Non-Stationary Functional Bilevel Optimization

Jason Bohne[1*] **Ieva Petrulionyte**[2*] **Michael Arbel**[2] **Julien Mairal**[2] **Paweł Polak**[1]
[1]Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY, USA
[2]Inria, CNRS, Grenoble INP, LJK, Université Grenoble Alpes

Abstract

Functional bilevel optimization (FBO) provides a powerful framework for hierarchical learning in function spaces, yet current methods are limited to static offline settings and perform suboptimally in online, non-stationary scenarios. We propose **SmoothFBO**, the first algorithm for non-stationary FBO with both theoretical guarantees and practical scalability. SmoothFBO introduces a time-smoothed stochastic hypergradient estimator that reduces variance through a window parameter, enabling stable outer-loop updates with sublinear regret. Importantly, the classical parametric bilevel case is a special reduction of our framework, making SmoothFBO a natural extension to online, non-stationary settings. Empirically, SmoothFBO consistently outperforms existing FBO methods in non-stationary hyperparameter optimization and model-based reinforcement learning, demonstrating its practical effectiveness. Together, these results establish SmoothFBO as a general, theoretically grounded, and practically viable foundation for bilevel optimization in online, non-stationary scenarios.

1 INTRODUCTION

Bilevel optimization has emerged as a powerful paradigm for solving complex nested optimization problems in various domains. Initially employed in model selection (Bennett et al., 2006) and sparse feature learning (Mairal et al., 2012), this approach subsequently emerged as an efficient alternative to grid search for hyperparameter optimization tuning (Feurer and Hutter, 2019; Lorraine et al., 2019; Franceschi et al., 2017). More recently, the application domain of bilevel optimization has expanded considerably to encompass meta-learning (Bertinetto et al., 2019), auxiliary task learning (Navon et al., 2021), inverse problems (Holler et al., 2018), invariant risk minimization (Arjovsky et al., 2019; Ahuja et al., 2020) and reinforcement learning (Hong et al., 2023; Liu et al., 2021; Nikishin et al., 2022). Traditional bilevel optimization approaches typically operate in parameter spaces, where the inner and outer optimization problems are formulated over finite-dimensional vectors. Recently, Petrulionyte et al. (2024) introduced Functional Bilevel Optimization (FBO), which extends this framework to function spaces, allowing for an expressive theoretical framework and a novel algorithmic approach.

Despite the success of bilevel optimization methods, current algorithms predominantly address static environments where loss functions are expectations over stationary data distributions. However, numerous real-world applications involve dynamic environments where data distributions change over time (Besbes et al., 2015). In reinforcement learning, for instance, environment dynamics may evolve over time, requiring agents to continuously adapt their policies (Besbes et al., 2015; Padakandla et al., 2020). Similarly, in online learning, the patterns in data often change systematically over time (Tarzanagh et al., 2024). These challenges motivate the development of bilevel optimization methods that can efficiently handle such non-stationary settings. While limited research has explored parametric bilevel approaches in non-stationary environments (Bohne et al., 2024; Tarzanagh et al., 2024), a comprehensive functional perspective remains undeveloped for such applications, representing a significant gap in the literature.

In this paper, we address this gap by introducing Non-Stationary Functional Bilevel Optimization (NS-FBO), a framework that extends functional bilevel optimization to time-varying settings. Formally, we aim to solve

[*]Equal contribution.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

$\forall t \in [1, T]$:

$$\begin{aligned} \min_{\boldsymbol{\lambda} \in \Lambda} \ & \mathcal{F}_t(\boldsymbol{\lambda}) := L_t^{out}\big(\boldsymbol{\lambda}, h_{t,\boldsymbol{\lambda}}^\star\big) \\ \text{s.t.} \quad & h_{t,\boldsymbol{\lambda}}^\star = \arg\min_{h \in \mathcal{H}} \ L_t^{in}\big(\boldsymbol{\lambda}, h\big). \end{aligned} \tag{NS-FBO}$$

Where we have a sequence of time-varying loss functions (L_t^{out}, L_t^{in}) for $t = 1, \ldots, T$ due to time-varying data distributions $\mathbb{P}_t, \mathbb{Q}_t$. With this definition, a special case of an NS-FBO problem is the following stochastic non-stationary bilevel optimization problem defined $\forall t \in [1, T]$,

$$\begin{aligned} \min_{\boldsymbol{\lambda} \in \Lambda} \ & L_t^{out}\big(\boldsymbol{\lambda}, h_{t,\boldsymbol{\lambda}}^\star\big) := \mathbb{E}_{x,y \sim \mathbb{P}_t}\big[\ell_{out}\big(\boldsymbol{\lambda}, h_{t,\boldsymbol{\lambda}}^\star(x), x, y\big)\big] \\ \text{s.t.} \quad & h_{t,\boldsymbol{\lambda}}^\star = \arg\min_{h \in \mathcal{H}} \ \mathbb{E}_{x,y \sim \mathbb{Q}_t}[\ell_{in}(\boldsymbol{\lambda}, h(x), x, y)]\,. \end{aligned} \tag{1}$$

where $\mathcal{F}_t$ represents the outer objective function at time t, which depends on the solution to the inner problem $h_{t,\boldsymbol{\lambda}}^\star$, and ℓ_{in}, ℓ_{out} are point-wise losses. Differently from classical parametric approaches, the inner variable $h_{t,\boldsymbol{\lambda}}$ is a function living in some function space $\mathcal{H}$. The inner and outer objectives involve expectations over potentially different data distributions $\mathbb{P}_t$ and $\mathbb{Q}_t$ that evolve over time. We denote $\Omega = \mathbb{P}_t \times \mathbb{Q}_t$ as the joint distribution from data samples $(x_t, y_t) \sim \mathbb{P}_t$ from the outer objective and $(x_t, y_t) \sim \mathbb{Q}_t$ from the inner objective.

Our work bridges the gap between functional bilevel optimization and online learning, enabling efficient optimization in non-stationary environments. The contributions of this paper are summarized as follows:

1. We formulate the Non-Stationary Functional Bilevel Optimization (NS-FBO) problem for stochastic settings with time-varying data distributions.
2. We develop *SmoothFBO*, an efficient algorithm that incorporates time-smoothing techniques to handle temporal dependencies and reduce variance in gradient estimates.
3. We provide theoretical convergence guarantees for our proposed algorithm.
4. We demonstrate the practical efficacy of our approach on a controlled synthetic non-stationary regression task and on model-based reinforcement learning in non-stationary environments.

The remainder of this paper is organized as follows: *Section 2* presents the preliminaries from functional and online bilevel optimization literature; *Sections 3 and 4* provide our proposed methods and the theoretical analysis of its convergence properties; *Section 5* presents experiments on a synthetic non-stationary regression and on model-based reinforcement learning.

2 RELATED WORK

Bilevel optimization and hypergradients. Bilevel optimization is a standard tool for hyperparameter tuning and meta-learning. Most practical methods rely on either differentiating through an inner solver ("unrolling") or using implicit differentiation to avoid storing the full trajectory (Franceschi et al., 2017; Lorraine et al., 2019; Shaban et al., 2019; Arbel and Mairal, 2022). Our work follows this line, but targets a setting where the objectives drift over time and where the inner problem is posed in a function space rather than in a finite-dimensional parameterization.

Functional bilevel optimization. Functional Bilevel Optimization represents a paradigm shift from traditional parameter-centric approaches to a function-space perspective for bilevel optimization problems. This framework, introduced by Petrulionyte et al. (2024), effectively addresses the ambiguity challenges that emerge when employing deep neural networks in bilevel optimization settings. The functional perspective provides a theoretical framework that accurately describes the actual techniques used by machine learning practitioners and derives a novel functional implicit differentiation rule. In this work we keep the same functional viewpoint, but move from an offline regime to a non-stationary one. Algorithmically, we introduce a time-smoothed stochastic estimator of the functional hypergradient and show how the window controls the stability–adaptation tradeoff.

Online / non-stationary bilevel optimization. Recent work has started to analyze bilevel learning with time-varying objectives in the parametric setting, typically under smoothness/strong-convexity assumptions and with regret guarantees. In particular, Lin et al. (2023) study nonconvex bilevel problems with time-varying objectives, and more recent online bilevel optimization methods use window-averaged hypergradients to reduce variance and improve tracking (Bohne et al., 2024; Tarzanagh et al., 2024). For instance, Bohne et al. (2024) showed that averaging past gradients mitigates the impact of stochastic noise, enabling better tracking of slowly changing optima in parametric bilevel problems. However, these approaches are limited to finite-dimensional parameter spaces, which may not capture the expressive power of function spaces required for complex tasks like those in reinforcement learning. *SmoothFBO* can be seen as a functional gen-

eralization of this idea: we apply time-smoothing at the level of functional hypergradients (estimated from data), which lets us treat parametric bilevel optimization as a special case while retaining the expressivity of function spaces needed for applications such as reinforcement learning.

Online / non-stationary multi-level optimization. Our regret analysis also connects to the broader online optimization literature, where sublinear dynamic regret typically requires controlling the temporal variation of the losses ("gradual variation") (Besbes et al., 2015; Chiang et al., 2012; Yang et al., 2016). A related viewpoint appears in time-varying multi-objective optimization, where one seeks guarantees for multiple competing objectives as they drift in time (Shafiei and Marecek, 2025). While our setting is bilevel (rather than multi-level), we share the same goal of making the stability-performance tradeoff transparent, here via an explicit window parameter. Finally, multi-level *convex* problems have been studied through monotone operator theory and fixed-point arguments. Shafiei et al. (2024) propose first-order fixed-point algorithms for nested convex programs and analyze convergence rates under monotonicity assumptions. This is complementary to our work: we focus on stochastic, non-stationary bilevel learning, and we emphasize function-space modeling, whereas monotone-operator analyses typically target deterministic, finite-dimensional convex formulations.

3 PRELIMINARIES

3.1 Functional Bilevel Optimization

Here we describe how Functional Bilevel Optimization (FBO) can be effectively used to compute the hypergradient for the non-stationary bilevel problem introduced in the previous section. The non-stationary functional bilevel problem (NS-FBO) involves an optimal prediction function $h^\star_{t,\boldsymbol{\lambda}}$ for each value of the outer-level parameter $\boldsymbol{\lambda}$. Solving NS-FBO by using a first-order method then requires characterizing the implicit dependence of $h^\star_{t,\boldsymbol{\lambda}}$ on the outer-level parameter $\boldsymbol{\lambda}$ to evaluate the hypergradient $\nabla\mathcal{F}_t$ in $\mathbb{R}^d$. Using functional implicit differentiation and the adjoint sensitivity method from Petrulionyte et al. (2024), under differentiability and standard optimization assumptions detailed in Appendix A.4, the functional hypergradient $\nabla\mathcal{F}_t$ is given by:

$$\begin{aligned} \nabla\mathcal{F}_t(\boldsymbol{\lambda}) &= \partial_{\boldsymbol{\lambda}} L_t^{out}(\boldsymbol{\lambda}, h^\star_{\boldsymbol{\lambda}}) + B_{\boldsymbol{\lambda}}\, a^\star_{t,\boldsymbol{\lambda}}, \\ \text{with} \quad B_{\boldsymbol{\lambda}} &:= \partial_{\boldsymbol{\lambda},h} L_t^{in}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}). \end{aligned} \tag{2}$$

with $a^\star_{t,\boldsymbol{\lambda}} := -C_{\boldsymbol{\lambda}}^{-1} d_{\boldsymbol{\lambda}}$ an element of $\mathcal{H}$ that minimizes the quadratic objective:

$$\begin{aligned} a^\star_{\boldsymbol{\lambda}} &= \arg\min_{a\in\mathcal{H}}\ L_{adj}(\boldsymbol{\lambda}, a) \\ &:= \tfrac{1}{2}\ \langle a, C_{\boldsymbol{\lambda}} a\rangle_{\mathcal{H}} + \langle a, d_{\boldsymbol{\lambda}}\rangle_{\mathcal{H}}, \\ \text{with} \quad C_{\boldsymbol{\lambda}} &:= \partial_h^2 L_t^{in}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}). \end{aligned} \tag{3}$$

FBO is a class of practical algorithms designed to estimate the functional hypergradient $\nabla\mathcal{F}_t(\boldsymbol{\lambda})$ in the stationary context. Our goal is to generalize this class of algorithms to non-stationary environments when both the outer and inner losses are expectations on time-varying probability distributions.

