Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Natural Functional Gradients for Smooth Trajectory Optimization

Kisang Park[1], Chanwoo Kim[1], Kyungjae Lee[2], Sungjoon Choi[1,3]
[1]Department of Artificial Intelligence, Korea University, Seoul, Republic of Korea
[2]Department of Statistics, Korea University, Seoul, Republic of Korea
[3]RLWRLD, Seoul, Republic of Korea
{qkrrltkd1626, chanwoo-kim, kyungjae_lee, sungjoon-choi}@korea.ac.kr

***Abstract*—Generating collision-free and smooth motions remains a central challenge in robotic manipulation, particularly in cluttered environments and narrow passages where feasible regions are highly constrained and fragmented. We propose a trajectory optimization framework that performs geometry-aware updates directly in function space using natural functional gradients. The method optimizes a Gaussian-smoothed surrogate objective that regularizes the optimization landscape through smooth trajectory perturbations while preserving trajectory-level structure. Because the updates are defined intrinsically in function space, trajectory regularity can be controlled independently of a particular time discretization. We derive a practical Monte-Carlo estimator of the natural functional gradient that requires only black-box trajectory evaluations, making the method applicable when analytic gradients are unavailable or unreliable due to collision checking and contact-rich simulation. Experiments on constrained robotic manipulation tasks demonstrate that the proposed method improves trajectory feasibility and produces smoother motions than representative planning and trajectory optimization baselines in environments with narrow geometric clearances. Additional results, videos, and implementation details are available at the project page.**

I. Introduction

Trajectory optimization is a fundamental component of robotic motion generation. In manipulation tasks, a planner must simultaneously satisfy collision avoidance, kinematic feasibility, and trajectory smoothness under limited geometric clearance. These requirements become particularly challenging in cluttered environments, where feasible motions occupy only a small subset of the trajectory space and small deviations can induce collisions. In such regimes, feasibility alone is insufficient: the temporal regularity of the solution directly impacts tracking accuracy and robustness during execution.

Existing approaches provide complementary advantages but expose a persistent trade-off between feasibility and smoothness. Sampling-based planners effectively explore highly non-convex configuration spaces and discover collision-free solutions, but often return piecewise-linear paths that require additional smoothing before execution [1, 2, 3]. Trajectory optimization methods instead generate smooth motions directly, but they typically rely on local information and can become sensitive to initialization in fragmented feasible regions [4, 5, 6]. Recent GPU-accelerated methods improve optimization throughput, yet trajectory generation under narrow geometric constraints remains challenging [7].

A central difficulty is that feasibility and trajectory regularity are often handled using separate representations and optimization procedures. Methods focused on feasibility commonly operate on sparse geometric paths and defer smoothness to post-processing, while smooth trajectory optimization methods directly optimize discretized trajectories and can struggle to escape poor local regions. As a result, maintaining trajectory stability while satisfying collision constraints remains difficult, particularly when optimization relies on black-box simulation costs or lacks reliable analytic gradients.

This paper adopts a function-space perspective on trajectory optimization. Rather than representing trajectories using a fixed finite-dimensional parameterization, we model trajectories as elements of a Hilbert space and perform updates directly in function space. This formulation enables trajectory regularity to be controlled through the geometry induced by Gaussian kernels and allows smooth perturbations to be generated independently of a particular trajectory discretization [8, 9, 10]. Building on this viewpoint, we propose a zeroth-order trajectory optimization framework based on natural functional gradients induced by Gaussian smoothing.

The proposed update direction admits a simple weight-times-noise form, enabling practical Monte-Carlo optimization even when analytic gradients are unavailable or unreliable due to collision checking and contact-rich simulation. We additionally analyze the Gaussian-smoothed objective directly in Hilbert space and establish convergence guarantees without defining the optimization geometry through a particular finite-dimensional trajectory parameterization. The resulting framework combines smooth function-space perturbations with gradient-free optimization for constrained robotic motion generation tasks.

The contributions of this paper are threefold. First, we introduce a function-space trajectory optimization framework based on natural functional gradients that performs geometry-aware trajectory updates directly in Hilbert space. Second, we develop a Gaussian smoothing formulation for zeroth-order trajectory optimization and provide convergence guarantees in function space. Third, we demonstrate on constrained robotic manipulation tasks that the proposed method improves trajectory feasibility and produces smoother trajectories than representative planning and trajectory optimization baselines in cluttered environments with narrow geometric clearances.

TABLE I: Conceptual comparisons of representative motion planning and trajectory optimization approaches.

Method family	Trajectory representation	Cost differentiability required	Gradient access	Smoothness handling
Sampling-based [1, 2]	Discrete path	No	None	Post-processing
Gradient-based [4, 5]	Finite-dimensional	Yes / approximated	Analytic	Explicit regularization
Stochastic optimization [6, 11]	Finite-dimensional	No	Zeroth-order	Implicit / penalty-based
Inference-based / GP planning [12, 13, 9]	Probabilistic trajectory model	Yes / model-dependent	Variational / analytic	Model-based prior
Evolutionary algorithms [14]	Finite-dimensional	No	Zeroth-order	Not intrinsic
Zeroth-order optimization [15, 16]	Finite-dimensional	No	Zeroth-order	Via smoothing
Homotopy / continuation [17, 18]	Varies	Varies	Varies	Via smoothing schedule
Proposed (natural functional gradients)	Function space	No	Zeroth-order	Intrinsic (geometry-induced)

II. Related Work

Trajectory optimization for robotic motion generation has been studied from multiple complementary perspectives, including sampling-based planning, gradient-based trajectory optimization, stochastic optimization, and zeroth-order black-box optimization. These approaches address different aspects of the core challenge: discovering feasible motions in highly nonconvex environments while maintaining trajectory smoothness and execution stability. Table I summarizes representative approaches according to trajectory representation, gradient access, differentiability requirements, and smoothness handling.

Sampling-based planners such as PRM [1] and RRT [2, 19, 3] explore configuration spaces without relying on gradient information and are effective for discovering collision-free motions in cluttered environments. However, the resulting solutions are typically represented as sparse geometric paths and often require additional shortcutting or smoothing before execution. Gradient-based trajectory optimization methods such as CHOMP and TrajOpt [20, 4, 21, 5] instead optimize smooth trajectories directly using local cost information. Related probabilistic and function-space formulations incorporate smoothness priors through Gaussian-process trajectory models and reproducing kernel Hilbert spaces [9, 10, 8]. Despite their effectiveness, these methods commonly rely on analytic or approximated gradients and can become sensitive to initialization in environments dominated by narrow feasible regions or nondifferentiable collision costs.

Stochastic and gradient-free approaches relax differentiability requirements by estimating updates from sampled perturbations or rollout trajectories. Representative examples include STOMP [6], path-integral methods [11, 22], evolutionary strategies [14, 23], and recent GPU-parallelized motion generation systems [7]. These methods are generally more flexible with respect to black-box objectives and simulation-based costs, but most operate on finite-dimensional trajectory parameterizations and enforce smoothness indirectly through discretization-dependent penalties or post-processing. More broadly, zeroth-order optimization theory and Gaussian smoothing methods provide principled tools for optimization using only function evaluations [15, 24, 16, 25]. Related continuation and homotopy methods further study how smoothed objectives can improve optimization behavior in nonconvex landscapes [17, 18].

