over the latest phantom iterations, this ratio provides an efficient estimate of the effective contraction factor using only the available gradient history.

The SE-PG Estimator. Instead of truncating at h_{T+k}, SE-PG extrapolates the gradient evolution to the infinite horizon. Leveraging the geometric contraction property of the linearized dynamics, we compute the fixed point using the current gradient state v_{T+k} and its latest increment:

$$G_{\text{SE}} \;=\; \underbrace{v_{T+k}}_{\text{Phantom Gradient Base}} \;+\; \underbrace{\frac{\hat{\rho}}{1-\hat{\rho}}(v_{T+k} - v_{T+k-1})}_{\text{Spectral Tail Correction}}. \tag{11}$$

Here, the second term approximates the remaining gradient updates beyond the phantom horizon without full solver unrolling.

4.3 Adaptive Hybrid Mechanism

To combine the effectiveness of SE-PG for transient solver states with the accuracy of IFT near convergence, we construct a sample-wise fusion strategy based on the convergence status of the current batch. Let $\mathcal{B}$ be a batch of size N.

1. Convergence Mask (m_i): For each sample $i \in \mathcal{B}$, we define a binary mask indicating if the solver has reached a tolerance δ_{conv}:
$$m_i = \mathbb{I}(\|h_T^{(i)} - \mathcal{F}(h_T^{(i)})\| < \delta_{conv}). \tag{12}$$
2. Batch Convergence Rate (α): We calculate the global stability of the current training batch:
$$\alpha = \frac{1}{N}\sum_{i=1}^{N} m_i. \tag{13}$$
3. Fusion Weight (w_i): We define the fusion weight for sample i as:
$$w_i = m_i \cdot \alpha^2. \tag{14}$$

The final gradient is obtained by sample-wise fusion:

$$G_{\text{Hybrid}}^{(i)} \leftarrow (1 - w_i)G_{\text{SE}}^{(i)} + w_i G_{\text{IFT}}^{(i)}. \tag{15}$$

4.4 Theoretical Analysis

Let $v^\star$ denote the gradient obtained by fully differentiating through the complete optimization trajectory and let v_k denote the gradient after truncating the backward pass at iteration k. The missing gradient propagation is defined as

$$r_k = v^\star - v_k. \tag{16}$$

As derived in Appendix B, the residual gradient propagation can be characterized by the local increment dynamics

$$\Delta v_t = \rho \Delta v_{t-1} + \epsilon_t, \tag{17}$$

where $\Delta v_t = v_t - v_{t-1}$, ρ denotes the local evolution factor, and ϵ_t represents the deviation from the local spectral evolution.

Under this assumption, the missing gradient tail can be approximated by the geometric accumulation of future increments. Therefore, SE-PG uses

$$\hat{r}_k = \frac{\hat{\rho}}{1 - \hat{\rho}} \Delta v_k \tag{18}$$

to estimate the residual propagation.

Theorem 4.1 (Residual Gradient Propagation Approximation)**.** *The SE-PG approximation satisfies*

$$\|r_k - \hat{r}_k\| \leq C_1 |\rho - \hat{\rho}| \|\Delta v_k\| + C_2 \sum_t \|\epsilon_t\|, \tag{19}$$

where C_1 and C_2 depend on the local gradient propagation dynamics. The proof is provided in Appendix C.

Interpretation. The approximation error of SE-PG depends on the accuracy of the estimated evolution factor and the deviation of the actual gradient trajectory from the assumed evolution model.

Implementation. Although the analysis considers a general local gradient propagation model, SE-PG does not explicitly construct the transition operator. Instead, it estimates the effective evolution factor $\hat{\rho}$ directly from the observed gradient trajectory, preserving the computational efficiency of truncated differentiation. The validation of $\hat{\rho}$ is provided in Appendix D.

5 EXPERIMENTS

In this section, we evaluate Hybrid SE-PG on gradient fidelity, optimization performance, and component-wise ablations. Experiments include classical control benchmarks and additional high-dimensional manipulation tasks.

All experiments were conducted on a workstation equipped with an Intel Core i9 processor (16 cores, 2.5 GHz), 32 GB RAM, and an NVIDIA GeForce RTX 4060 GPU (CUDA 12.1).