3.2 Online Bilevel Optimization

Online Bilevel Optimization (OBO) extends the bilevel optimization framework to dynamic, non-stationary environments where objectives evolve over time due to shifting data distributions or environmental changes. Unlike traditional bilevel optimization, which relies on static datasets, OBO involves sequential learning in response to streaming data, making it critical for applications like online reinforcement learning (Padakandla et al., 2020). In such settings, the inner and outer objectives, defined over time-varying distributions $\mathbb{P}_t$ and $\mathbb{Q}_t$, require algorithms to adaptively track drifting optima while maintaining optimization stability.

To ensure meaningful regret bounds in non-stationary environments, such as the stochastic bilevel optimization problem in (1), regularity constraints on the sequence of objectives are essential (Besbes et al., 2015). These constraints, often expressed as sublinear comparator sequences, quantify the temporal variation in objectives. One key metric in OBO is the p-th order outer-level function variation, defined as:

$$\begin{aligned} V_{p,T} := \sum_{t=1}^{T} \sup_{\boldsymbol{\lambda}\in\Lambda} \Big| &\mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\mathcal{F}_{t+1}(\boldsymbol{\lambda}, h^\star_{t+1,\boldsymbol{\lambda}}, x, y)\right] \\ &- \mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\mathcal{F}_t(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}, x, y)\right]\Big|^p, \end{aligned} \tag{4}$$

where $\mathcal{F}_t(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}, x, y) = \ell_{out}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}(x), x, y)$ is the outer objective from (1), with $\mathcal{F}_t = 0$ for $t < 0$. The metric $V_{p,T}$ (Lin et al., 2023) measures changes in the outer objective's expected value. In this work, we focus on first-order outer-level function variation $V_{1,T} = o(T)$ to derive sublinear regret bounds for our proposed Algorithm 3.

4 SmoothFBO: A GENERALIZED FBO ALGORITHM

This section introduces the generalized Functional Bilevel Optimization (FBO) algorithm, *SmoothFBO*

in Algorithm 1. Our proposed method extends current algorithms for functional bilevel optimization Petrulionyte et al. (2024) to non-stationary environments by introducing a stochastic hypergradient estimator constructed via time smoothing, a technique commonly employed in online algorithms (Hazan et al., 2017; Lin et al., 2023). For simplicity, we first introduce a stochastic functional hypergradient oracle that allows us to conveniently analyze the bias and variance properties of our time-smoothed hypergradient estimator in Algorithm 1, independent of confounding factors arising from hypergradient estimation.

Definition 4.1 (Stochastic Hypergradient Oracle)**.** We define a stochastic functional hypergradient oracle $\mathcal{O}(\boldsymbol{\lambda})$ that returns an i.i.d. vector $\widehat{\nabla \mathcal{F}}_t(\boldsymbol{\lambda})$ such that

$$\mathbb{E}_{\Omega_t}\left[\widehat{\nabla \mathcal{F}}_t(\boldsymbol{\lambda})\right] = \nabla \mathcal{F}_t(\boldsymbol{\lambda}),$$
$$\mathrm{Var}_{\Omega_t}\left[\widehat{\nabla \mathcal{F}}_t(\boldsymbol{\lambda})\right] := \mathbb{E}_{\Omega_t}\left[\|\widehat{\nabla \mathcal{F}}_t(\boldsymbol{\lambda})\|^2\right] - \left\|\mathbb{E}_{\Omega_t}\left[\widehat{\nabla \mathcal{F}}_t(\boldsymbol{\lambda})\right]\right\|^2 \leq \sigma_f^2,$$

where $\nabla \mathcal{F}_t(\boldsymbol{\lambda})$ denotes the true hypergradient given in (2), and $\Omega_t = \mathbb{P}_t \times \mathbb{Q}_t$.

At each round t, Algorithm 1 queries the stochastic hypergradient oracle to obtain an estimate $\widehat{\nabla \mathcal{F}}_t(\boldsymbol{\lambda})$ of the true hypergradient $\nabla \mathcal{F}_t(\boldsymbol{\lambda})$. While single-round stochastic estimates are effective in stationary settings, such estimates are less suitable for non-stationary environments where gradients between rounds may change rapidly. Lemma 5.1 introduces a time-smoothed stochastic estimate, constructed over a window of length w, which preserves unbiasedness and substantially reduces variance by a factor of w.

Lemma 4.2 (Time-Smoothed Hypergradient Estimator)**.** *Let $\widehat{\nabla \mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})$ denote the estimate from the oracle $\mathcal{O}(\boldsymbol{\lambda}_{t-i})$ for $i = 0, \ldots, w-1$. Define*

$$\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) := \frac{1}{w}\sum_{i=0}^{w-1} \widehat{\nabla \mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}),$$

with $\mathcal{F}_t = 0$ for $t < 0$ and $\mathcal{Z}_{t,w} = \prod_{i=0}^{w-1} \Omega_{t-i}$. Then

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right] = \frac{1}{w}\sum_{i=0}^{w-1} \nabla \mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i}),$$
$$Var_{\mathcal{Z}_{t,w}}\left[\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right] \leq \frac{\sigma_f^2}{w}.$$

Proof. Expectation follows by linearity such that we have $\mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla \mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})\right] = \nabla \mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})$ and further $\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right] = \frac{1}{w}\sum_{i=0}^{w-1} \nabla \mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})$. For variance, we have $\mathrm{Var}_{\mathcal{Z}_{t,w}}\left[\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right] = \frac{1}{w^2}\sum_{i=0}^{w-1} \mathrm{Var}_{\Omega_{t-i}}\left[\widehat{\nabla \mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})\right] \leq \frac{\sigma_f^2}{w}$. □

Algorithm 1 ***SmoothFBO*** (Smooth Functional Bilevel Optimization)

Require: Initial outer parameter $\boldsymbol{\lambda}_1$, step size $\alpha > 0$, window size $w \geq 1$, stochastic oracle $\mathcal{O}(\boldsymbol{\lambda})$

for $t = 1, \ldots, T$ **do**
  $\widehat{\nabla \mathcal{F}}_t(\boldsymbol{\lambda}_t) \leftarrow \mathcal{O}(\boldsymbol{\lambda}_t)$
  $\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) \leftarrow \frac{1}{w}\sum_{i=0}^{w-1} \widehat{\nabla \mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})$
  $\boldsymbol{\lambda}_{t+1} \leftarrow \boldsymbol{\lambda}_t - \alpha \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$
end for

Lemma 4.2 establishes that averaging w stochastic estimates reduces variance by a factor of w while remaining unbiased with respect to the windowed average hypergradients. This variance reduction stabilizes Algorithm 1 in non-stationary environments.

Definition 4.3 (Bilevel Local Regret)**.** Given window length $w \geq 1$ and outer variables $\{\boldsymbol{\lambda}_t\}_{t=1}^T$, define

$$\mathrm{BLR}_w(T) := \sum_{t=1}^{T} \left\|\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2,$$

where $\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) := \frac{1}{w}\sum_{i=0}^{w-1} \mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})$ with $\mathcal{F}_t = 0$ for $t < 0$.

Theorem 4.4. *Under the assumptions of Section A.1, Algorithm 1 with step size $\alpha = 1/L$ satisfies*

$$BLR_w(T) = \sum_{t=1}^{T} \mathbb{E}_{\mathcal{Z}_{t,w}}\left[\|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2\right] \leq 2L\left(\frac{2TQ}{w} + V_{1,T} + \frac{T\sigma_f^2}{2Lw}\right).$$

Corollary 4.5. *For window size $w = o(T)$, the regret $BLR_w(T)$ is sublinear. In particular,*

$$\sum_{t=1}^{T} \mathbb{E}_{\mathcal{Z}_{t,w}}\left[\|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2\right] \leq O\left(\frac{TQ}{w} + V_{1,T} + \frac{T\sigma_f^2}{w}\right).$$

5 SmoothFBO WITH HYPERGRADIENT ESTIMATION

This section presents Algorithm 2, an extension of **SmoothFBO** (Algorithm 1), which replaces the hypergradient oracle with a time-smoothed stochastic functional hypergradient estimator for the outer objective $\mathcal{F}_t(\boldsymbol{\lambda}) = \mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}(x), x, y)\right]$. Building on the original time-smoothed estimator of Lemma (5.1), this approach averages stochastic functional hypergradient estimates from the following algorithm (Algorithm 2) over a window of size w, achieving variance reduction for approximate stochastic gradients similar to the oracle setting, as established in Theorem 5.3. Algorithm 2 is for functional hypergradient estimation

from Equation (2), the subroutines for finding the adjoint function (`AdjointOpt`) and the inner prediction function (`InnerOpt`) can be found in Appendix A.3.

Algorithm 2 FuncGrad$(\boldsymbol{\lambda}, h, a, \mathcal{D})$

Require: current outer, inner, and adjoint models $\boldsymbol{\lambda}$, h, a, dataset $\mathcal{D} = (\mathcal{D}_{in}, \mathcal{D}_{out})$
 # Inner-level optimization
$\hat{h}_{\boldsymbol{\lambda}} \leftarrow \texttt{InnerOpt}(\boldsymbol{\lambda}, h, \mathcal{D}_{in})$
 # Adjoint optimization
$\hat{a}_{\boldsymbol{\lambda}} \leftarrow \texttt{AdjointOpt}(\boldsymbol{\lambda}, a, \hat{h}_{\boldsymbol{\lambda}}, \mathcal{D})$
 # Hypergradient estimation
Sample a mini-batch $\mathcal{B} = (\mathcal{B}_{out}, \mathcal{B}_{in})$ from $\mathcal{D} = (\mathcal{D}_{out}, \mathcal{D}_{in})$
$g_{Exp} \leftarrow \partial_{\boldsymbol{\lambda}} \hat{L}_{out}(\boldsymbol{\lambda}, \hat{h}_{\boldsymbol{\lambda}}, \mathcal{B}_{out})$
$g_{Imp} \leftarrow \frac{1}{|\mathcal{B}_{in}|} \sum_{(x,y)\in\mathcal{B}_{in}} \partial_{\boldsymbol{\lambda},v} \ell_{in}(\boldsymbol{\lambda}, \hat{h}_{\boldsymbol{\lambda}}(x), x, y)\, \hat{a}_{\boldsymbol{\lambda}}(x)$
return $g_{Exp} + g_{Imp}, \hat{h}_{\boldsymbol{\lambda}}, \hat{a}_{\boldsymbol{\lambda}}$

Lemma 5.1 (Time-Smoothed Hypergradient Estimator). *Define $\mathcal{F}_t = 0\ \forall t < 0$. Let $\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})$ denote the stochastic hypergradient estimate from Algorithm 2 for each round $i = 0, \dots, w-1$, using a mini-batch $\mathcal{B}_{t-i} \sim \mathbb{P}_{t-i}$, evaluated at $\boldsymbol{\lambda}_{t-i}$. The time-smoothed stochastic estimator is constructed:*

$$\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) := \frac{1}{w}\sum_{i=0}^{w-1} \widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}). \tag{5}$$

and the bias and variance of this estimator satisfies for $\mathcal{Z}_{t,w} = \prod_{i=0}^{w-1} \Omega_{t-i}$

$$\left\| \mathbb{E}_{\mathcal{Z}_{t,w}}\left[\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) \mid \boldsymbol{\lambda}_t\right] - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\| \le \frac{1}{w}\sum_{i=0}^{w-1} \|b_{t-i}(\boldsymbol{\lambda}_{t-i})\|, \quad \textit{and}$$

$$Var_{\mathcal{Z}_{t,w}}\left[\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) \mid \boldsymbol{\lambda}_t\right] = \frac{1}{w^2}\sum_{i=0}^{w-1} Var_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) \mid \boldsymbol{\lambda}_{t-i}\right] \le \frac{\sigma^2_{\mathcal{F}_t}}{w}, \tag{6}$$

where $b_{t-i}(\boldsymbol{\lambda}_{t-i}) := \mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) \mid \boldsymbol{\lambda}_{t-i}\right] - \nabla\mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})$ is the bias and $\sigma^2_{\mathcal{F}_t}$ is the variance of individual stochastic hypergradient estimates at time $t-i$, detailed in the Appendix.