Our work combines these perspectives through a function-space formulation of zeroth-order trajectory optimization. Rather than optimizing a finite-dimensional trajectory parameterization, we perform geometry-aware updates directly in Hilbert space using natural functional gradients induced by Gaussian smoothing. This formulation enables smooth trajectory perturbations to be generated intrinsically in function space while remaining compatible with black-box trajectory evaluation. Unlike prior smooth trajectory optimization methods, the proposed approach does not rely on analytic gradients or discretization-tuned smoothness penalties. At the same time, it differs from conventional stochastic optimization methods by explicitly incorporating function-space geometry into the update rule and smoothing process.

We additionally analyze the Gaussian-smoothed objective directly in function space and establish convergence guarantees without reducing the optimization problem to a finite-dimensional representation. As a result, the proposed method occupies a distinct position at the intersection of function-space trajectory optimization, Gaussian smoothing, and gradient-free robotic motion generation.

III. Preliminaries

This section introduces the minimum notation and background required to describe our method. Our goal is to formalize trajectories as elements of a function space and to fix the probabilistic and geometric structures used later for defining geometry-aware, zeroth-order updates. All methodological contributions begin in Section IV; the material here is standard and is included only to establish a common framework.

A. Trajectories in a Hilbert space

Let $T > 0$ be a fixed time horizon and let $\xi : [0, T] \to \mathbb{R}^d$ denote a continuous-time trajectory. We model trajectories as elements of a separable real Hilbert space $\mathcal{H}$ equipped with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and norm $\| \cdot \|_{\mathcal{H}}$. This viewpoint is deliberately resolution-independent: time discretization is used only for numerical approximation and does not define the optimization geometry. In particular, the choice of $\mathcal{H}$ encodes a notion of regularity (e.g., smoothness) at the level of functions rather than finite-dimensional vectors.

In robotic motion generation, this abstraction is natural because physically executable trajectories must satisfy temporal regularity constraints, and many objectives are most naturally expressed as functionals of the entire path. Sobolev spaces and reproducing kernel Hilbert spaces (RKHSs) are

commonly used to encode smoothness and prior structure in trajectory optimization and motion planning [8, 9, 10]. Although our theoretical development is stated for a general Hilbert space, later sections will specialize to an RKHS-based implementation where the induced geometry directly shapes the smoothness of updates.

B. Zeroth-order objective

Let $f : \mathcal{H} \to \mathbb{R} \cup \{-\infty\}$ be a trajectory objective functional. In robotic manipulation and motion planning, f may aggregate task objectives, control effort, collision penalties, and environment-specific costs. We do not assume that f is differentiable with respect to ξ, nor that analytic gradients are available or reliable. Instead, we assume only zeroth-order access: given a candidate trajectory ξ, the value $f(\xi)$ can be evaluated, for example by collision checking and simulation-based rollout.

This assumption reflects common practice in robotics and simulation-based control. Collision checking and contact-rich dynamics often introduce nondifferentiabilities, and many pipelines incorporate black-box components or hard feasibility checks. As a result, gradient-free and zeroth-order optimization methods are widely used when analytic derivatives are unavailable or unstable [15, 16, 24]. Our method builds on this zeroth-order setting while explicitly addressing the need for smooth, executable trajectories.

C. Gaussian perturbations and Cameron–Martin space

Let $\Sigma : \mathcal{H} \to \mathcal{H}$ be a self-adjoint, positive, trace-class, and injective operator, and let $\sigma > 0$. We consider Gaussian perturbations $\varepsilon \sim \mathcal{N}(0, \sigma^2\Sigma)$ taking values in $\mathcal{H}$. Such perturbations are the basis of Gaussian smoothing, a classical technique for regularizing nonsmooth or nonconvex objectives by averaging over local perturbations [24, 25]. In the context of trajectories, the covariance operator Σ shapes how perturbations vary over time and across dimensions, thereby directly influencing the smoothness of induced updates.

A distinctive feature of Gaussian measures in infinite-dimensional spaces is that differentiation is naturally restricted to a subspace determined by the covariance. The associated Cameron–Martin space is defined as $\mathcal{H}_\Sigma = \mathrm{Range}(\Sigma^{1/2})$, equipped with the inner product

$$\langle h_1, h_2 \rangle_{\mathcal{H}_\Sigma} = \langle \Sigma^{-1/2} h_1, \Sigma^{-1/2} h_2 \rangle_{\mathcal{H}}, \quad h_1, h_2 \in \mathcal{H}_\Sigma. \quad (1)$$

The Cameron–Martin construction characterizes precisely the directions along which shifts of a Gaussian measure remain absolutely continuous and thus differentiable [26, 27, 28]. In Section IV, this geometry will induce a natural functional gradient for Gaussian-smoothed objectives, yielding smooth trajectory updates aligned with the covariance structure.

IV. Trajectory Optimization via Natural Functional Gradient

We now present our trajectory optimization method based on natural functional gradients. We treat a trajectory as a time-indexed function (e.g., $t \mapsto q(t)$ or $t \mapsto x(t)$), rather than a finite vector of waypoints, and formulate the optimization in a Hilbert space where the inner product encodes trajectory smoothness. In this setting, the kernel defines the geometry of the trajectory space, determining how trajectories are compared and how updates are shaped.

The central idea is to perform zeroth-order optimization directly in this function space, rather than on a finite-dimensional parameterization. By combining exponential transformation with Gaussian smoothing in a Hilbert space, we obtain a geometry-aware update rule that produces smooth trajectories intrinsically while requiring only black-box evaluations of the underlying cost functional. This resulting update can be interpreted as a smooth deformation of the entire trajectory.

Throughout this section, we emphasize a clear separation between three conceptual components: (i) the definition of a Gaussian-smoothed surrogate objective in function space, (ii) the derivation of its natural functional gradient under the Cameron–Martin geometry, and (iii) a practical discretization that enables efficient Monte-Carlo estimation and implementation. This organization mirrors the logical flow from problem formulation to theory and finally to computation.

A. Gaussian-Smoothed Surrogate Objective

We consider trajectory optimization over a feasible set $S \subset \mathcal{H}$, where $\mathcal{H}$ is a separable real Hilbert space of trajectories $\xi : [0, T] \to \mathbb{R}^d$. The feasible set S encodes boundary conditions and admissibility constraints such as collision avoidance and joint limits. We assume only zeroth-order access to an objective functional $f : S \to \mathbb{R}$, evaluated through collision checking and/or simulation.

To handle feasibility within a unified objective, we extend f to all of $\mathcal{H}$ by setting $f(\xi) = -\infty$ for $\xi \notin S$. This allows us to write all subsequent objectives as expectations over $\mathcal{H}$ without introducing explicit constraints. For a power parameter $N_{\mathrm{pow}} > 0$, we define the exponentially transformed functional

$$g_{N_{\mathrm{pow}}}(\xi) := \exp\big(N_{\mathrm{pow}}\, f(\xi)\big), \quad (2)$$

which amplifies relative differences between trajectories while enforcing feasibility via the indicator.