5.1 GRADIENT FIDELITY ANALYSIS

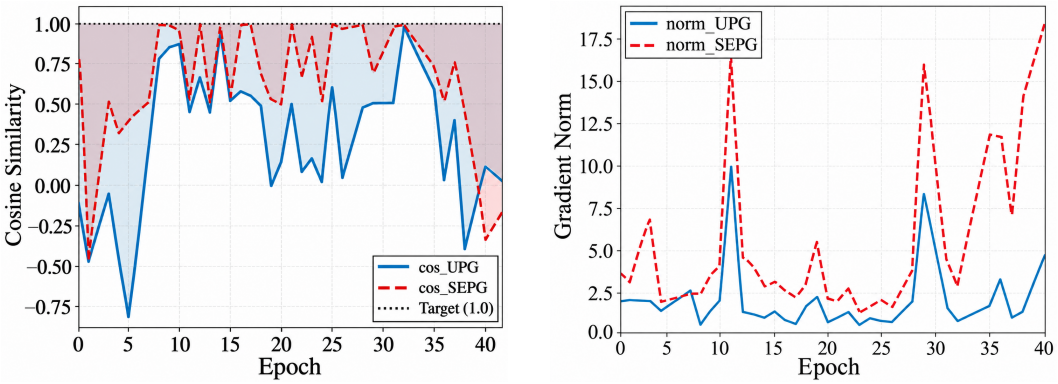
We first evaluate gradient quality against the converged IFT gradient as the reference during training. SE-PG ($k = 2$) is compared with Unrolling-based Phantom Gradient (UPG). As shown in Figure 2, SE-PG achieves higher cosine similarity, indicating improved gradient directional alignment. It also better preserves gradient magnitude compared with UPG, which suffers from finite-horizon truncation. These results demonstrate that SE-PG provides more accurate gradient estimation under pre-convergence solver states.

To further verify that the improvement is consistent across different initializations, we conduct a multi-seed evaluation on the Pendulum task. As summarized in Table 1, SE-PG achieves a mean cosine similarity of 0.7735 ± 0.0316 over five random seeds, substantially higher than UPG (0.4203 ± 0.0488). The lower variance of SE-PG also indicates more stable gradient estimation under solver non-convergence.

Table 1: Gradient direction fidelity comparison on Pendulum over five random seeds. Higher values indicate better alignment with the converged IFT gradient.

Method	Cosine Similarity (↑)
UPG	0.4203 ± 0.0488
SE-PG	$\mathbf{0.7735 \pm 0.0316}$

We further visualize the gradient behavior during actual training in Figure 3. The results show that SE-PG provides useful gradients especially in the early training stage, while Hybrid SE-PG maintains strong gradient alignment throughout training.



(a) Gradient Direction.

(b) Gradient Magnitude.

Figure 2: Gradient fidelity comparison under solver non-convergence. SE-PG consistently outperforms the unrolling-based phantom gradient (UPG) in both direction alignment and norm recovery.

5.2 PERFORMANCE EVALUATION ON CONTROL BENCHMARKS

Experimental Setup & Baselines. We evaluate on two benchmarks: (1) Cartpole (Cost Learning, $H = 20$) via inverse optimal control; and (2) Pendulum (Dynamics Learning, $H = 20$) via system identification. To rigorously test the pre-convergence regime, we enforce a strict solver budget of

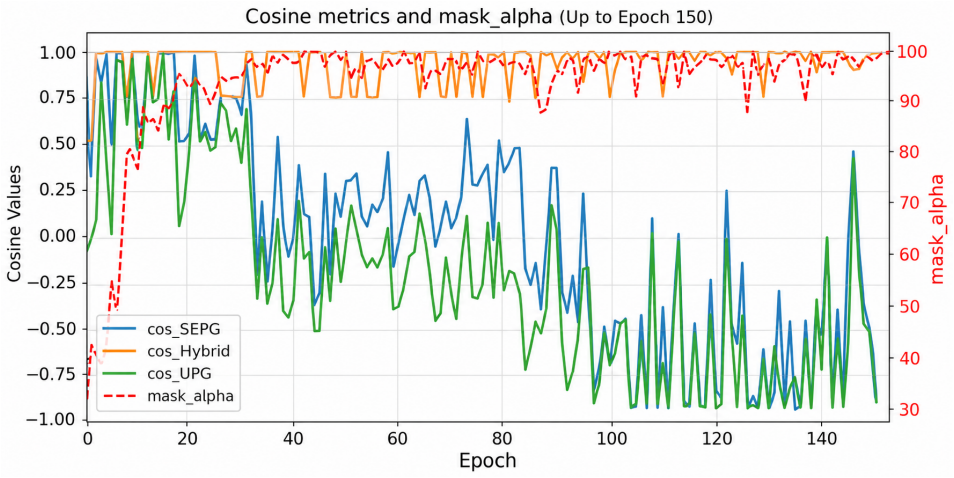


Figure 3: Gradient behavior during training. The red dashed curve shows the convergence-related mask statistic. Hybrid SE-PG maintains reliable gradients throughout optimization, while SE-PG improves early-stage estimation.