Proof. Expanding via linearity and noting that bias is $\frac{1}{w}\sum_{i=0}^{w-1}\left(\mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) \mid \boldsymbol{\lambda}_{t-i}\right] - \nabla\mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})\right) = \frac{1}{w}\sum_{i=0}^{w-1} b_{t-i}(\boldsymbol{\lambda}_{t-i})$, where $b_{t-i}(\boldsymbol{\lambda}_{t-i}) = \mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) \mid \boldsymbol{\lambda}_{t-i}\right] - \nabla\mathcal{F}_{t-i}(\boldsymbol{\lambda}_t)$, bounded by $\frac{1}{w}\sum_{i=0}^{w-1}\|b_{t-i}(\boldsymbol{\lambda}_{t-i})\|$. The variance is $\mathrm{Var}_{\mathcal{Z}_{t,w}}\left[\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) \mid \boldsymbol{\lambda}_t\right] = \frac{1}{w^2}\sum_{i=0}^{w-1}\mathrm{Var}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) \mid \boldsymbol{\lambda}_{t-i}\right] \le \frac{\sigma^2_{\mathcal{F}_t}}{w}$. □

Lemma 5.1 establishes that our time-smoothed hypergradient estimator, by averaging w stochastic estimates, achieves a bias bounded by the average of individual biases relative to the true hypergradients over a window of length w, while reducing variance by a factor of w. The variance decomposition shows the total variance is the scaled sum of individual estimate variances, each bounded by $\sigma^2_{\mathcal{F}_t}$. Consistent with the oracle setting, the window size improves the stability of SmoothFBO with hypergradient estimation (Algorithm 3) in non-stationary environments.

Algorithm 3 SmoothFBO (Smooth Functional Bilevel Optimization)

Require: Step $\alpha > 0$, window size $w \ge 1$, data distribution $\mathbb{P}_t$, hypergradient estimator **FuncGrad**$(\boldsymbol{\lambda}, h, a, \mathbb{P}_t)$, initial $\boldsymbol{\lambda}_1$ and models $h_{\boldsymbol{\lambda}_1}, a_{\boldsymbol{\lambda}_1}$
for $t = 1, \dots, T$ **do**
  # Query stochastic hypergradient estimate
  $\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t), h_{\boldsymbol{\lambda}_{t+1}}, a_{\boldsymbol{\lambda}_{t+1}} \leftarrow$ **FuncGrad**$(\boldsymbol{\lambda}_t, \dots)$
  # Compute time-smoothed stochastic estimator
  $\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) \leftarrow \frac{1}{w}\sum_{i=0}^{w-1}\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})$
  # Update outer parameter
  $\boldsymbol{\lambda}_{t+1} \leftarrow \boldsymbol{\lambda}_t - \alpha\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$
end for

Lemma 5.2 (Expected Squared Error of Time-Smoothed Hypergradient Estimator). *Let $\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$ denote the time-smoothed hypergradient estimator defined in 5.1, where $\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})$ is the stochastic hypergradient estimate from Algorithm 2 at time $t-i$. The expected error is bounded by*

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right] \le C_1\frac{T\sigma^2_{\mathcal{F}_t}}{w} + C_2\sum_{t=1}^{T}\epsilon^2_{in,t} + C_3\sum_{t=1}^{T}\epsilon^2_{adj,t},$$

where C_1, C_2, and C_3 are constants defined in the Appendix, $\epsilon_{in,t}$ and $\epsilon_{adj,t}$ represent the inner and adjoint approximation errors, respectively, and w is the window size. A proof is in the Appendix.

This bound on the expected squared error of $\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$ ensures that the time-smoothed hypergradient estimator remains accurate under sublinear approximation errors $\epsilon_{\mathrm{in},t}$ and $\epsilon_{\mathrm{adj},t}$, enabling us to leverage these properties in the subsequent analysis to establish sublinear $\mathrm{BLR}_w(T)$ in the convergence of Algorithm 3. Next, we introduce an additional assumption required for our regret analysis.

1. **Approximate Optimality with Sublinear Errors**: The inner optimization and adjoint problems

have sublinear approximation errors $\epsilon_{\text{in},t}$ and $\epsilon_{\text{adj},t}$ across time, satisfying:

$$\sum_{t=1}^{T} \epsilon_{\text{in},t}^2 = o(T) \quad \text{and} \quad \sum_{t=1}^{T} \epsilon_{\text{adj},t}^2 = o(T).$$

Theorem 5.3. *Under the assumptions of Section A.4, the bilevel local regret of Algorithm 3, using the time-smoothed hypergradient estimator $\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$, achieves an upper bound with step size $\alpha = \frac{4}{5L}$:*

$$\begin{aligned} BLR_w(T) &= \sum_{t=1}^{T} \mathbb{E}_{\mathcal{Z}_{t,w}} \left[\|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 \right] \\ &\leq C_4 \left(\frac{2TQ}{w} + V_{1,T} \right) + C_5 \mathbb{E}_{\mathcal{Z}_{t,w}} \Gamma_{t,w} \\ &\leq C_4 \left(\frac{2TQ}{w} + V_{1,T} \right) + C_5 \tilde{\Gamma}_{t,w}, \end{aligned} \tag{7}$$

where for shorthand we denote $\Gamma_{t,w} := \sum_{t=1}^{T} \left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2$, with an upper bound $\tilde{\Gamma}_{t,w} := C_1 \frac{T\sigma_{\mathcal{F}_t}^2}{w} + C_2 \sum_{t=1}^{T} \epsilon_{in,t}^2 + C_3 \sum_{t=1}^{T} \epsilon_{adj,t}^2$, L is the Lipschitz constant of $\nabla \mathcal{F}_t$, $\sigma_{\mathcal{F}_t}^2$ is the variance bound of the hypergradient estimates, Q bounds the outer objective, $V_{1,T} = o(T)$ quantifies the variation in the comparator sequence, and C_1, C_2, C_3 are constants from Lemma 5.2 associated with the approximation errors $\epsilon_{in,t}$ and $\epsilon_{adj,t}$. For window size $w = o(T)$, the regret $BLR_w(T)$ of Algorithm 3 is sublinear when $\sum_{t=1}^{T} \epsilon_{in,t}^2 = o(T)$ and $\sum_{t=1}^{T} \epsilon_{adj,t}^2 = o(T)$. The proof with constants C_4, C_5 is provided in the Appendix.

Having established an upper bound on the bilevel local regret $\text{BLR}_w(T)$ for Algorithm 3, which generalizes the oracle setting by incorporating the hypergradient estimation errors of $\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$, the following corollary examines how increasing the window parameter w mitigates the variance and error contributions to achieve improved convergence.

Corollary 5.4. *Increasing the window parameter w in Algorithm 3 reduces the variance and error contributions to the bilevel local regret, as given by:*

$$\begin{aligned} &\sum_{t=1}^{T} \mathbb{E}_{\mathcal{Z}_{t,w}} \left[\|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 \right] \\ &\leq \mathcal{O}\left(\frac{TQ}{w} + V_{1,T} + \frac{T\sigma_{\mathcal{F}_t}^2}{w} + \sum_{t=1}^{T} \epsilon_{in,t}^2 + \sum_{t=1}^{T} \epsilon_{adj,t}^2 \right), \end{aligned} \tag{8}$$

where a larger w diminishes the impact of the variance term $\frac{T\sigma_{\mathcal{F}_t}^2}{w}$, and a sublinear rate can be achieved providing $\sum_{t=1}^{T} \epsilon_{in,t}^2$ and $\sum_{t=1}^{T} \epsilon_{adj,t}^2$ remain sufficiently small, that is sublinear.

Lemma 5.5 (Reduction of Rates with Linear Inner Predictor)**.** *Consider the case where the inner predictor is linear, $h_{t,\boldsymbol{\lambda}}^{\star}(x) = \Phi(x)\, \theta_{t,\boldsymbol{\lambda}}^{\star}$, where $\theta_{t,\boldsymbol{\lambda}}^{\star}$ is the optimal parameter obtained from the inner optimization problem and $\Phi(x)$ is a linear mapping of x. In this setting, the online functional bilevel optimization problem (NS-FBO) reduces to the parametric special case, analyzed within Bohne et al. (2024). Under the assumptions of Section A.4, the bilevel local regret of Algorithm 3 then satisfies*

$$\begin{aligned} &\sum_{t=1}^{T} \mathbb{E}_{\mathcal{Z}_{t,w}} \left[\|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 \right] \\ &\leq \mathcal{O}\left(\frac{TQ}{w} + V_{1,T} + \frac{T\sigma_{\mathcal{F}_t}^2}{w} + H_{2,T} \right) \end{aligned} \tag{9}$$

where the comparator sequence of $H_{2,T}$ is the second-order path variation from the parametric OBO setting defined as $H_{2,T} := \sum_{t=1}^{T} \sup_{\boldsymbol{\lambda} \in \mathcal{X}} \left\| \theta_{t-1,\boldsymbol{\lambda}}^{*} - \theta_{t,\boldsymbol{\lambda}}^{*} \right\|^2$. For window size $w = o(T)$, the regret $BLR_w(T)$ of Algorithm 3 is sublinear under the standard conditions that comparator sequences satisfy regularity constraints $V_{1,T} = o(T)$, $H_{2,T} = o(T)$, see Tarzanagh et al. (2024); Lin et al. (2023). A proof for this lemma is found in the Appendix.

6 EXPERIMENTS

6.1 Importance–weight tuning for non-stationary regression

We begin with a controlled regression benchmark that instantiates the non-stationary functional setup in (1). The ground-truth data-generating process is a single-neuron sigmoid network:

$$f_{\text{true}}(x_t) = \sigma(W_t^{\top} x + b_t), \tag{10}$$

where the underlying parameters (W_t, b_t) evolve non-stationarily over time. Figure 2 (Fig. 2) summarizes the temporal drift of these parameters in the data-generating process (DGP) via a sinusoidal drift $\beta \sin(\omega t)$. Inputs are drawn i.i.d. as $\mathbf{X}_t \sim \mathcal{N}(0, \mathbf{I})$, and observed targets are

$$\mathbf{Y}_t = f_{\text{true}}(\mathbf{X}_t) + \boldsymbol{\zeta}_t, \tag{11}$$

with Gaussian noise $\boldsymbol{\zeta}_t \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. The inner predictor is a three-layer MLP $h \in \mathcal{H}$ with GeLU activations. The outer variable is a nonnegative scalar importance weight $\lambda \in \mathbb{R}_{\geq 0}$ that controls the emphasis on recent versus older minibatches. Given a window $W_t = \{t - w, \ldots, t - 1\}$ of length $w \geq 1$, we solve the

bilevel problem

$$\min_{\lambda\in\Lambda} \quad L_t^{out}(\lambda, h^\star_{t,\lambda}) := \tfrac{1}{B}\sum_{i=1}^{B} \left\|Y_{t,i} - h^\star_{t,\lambda}(X_i)\right\|_2^2, \quad (12)$$

$$\text{s.t.} \quad h^\star_{t,\lambda} \approx \arg\min_{h\in\mathcal{H}} \sum_{s\in W_t} \lambda_{t,s} \cdot \tfrac{1}{B}\sum_{i=1}^{B} \left\|Y_{s,i} - h(X_i)\right\|_2^2, \quad (13)$$

where nonnegative weights $\lambda_{t,s}$ are projected as $\lambda \leftarrow \max(\lambda, 0)$ after each update. The outer loss (12) uses a holdout minibatch at time t, while the inner loss (13) aggregates minibatches from the sliding window.

Baselines. We compare the following hypergradient estimators:

1. **Parametric** (unrolling truncated backprop through optimization),
2. **FBO** (functional bilevel optimization) without window-smoothing of Petrulionyte et al. (2024)
3. **SmoothFBO** (ours), which averages stochastic functional hypergradients over a window of length w before the outer update.

Further comparisons to offline and online parametric baselines such as *Approximate Implicit Differentiation (AID)* are deferred to the appendix.

Results. Figure 1 (Fig. 1) reports bilevel local regret (BLR_ω) versus rounds. Consistent with our main theorem, **SmoothFBO** achieves *sublinear regret*, as highlighted in the zoomed-in panel. Moreover, increasing the window size w (from 5 to 500) further reduces regret, reflecting as highlighted by our Theorem (5.3). In contrast, FBO and parametric methods incur substantially larger regret.

Figure 2 (Fig. 2) highlights the temporal evolution of (W_t, b_t) in the data-generating process, which induces the nonstationary regression challenge. Further analysis in the Appendix analyzes the loss and gradient norms across considered algorithms as well additional ablation studies on the experiment design.

6.2 Non-Stationary Model-based Reinforcement Learning

Non-stationary reinforcement learning environments present significant challenges as the underlying system dynamics evolve over time, rendering traditional RL approaches ineffective due to their stationarity assumptions. Our experiments demonstrate that the non-stationary functional bilevel optimization framework effectively captures these time-varying dynamics.