We smooth $g_{N_{\mathrm{pow}}}$ by Gaussian perturbations in $\mathcal{H}$. Let $\Sigma : \mathcal{H} \to \mathcal{H}$ be a self-adjoint, positive, trace-class, and injective covariance operator, and let

$$\varepsilon \sim \mathcal{N}(0, \sigma^2\Sigma), \quad (3)$$

where $\sigma > 0$ controls the magnitude of perturbations. The Gaussian-smoothed surrogate objective is then defined as

$$F_{N_{\mathrm{pow}}}(\mu) := \mathbb{E}\big[g_{N_{\mathrm{pow}}}(\mu + \varepsilon)\big], \qquad \mu \in \mathcal{H}, \quad (4)$$

where μ denotes the current mean trajectory.

The role of this construction is twofold. First, Gaussian smoothing regularizes the optimization landscape by averaging over local perturbations in trajectory space. Second, the exponential transformation induces a homotopy-like effect: as N_{pow} increases, the surrogate objective increasingly concentrates around trajectories with high original cost. Crucially, evaluating $F_{N_{\mathrm{pow}}}$ requires only evaluations of $f(\mu + \varepsilon)$ and does not rely on analytic gradients.

Algorithm 1 Natural Functional Gradient

Require: Initial mean trajectory μ_0 (discretized on grid $\{t_i\}_{i=1}^m$); trajectory objective functional $f(\cdot)$ (black-box evaluation); kernel $k(\cdot,\cdot)$ and regularization $\lambda > 0$; noise scale $\sigma > 0$; power parameter $N_{\text{pow}} > 0$; Monte-Carlo samples B; stepsizes $\{\eta_k\}_{k=0}^{K-1}$; iterations K.

Ensure: Optimized mean trajectory μ_K.

Construct kernel matrix $K \in \mathbb{R}^{m\times m}$ by $K_{ij} = k(t_i, t_j)$ and set $K_\lambda \leftarrow K + \lambda I$.
Compute a matrix square root L such that $LL^\top = K_\lambda$ (e.g., Cholesky factor).
Initialize $\mu \leftarrow \mu_0$.
for $k = 0$ to $K-1$ **do**
  $\widehat{g} \leftarrow 0 \in \mathbb{R}^m$.
  for $s = 1$ to B **do**
    Sample $z^{(s)} \sim \mathcal{N}(0, I_m)$.
    $\varepsilon^{(s)} \leftarrow \sigma L z^{(s)}$ ▷ $\varepsilon^{(s)} \sim \mathcal{N}(0, \sigma^2 K_\lambda)$
    Evaluate weight $w^{(s)} \leftarrow \exp\big(N_{\text{pow}}\, f(\mu + \varepsilon^{(s)})\big)$.
    $\widehat{g} \leftarrow \widehat{g} + \frac{1}{B\sigma^2}\, w^{(s)}\, \varepsilon^{(s)}$.
  end for
  $\mu \leftarrow \mu + \eta_k\, \widehat{g}$.
end for
return μ.

B. Natural Functional Gradient

Gaussian measures in infinite-dimensional spaces admit differentiation only along a covariance-dependent subspace. The Gaussian measure $\mathcal{N}(0, \sigma^2\Sigma)$ induces the Cameron–Martin space

$$\mathcal{H}_\Sigma := \text{Range}(\Sigma^{1/2}), \tag{5}$$

equipped with the inner product

$$\langle h_1, h_2\rangle_{\mathcal{H}_\Sigma} = \langle \Sigma^{-1/2} h_1, \Sigma^{-1/2} h_2\rangle_{\mathcal{H}}, \qquad h_1, h_2 \in \mathcal{H}_\Sigma. \tag{6}$$

This geometry characterizes the admissible directions along which the smoothed objective $F_{N_{\text{pow}}}$ can be differentiated.

Lemma 1 (Directional derivative identity)**.** *For any $\mu \in \mathcal{H}$ and any $h \in \mathcal{H}_\Sigma$, the directional derivative of $F_{N_{\text{pow}}}$ exists and satisfies*

$$D_h F_{N_{\text{pow}}}(\mu) = \mathbb{E}\left[g_{N_{\text{pow}}}(\mu + \varepsilon) \sum_{k:\lambda_k>0} \frac{\langle h, e_k\rangle_{\mathcal{H}}}{\sigma\sqrt{\lambda_k}} \xi_k \right], \tag{7}$$

where $\varepsilon = \sigma \sum_{k:\lambda_k>0} \sqrt{\lambda_k}\, \xi_k e_k$ with $\xi_k \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ is the Karhunen–Loève representation associated with the eigendecomposition $\Sigma e_k = \lambda_k e_k$.

The full proof can be found in the supplementary materials. The result follows by rewriting $F_{N_{\text{pow}}}(\mu + th)$ as an expectation with respect to a translated Gaussian measure and applying the Cameron–Martin theorem to obtain the corresponding Radon–Nikodym derivative (see Chapter 2 in [27]).

Lemma 2 (Closed form of the natural functional gradient)**.** *There exists a unique $\nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu) \in \mathcal{H}_\Sigma$ such that*

$$D_h F_{N_{\text{pow}}}(\mu) = \langle \nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu), h\rangle_{\mathcal{H}_\Sigma} \quad \textit{for all } h \in \mathcal{H}_\Sigma,$$

and it admits the closed-form representation

$$\nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu) = \sigma^{-2}\, \mathbb{E}\big[g_{N_{\text{pow}}}(\mu + \varepsilon)\, \varepsilon \big]. \tag{8}$$

Lemma 2 shows that the directional derivative of $F_{N_{\text{pow}}}$ admits a Riesz representer in the Cameron–Martin space, yielding the closed-form expression in (8) [29]. This representation leads to a practical estimator, where Gaussian perturbations are weighted by their exponentiated objective values. Crucially, the resulting update depends only on zeroth-order evaluations of f, making it applicable in nondifferentiable or simulation-based settings.

C. Monte-Carlo Estimation and Update Rule

In practice, the expectation in (8) is approximated using Monte-Carlo sampling. Given i.i.d. samples $\varepsilon^{(1)}, \ldots, \varepsilon^{(B)} \sim \mathcal{N}(0, \sigma^2\Sigma)$, we use the unbiased estimator

$$\widehat{\nabla}_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu) = \frac{1}{B\sigma^2} \sum_{b=1}^{B} g_{N_{\text{pow}}}(\mu + \varepsilon^{(b)})\, \varepsilon^{(b)}. \tag{9}$$

The number of samples B controls the variance of the estimator, while the covariance operator Σ determines the smoothness of sampled perturbations.

We perform an explicit natural-gradient ascent step

$$\mu_{k+1} = \mu_k + \eta_k\, \widehat{\nabla}_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu_k), \tag{10}$$

where $\eta_k > 0$ is a step size. Because the update direction lies in $\mathcal{H}_\Sigma$, trajectory smoothness is preserved intrinsically by construction.