$T = 10$ iterations. We compare Hybrid SE-PG against: (1) Exact DiLQR (Oracle): Solves to strict convergence, representing the performance upper bound. (2) UPG (Baseline): Operates under the restricted budget ($T = 10$). We select UPG as the representative truncated method as standard TBPTT is numerically unstable in this regime.

Figures 4a and 4b report the training loss against wall-clock time. The results reveal a clear trade-off: UPG (Blue) yields high throughput but stagnates at a biased floor due to truncation. Exact DiLQR (Red) achieves optimal loss but incurs high computational cost. Hybrid SE-PG (Green) successfully bridges this gap, matching UPG's initial speed (via efficient $k = 2$ rollout) while continuously descending to match the Oracle's asymptotic accuracy.

We further challenged the method by increasing the planning horizon $H \in \{15, 20, 25\}$, which significantly increases difficulty. Across all settings (detailed in Appendix E), Hybrid SE-PG consistently converged to the optimal loss, demonstrating robustness to the horizon length.

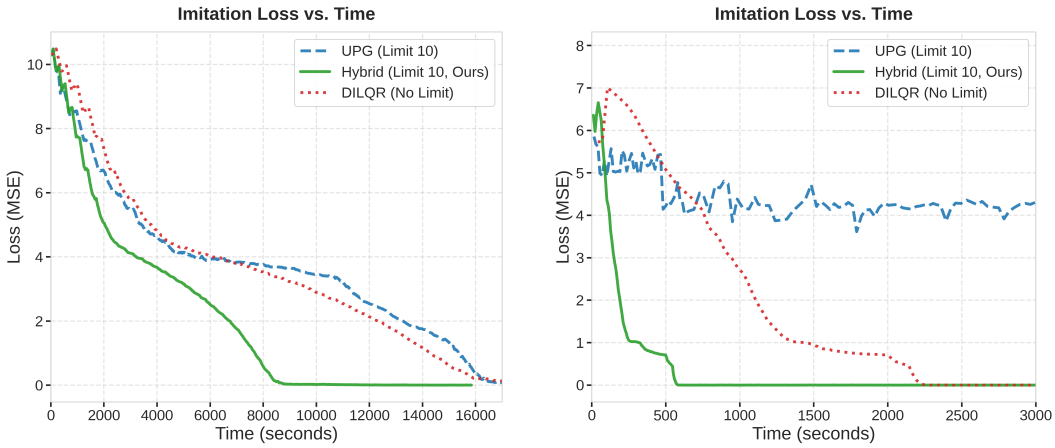
Additional evaluations on high-dimensional manipulation tasks are provided in Appendix F. We integrate Hybrid SE-PG into the differentiable optimization pipeline of DiffTORI Wan et al. (2024), replacing its original Theseus-based gradient estimation module Pineda et al. (2022). The results show that Hybrid SE-PG maintains effective task performance across diverse manipulation scenarios, demonstrating its compatibility with high-dimensional differentiable optimization pipelines.

5.3 Ablation Study: Effectiveness of Hybrid Mechanism

To analyze the contribution of SE-PG and the adaptive hybrid mechanism, we conduct an ablation study under the same experimental protocol as Sec. 5.2. We compare three variants: (1) Limited DiLQR, which uses IFT only for samples satisfying the convergence criterion; (2) Pure SE-PG, which applies SE-PG to all samples; and (3) Hybrid SE-PG, adaptive combination of SE-PG and IFT.

Figures 5a and 5b show the training trajectories. Limited DiLQR exhibits slower optimization when solver states remain far from convergence, since IFT gradients are unavailable for such transient states. In contrast, Pure SE-PG provides informative gradients throughout the transient regime, while Hybrid SE-PG combines transient-state gradient estimation with accurate IFT gradients as the solver approaches convergence, achieving the best overall optimization behavior.