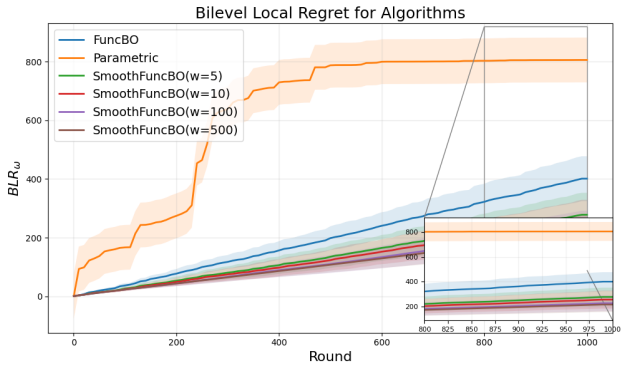

Figure 1: Bilevel local regret (BLR_ω) vs. rounds (Fig. 1). SmoothFBO achieves sublinear regret, consistent with our theorem. The zoomed-in component highlights the sublinear trend, while increasing the window w ($5 \to 500$) further reduces regret.

Problem formulation. In model-based RL with non-stationary environments, the time-varying Markov Decision Process (MDP) is approximated by a probabilistic model $q_{\boldsymbol{\lambda},t}$ with parameters $\boldsymbol{\lambda}$. The model predicts the next state $s_{\boldsymbol{\lambda},t}(x)$ and reward $r_{\boldsymbol{\lambda},t}(x)$, given a pair $x := (s, a)$ where s is the current environment state and a is the agent's action.

A second model approximates the action-value function $h_t(x)$ that computes the expected cumulative reward given the current state-action pair at time t. Traditionally, the action-value function is learned using the current MDP model, while the MDP model is learned independently using Maximum Likelihood Estimation (MLE) (Sutton, 1991). However, in recent work, Nikishin et al. (2022) showed that casting model-based RL as a bilevel problem can result in better performance and tolerance to model-misspecification (see B).

In our online bilevel formulation, the inner-level problem at time t involves learning the optimal action-value function $h^\star_{t,\boldsymbol{\lambda}}$ with the current MDP model $q_{\boldsymbol{\lambda},t}$ by minimizing the Bellman error. The inner-level objective can be expressed as an expectation of a point-wise loss f with samples $(x, r', s') \sim \mathbb{Q}_t$, derived from the agent-environment interaction at time t:

$$h^\star_{t,\boldsymbol{\lambda}} = \arg\min_{h\in\mathcal{H}} \mathbb{E}_{\mathbb{Q}_t}\left[f(h(x), r_{\boldsymbol{\lambda},t}(x), s_{\boldsymbol{\lambda},t}(x))\right]. \quad (14)$$

Here, the future state and reward (r', s') are replaced by the time-varying MDP model predictions $r_{\boldsymbol{\lambda},t}(x)$ and $s_{\boldsymbol{\lambda},t}(x)$. In practice, samples from $\mathbb{Q}_t$ are obtained using a replay buffer that adapts to the changing environment dynamics. The buffer accumulates data by interacting with the environment at time t. The non-stationarity in our environment is modeled by shifting the pole angle reward zones with time, which fundamentally

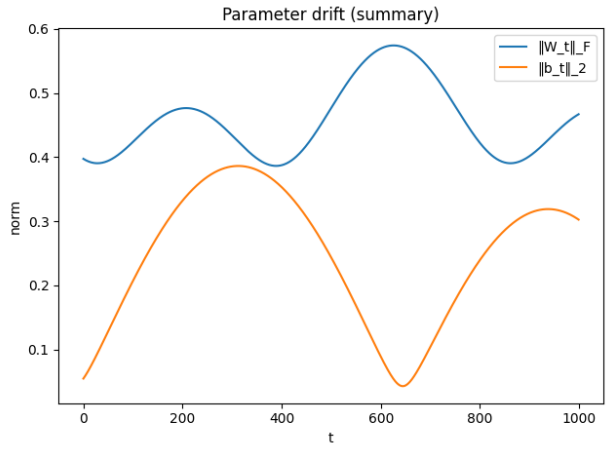

Figure 2: Parameter drift in the underlying DGP (Fig. 2). The weights (W_t, b_t) evolve nonstationarily, driving the outer-loop adaptation challenge.

alters the system's dynamics at each time step t. This creates a sequence of time-varying MDPs that the agent must continuously adapt to, the exact setup is further detailed in B.

The point-wise loss function f represents the error between the action-value function prediction and the expected cumulative reward given the current state-action pair:

$$f(v, r', s') := \frac{1}{2} \left\| v - r' - \gamma \log \sum_{a'} e^{\bar{h}_t(s', a')} \right\|^2,$$

with $\bar{h}_t$ a lagged version of h_t and γ a discount factor.

The time-varying MDP model is learned implicitly using the optimal function $h^\star_{t,\boldsymbol{\lambda}}$, by minimizing the Bellman error w.r.t. $\boldsymbol{\lambda}$ at each time step t:

$$\min_{\boldsymbol{\lambda} \in \Lambda} \mathbb{E}\left[f(h^\star_{t,\boldsymbol{\lambda}}(x), r', s') \right]. \tag{15}$$

Equations 15 and 14 define a non-stationary bilevel problem as in the general framework of equation 1, where at each time step t, we have data distribution $\mathbb{Q}_t = \mathbb{P}_t$, $y = (r', s')$, and the point-wise losses ℓ_{in} and ℓ_{out} are given by: $\ell_{in}(\boldsymbol{\lambda}, v, x, y) = f(v, r_{\boldsymbol{\lambda},t}(x), s_{\boldsymbol{\lambda},t}(x))$ and $\ell_{out}(\boldsymbol{\lambda}, v, x, y) = f(v, r', s')$. Therefore, we can directly apply our SmoothFBO Algorithm 3 to learn both the time-varying MDP model $q_{\boldsymbol{\lambda},t}$ and the optimal action-value function $h^\star_{t,\boldsymbol{\lambda}}$.

Experimental details. We evaluate the proposed *SmoothFBO* algorithm against three baselines and their time-smoothed variants:

1. **Maximum Likelihood Estimation (MLE):** The standard approach that updates the world model by direct likelihood maximization (Sutton, 1991).
2. **Optimal Model Design (OMD):** A parametric bilevel method for RL following implicit differentiation (Nikishin et al., 2022).
3. **Iterative Differentiation (ITD):** Differentiating through the inner optimizer (Lorraine et al., 2019).
4. **Functional Bilevel Optimization (FBO):** The functional approach of Petrulionyte et al. (2024) without temporal smoothing.
5. **SmoothFBO (ours):** An extension of FBO with time-smoothed hypergradient estimation.

To create a challenging non-stationary testbed, we implemented a modified CartPole environment (Brockman et al., 2016) where the reward structure drifts gradually over time. In the stationary environment, the agent is rewarded for maintaining the pole angle within a fixed optimal interval. In our non-stationary variant, this interval interpolates smoothly between two different regions, requiring the agent to adapt both its world model and policy continually. Figure 3 shows the evolving pole angle target region during training. This design makes adaptation essential: the change is large enough to invalidate static models, but gradual enough that tracking is feasible.

Our evaluation protocol consists of two phases: (1) hyperparameter tuning via grid search in the stationary environment, and (2) comprehensive evaluation of the best configurations across multiple random seeds in both stationary and non-stationary settings. All results are averaged across 5–10 seeds, with shaded regions in the plots indicating variability. Implementation and hardware details are deferred to §B.

6.3 Results and Analysis

Evaluation. After tuning in the stationary setting, we evaluate each algorithm's best configuration in both stationary and non-stationary environments. Performance is measured by cumulative episode reward. Figure 4 summarizes the results.

On the left, we observe that in the stationary case both FBO and *SmoothFBO* perform competitively; however, when the reward structure drifts, *SmoothFBO* is significantly more robust and maintains higher cumulative rewards. The right panel compares all baselines: *SmoothFBO* matches or exceeds their performance, highlighting its advantage in adapting to non-stationary dynamics.

We additionally compare against parametric unrolling/ITD (Baydin et al., 2017) with and without

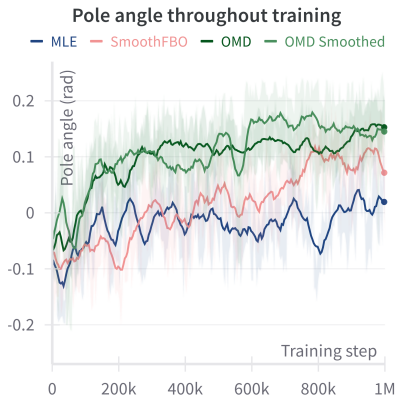

Figure 3: The changing pole angle throughout training. In the non-stationary CartPole experiment, the target pole angle shifts gradually over training steps, forcing the agent to adapt its learned dynamics and policy.

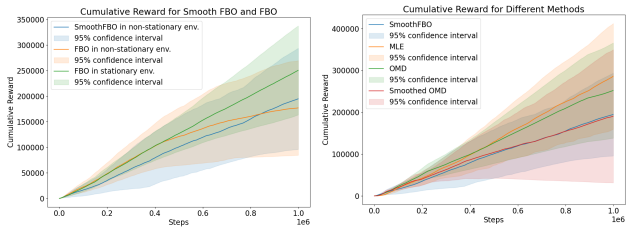

Figure 4: Cumulative reward for the non-stationary CartPole evaluation environment over 1 million environment steps. Each curve represents the mean cumulative episode reward across 10 random seeds, with shaded regions indicating 95% confidence intervals. **Left**: The FBO method in stationary and non-stationary environments compared to *SmoothFBO*. **Right**: Comparison with baseline methods where *SmoothFBO* matches their performance in adapting to the non-stationary dynamics with an averaging window of 100.

time-smoothing, and report the final episode reward on the evaluation environment in Table 6.3. In our non-stationary CartPole setup, while ITD can occasionally achieve partial convergence (mean final reward 56.87, max 336.05), it remains much less stable across seeds compared to *SmoothFBO*. While performing a grid search on ITD, we obtained notoriously unstable hypergradients with Adam, the optimizer used for all other methods, and had to switch to SGD. Even then, ITD remains less stable than the other approaches.

These findings emphasize two key points: (i) temporal smoothing in *SmoothFBO* improves stability under drift, and (ii) robustness to non-stationarity does not compromise efficiency in the stationary setting.

Method	Max	Mean	Min
Smoothed ITD	23.40	9.56	4.20
ITD	336.05	56.87	4.20
Smoothed OMD	147.75	31.55	5.00
SmoothFBO	**500.00**	**232.19**	**5.05**

Table 1: Final episode reward on the evaluation environment in the non-stationary CartPole experiment, over 10 seeds, for additional parametric bilevel baselines and representative implicit baselines. ITD/unrolling exhibits high variability and lower reliability than *SmoothFBO*.

7 CONCLUSION

This work presents a non-stationary functional bilevel optimization framework *SmoothFBO*. This method extends functional bilevel optimization to non-stationary environments through time-smoothing techniques that reduce variance in the outer loop. This enables more stable convergence with proven sublinear regret bounds. Despite these advances, achieving optimal performance in dynamic reinforcement learning environments remains challenging, as reinforcement learning models remain sensitive to initialization and prone to catastrophic forgetting. Nevertheless, experimental results demonstrate SmoothFBO's practical effectiveness in adapting to changing dynamics where standard FBO methods struggle. Our work opens the door to future research in new variance reduction strategies for bilevel algorithms in non-stationary environments.

Acknowledgments

This work was supported by the ERC grant number 101087696 (APHELAIA project) and by ANR 3IA MIAI@Grenoble Alpes (ANR-19-P3IA-0003) and the ANR project BONSAI (grant ANR-23-CE23-0012-01).

References

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. *International Conference on Machine Learning (ICML)*, 2020.

Michael Arbel and Julien Mairal. Amortized implicit differentiation for stochastic bilevel optimization. *International Conference on Learning Representations (ICLR)*, 2022.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint 1907.02893*, 2019.

Atılım Günes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind.

Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research (JMLR)*, 18(153):1–43, 2017.

Kristin P. Bennett, Jing Hu, Xiaoyun Ji, Gautam Kunapuli, and Jong-Shi Pang. Model selection via bilevel optimization. *IEEE International Joint Conference on Neural Network Proceedings*, 2006.