D. Discretization for Implementation

For numerical implementation, we discretize trajectories on a time grid $\{t_i\}_{i=1}^m \subset [0, T]$ and represent μ by its values on the grid. We implement the covariance operator Σ via a reproducing kernel Hilbert space (RKHS) kernel $k : [0,T] \times [0,T] \to \mathbb{R}$ and its kernel matrix $K \in \mathbb{R}^{m\times m}$ defined by $K_{ij} = k(t_i, t_j)$. We use a regularized kernel matrix $K_\lambda = K + \lambda I$ with $\lambda > 0$ to ensure numerical stability.

Gaussian perturbations are sampled as

$$\varepsilon^{(s)} = \sigma K_\lambda^{1/2} z^{(s)}, \qquad z^{(s)} \sim \mathcal{N}(0, I_m), \tag{11}$$

so that $\varepsilon^{(s)} \sim \mathcal{N}(0, \sigma^2 K_\lambda)$. A direct discretization of (8) yields the estimator

$$\widehat{g}(\mu) = \frac{1}{B\sigma^2} \sum_{s=1}^{B} \exp\big(N_{\text{pow}} f(\mu + \varepsilon^{(s)})\big)\, \varepsilon^{(s)}. \tag{12}$$

Then, the mean trajectory is updated according to (10).

Algorithm 1 summarizes the proposed optimizer in its practical RKHS discretization. At each iteration, we draw smooth Gaussian perturbations by sampling $z^{(s)} \sim \mathcal{N}(0, I_m)$ and mapping them through the Cholesky factor L of the

regularized kernel matrix K_λ, which implements the covariance structure that controls trajectory regularity. The update direction is estimated via (12): each perturbation is weighted by $\exp(N_{\text{pow}} f(\mu+\varepsilon^{(s)}))$ and multiplied by the corresponding direction $Lz^{(s)}$, avoiding explicit matrix inversion. The resulting Monte-Carlo estimate $\widehat{g}$ is then used for an explicit ascent step $\mu \leftarrow \mu + \eta_k \widehat{g}$, yielding smooth trajectory updates whose structure is determined by the kernel while remaining compatible with black-box, potentially nondifferentiable objectives.

Remark (Resolution-agnostic discretization)**.** *Although Algorithm 1 is implemented on a finite time grid, the optimization problem is formulated in the continuous Hilbert space $\mathcal{H}$. The discretization serves only as a numerical representation and can therefore be refined or changed without altering the underlying trajectory optimization problem.*

Remark (Finite-dimensional approximation and covariance truncation)**.** *Although the proposed optimization method is formulated intrinsically in the infinite-dimensional Hilbert space $\mathcal{H}$, the practical implementation operates on a finite-dimensional approximation induced by trajectory discretization and RKHS kernel representations.*

Let $(e_k)_{k\geq 1}$ denote the eigenbasis of the covariance operator Σ, satisfying $\Sigma e_k = \lambda_k e_k$, for $\lambda_k > 0$. For a truncation level $d_{\text{eff}} \geq 1$, define the finite-dimensional subspace $\mathcal{H}_{d_{\text{eff}}} := \text{span}\{e_1, \ldots, e_{d_{\text{eff}}}\} \subset \mathcal{H}$, and let $P_{d_{\text{eff}}} : \mathcal{H} \to \mathcal{H}_{d_{\text{eff}}}$ denote the $\mathcal{H}$-orthogonal projection $P_{d_{\text{eff}}} h = \sum_{k=1}^{d_{\text{eff}}} \langle h, e_k\rangle_{\mathcal{H}} e_k$. Here, d_{eff} is interpreted as an effective dimension of the trajectory optimization problem and is closely related to the trajectory discretization level m used in the RKHS implementation. The projection operator introduced later plays an analogous role in the stochastic analysis and iteration-complexity guarantees.

V. Iteration Complexity of Natural Functional Gradient

This subsection establishes theoretical guarantees for Gaussian-smoothed natural functional gradient ascent. In particular, we show that exponential smoothing induces an optimization landscape oriented toward global optimality that eliminates spurious stationary points, and we derive a global iteration-complexity bound based on Monte–Carlo estimation. ξ^* denotes a unique optimizer of the original objective $f(\xi)$. We optimize a Gaussian-smoothed surrogate that serves as a tractable proxy in the zeroth-order setting.

Theorem 1 (Directional ascent toward optimal point)**.** *Assume that $g_N(x) := e^{Nf(x)}\mathbf{1}_{\{x\in S\}}$ is bounded for each N. For $M > 0$, define the ellipsoid $K_M := \overline{\{\mu \in \mathcal{H}_\Sigma : \|\mu\|_{\mathcal{H}_\Sigma} \leq M\}}$. Then, for every $\epsilon > 0$ and $M > 0$, there exists $N_0 = N_0(\epsilon, M, \sigma, \Sigma) < \infty$ such that for all $N \geq N_0$, for every $\mu \in K_M$ satisfying $\|\mu - \xi^\star\|_{\mathcal{H}} \geq \epsilon$, letting*

$$u := (\xi^\star - \mu)/\|\xi^\star - \mu\|_{\mathcal{H}}, \tag{13}$$

we have $D_{\Sigma u} F_N(\mu) > 0$.

The proof can be found in the supplementary materials. Theorem 1 shows that, under stated conditions, the natural functional gradient of the surrogate is directionally aligned with ξ^*: outside an ϵ-neighborhood of ξ^*, the update provides a strict ascent direction toward ξ^* for sufficiently large N. Consequently, the optimization is guided toward an ϵ neighborhood of the true optimum.

Remark (Eigen-directional ascent outside an ϵ-ball)**.** *Let e_i be an eigenfunction of the covariance operator Σ, i.e., $\Sigma e_i = \lambda_i e_i$ with $\lambda_i > 0$. Consider an iterate of the form $\mu = \xi^\star - t\epsilon e_i$ with $t > 1$, so that $\|\mu - \xi^\star\|_{\mathcal{H}} > \epsilon$. Then, by Theorem 1, the Cameron–Martin direction Σe_i is an ascent direction of the smoothed objective F_N for all sufficiently large N. Equivalently, since directional derivatives are linear and $\lambda_i > 0$, the eigenfunction direction e_i itself satisfies $D_{e_i} F_N(\mu) > 0$. Hence, outside the ϵ-ball around $\xi^\star$, the natural functional gradient always admits an ascent component toward the optimizer along the corresponding eigenmode. In particular, no point outside this neighborhood can be stationary for the smoothed objective.*

We now derive an iteration complexity bound for the proposed stochastic natural functional gradient method by adapting standard stochastic gradient analysis to the Cameron–Martin geometry induced by Gaussian smoothing.