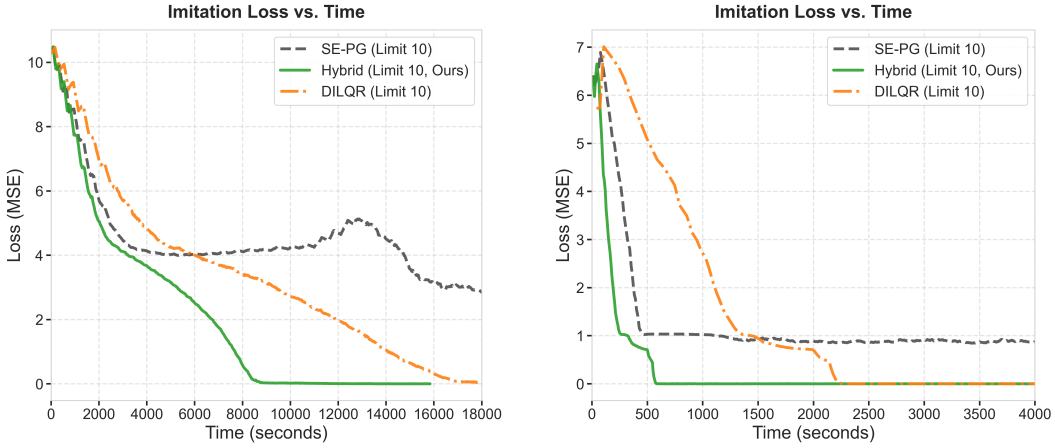
We further investigate the sensitivity of key hyperparameters that govern the gradient estimation and hybrid fusion mechanisms, including the phantom horizon k controlling the extrapolation window, the damping factor λ regulating phantom dynamics, the spectral clipping parameter controlling the stability of the estimated contraction factor, and the fusion schedule exponent determining the transition between SE-PG and IFT. Detailed results are provided in Appendix G.



(a) Cartpole Cost Learning.

(b) Pendulum Dynamics Learning.

Figure 4: Training efficiency comparison on classical control benchmarks. Hybrid SE-PG achieves a favorable trade-off between optimization speed and gradient fidelity by combining efficient transient gradients with accurate asymptotic gradients.



(a) Cartpole Ablation.

(b) Pendulum Ablation.

Figure 5: Ablation study of the hybrid mechanism. Across both benchmarks, Hybrid SE-PG combines the early optimization advantage of SE-PG with the asymptotic refinement of DiLQR, leading to improved training behavior over either component alone.

6 Conclusion

In this work, we introduced **Hybrid SE-PG**, a differentiable optimization framework for gradient estimation under pre-convergence solver states. The framework addresses the trade-off between gradient fidelity and computational cost by combining two complementary components: **SE-PG**, which estimates the missing gradient propagation beyond phantom rollouts, and an adaptive hybrid mechanism that incorporates IFT when the solver approaches convergence. Our theoretical analysis characterizes the approximation error of SE-PG under locally contractive gradient evolution. Experiments on classical control benchmarks demonstrate that Hybrid SE-PG provides more accurate gradient estimation under transient solver states while maintaining effective learning performance. Evaluations on high-dimensional manipulation tasks further demonstrate its compatibility with differentiable optimization pipelines. Future work will explore extending this gradient estimation perspective to broader differentiable optimization and model-based learning problems.

7 Ethics Statement

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used, were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

8 Reproducibility Statement

We have made every effort to ensure the reproducibility of the results presented in this paper. The experimental setup, including training procedures, model architectures, hyperparameters, optimization settings, and hardware configurations, is described in detail in the paper and appendices. We provide sufficient implementation details to enable independent reproduction of our experiments. We believe these materials will facilitate further investigation and development of differentiable optimization methods.

References

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pp. 136–145. PMLR, 2017.

Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J Zico Kolter. Differentiable mpc for end-to-end planning and control. *Advances in neural information processing systems*, 31, 2018.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18 (153):1–43, 2018.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in neural information processing systems*, 35:5230–5242, 2022.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31, 2018.

Justin Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, pp. 318–326. PMLR, 2012.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pp. 1568–1577. PMLR, 2018.

Samy Wu Fung, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. Jfb: Jacobian-free backpropagation for implicit networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6648–6656, 2022.

Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. *Advances in Neural Information Processing Systems*, 34:24247–24260, 2021.

Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

Wanxin Jin, Zhaoran Wang, Zhuoran Yang, and Shaoshuai Mou. Pontryagin differentiable programming: An end-to-end learning and control framework. *Advances in Neural Information Processing Systems*, 33:7979–7992, 2020.

Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. *Advances in neural information processing systems*, 30, 2017.

Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *First International Conference on Informatics in Control, Automation and Robotics*, volume 2, pp. 222–229. SciTePress, 2004.

Renjie Liao, Yuwen Xiong, Ethan Fetaya, Lisa Zhang, KiJung Yoon, Xaq Pitkow, Raquel Urtasun, and Richard Zemel. Reviving and improving recurrent back-propagation. In *International conference on machine learning*, pp. 3082–3091. PMLR, 2018.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pp. 1540–1552. PMLR, 2020.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pp. 2113–2122. PMLR, 2015.