Luca Bertinetto, João F. Henriques, Philip H.S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *International Conference on Learning Representations (ICLR)*, 2019.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.

Jason Bohne, David S Rosenberg, Gary Kazantsev, and Pawel Polak. Online nonconvex bilevel optimization with bregman divergences. In *OPT 2024: Optimization for Machine Learning*, 2024.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint 1606.01540*, 2016.

Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Conference on Learning Theory (COLT)*, 2012.

Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated Machine Learning: Methods, Systems, Challenges*, pages 3–33. Springer International Publishing, 2019.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. *International Conference on Machine Learning (ICML)*, 2017.

Elad Hazan, Karan Singh, and Cyril Zhang. Efficient regret minimization in non-convex games. *Proceedings of Machine Learning Research (PMLR)*, 2017.

Gernot Holler, Karl Kunisch, and Richard C. Barnard. A bilevel approach for parameter learning in inverse problems. *Inverse Problems*, 34(11):115012, 2018.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023.

Sen Lin, Daouda Sow, Kaiyi Ji, Yingbin Liang, and Ness Shroff. Non-convex bilevel optimization with time-varying objective functions. *Advances in Neural Information Processing Systems*, 36:29692–29717, 2023.

Risheng Liu, Xuan Liu, Shangzhi Zeng, Jin Zhang, and Yixuan Zhang. Value-function-based sequential minimization for bi-level optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 45:15930–15948, 2021.

Jonathan Lorraine, Paul Vicol, and David Kristjanson Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(4):791–804, 2012.

Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. Auxiliary learning by implicit differentiation. *International Conference on Learning Representations (ICLR)*, 2021.

Evgenii Nikishin, Romina Abachi, Rishabh Agarwal, and Pierre-Luc Bacon. Control-oriented model-based reinforcement learning with implicit differentiation. *AAAI Conference on Artificial Intelligence*, 2022.

Sindhu Padakandla, Prabuchandran KJ, and Shalabh Bhatnagar. Reinforcement learning algorithm for non-stationary environments. *Applied Intelligence*, 50(11):3590–3606, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Ieva Petrulionyte, Julien Mairal, and Michael Arbel. Functional bilevel optimization for machine learning. *arXiv preprint arXiv:2403.20233*, 2024.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Allahkaram Shafiei and Jakub Marecek. Time-varying multi-objective optimization: Tradeoff regret bounds.

In Yingqian Zhang, Milan Hladik, and Hossein Moosaei, editors, *Learning and Intelligent Optimization - 19th International Conference, LION 19, Prague, Czech Republic, June 15–19, 2025, Proceedings, Part I*, volume 15744, pages 253–264, 2025.

Allahkaram Shafiei, Vyacheslav Kungurtsev, and Jakub Marecek. Trilevel and multilevel optimization using monotone operator theory. *Mathematical Methods of Operations Research*, 99(1):77–114, 2024.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Davoud Ataee Tarzanagh, Parvin Nazari, Bojian Hou, Li Shen, and Laura Balzano. Online bilevel optimization: Regret analysis of online alternating gradient methods. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning (ICML)*, 2016.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**] *(NS-FBO is formalized in the Introduction; SmoothFBO is defined with Algorithms 1-3 and supporting definitions/lemmas in Secs.2-4.)*
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**] *(Property and complexity details are provided with Algorithms 1–3*
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [*Yes, the full source code will be released upon acceptance*]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**] *(Assumptions for the oracle setting in Appendix A.1 and for estimated hypergradients in Appendix A.4 are explicitly listed.)*
   (b) Complete proofs of all theoretical results. [**Yes**] *(Proofs and supporting lemmas are provided for regret/convergence results, including Theorem 3.4/Corollary 3.5 and Appendix A.1–A.7.)*
   (c) Clear explanations of any assumptions. [**Yes**] *(Assumptions are explained before the main results, with detailed context in Appendix A.)*

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**] *(Data is open-source and sufficient details are provided to reproduce.)*
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**] *(Appendix B provides grid-search ranges, buffer sizes, training steps, and method configuration.)*
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**] *(Figures report mean across seeds with confidence intervals or standard errors for regression and RL.)*
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**] *(Experiments report using 24GB NVIDIA RTX A5000 GPUs and ∼6 hours per configuration.)*

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator if your work uses existing assets. [**Yes**] *(References include OpenAI Gym, PyTorch, and JAX used in experiments.)*
   (b) The license information of the assets, if applicable. [**Yes**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**No**]
   (d) Information about consent from data providers/curators. [**N/A**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**N/A**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [**N/A**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**N/A**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**N/A**]

A CONVERGENCE ANALYSIS

A.1 Assumptions for Algorithm 1

For the theoretical convergence analysis of Algorithm 1 in the stochastic bilevel optimization problem (1), we impose the following assumptions:

1. **Differentiability**: For $t = 1, \ldots, T$, the objective $\mathcal{F}_t(\boldsymbol{\lambda}) = \mathbb{E}_{\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}(x), x, y)\right]$ is continuously differentiable in $\boldsymbol{\lambda} \in \Lambda$, with hypergradient $\nabla\mathcal{F}_t(\boldsymbol{\lambda})$ being L-Lipschitz.
2. **Bounded Variance**: The hypergradient estimates $\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t)$ from oracle $\mathcal{O}(\boldsymbol{\lambda}_t)$ satisfy $\mathbb{E}_{\Omega_t}\left[\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t)\right] = \nabla\mathcal{F}_t(\boldsymbol{\lambda}_t)$ and $\mathrm{Var}_{\Omega_t}\left[\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t)\right] \le \sigma_f^2$, with $\Omega_t = \mathbb{P}_t \times \mathbb{Q}_t$.
3. **Bounded Objective**: The outer objective satisfies $|\mathcal{F}_t(\boldsymbol{\lambda})| \le Q$ for all $\boldsymbol{\lambda} \in \Lambda$ and $t \ge 1$.
4. **Gradual Non-stationarity**: The outer objective variation is bounded by:

$$V_{1,T} = \sum_{t=1}^{T} \sup_{\boldsymbol{\lambda}\in\Lambda} |\mathcal{F}_{t+1}(\boldsymbol{\lambda}) - \mathcal{F}_t(\boldsymbol{\lambda})| = o(T).$$

A.2 Convergence Analysis for Algorithm 1

First we introduce a necessary lemma characterizing the expected differences in the time-smoothed cumulative function evaluations

Lemma A.1. *Suppose the outer objective $\mathcal{F}_t(\boldsymbol{\lambda}) = \mathbb{E}_{(x,y)\sim\mathbb{P}_t}[\ell_{out}(\boldsymbol{\lambda})]$ is bounded such that $|\mathcal{F}_t(\boldsymbol{\lambda})| \le Q$ for all $\boldsymbol{\lambda} \in \Lambda$ and $t \ge 1$. If Algorithm 1 (**SmoothFBO**) with window size $w \ge 1$ generates the sequence $\{\boldsymbol{\lambda}_t\}_{t=1}^T$, then:*

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})\right)\right] \le \frac{2TQ}{w} + V_{1,T}, \tag{16}$$

where $\mathcal{Z}_{t,w} = \prod_{i=0}^{w-1}\Omega_{t-i}$, and $\mathcal{F}_{t,w}(\boldsymbol{\lambda}) = \frac{1}{w}\sum_{i=0}^{w-1}\mathcal{F}_{t-i}(\boldsymbol{\lambda})$, $\mathcal{F}_t(\boldsymbol{\lambda}) = 0$ for $t < 0$.

Proof. To simplify notation, we define the outer objective as $\mathcal{F}_t(\boldsymbol{\lambda}) = \mathbb{E}_{(x,y)\sim\mathbb{P}_t}[\ell_{out}(\boldsymbol{\lambda})]$, where $\ell_{out}(\boldsymbol{\lambda})$ is the pointwise outer objective at time t, abstracting the inner predictor and data dependencies compared to the full form $\mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}(x), x, y)\right]$. Then:

$$\begin{aligned}\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})\right) &= \sum_{t=1}^{T}\frac{1}{w}\sum_{i=0}^{w-1}\left(\mathcal{F}_{t-i}(\boldsymbol{\lambda}_t) - \mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t+1})\right)\\ &= \sum_{t=1}^{T}\frac{1}{w}\sum_{i=0}^{w-1}\mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}_t) - \ell_{out}(\boldsymbol{\lambda}_{t+1})\right].\end{aligned} \tag{17}$$

Taking expectation over $\mathcal{Z}_{t,w}$:

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})\right)\right] = \mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\frac{1}{w}\sum_{i=0}^{w-1}\mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}_t) - \ell_{out}(\boldsymbol{\lambda}_{t+1})\right]\right]. \tag{18}$$

Rewrite this as:

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\frac{1}{w}\sum_{i=0}^{w-1}\mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}_t) - \ell_{out}(\boldsymbol{\lambda}_{t+1})\right]\right] \tag{19}$$

$$+\,\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\frac{1}{w}\sum_{i=0}^{w-1}\mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}_{t+1}) - \ell_{out}(\boldsymbol{\lambda}_{t+1})\right]\right]. \tag{20}$$

For (19), apply linearity of expectation:

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\frac{1}{w}\sum_{i=0}^{w-1}\mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}_t)-\ell_{out}(\boldsymbol{\lambda}_{t+1})\right]\right]$$
$$=\frac{1}{w}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\mathcal{F}_{t+1-w}(\boldsymbol{\lambda}_{t+1-w})-\mathcal{F}_{t+1}(\boldsymbol{\lambda}_{t+1})\right]\leq\frac{2Q}{w}, \tag{21}$$

since $|\mathcal{F}_t(\boldsymbol{\lambda})|\leq Q$. For (20):

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\frac{1}{w}\sum_{i=0}^{w-1}\mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}_{t+1})-\ell_{out}(\boldsymbol{\lambda}_{t+1})\right]\right]$$
$$\leq\sum_{t=1}^{T}\sup_{\boldsymbol{\lambda}\in\Lambda}\left[\mathcal{F}_{t+1}(\boldsymbol{\lambda})-\mathcal{F}_t(\boldsymbol{\lambda})\right]=V_{1,T}. \tag{22}$$

Combining (21) and (22):

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)-\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})\right)\right]\leq\frac{2TQ}{w}+V_{1,T}.$$

□

Our next theorem provides an upper bound on the bilevel local regret of Algorithm 1 and outlines the conditions required to achieve a sublinear rate.

Theorem A.2. *Under the assumptions of section A.1, the bilevel local regret of Algorithm 1, leveraging a hypergradient oracle $\mathcal{O}(\boldsymbol{\lambda})$, can achieve an upper bound, with step size $\alpha=\frac{1}{L}$ of:*

$$BLR_w(T)=\sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}[\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2]\leq 2L\left(\frac{2TQ}{w}+V_{1,T}+\frac{T\sigma_f^2}{2Lw}\right),$$

where L is the Lipschitz constant of $\nabla\mathcal{F}_t$, expectation is computed with respect to $\mathcal{Z}_{t,w}=\prod_{i=0}^{w-1}\Omega_{t-i}$, σ_f^2 is the variance bound of the hypergradient estimates, Q bounds the outer function, and $V_{1,T}=o(T)$ is the variation in the outer objectives.