Lemma 3 (Second directional derivative identity)**.** *Assume that $g_{N_{\text{pow}}}$ is bounded. Then, for any $h, k \in \mathcal{H}_\Sigma$, the mixed second directional derivative exists and satisfies*

$$D^2 F_{N_{\text{pow}}}(\mu)[h,k] = \frac{1}{\sigma^2}\mathbb{E}\Big[g_{N_{\text{pow}}}(\mu+\varepsilon)\left(\widehat{h}(\varepsilon)\widehat{k}(\varepsilon) - \langle h,k\rangle_{\mathcal{H}_\Sigma}\right)\Big], \tag{14}$$

where $\widehat{h}(\varepsilon) := \sum_{j:\lambda_j>0}\langle h, e_j\rangle_{\mathcal{H}}\xi_j/\sqrt{\lambda_j}$ and $\widehat{k}(\varepsilon) := \sum_{j:\lambda_j>0}\langle k, e_j\rangle_{\mathcal{H}}\xi_j/\sqrt{\lambda_j}$. $D^2 F_{N_{\text{pow}}}(\mu)[h,k] = D^2 F_{N_{\text{pow}}}(\mu)[k,h]$ also hold.

The proof can be found in the supplementary materials. Lemma 3 provides a closed-form second-order identity via Gaussian integration by parts. This identity directly yields a uniform bound on the operator norm of the Hessian along Cameron–Martin directions, leading to a Lipschitz constant that depends only on $\|g_{N_{\text{pow}}}\|_\infty$ and σ.

Lemma 4 (Lipschitz continuity of the natural functional gradient)**.** *Assume that $g_{N_{\text{pow}}}$ is bounded, and define $\|g_{N_{\text{pow}}}\|_\infty := \sup_{\xi\in\mathcal{H}} |g_{N_{\text{pow}}}(\xi)| < \infty$ Then the natural functional gradient $\nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}$ is Lipschitz continuous along Cameron–Martin directions: for any $\mu, \nu \in \mathcal{H}$ such that $\mu - \nu \in \mathcal{H}_\Sigma$, we have*

$$\left\|\nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu) - \nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\nu)\right\|_{\mathcal{H}_\Sigma} \leq \frac{2\|g_{N_{\text{pow}}}\|_\infty}{\sigma^2}\|\mu-\nu\|_{\mathcal{H}_\Sigma}. \tag{15}$$

The proof can be found in the supplementary materials. We then control the stochasticity introduced by Monte-Carlo estimation. The following lemma shows unbiasedness and a second-moment bound that scales with d_{eff}, capturing the intrinsic effective dimension of the perturbations.

Lemma 5 (Second moment of the projected Monte-Carlo natural gradient estimator)**.** *Let $(e_k)_{k\geq 1}$ be an eigenbasis of Σ, and define $\mathcal{H}_{d_{\text{eff}}} := \text{span}\{e_1, \ldots, e_{d_{\text{eff}}}\}$. Let $P_{d_{\text{eff}}}$:*

$\mathcal{H} \to \mathcal{H}_{d_{\text{eff}}}$ *be the* $\mathcal{H}$*-orthogonal projection. Let* $\widehat{g}(\mu)$ *be the discretized Monte-Carlo estimator defined in* (12)*, where* $\varepsilon^{(b)} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2 K_\lambda)$*. Then* $\widehat{g}(\mu)$ *is an unbiased estimator of* $P_{d_{\text{eff}}} \nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu)$*, and satisfies*

$$\mathbb{E}\Big[\big\|\widehat{g}(\mu)\big\|_{\mathcal{H}_\Sigma}^2\Big] \le \frac{\|g_{N_{\text{pow}}}\|_\infty^2 d_{\text{eff}}}{B\sigma^2}. \tag{16}$$

The proof can be found in the supplementary materials. Combining the Lipschitz property with the second-moment bound yields a one-step expected ascent inequality.

Lemma 6 (One-step expected ascent)**.** *Let* $P_{d_{\text{eff}}} : \mathcal{H} \to \mathcal{H}_{d_{\text{eff}}}$ *be the* $\mathcal{H}$*-orthogonal projection. Suppose that* $\{\mu_t\}$ *is generated by the projected update rule. Then, conditioning on* μ_t*,*

$$\mathbb{E}\big[F_{N_{\text{pow}}}(\mu_{t+1}) \,\big|\, \mu_t\big] \ge F_{N_{\text{pow}}}(\mu_t) + \eta_t \big\|P_{d_{\text{eff}}} \nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu_t)\big\|_{\mathcal{H}_\Sigma}^2 - \frac{\|g_{N_{\text{pow}}}\|_\infty^3 m}{B\sigma^4} \eta_t^2. \tag{17}$$

The proof can be found in the supplementary materials. Finally, we conclude with an iteration-complexity bound for the covariance-truncated stochastic dynamics. The result gives a zeroth-order convergence rate, expressed in terms of $\|g_{N_{\text{pow}}}\|_\infty$, σ, and d_{eff}.

Theorem 2 (Iteration complexity in a Hilbert space)**.** *Let* $\{\mu_t\}_{t\ge 0} \subset \mathcal{H}$ *be generated by the projected update rule with a constant stepsize* $\eta > 0$*. Then the iterates satisfy*

$$\min_{0\le t\le T-1} \mathbb{E}\Big[\big\|P_{d_{\text{eff}}} \nabla_{\mathcal{H}_\Sigma} F_{N_{\text{pow}}}(\mu_t)\big\|_{\mathcal{H}_\Sigma}\Big] \le \varepsilon \tag{18}$$

provided that $T \ge \frac{4\,\|g_{N_{\text{pow}}}\|_\infty^4\, d_{\text{eff}}}{B\,\sigma^4 \varepsilon^4}$.

The proof is provided in the supplementary materials. Here, d_{eff} denotes the effective dimension induced by the truncated covariance eigenbasis. The iteration complexity scales linearly with d_{eff}, yielding an overall complexity of $\mathcal{O}(d_{\text{eff}}\varepsilon^{-4})$ for reaching ε-stationarity. To our knowledge, this is the first global convergence-rate result for trajectory optimization formulated directly in function space.

Remark (Comparison with Existing Trajectory Optimization Methods)**.** *The proposed method is related to sampling-based planners such as RRT*, which are asymptotically optimal but often require additional smoothing for discrete paths [3]. In contrast, our method directly optimizes smooth trajectories through the Cameron–Martin geometry.*

Unlike CHOMP and TrajOpt, which rely on analytic gradients and discretization-dependent regularization [4, 5], the proposed approach does not require differentiable costs and remains applicable under nondifferentiable collision checking.

From a zeroth-order optimization viewpoint, the resulting $\mathcal{O}(\varepsilon^{-4})$ *complexity matches standard Gaussian smoothing rates while explicitly revealing how the covariance operator controls exploration and trajectory regularity [24, 25].*

VI. Experiments

This section evaluates the proposed natural functional gradient (NFG) trajectory optimizer across progressively more constrained settings. We first study optimization behavior in a synthetic narrow-passage environment (§VI-A), where the feasible region is fragmented and highly sensitive to trajectory perturbations. We then evaluate manipulation performance in cluttered cabinet insertion tasks (§VI-B), including unified execution validation under shared simulation conditions. Finally, we demonstrate real-world execution on a physical robotic platform (§VI-C) to illustrate the practical feasibility and smooth execution behavior of the proposed function-space trajectory optimization framework.