Masashi Okada, Luca Rigazio, and Takenobu Aoshima. Path integral networks: End-to-end differentiable optimal control. *arXiv preprint arXiv:1706.09597*, 2017.

Marcus Pereira, David D Fan, Gabriel Nakajima An, and Evangelos Theodorou. Mpc-inspired neural network policies for sequential decision making. *arXiv preprint arXiv:1802.05803*, 2018.

Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, et al. Theseus: A library for differentiable nonlinear optimization. *Advances in Neural Information Processing Systems*, 35: 3801–3818, 2022.

Damien Scieur, Alexandre d'Aspremont, and Francis Bach. Regularized nonlinear acceleration. *Advances In Neural Information Processing Systems*, 29, 2016.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1723–1732. PMLR, 2019.

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *International conference on machine learning*, pp. 4732–4741. PMLR, 2018.

Corentin Tallec and Yann Ollivier. Unbiasing truncated backpropagation through time. *arXiv preprint arXiv:1705.08209*, 2017.

Yuval Tassa, Nicolas Mansard, and Emo Todorov. Control-limited differential dynamic programming. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1168–1175. IEEE, 2014.

Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

Weikang Wan, Ziyu Wang, Yufei Wang, Zackory Erickson, and David Held. Difftori: Differentiable trajectory optimization for deep reinforcement and imitation learning. *Advances in Neural Information Processing Systems*, 37:109430–109459, 2024.

Shuyuan Wang, Philip D Loewen, Michael Forbes, Bhushan Gopaluni, and Wei Pan. DiLQR: Differentiable iterative linear quadratic regulator via implicit differentiation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=m2EfTrbv4o.

Ronald J Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural computation*, 2(4):490–501, 1990.

A LLM USAGE

Large Language Models (LLMs) were used during the preparation of this manuscript to assist with literature discovery, research discussion, and writing refinement. Specifically, LLMs were used to help identify relevant background literature, explore potential research directions, and improve the clarity, readability, and organization of the manuscript.

The use of LLMs was limited to assisting the research process and manuscript preparation. All research ideas, methodological decisions, theoretical analyses, experimental designs, implementations, and interpretations of results were developed, verified, and conducted by the authors. The authors take full responsibility for the content of the manuscript and all claims presented herein.

B DERIVATION OF LINEARIZED GRADIENT PROPAGATION

In this section, we derive the local gradient propagation model used in the theoretical analysis.

Consider the damped phantom update:

$$h^{(t)} = (1-\lambda)h^{(t-1)} + \lambda \mathcal{F}(h^{(t-1)}, \theta). \tag{20}$$

We define the propagated gradient variable as

$$v_t := \frac{dh^{(t)}}{d\theta}. \tag{21}$$

Applying the chain rule yields

$$v_t = (1-\lambda)v_{t-1} + \lambda\left(\frac{\partial \mathcal{F}}{\partial h}v_{t-1} + \frac{\partial \mathcal{F}}{\partial \theta}\right) \tag{22}$$

$$= \left((1-\lambda)I + \lambda\frac{\partial \mathcal{F}}{\partial h}\right)v_{t-1} + \lambda\frac{\partial \mathcal{F}}{\partial \theta}. \tag{23}$$

Around the anchor point h_{anchor}, the Jacobian terms are locally approximated as constants:

$$B = (1-\lambda)I + \lambda J_{\text{anchor}}, \tag{24}$$

and

$$g_{\text{anchor}} = \left.\frac{\partial \mathcal{F}}{\partial \theta}\right|_{h_{\text{anchor}}}. \tag{25}$$

Therefore, the local gradient propagation follows

$$v_t \approx Bv_{t-1} + \lambda g_{\text{anchor}}. \tag{26}$$

Considering the gradient increment

$$\Delta v_t = v_t - v_{t-1}, \tag{27}$$

we obtain

$$\Delta v_t = (Bv_{t-1} + \lambda g_{\text{anchor}}) - (Bv_{t-2} + \lambda g_{\text{anchor}}) \tag{28}$$

$$= B(v_{t-1} - v_{t-2}) \tag{29}$$

$$= B\Delta v_{t-1}. \tag{30}$$

Thus, the local evolution of gradient increments is governed by the transition operator B. In practice, SE-PG estimates the effective evolution factor from the observed gradient increments rather than explicitly computing the operator.

C THEORETICAL PROOF OF THEOREM 4.1

We prove the residual gradient propagation approximation error bound.