A.2.1 Proof of A.2

Proof. Under standard smoothness assumptions one can show $\mathcal{F}_{t,w}$ is L-smooth:

$$\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})-\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\leq\langle\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t),\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\rangle+\frac{L}{2}\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\|^2. \tag{23}$$

Substitute $\boldsymbol{\lambda}_{t+1}=\boldsymbol{\lambda}_t-\alpha\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$:

$$\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})-\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\leq-\alpha\langle\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t),\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\rangle+\frac{L}{2}\alpha^2\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\|^2. \tag{24}$$

Taking expectation conditioned on $\boldsymbol{\lambda}_t$ and using the unbiasedness and variance bound from Lemma 5.1, we obtain:

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})-\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\mid\boldsymbol{\lambda}_t\right]\leq-\alpha\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\langle\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t),\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\rangle\right] \tag{25}$$
$$+\frac{L\alpha^2}{2}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\|^2\right] \tag{26}$$
$$=-\alpha\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 \tag{27}$$
$$+\frac{L\alpha^2}{2}\left(\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2+\frac{\sigma_f^2}{w}\right). \tag{28}$$

Simplify:

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1}) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \mid \boldsymbol{\lambda}_t\right] \leq -\alpha\left(1 - \frac{L\alpha}{2}\right)\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 + \frac{L\alpha^2}{2}\frac{\sigma_f^2}{w}. \tag{29}$$

Telescope over $t = 1$ to T:

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})\right)\right] \geq \alpha\left(1 - \frac{L\alpha}{2}\right)\sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2\right] - \frac{L\alpha^2}{2}\frac{T\sigma_f^2}{w}. \tag{30}$$

Choose the fixed step size $\alpha = \frac{1}{L}$, so $1 - \frac{L\alpha}{2} = \frac{1}{2}$:

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})\right)\right] \geq \frac{\alpha}{2}\sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2\right] - \frac{T\sigma_f^2}{2Lw}. \tag{31}$$

Given the upper bound $\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})\right) \leq \frac{2TQ}{w} + V_{1,T}$, in expectation:

$$\frac{\alpha}{2}\sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2\right] \leq \frac{2TQ}{w} + V_{1,T} + \frac{T\sigma_f^2}{2Lw}.$$

With $\alpha = \frac{1}{L}$, multiply by $\frac{2}{\alpha} = 2L$:

$$\sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}[\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2] \leq 2L\left(\frac{2TQ}{w} + V_{1,T} + \frac{T\sigma_f^2}{2Lw}\right), \tag{32}$$

□

A.3 Functional Hypergradient

In functional implicit differentiation, the function `InnerOpt` (defined in Algorithm 4) optimizes inner model parameters for a given $\boldsymbol{\lambda}$, initialization θ_0, and data $\mathcal{D}_{in}$, using M gradient updates. It returns the inner model $\hat{h}_{\boldsymbol{\lambda}}$, usually a neural network, parameterized by parameters θ_M, approximating the inner-level solution. Similarly, `AdjointOpt` (defined in Algorithm 5) optimizes adjoint model parameters with K gradient updates, producing the approximate adjoint function $\hat{a}_{\boldsymbol{\lambda}}$. Other optimization procedures may also be used, especially when closed-form solutions are available, as exploited in the experiments in Section 6. Operations requiring differentiation can be implemented using standard optimization procedures with automatic differentiation packages like PyTorch (Paszke et al., 2019) or Jax (Bradbury et al., 2018).

Algorithm 4 `InnerOpt`

Require: outer variable $\boldsymbol{\lambda}$, inner model h parameterized by θ_0, dataset $\mathcal{D}_{in}$
  for $m = 0, \ldots, M-1$ **do**
    Sample batch $\mathcal{B}_{in}$ from $\mathcal{D}_{in}$
    $h_m \leftarrow$ inner model parameterized by θ_m
    $g_{in} \leftarrow \nabla_\theta[\hat{L}_{in}(\boldsymbol{\lambda}, h_m, \mathcal{B}_{in}) + R_{in}(\theta_m)]$
    $\theta_{m+1} \leftarrow$ Update θ_m using g_{in}
  end for
  $\hat{h}_{\boldsymbol{\lambda}} \leftarrow$ inner model parameterized by θ_{m+1}
  Return $\hat{h}_{\boldsymbol{\lambda}}$

Algorithm 5 `AdjointOpt`

Require: outer variable $\boldsymbol{\lambda}$, adjoint model a parameterized by ξ_0, inner model $\hat{h}_{\boldsymbol{\lambda}}$, dataset $\mathcal{D}$
  for $k = 0, \ldots, K-1$ **do**
    Sample batch $\mathcal{B}$ from $\mathcal{D}$
    $a_k \leftarrow$ inner model parameterized by ξ_k
    $g_{adj} \leftarrow \nabla_\xi[\hat{L}_{adj}(\boldsymbol{\lambda}, a_k, \hat{h}_{\boldsymbol{\lambda}}, \mathcal{B}) + R_{adj}(\xi_k)]$
    $\xi_{k+1} \leftarrow$ Update ξ_k using g_{adj}
  end for
  $\hat{a}_{\boldsymbol{\lambda}} \leftarrow$ adjoint model parameterized by ξ_{m+1}
  Return $\hat{a}_{\boldsymbol{\lambda}}$

Since we cannot perform first-order optimization techniques directly in function spaces, we assume that, in practice, h and a are models parameterized by some finite dimensional parameter vectors θ and ξ, rather than functions in L_2. As discussed in the theoretical analysis of the algorithm, we assume that these models map finite dimensional parameter vectors to a functions that are ϵ-close to the true predictions functions. Together with empirical objectives, commonly used regularization techniques $R_{in}(\theta)$ and $R_{adj}(\xi)$ may be introduced in inner and adjoint optimization subroutines, such as ridge penalty.

We approximate the hypergradient $\nabla\mathcal{F}_t$ after computing the approximate solutions $\hat{h}_{\boldsymbol{\lambda}}$ and $\hat{a}_{\boldsymbol{\lambda}}$. We decompose the gradient into two terms: g_{Exp}, an empirical approximation of $g_{\boldsymbol{\lambda}} := \partial_{\boldsymbol{\lambda}} L_t^{out}(\boldsymbol{\lambda}, h^\star_{\boldsymbol{\lambda}})$ representing the explicit dependence of L_t^{out} on the outer variable $\boldsymbol{\lambda}$, and g_{Imp}, an approximation to the implicit gradient term $B_{\boldsymbol{\lambda}} a^\star_{\boldsymbol{\lambda}}$. Both terms are obtained by replacing the expectations by empirical averages over batches $\mathcal{B}_{in}$ and $\mathcal{B}_{out}$, and using the approximations $\hat{h}_{\boldsymbol{\lambda}}$ and $\hat{a}_{\boldsymbol{\lambda}}$ instead of the exact solutions.

A.4 Assumptions for Algorithm 3 Convergence

Compared to the assumptions of A.1, we include two additional assumptions of Biased Gradient Estimators with Bounded Moments and Approximate Optimality under Sublinear Errors.

1. **Differentiability and Smoothness**: For $t = 1, \dots, T$, the point-wise inner and outer objectives $\ell_{in}(\boldsymbol{\lambda}, h(x), x, y)$ and $\ell_{out}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}(x), x, y)$ are continuously differentiable in $\boldsymbol{\lambda} \in \Lambda$ and $h(x) \in \mathcal{H}$ for all (x, y). Consequently, the outer objective $\mathcal{F}_t(\boldsymbol{\lambda}) = \mathbb{E}_{(x,y)\sim\mathbb{P}_t}\left[\ell_{out}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}(x), x, y)\right]$ and its time-smoothed version $\mathcal{F}_{t,w}(\boldsymbol{\lambda}) = \frac{1}{w}\sum_{i=0}^{w-1} \mathcal{F}_{t-i}(\boldsymbol{\lambda})$ for any $w > 0$ are continuously differentiable in $\boldsymbol{\lambda} \in \Lambda$, with true hypergradient $\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda})$ being L-Lipschitz. These and other technical assumptions, necessary to derive the functional hypergradient, are discussed in detail in Appendix D and E of Petrulionyte et al. (2024).

2. **Biased Gradient Estimators with Bounded Moments**: The hypergradient estimates $\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t)$, computed as:

$$\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t) = \frac{1}{|\mathcal{B}_{\text{out}}|}\sum_{(\tilde{x},\tilde{y})\in\mathcal{B}_{\text{out}}} \partial_{\boldsymbol{\lambda}}\ell_{out}\left(\boldsymbol{\lambda}_t, \hat{h}_{\boldsymbol{\lambda}_t}(\tilde{x}), \tilde{x}, \tilde{y}\right) + \frac{1}{|\mathcal{B}_{\text{in}}|}\sum_{(x,y)\in\mathcal{B}_{\text{in}}} \partial_{\boldsymbol{\lambda},v}\ell_{in}\left(\boldsymbol{\lambda}_t, \hat{h}_{\boldsymbol{\lambda}_t}(x), x, y\right)\hat{a}_{\boldsymbol{\lambda}_t}(x),$$

   where $\mathcal{B}_{\text{in}}$ and $\mathcal{B}_{\text{out}}$ are independent samples from $\mathbb{P}_t$ and $\mathbb{Q}_t$, respectively, are biased due to suboptimal solutions $\hat{h}_{\boldsymbol{\lambda}_t}$ and $\hat{a}_{\boldsymbol{\lambda}_t}$. The distributions $\mathbb{P}_t$ and $\mathbb{Q}_t$ have bounded second moments, and the point-wise losses ℓ_{in} and ℓ_{out} are differentiable and smooth, as per Petrulionyte et al. (2024), Appendix D.

3. **Bounded Objective**: The outer objective satisfies $|\mathcal{F}_t(\boldsymbol{\lambda})| \le Q$ for all $\boldsymbol{\lambda} \in \Lambda$ and $t \ge 1$.

4. **Gradual Non-stationarity**: The outer objective variation is bounded by:

$$V_{1,T} = \sum_{t=1}^{T} \sup_{\boldsymbol{\lambda}\in\Lambda} |\mathcal{F}_{t+1}(\boldsymbol{\lambda}) - \mathcal{F}_t(\boldsymbol{\lambda})| = o(T).$$

5. **Approximate Optimality with Sublinear Errors**: The inner optimization and adjoint problems have sublinear approximation errors $\epsilon_{\text{in},t}$ and $\epsilon_{\text{adj},t}$ across time, satisfying:

$$\sum_{t=1}^{T} \epsilon_{\text{in},t} = o(T) \quad \text{and} \quad \sum_{t=1}^{T} \epsilon_{\text{adj},t} = o(T).$$

A.5 Convergence Analysis of Algorithm 5.3

A.5.1 Preliminary Lemmas

We use the two following lemmas proven in Petrulionyte et al. (2024) for bias-variance decomposition.

Lemma A.3 (Lemma E.4 in Petrulionyte et al. (2024)). *Let the assumptions from Appendix A.4 hold $\forall t \in [1, T]$. Then we have the following bias from the stochastic hypergradient estimation of FuncGrad in Algorithm 2:*

$$\mathbb{E}_{\Omega_t}\left[\left\|\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_t(\boldsymbol{\lambda}_t)\right\|\right] \le c_1\epsilon_{in,t} + c_2\epsilon_{adj,t},$$

where c_1 and c_2 are constants defined in Equation 50 of Petrulionyte et al. (2024), and $\Omega_t = \mathbb{P}_t \times \mathbb{Q}_t$. Here, $\epsilon_{in,t}$ and $\epsilon_{adj,t}$ denote the inner and adjoint approximation errors.

Lemma A.4 (Lemma E.5 in Petrulionyte et al. (2024)). *Let the assumptions from Appendix A.4 hold* $\forall t \in [1, T]$. *Then the variance stochastic functional hypergradient estimation in Algorithm 2:*

$$\mathbb{E}_{\Omega_t}\left[\left\|\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t) - \mathbb{E}_{\Omega_t}\left[\widehat{\nabla\mathcal{F}}_t(\boldsymbol{\lambda}_t)\right]\right\|\right] \leq \sigma^2_{\mathcal{F}_t},$$

where $\sigma^2_{\mathcal{F}_t}$ *is a positive constant given by:*

$$\sigma^2_{\mathcal{F}_t} := \frac{2}{|\mathcal{B}_{out}|}\left(2c_3^2\mu^{-1}\epsilon_{in,t} + \sigma^2_{out}\right) + \frac{4B_2^2}{|\mathcal{B}_{in}|}\left(\mu^{-1}\epsilon_{adj,t} + 2\mu^{-3}c_4^2M^2\epsilon_{in,t} + \mu^{-2}B_3^2\right), \tag{33}$$

and c_3, c_4, B_2, B_3, M, μ, σ^2_{out} *are additional constants defined in Petrulionyte et al. (2024).*

Lemma A.5 (Expected Squared Error of Time-Smoothed Hypergradient Estimator). *Let* $\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$ *denote the time-smoothed hypergradient estimator defined as:*

$$\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) := \frac{1}{w}\sum_{i=0}^{w-1}\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}),$$

where $\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})$ *is the stochastic hypergradient estimate at time* $t-i$. *The expected error from the true hypergradient is then bounded by:*

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right] \leq C_1\frac{T\sigma^2_{\mathcal{F}_t}}{w} + C_2\sum_{t=1}^{T}\epsilon^2_{in,t} + C_3\sum_{t=1}^{T}\epsilon^2_{adj,t},$$

where C_1, C_2, *and* C_3 *are constants,* $\epsilon_{in,t}$ *and* $\epsilon_{adj,t}$ *are time-dependent inner and adjoint approximation errors, respectively,* $\sigma^2_{\mathcal{F}_t}$ *is the variance bound of the hypergradient estimates,* w *is the window size, and we denote* $\mathcal{Z}_{t,w} = \prod_{i=0}^{w-1}\Omega_{t-i}$.