A. Synthetic Narrow-Passage Experiment

We first evaluate the proposed method in a synthetic narrow-passage trajectory optimization problem designed to study optimization behavior under fragmented feasible regions. The environment consists of multiple box-shaped obstacles arranged along the temporal axis, producing a narrow collision-free corridor that occupies only a small subset of the trajectory space. Because small trajectory perturbations can easily induce collision, the environment provides a controlled benchmark for evaluating optimization behavior under constrained geometry.

The task is formulated as a one-dimensional trajectory optimization problem over a fixed $1\,\text{s}$ horizon discretized at $100\,\text{Hz}$. The objective combines collision avoidance with a trajectory regularity bonus based on the average absolute jerk of the trajectory. Additional details and environment specifications are provided in the supplementary material.

For the proposed method, trajectory perturbations are generated using a squared exponential (SE) kernel, defined as $k_{\text{SE}}(x, x') = g^2 \exp\big(-\frac{1}{2l^2}|x - x'|^2\big)$, with variance $g^2 = 0.29$ and length-scale $l = 0.22$. The exponential smoothing parameter is set to $N_{\text{pow}} = 100$, and each optimization step uses 100 Monte-Carlo trajectory perturbations. We compare against representative trajectory optimization baselines including CHOMP [4], STOMP [6], and MPPI [30]. All methods are initialized using the same linear interpolation trajectory between the initial and final states and evaluated under the same trajectory horizon and temporal discretization. NFG, CHOMP [4], and STOMP [6] are executed for 100 iterations, while MPPI [30] is evaluated using its standard online planning configuration. All experiments are repeated over five random seeds while using the same obstacle configuration and optimization setup across seeds. Performance is evaluated using success rate, runtime, path length, and average jerk. A trajectory is counted as successful if it remains collision-free throughout the trajectory horizon. Average jerk is computed from successful trajectories only.

Table II summarizes the quantitative results. The proposed method achieves the highest success rate among the baselines, consistently discovering feasible trajectories despite the fragmented feasible region. In contrast, local trajectory optimization methods, e.g., CHOMP [4] and STOMP [6], frequently fail to enter the narrow feasible corridor, while MPPI [30] occasionally discovers feasible solutions but produces trajectories with substantially larger jerk. Although some baselines achieve lower jerk values in individual successful trials, the

TABLE II: Comparison with the baseline planners in the synthetic box-avoidance scenario.

Method	Success Rate ↑ (%)	Time ↓ (s)	Path Length ↓ (m)	Avg Jerk ↓ (m/s^3)
NFG (Ours)	**100.0**	0.73 ± 0.03	**10.06 ± 1.25**	90.91 ± 33.43
STOMP [6]	20.0	0.63 ± 0.01	10.62 ± 0.00	**34.67 ± 0.00**
CHOMP [4]	20.0	**0.02 ± 0.00**	10.82 ± 0.00	43.52 ± 0.00
MPPI [30]	60.0	0.37 ± 0.00	7.68 ± 0.43	621.22 ± 86.07

proposed method maintains a more consistent balance between feasibility and trajectory regularity across repeated runs.

B. Simulation Experiments in Cabinet Environments

We next evaluate the proposed method on robotic manipulation tasks in cluttered cabinet environments with progressively constrained geometry. The experiments study trajectory generation under narrow collision-free clearances, where feasible motions require stable trajectory deformation near the insertion region. We first describe the cabinet environments and optimization setup (§VI-B1), followed by the execution-level evaluation protocol (§VI-B2) and quantitative comparisons against representative planning and trajectory optimization baselines (§VI-B3).

1) Environment Setup: We evaluate the proposed method using a Franka Research 3 manipulator in a cabinet insertion task. The robot inserts a cylindrical object into a cabinet while avoiding collisions with surrounding obstacles and cabinet walls. Because the insertion region contains limited geometric clearance, small trajectory deviations near the cabinet opening can induce collision during execution. To study performance under progressively constrained geometry, we consider four environments with increasing levels of difficulty: free space, fully open, quarter-closed, and half-closed, as illustrated in Fig. 1. The free space environment is included as a sanity-check setting without cabinet constraints, while the remaining environments introduce progressively narrower cabinet openings around the insertion region. All environments share the same initial and final joint configurations, allowing the effect of geometric constraints on trajectory generation to be evaluated consistently across environments.

The trajectory objective combines collision avoidance with joint-space motion regularization. Collision costs are computed from penetration depth and contact information obtained during simulation, while an additional path-length regularization term penalizes excessive joint-space motion along the trajectory horizon. This formulation encourages trajectories that avoid collision while limiting unnecessarily aggressive configuration changes during execution. All methods are initialized using the same joint-space linear interpolation trajectory between the start and goal configurations and evaluated under the same simulation and execution settings.

Following the synthetic narrow-passage experiment, the proposed method uses a squared exponential (SE) kernel to generate smooth trajectory perturbations in function space. The kernel variance is set to $g^2 = 1.0$, and the length-scale is chosen as half of the trajectory horizon. Trajectory

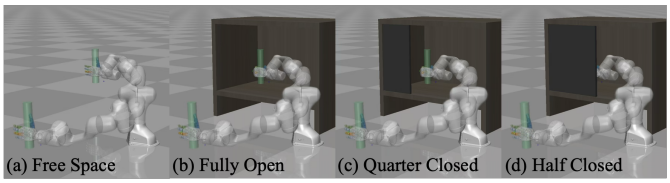

Fig. 1: Cabinet insertion environments with increasing geometric constraints: (a) free space, (b) fully open, (c) quarter-closed, and (d) half-closed.

optimization is performed over a fixed $5\,\mathrm{s}$ horizon discretized at $100\,\mathrm{Hz}$ using 100 sampled trajectories per optimization step with $N_{\mathrm{pow}} = 20$. Each optimization run is executed for at most 30 iterations or $300\,\mathrm{s}$ and terminates early once a collision-free trajectory is found.

We compare the proposed method against representative motion planning and trajectory optimization baselines, including BiTRRT [31], CHOMP [4], STOMP [6], and cuRobo [7]. These methods cover sampling-based planning, gradient-based trajectory optimization, stochastic trajectory optimization, and GPU-accelerated motion generation, respectively. All baselines are evaluated using the same start and goal configurations, trajectory discretization, and execution settings. For each cabinet environment, all methods are evaluated over five random seeds under the same environment geometry and task configuration. Additional implementation details and evaluation settings are provided in the supplementary material.

2) Unified Execution Evaluation: The compared planners employ different internal trajectory representations, optimization strategies, and collision models. Some methods generate sparse geometric waypoints [31], while others directly optimize discretized joint trajectories [4, 6] or rely on planner-specific collision approximations during optimization [7]. As a result, the native outputs of different planners are not directly comparable under a shared execution setting without additional trajectory processing and validation.

To enable a consistent comparison, all generated trajectories are converted into a unified time-parameterized joint-space trajectory and resampled at a fixed execution frequency of $100\,\mathrm{Hz}$. The resampled trajectories are then evaluated in the MuJoCo simulator using the same collision-checking and execution settings across all methods. A rollout is considered successful if the resampled execution trajectory remains collision-free throughout the full trajectory horizon.