The missing gradient propagation after truncation is defined as

$$r_k = \sum_{i=1}^{\infty} \Delta v_{k+i}. \tag{31}$$

According to the local gradient evolution model,

$$\Delta v_t = \rho \Delta v_{t-1} + \epsilon_t, \tag{32}$$

where ϵ_t represents the deviation from the ideal local evolution.

By recursively expanding the above equation, we obtain

$$\Delta v_{k+i} = \rho^i \Delta v_k + \sum_{j=1}^{i} \rho^{i-j} \epsilon_{k+j}. \tag{33}$$

Therefore, the residual propagation can be decomposed as

$$r_K = \sum_{i=1}^{\infty} \rho^i \Delta v_k + \sum_{i=1}^{\infty} \sum_{j=1}^{i} \rho^{i-j} \epsilon_{k+j} \tag{34}$$

$$= r_k^{\text{ideal}} + E_k, \tag{35}$$

where

$$r_k^{\text{ideal}} = \frac{\rho}{1-\rho} \Delta v_k \tag{36}$$

is the ideal residual propagation and E_k denotes the accumulated local evolution error.

For stable local evolution with $|\rho| < 1$, the accumulated error satisfies

$$\|E_k\| \le C_2 \sum_t \|\epsilon_t\|. \tag{37}$$

The SE-PG estimator uses

$$\hat{r}_k = \frac{\hat{\rho}}{1-\hat{\rho}} \Delta v_k. \tag{38}$$

Thus,

$$\|r_k - \hat{r}_k\| \le \|r_k - r_k^{\text{ideal}}\| + \|r_k^{\text{ideal}} - \hat{r}_k\| \tag{39}$$

$$\le \|E_k\| + \|r_k^{\text{ideal}} - \hat{r}_k\|. \tag{40}$$

For the second term, define

$$f(x) = \frac{x}{1-x}. \tag{41}$$

Then,

$$r_k^{\text{ideal}} - \hat{r}_k = (f(\rho) - f(\hat{\rho})) \Delta v_k. \tag{42}$$

Using the mean value theorem,

$$|f(\rho) - f(\hat{\rho})| \le C_1 |\rho - \hat{\rho}|, \tag{43}$$

where C_1 depends on the local contraction factor.

Therefore,

$$\|r_k^{\text{ideal}} - \hat{r}_k\| \le C_1 |\rho - \hat{\rho}| \|\Delta v_k\|. \tag{44}$$

Combining the two terms gives

$$\|r_k - \hat{r}_k\| \le C_1 |\rho - \hat{\rho}| \|\Delta v_k\| + C_2 \sum_t \|\epsilon_t\|. \tag{45}$$

This completes the proof.

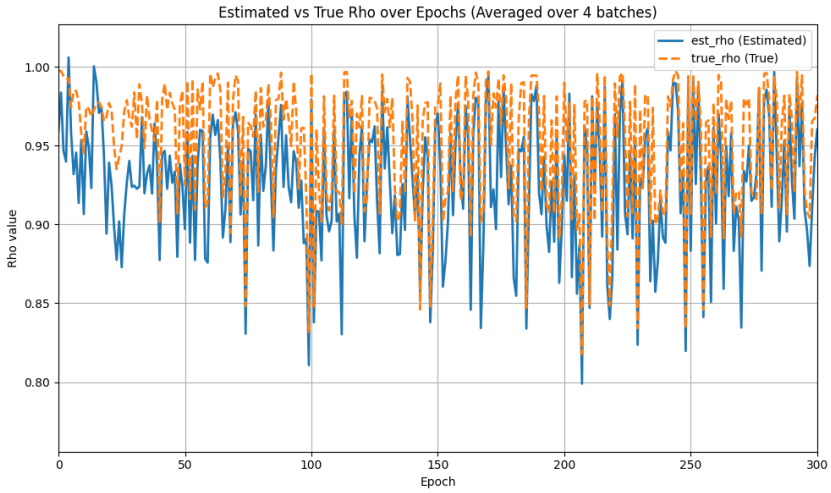


Figure 6: Validation of the estimated contraction factor. The estimated $\hat{\rho}$ is compared with the ground-truth spectral radius ρ_B of the local gradient propagation operator.

D Validation of Spectral Contraction Estimation

The SE-PG estimator relies on the local contraction assumption that the gradient increments approximately follow a dominant contraction mode. To validate this assumption, we compare the estimated contraction factor $\hat{\rho}$ with the ground-truth spectral radius of the local transition operator.

Specifically, given the linearized gradient propagation operator B, the reference contraction factor is computed as

$$\rho_B = \rho(B), \tag{46}$$

where $\rho(\cdot)$ denotes the spectral radius. Note that ρ_B is only used for evaluation and is not required during SE-PG computation. In contrast, SE-PG estimates $\hat{\rho}$ directly from observed gradient increments without explicitly constructing B.