Proof. The time-smoothed hypergradient estimator is defined as:

$$\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) := \frac{1}{w}\sum_{i=0}^{w-1}\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}).$$

Note the expansion:

$$\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right] = \sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\frac{1}{w}\sum_{i=0}^{w-1}\left(\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t-i})\right)\right\|^2\right].$$

Using the provided inequality:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\frac{1}{w}\sum_{i=0}^{w-1}\left(\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) - \nabla\mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})\right)\right\|^2\right] \\
&\leq 2\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\frac{1}{w}\sum_{i=0}^{w-1}\left(\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) - \mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})\right]\right)\right\|^2\right] \\
&+ 2\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\frac{1}{w}\sum_{i=0}^{w-1}\left(\mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})\right] - \nabla\mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})\right)\right\|^2\right].
\end{aligned}$$

For the variance, we apply Lemma A.4 to get:

$$\begin{aligned}
\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\frac{1}{w}\sum_{i=0}^{w-1}\left(\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) - \mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})\right]\right)\right\|^2\right] &\leq \frac{1}{w^2}\sum_{i=0}^{w-1}\mathbb{E}_{\Omega_{t-i}}\left[\left\|\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i}) - \mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})\right]\right\|^2\right] \\
&\leq \frac{\sigma^2_{\mathcal{F}_t}}{w}.
\end{aligned}$$

Thus,

$$\sum_{t=1}^{T} \frac{2\sigma_{\mathcal{F}_t}^2}{w} = \frac{2\sigma_{\mathcal{F}_t}^2 T}{w}.$$

For the bias, we apply Lemma A.3 to get:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\frac{1}{w}\sum_{i=0}^{w-1}\left(\mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})\right]-\nabla\mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})\right)\right\|^2\right] \\
&\leq \frac{1}{w}\sum_{i=0}^{w-1}\left\|\mathbb{E}_{\Omega_{t-i}}\left[\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})\right]-\nabla\mathcal{F}_{t-i}(\boldsymbol{\lambda}_{t-i})\right\|^2 \\
&\leq \frac{1}{w}\sum_{i=0}^{w-1}\left(2c_1^2\epsilon_{\text{in},t-i}^2+2c_2^2\epsilon_{\text{adj},t-i}^2\right).
\end{aligned}$$

Thus,

$$\sum_{t=1}^{T}\left(\frac{4c_1^2}{w}\sum_{i=0}^{w-1}\epsilon_{\text{in},t-i}^2+\frac{4c_2^2}{w}\sum_{i=0}^{w-1}\epsilon_{\text{adj},t-i}^2\right) \leq 4c_1^2\sum_{t=1}^{T}\epsilon_{\text{in},t}^2+4c_2^2\sum_{t=1}^{T}\epsilon_{\text{adj},t}^2.$$

Combining terms:

$$\begin{aligned}
\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)-\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right] &\leq \frac{2\sigma_{\mathcal{F}_t}^2 T}{w}+4c_1^2\sum_{t=1}^{T}\epsilon_{\text{in},t}^2+4c_2^2\sum_{t=1}^{T}\epsilon_{\text{adj},t}^2 \\
&= C_1\frac{T\sigma_{\mathcal{F}_t}^2}{w}+C_2\sum_{t=1}^{T}\epsilon_{\text{in},t}^2+C_3\sum_{t=1}^{T}\epsilon_{\text{adj},t}^2,
\end{aligned}$$

where $C_1 = 2$, $C_2 = 4c_1^2$, $C_3 = 4c_2^2$. □

A.5.2 Convergence Theorem

Our next theorem utilizes the aforementioned lemma to derive an upper bound on the bilevel local regret and the conditions required to achieve a sublinear rate.

Theorem A.6. *Under the assumptions of A.4, the bilevel local regret of Algorithm 3, using the time-smoothed hypergradient estimator $\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$, achieves an upper bound with step size $\alpha = \frac{4}{5L}$:*

$$\begin{aligned}
&BLR_w(T) = \sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right] \\
&\leq C_4\left(\frac{2TQ}{w}+V_{1,T}\right)+C_5\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)-\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right] \\
&\leq C_4\left(\frac{2TQ}{w}+V_{1,T}\right)+C_5\left(C_1\frac{T\sigma_{\mathcal{F}_t}^2}{w}+C_2\sum_{t=1}^{T}\epsilon_{in,t}^2+C_3\sum_{t=1}^{T}\epsilon_{adj,t}^2\right),
\end{aligned} \tag{34}$$

where L is the Lipschitz constant of $\nabla\mathcal{F}_t$, $\sigma_{\mathcal{F}_t}^2$ is the variance bound of the hypergradient estimates, Q bounds the outer objective, $V_{1,T} = o(T)$ quantifies the variation in the comparator sequence, and C_1, C_2, C_3 are constants from Lemma A.5 associated with the approximation errors $\epsilon_{in,t}$ and $\epsilon_{adj,t}$.

Proof. By L-smoothness of $\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)$:

$$\begin{aligned}
\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})-\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \leq \langle\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t),\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\rangle \\
+\frac{L}{2}\left\|\boldsymbol{\lambda}_{t+1}-\boldsymbol{\lambda}_t\right\|^2.
\end{aligned}$$

With the update rule $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t - \alpha\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$, where $\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) = \frac{1}{w}\sum_{i=0}^{w-1}\widehat{\nabla\mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})$:

$$\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1}) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \leq -\alpha\langle\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t), \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\rangle + \frac{L}{2}\alpha^2\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2.$$

Taking conditional expectations over $\mathcal{Z}_{t,w} = \prod_{i=0}^{w-1}\Omega_{t-i}$, given $\boldsymbol{\lambda}_t$, and applying Lemmas A.8 and A.9 with $\alpha = \frac{4}{5L}$:

$$\begin{aligned}&\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1}) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \mid \boldsymbol{\lambda}_t\right]\\ &\leq \frac{43}{25L}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2 \mid \boldsymbol{\lambda}_t\right]\\ &+ \frac{1}{50L}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2 \mid \boldsymbol{\lambda}_t\right].\end{aligned}$$

Summing over $t = 1$ to T and taking total expectations:

$$\begin{aligned}&\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1}) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right)\right]\\ &\leq \frac{43}{25L}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right]\\ &+ \frac{1}{50L}\sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right].\end{aligned}$$

By Lemma A.1, $\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left(\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) - \mathcal{F}_{t,w}(\boldsymbol{\lambda}_{t+1})\right)\right] \leq \frac{2TQ}{w} + V_{1,T}$, so:

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right] &\leq 50L\left(\frac{2TQ}{w} + V_{1,T}\right)\\ &+ 86\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right].\end{aligned}$$

By Lemma A.5:

$$\begin{aligned}&\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\sum_{t=1}^{T}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right]\\ &\leq C_1\frac{T\sigma^2_{\mathcal{F}_t}}{w} + C_2\sum_{t=1}^{T}\epsilon^2_{\text{in},t} + C_3\sum_{t=1}^{T}\epsilon^2_{\text{adj},t}.\end{aligned}$$

Thus:

$$\begin{aligned}&\text{BLR}_w(T) = \sum_{t=1}^{T}\mathbb{E}_{\mathcal{Z}_{t,w}}\left[\left\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2\right]\\ &\leq C_4\left(\frac{2TQ}{w} + V_{1,T}\right) + C_5\left(C_1\frac{T\sigma^2_{\mathcal{F}_t}}{w} + C_2\sum_{t=1}^{T}\epsilon^2_{\text{in},t} + C_3\sum_{t=1}^{T}\epsilon^2_{\text{adj},t}\right).\end{aligned}$$

where $C_4 = 50L, C_5 = 86$. □

A.6 Reduction of Rates in Linear Predictor Setting

Lemma A.7 (Reduction of Rates with Linear Inner Predictor)**.** *Consider the case where the inner predictor is linear, $h^\star_{t,\boldsymbol{\lambda}}(x) = \Phi(x)\,\theta^\star_{t,\boldsymbol{\lambda}}$, where $\theta^\star_{t,\boldsymbol{\lambda}}$ is the optimal parameter obtained from the inner optimization problem*

and $\Phi(x)$ is a linear mapping of x. In this setting, the online functional bilevel optimization problem (NS-FBO) reduces to the parametric special case, analyzed within Bohne et al. (2024). Under the assumptions of Section A.4, the bilevel local regret of Algorithm 3 then satisfies

$$\sum_{t=1}^{T} \mathbb{E}_{\mathcal{Z}_{t,w}} \left[\|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 \right] \leq \mathcal{O}\left(\frac{TQ}{w} + V_{1,T} + \frac{T\sigma^2_{\mathcal{F}_t}}{w} + H_{2,T} \right) \tag{35}$$

where the comparator sequence of $H_{2,T}$ is the second-order path variation from the parametric OBO setting defined as $H_{2,T} := \sum_{t=1}^{T} \sup_{\boldsymbol{\lambda} \in \mathcal{X}} \left\| \theta^*_{t-1,\boldsymbol{\lambda}} - \theta^*_{t,\boldsymbol{\lambda}} \right\|^2$ where we denote $\theta^*_{t,\boldsymbol{\lambda}} := \theta^*_t(\boldsymbol{\lambda})$. For window size $w = o(T)$, the regret $BLR_w(T)$ of Algorithm 3 is sublinear under the standard conditions that comparator sequences satisfy regularity $V_{1,T} = o(T)$, $H_{2,T} = o(T)$, see Tarzanagh et al. (2024); Lin et al. (2023).

Proof. For a linear inner predictor $h^\star_{t,\boldsymbol{\lambda}}(x)$ with linear mapping $\Phi(x)$, we can write

$$h^\star_{t,\boldsymbol{\lambda}}(x) = \Phi(x)\, \theta^\star_{t,\boldsymbol{\lambda}},$$

so that (NS-FBO) reduces to a parametric bilevel problem over θ. The outer objective becomes

$$\mathcal{F}_t(\boldsymbol{\lambda}) = L_t^{out}(\boldsymbol{\lambda}, h^\star_{t,\boldsymbol{\lambda}}) = L_t^{out}(\boldsymbol{\lambda}, \Phi(\cdot)\theta^\star_{t,\boldsymbol{\lambda}}).$$

In the parametric setting, the error between the stochastic hypergradient estimate $\widehat{\nabla \mathcal{F}}_{t-i}(\boldsymbol{\lambda}_{t-i})$ and the true hypergradient at each time step $t-i$ satisfies the bound from Theorem 5.6 of Lin et al. (2023), restated in our notation:

$$\left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2 \leq \mathcal{O}\left(\frac{\sigma^2_{\mathcal{F}_t}}{w} + \sup_{\boldsymbol{\lambda} \in \mathcal{X}} \left\| \theta^*_{t-1,\boldsymbol{\lambda}} - \theta^*_{t,\boldsymbol{\lambda}} \right\|^2 \right).$$

or cumulatively across $t = 1, \ldots, T$ rounds

$$\mathbb{E}_{\mathcal{Z}_{t,w}} \left[\sum_{t=1}^{T} \left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2 \right] \leq \mathcal{O}\left(\frac{T\sigma^2_{\mathcal{F}_t}}{w} + H_{2,T} \right).$$

Substituting this hypergradient error bound into the proof of Theorem A.6 gives

$$\sum_{t=1}^{T} \mathbb{E}_{\mathcal{Z}_{t,w}} \left[\|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 \right] \leq \mathcal{O}\Big(\frac{TQ}{w} + V_{1,T} + \frac{T\sigma^2_{\mathcal{F}_t}}{w} + H_{2,T} \Big), \tag{36}$$

which for $w = o(T)$ yields sublinear regret under the regularity conditions $V_{1,T} = o(T)$ and $H_{2,T} = o(T)$.