In addition to success rate, we report total optimization time and average jerk computed from the executed trajectories. Average jerk is reported only for successful trajectories and is used to evaluate trajectory regularity during execution.

3) Quantitative Results: Table III summarizes the quantitative results across the cabinet environments. We report success rate, total optimization time, and average jerk measured from the executed trajectories after trajectory resampling and simulation-based validation. A trajectory is counted as successful if the resampled execution trajectory remains collision-free throughout the full trajectory horizon. Average jerk is reported only for successful trajectories and is used to evaluate

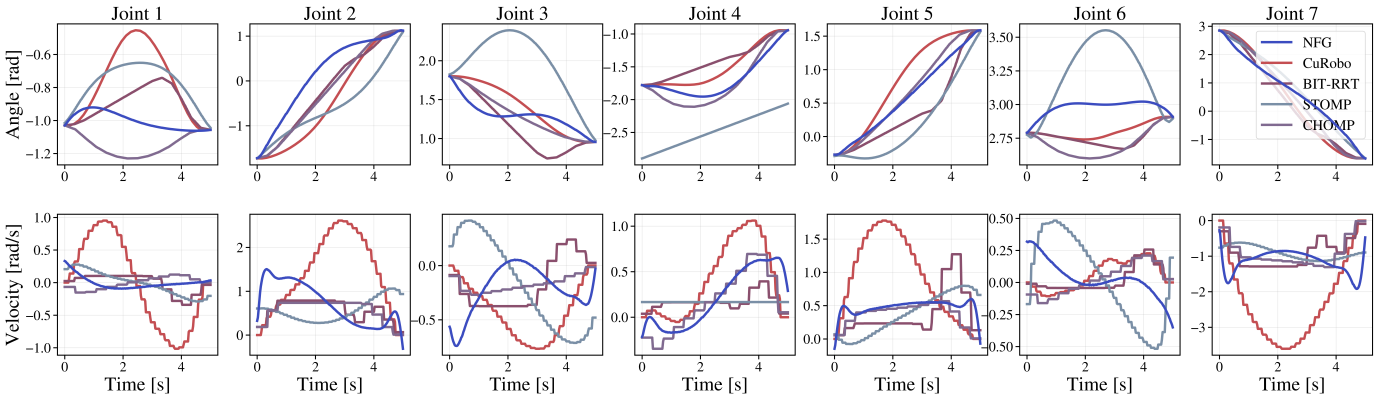

Fig. 2: Joint trajectories for the fully open cabinet environment. The top and bottom rows show the joint positions (rad) and joint velocities $(\mathrm{rad/s})$ of the Franka Research 3 manipulator, respectively. STOMP [6] fails to generate a feasible trajectory.

TABLE III: Comparison with baseline planners across cabinet environments.

(a) **Success Rate** (↑, %)

Method	Free Space	Fully Open	Quarter-Closed	Half-Closed
NFG (Ours)	**100.0**	**100.0**	**100.0**	**100.0**
BiTRRT [31]	**100.0**	40.0	13.3	16.7
STOMP [6]	**100.0**	0.0	0.0	0.0
CHOMP [4]	**100.0**	**100.0**	0.0	0.0
cuRobo [7]	**100.0**	**100.0**	**100.0**	**100.0**

(b) **Optimization Time** (↓, s)

NFG (Ours)	0.31 ± 0.01	4.10 ± 2.89	103.68 ± 75.29	166.10 ± 91.63
BiTRRT [31]	0.19 ± 0.02	**0.30 ± 0.04**	**0.51 ± 0.29**	**0.82 ± 0.18**
STOMP [6]	**0.00 ± 0.00**	-	-	-
CHOMP [4]	0.13 ± 0.01	0.31 ± 0.10	-	-
cuRobo [7]	0.01 ± 0.00	0.020 ± 0.01	14.24 ± 0.17	128.87 ± 0.18

(c) **Average Jerk** (↓, $\mathrm{rad/s^3}$)

NFG (Ours)	461.61 ± 178.53	461.61 ± 178.52	**694.06 ± 275.11**	**840.40 ± 370.27**
STOMP [6]	**0.82 ± 0.18**	-	-	-
CHOMP [4]	57.62 ± 0.05	**109.21 ± 6.39**	-	-
cuRobo [7]	1141.31 ± 39.28	859.73 ± 11.19	1219.58 ± 54.75	1177.27 ± 79.62

trajectory regularity during execution.

In the free space and fully open cabinet environments, multiple planners, including the proposed method, BiTRRT [31], and cuRobo [7], are able to generate collision-free trajectories under the shared execution setting. Because the geometric constraints are relatively mild in these scenarios, feasible motions can often be discovered without substantial trajectory deformation. As the cabinet opening becomes smaller, however, the feasible region becomes increasingly narrow, making feasible trajectory generation significantly more difficult in the quarter-closed and half-closed environments.

Under these more constrained settings, the proposed method maintains successful trajectory generation across all cabinet configurations while producing lower average jerk than cuRobo [7]. The joint trajectories in Fig. 2 exhibit gradual configuration changes throughout the insertion motion without abrupt transitions near the constrained region. In contrast, sampling-based and optimization-based baselines show reduced success rates as the cabinet opening becomes narrower.

cuRobo [7] also achieves high feasibility across the evaluated cabinet environments while demonstrating substantially faster optimization times. However, the resulting trajectories exhibit larger execution jerk under the shared evaluation setting. Overall, the results suggest that the proposed method provides stable trajectory generation in cluttered manipulation environments while maintaining smooth trajectory evolution near narrow geometric constraints.

C. Real-World Robotic Demonstration

We further evaluate the proposed method on a real-world manipulation task using a Doosan Robotics A0912 manipulator in a cluttered bookshelf environment. The task requires the robot to transfer a cylindrical object from a second-level shelf to a target placement region located inside a narrow compartment on the third level of the bookshelf. Because the target region is partially enclosed by surrounding shelf structures, feasible motions require accurate trajectory generation near constrained workspace boundaries.

The initial and final robot configurations are determined from inverse kinematics solutions corresponding to the grasp pose and target placement pose of the cylindrical object, respectively. Trajectory optimization is initialized using a joint-space linear interpolation trajectory between the initial and final configurations. The optimized trajectories are generated over a fixed $5\,\mathrm{s}$ horizon discretized at $100\,\mathrm{Hz}$ and executed directly on the robot using the same temporal discretization.

Figure 3 shows representative snapshots from the real-world experiment. The robot successfully completes the transfer and insertion task while maintaining stable motion near the constrained shelf geometry. These results demonstrate that the proposed method can generate feasible and smooth motions in cluttered real-world manipulation environments.

VII. Discussion and Limitations

This work demonstrates that natural functional gradient optimization provides a practical framework for trajectory generation in constrained robotic manipulation environments. By operating directly in function space and applying Gaussian smoothing, the proposed method generates smooth trajectory

Fig. 3: Real-world robotic manipulation in a cluttered environment with a narrow box constraint.

perturbations while remaining compatible with zeroth-order optimization and black-box trajectory evaluation. The resulting framework allows feasibility and trajectory regularity to be handled within a unified optimization procedure, particularly in environments with narrow geometric clearances and fragmented feasible regions.