Figure 6 compares $\hat{\rho}$ and ρ_B during optimization. The estimated contraction factor follows the reference value with small deviation, supporting the local contraction assumption adopted in SE-PG.

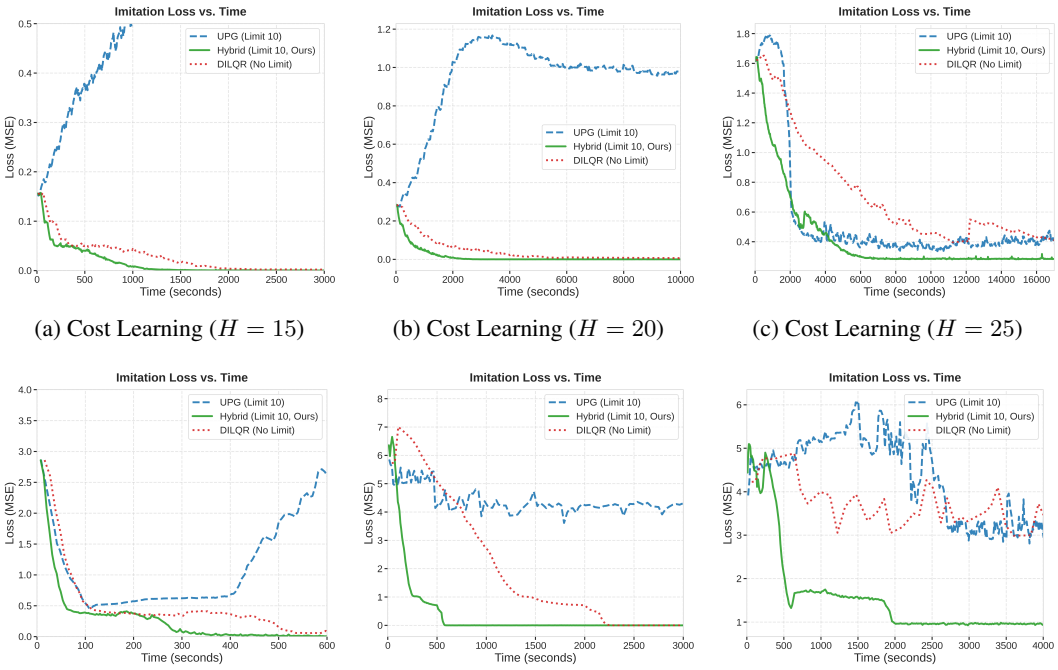
E Additional Analysis on Planning Horizon (Pendulum)

Figure 7 presents the training efficiency on the Pendulum task across varying planning horizons $H \in \{15, 20, 25\}$. A longer horizon H corresponds to a more difficult optimization problem. The results confirm that **Hybrid SE-PG** consistently achieves asymptotic optimality comparable to the Oracle (DiLQR) across all difficulty levels, whereas the UPG baseline struggles with higher bias.

F Additional Evaluation on High-Dimensional Tasks

To further evaluate the applicability of Hybrid SE-PG beyond low-dimensional control benchmarks, we integrate the proposed gradient estimation method into a high-dimensional manipulation pipeline based on DiffTORI. Specifically, we replace the original Theseus-based differentiable optimization module with Hybrid SE-PG, while keeping the remaining training pipeline and task settings unchanged.

We evaluate the two optimization modules on a set of manipulation tasks with different levels of difficulty. The objective of this experiment is not to introduce a new manipulation benchmark,



(a) Cost Learning ($H = 15$) (b) Cost Learning ($H = 20$) (c) Cost Learning ($H = 25$)

(d) Dynamics Learning ($H = 15$) (e) Dynamics Learning ($H = 20$) (f) Dynamics Learning ($H = 25$)

Figure 7: **Robustness to Task Difficulty (Pendulum).** Training curves for Cost Learning (Top) and Dynamics Learning (Bottom) with increasing planning horizons $H \in \{15, 20, 25\}$. Our method (Green) maintains optimality as task difficulty increases.

Table 2: **Success Rates on Additional High-Dimensional Manipulation Tasks.** Hybrid SE-PG replaces the Theseus-based differentiable optimization module. Results are reported as mean $\pm$ standard deviation across multiple seeds.