□

A.7 Additional Lemmas

Lemma A.8 (Approximate Gradient Norm Bound)**.** *Let $\widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$ be an approximate gradient and $\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)$ the true gradient of the loss $\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)$ at iterate $\boldsymbol{\lambda}_t$. We have*

$$-\left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2 \leq -\frac{1}{2} \|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 + \left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2.$$

Proof. Consider the norm of the true gradient:

$$\|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 \leq 2 \left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2 + 2 \left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2.$$

Rearrange:

$$-\frac{1}{2} \|\nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\|^2 \geq -\left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2 - \left\| \widetilde{\nabla \mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla \mathcal{F}_{t,w}(\boldsymbol{\lambda}_t) \right\|^2.$$

Add $\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2$ to both sides:

$$-\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2 \leq -\frac{1}{2}\left\|\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2 + \left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2.$$

□

Lemma A.9 (Generalized Projection Inequality)**.** *Let $\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$ be an approximate gradient and $\nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)$ the true gradient of the loss $\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)$ at iterate $\boldsymbol{\lambda}_t$. We have*

$$\left\langle \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t), \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right\rangle \leq \left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2 + \frac{1}{4}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2.$$

Proof. Consider the inner product:

$$\left\langle \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t), \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right\rangle.$$

By Young's inequality, for any $\eta > 0$:

$$\langle \mathbf{a}, \mathbf{b}\rangle \leq \frac{\eta}{2}\|\mathbf{a}\|^2 + \frac{1}{2\eta}\|\mathbf{b}\|^2.$$

Set $\mathbf{a} = \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)$, $\mathbf{b} = \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)$, and $\eta = 2$:

$$\left\langle \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t), \widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right\rangle \leq \left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t) - \nabla\mathcal{F}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2 + \frac{1}{4}\left\|\widetilde{\nabla\mathcal{F}}_{t,w}(\boldsymbol{\lambda}_t)\right\|^2.$$

□

B ADDITIONAL EXPERIMENT DETAILS

B.1 Non-stationary Regression

To further validate the mechanism underlying time–smoothing, we analyze how the smoothing window w affects hypergradient variance and bilevel local regret (BLR_ω). Figure 5 summarizes these effects: the left panel reports the (cumulative) BLR_ω and the right panel shows that larger w reduces the variance of the outer hypergradient. This confirms that temporal smoothing stabilizes the outer optimization, yielding smoother updates and lower BLR_ω. In agreement with Corollary 5.4, variance accumulation slows from near-linear to sublinear as w increases.

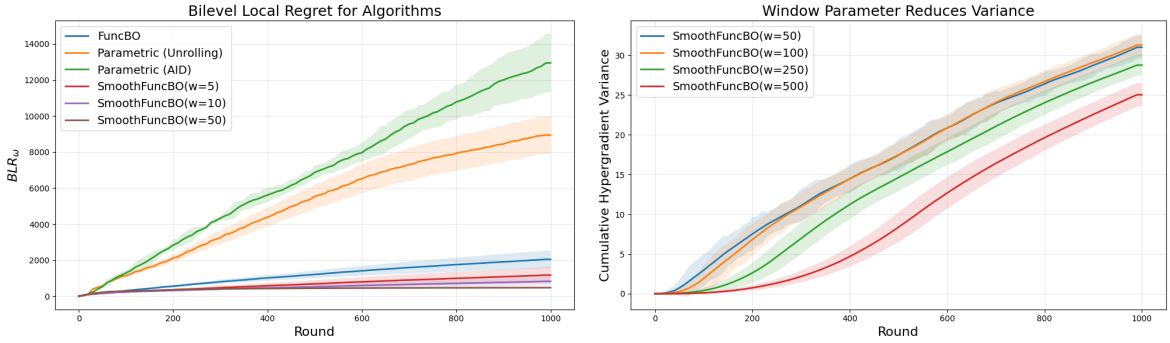

Figure 5: **Effect of smoothing on regret and hypergradient variance.** (Left) Cumulative bilevel local regret (BLR_ω) on the sinusoidal-drift task, showing sublinear regret for **SmoothFBO**; curves are averaged over seeds. (Right) Variance of the hypergradient as a function of the smoothing window w; increasing w reduces variance. These results demonstrate that temporal smoothing stabilizes bilevel optimization and improves bilevel local regret. Shaded regions indicate 95% confidence intervals across seeds.

To complement the sinusoidal drift analyzed in the main text (Fig. 2), we also examine a *discrete jump–based* non-stationarity where (W_t, b_t) undergo abrupt changes at fixed intervals. Figure 6 reports BLR_ω under jump–based drift and visualizes the parameter trajectory. **SmoothFBO** attains substantially lower cumulative regret than **FBO** and parametric baselines (AID, Unrolling), exhibiting sublinear regret.

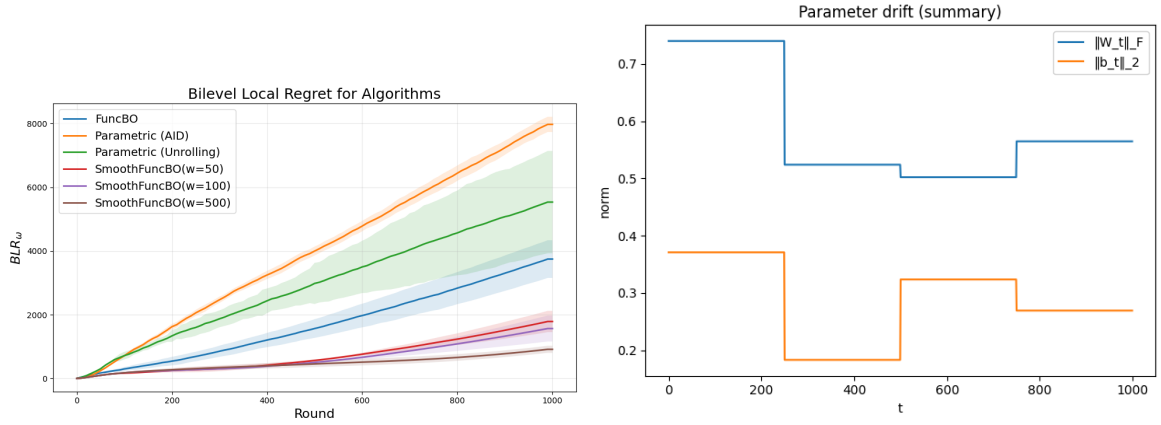

Figure 6: **Behavior under discrete jump–based non-stationarity.** (Left) Cumulative BLR_ω under abrupt parameter jumps. (Right) Evolution of (W_t, b_t) with discrete jumps that induce the drift. **SmoothFBO** recovers quickly after each jump and maintains lower cumulative regret.

Hyperparameter selection. All results are averaged over three random seeds, with shaded regions indicating standard error. Our method demonstrates robust performance across a broad range of hyperparameter configurations. Unless otherwise stated, reported results use an inner learning rate of 10^{-4}, an outer learning rate of 10^{-3}, a batch size of 32, and 5 inner steps. We observe consistent **SmoothFBO** performance across the following ranges:

- Inner learning rate: $\{10^{-2}, 10^{-3}, 10^{-4}\}$.
- Outer learning rate: $\{10^{-3}, 10^{-2}\}$.
- Batch size: $\{16, 32, 64, 128\}$.
- Number of inner steps: $\{5, 10\}$.

Training is performed for 1000 outer steps while varying the window parameter, which controls the time smoothing of hypergradients, over $\{1, 5, 10, 50, 100, 250, 500\}$. In the **SmoothFBO** implementation, a gradient buffer maintains recent hypergradients to enable this smoothing mechanism. Stable performance across these settings highlights the method's robustness to hyperparameter choice.

B.2 Reinforcement Learning

Non-stationary CartPole environment. As described in the main text, our non-stationary variant of CartPole modifies the reward interval associated with the pole angle. In the stationary environment, the reward is 1 when the pole angle lies in a fixed optimal interval and 0 otherwise. In the non-stationary setting, this interval shifts gradually throughout training. Formally, we interpolate linearly between two distinct pole-angle intervals:

$$(-0.2095,\ 0.06) \quad \longrightarrow \quad (-0.06,\ 0.2095).$$

This interpolation induces a continuous drift in the reward boundaries, requiring the agent to track the changing environment. The main text (Fig. 3) visualizes the pole angle drift for different agents.

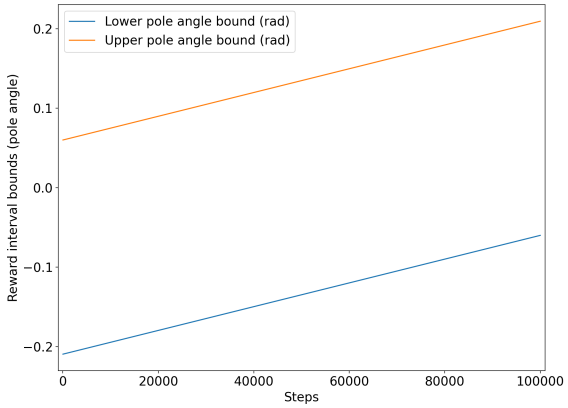

Figure 7: Non-stationary CartPole reward interval. This illustrates how the reward dynamics shift over training steps. The agent is then forced to update its policy accordingly as illustrated in Fig. 3.

Implementation details. Experiments were implemented in PyTorch using the OpenAI Gym CartPole environment (Brockman et al., 2016). Runs were executed on *24GB NVIDIA RTX A5000* GPUs. A single configuration requires approximately 8 hours to complete one million environment steps. We used Adam optimizers throughout.

B.2.1 Additional Results

Figure 8 complements the main RL results by isolating the effect of hypergradient time-smoothing on (left) bilevel local regret and (right) cumulative reward over 100k environment steps. Increasing the smoothing window w consistently lowers BLR_ω and reduces variability in the outer updates. Overall, temporal smoothing improves outer-level stability.

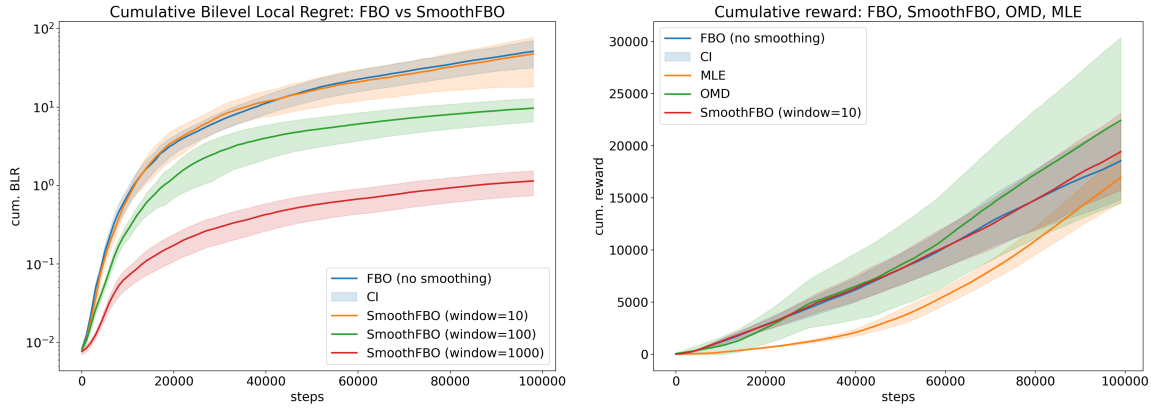

Figure 8: **Effect of hypergradient smoothing.** (Left) Bilevel local regret (BLR_ω) in the training environment for **SmoothFBO** with varying window w versus **FBO** (no smoothing); larger w lowers regret and reduces outer-update variability. (Right) Cumulative reward in the evaluation environment for **SmoothFBO** (w=10) compared to **FBO**, **OMD**, and **MLE**. The results are averaged over 20 seeds. Shaded regions show 95% confidence intervals.

B.2.2 Algorithm Configuration

For all three methods described in 6, we used neural networks with hidden dimensions of 3 for the world model, resulting in the under-specified setting as in Nikishin et al. (2022). We use soft Q-learning with temperature parameter α to encourage exploration. Each method was tuned in the stationary environment to ensure fair comparison, with the same hyperparameters used for the non-stationary evaluation.

Hyperparameter selection. We conduct a grid search in the stationary scenario using seed 1 for all three methods with the following values:

- Inner learning rate: $\{3 \cdot 10^{-3}, 3 \cdot 10^{-4}\}$
- Parameter τ: $\{10^{-1}, 10^{-2}, 10^{-3}\}$
- Temperature parameter α: $\{10^{-1}, 10^{-2}, 10^{-3}\}$

We use a replay buffer with capacity 100000 and perform 1000000 training steps during tuning. In the *SmoothFBO* implementation, we introduce a gradient buffer that maintains a history of recent hypergradients. The gradient buffer size and the smoothing parameter θ control the level of temporal averaging. These parameters were tuned to balance adaptation speed and stability over $\theta = 0.4, 0.6, 0.8$ and hypergradient buffer size of $1000, 10000$.