Table IV summarizes conceptual differences between the proposed method and representative planning and trajectory optimization approaches. Unlike methods that rely on geometric primitives or approximated collision representations, the proposed framework directly incorporates simulator-based contact evaluation during optimization. This allows trajectory updates to be computed using the same simulation environment employed during execution without requiring additional analytical approximations of collision geometry. The formulation also operates as a unified framework that jointly handles smooth trajectory generation and collision-aware optimization while remaining compatible with parallel rollout-based evaluation.

The primary limitation of the proposed approach is the additional computational cost introduced by Monte-Carlo estimation of the natural functional gradient. Although the optimization procedure is naturally parallelizable, the required number of sampled trajectory perturbations can still become expensive in high-dimensional problems or time-critical settings. Improving sample efficiency and exploiting large-scale parallel computation, similar to recent GPU-accelerated motion generation systems [7], remain important directions.

The performance of the method additionally depends on the covariance operator used to generate trajectory perturbations. Although fixed squared exponential kernels are effective across the evaluated tasks, performance can vary depending on the smoothing scale and covariance structure [24, 25]. Designing adaptive or task-dependent covariance operators that better reflect environmental geometry, contact structure, or task constraints remains an important open problem.

TABLE IV: Comparison of motion planning methods

Method	Smooth Trajectory	No Primitive Approx.	Simulator Contact Solver	Single Package	Parallel -izable
Bi-RRT [31]		✓			✓
STOMP [6]	✓			✓	✓
CHOMP [4]	✓			✓	
MPPI [30]	✓	✓	✓		✓
CuRobo [7]	✓			✓	✓
Ours	✓	✓	✓	✓	✓

VIII. Conclusion

This paper presented a trajectory optimization framework that performs geometry-aware updates directly in function space using natural functional gradients induced by Gaussian smoothing. The proposed method enables zeroth-order trajectory optimization with smooth function-space perturbations without relying on analytic gradients or discretization-dependent smoothness penalties. We additionally analyzed the Gaussian-smoothed objective directly in Hilbert space and established convergence guarantees for the resulting optimization procedure. Experiments on constrained robotic manipulation tasks demonstrated that the proposed method improves success rate and produces smoother execution trajectories than representative planning and trajectory optimization baselines in cluttered environments with narrow geometric clearances. These results suggest that function-space natural functional gradient optimization provides a practical framework for trajectory generation when feasible regions are sparse and reliable gradient information is difficult to obtain.

Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II190079, Artificial Intelligence Graduate School Program (Korea University), 25%); (No. RS-2022-II220480, Development of Training and Inference Methods for Goal-Oriented Artificial Intelligence Agents, 25%); (No. RS-2024-00336738, Development of Complex Task Planning Technologies for Autonomous Agents, 25%); This research was also financially supported by the Ministry of Trade, Industry, and Energy (MOTIE), Korea, under the "Global Industrial Technology Cooperation Center program" supervised by the Korea Institute for Advancement of Technology (KIAT) (Grant No. P0028435, 25%). This research was also supported by a Korea University Grant (K2612161). We thank RLWRLD for generously supporting our experiments by providing access to the Franka Research 3 robot platform and the Inspire Hand system.

References

[1] L. E. Kavraki, P. Švestka, J. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[2] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep. TR 98-11, Computer Science Department, Iowa State University, Oct. 1998.

[3] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[4] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[5] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[6] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4569–4574, 2011.

[7] B. Sundaralingam, S. Hari, A. Fishman, C. Garrett, K. V. Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox, "Curobo: Parallelized collision-free minimum-jerk robot motion generation," *arXiv preprint arXiv:2310.17274*, 2023.

[8] Z. Marinho, B. Boots, A. D. Dragan, A. Byravan, G. J. Gordon, and S. S. Srinivasa, "Functional gradient motion planning in reproducing kernel hilbert spaces," in *Proceedings of Robotics: Science and Systems (RSS)*, 2016.

[9] M. Mukadam, X. Yan, and B. Boots, "Gaussian process motion planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9–15, 2016.

[10] J. Dong, M. Mukadam, F. Dellaert, and B. Boots, "Motion planning as probabilistic inference using gaussian processes and factor graphs," in *Proceedings of Robotics: Science and Systems (RSS)*, 2016.

[11] E. A. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *Journal of Machine Learning Research*, vol. 11, pp. 3137–3154, 2010.

[12] M. Toussaint, "Robot trajectory optimization using approximate inference," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pp. 1049–1056, 2009.

[13] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," *arXiv preprint arXiv:1805.00909*, 2018.

[14] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[15] A. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, Vancouver, British Columbia, Canada, January 23-25, 2005*, pp. 385–394, SIAM, 2005.

[16] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2788–2806, 2015.

[17] H. Mobahi and J. W. Fisher III, "A theoretical analysis of optimization by gaussian continuation," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA* (B. Bonet and S. Koenig, eds.), pp. 1205–1211, AAAI Press, 2015.

[18] E. Hazan, K. Y. Levy, and S. Shalev-Shwartz, "On graduated optimization for stochastic non-convex problems," in *Proceedings of the 33nd International Conference on Machine Learning, ICML, New York City, NY, USA, June 19-24, 2016* (M. Balcan and K. Q. Weinberger, eds.), vol. 48, pp. 1833–1841, JMLR.org, 2016.

[19] J. J. K. Jr. and S. M. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 995–1001, 2000.

[20] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 489–494, May 2009.

[21] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *Proceedings of Robotics: Science and Systems (RSS)*, 2013.

[22] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control using covariance variable importance sampling," *arXiv preprint arXiv:1509.01149*, 2015.

[23] N. Hansen, Y. Akimoto, and P. Baudis, "CMA-ES/pycma on Github." Zenodo, DOI:10.5281/zenodo.2559634, Feb. 2019.

[24] Y. E. Nesterov and V. G. Spokoiny, "Random gradient-free minimization of convex functions," *Found. Comput. Math.*, vol. 17, no. 2, pp. 527–566, 2017.

[25] K. Gao and O. Sener, "Generalizing gaussian smoothing for random search," in *International Conference on Machine Learning, 17-23 July 2022, Baltimore, Maryland, USA*, vol. 162 of *Proceedings of Machine Learning Research*, pp. 7077–7101, PMLR, 2022.

[26] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276,

1998.
[27] V. I. Bogachev, *Gaussian measures*. No. 62, American Mathematical Soc., 1998.
[28] G. Da Prato and J. Zabczyk, *Stochastic equations in infinite dimensions*, vol. 152. Cambridge university press, 2014.
[29] F. Riesz and B. S. Nagy, *Functional analysis*. Courier Corporation, 2012.
[30] G. Williams, A. Aldrich, and E. Theodorou, "Model predictive path integral control using covariance variable importance sampling," *arXiv preprint arXiv:1509.01149*, 2015.
[31] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.