Method	Shelf Place	Bin Picking	Assembly	Sweep	Hand Insert
Theseus	0.300 ± 0.038	0.176 ± 0.043	0.928 ± 0.053	0.864 ± 0.054	0.136 ± 0.056
Hybrid SE-PG (Ours)	0.300 ± 0.077	0.184 ± 0.029	0.940 ± 0.036	0.876 ± 0.037	0.124 ± 0.034

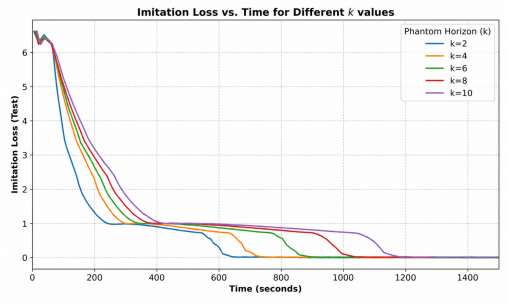
Method	Disassemble	Stick Push	Pick Place Wall	Coffee Pull	Coffee Push	Average
Theseus	0.696 ± 0.065	0.996 ± 0.008	0.648 ± 0.037	0.472 ± 0.035	0.812 ± 0.065	0.603
Hybrid SE-PG (Ours)	0.724 ± 0.043	0.988 ± 0.016	0.660 ± 0.031	0.440 ± 0.083	0.832 ± 0.070	0.607

but to verify whether Hybrid SE-PG can serve as a drop-in replacement for existing differentiable optimization components in high-dimensional tasks.

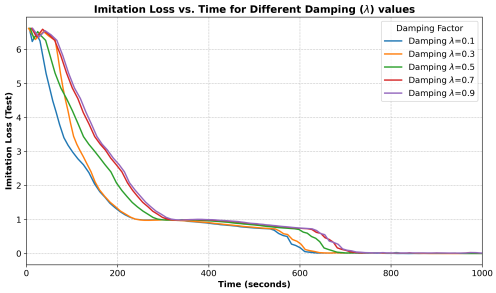
As shown in Table 2, Hybrid SE-PG achieves comparable task success rates with the Theseus-based baseline across diverse manipulation scenarios. The results indicate that improving gradient estimation under solver non-convergence does not compromise the final task performance, and the proposed method can be effectively integrated into more complex optimization pipelines.

G Hyperparameter Sensitivity Analysis

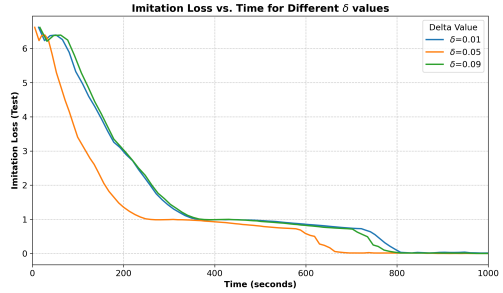
We further investigate the sensitivity of the key hyperparameters in Hybrid SE-PG, including the phantom horizon k, damping factor λ, spectral clipping parameter δ, and fusion schedule exponent p.



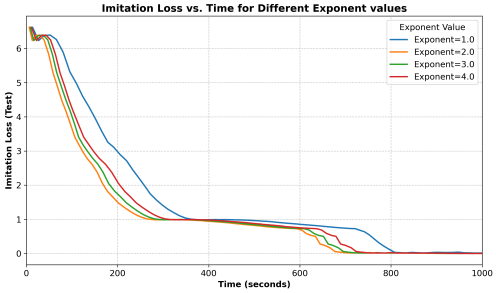
(a) Phantom horizon k.



(b) Damping factor λ.



(c) Spectral clipping parameter δ.



(d) Fusion exponent p.

Figure 8: Hyperparameter sensitivity analysis of Hybrid SE-PG on Pendulum dynamics learning.

All experiments are conducted on the Pendulum dynamics learning task under the same settings as the main experiments.

As shown in Figure 8, the selected default parameters provide a favorable trade-off between optimization efficiency and gradient reliability. For the phantom horizon, increasing k introduces additional computational overhead, while $k = 2$ achieves the best wall-clock performance. For the damping factor, smaller values result in more conservative phantom updates and slower convergence, with $\lambda = 0.1$ performing best in our experiments. The spectral clipping parameter controls the stability of the spectral extrapolation, where $\delta = 0.05$ provides the best balance between extrapolation accuracy and stability. Finally, we evaluate different fusion schedule exponents and find that the quadratic schedule ($p = 2$) achieves the best overall performance, while smaller or larger exponents lead to either aggressive or delayed transitions between SE-PG and IFT.

Based on these results, we use $k = 2$, $\lambda = 0.1$, $\delta = 0.05$, and $p = 2$ as the default configuration throughout the paper.