Beyond the Mean: Gaussian Distributional Successor Features for Zero-Shot Non-Linear Reward Adaptation

Yong Zhao[1], **Wen Sun**[1†], **Jianhua He**[2] and **Peng Wang**[3], **Qubeijian Wang**[1]
[1]School of Cybersecurity, Northwestern Polytechnical University, Xi'an, China
[2]School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K.
[3]Generative AI Research & Development Center, HKGAI, Hong Kong, China
fndr_zhao@mail.nwpu.edu.cn, {sunwen, qubeijian.wang}@nwpu.edu.cn, j.he@essex.ac.uk, pengwcityu@gmail.com

Abstract

Zero-shot transfer for offline reinforcement learning involves generalizing to a wide range of tasks that are often risk-sensitive, without new interaction. We show here that although Successor Features (SFs) provide a principled framework for transfer through abstracting dynamics from rewards, their typical form is restricted to expected cumulative returns and is thus indifferent to variance in returns. We contend that such a mean-centric approach introduces bias by conflating risky and safe trajectories with identical expected outcomes, as it ignores outcome dispersion critical for risk-aware evaluation. Additionally, we note that distributional methods can be employed to model outcome distributions, but frequently do so at a high computational cost due to the need for discretization or elaborate sampling. To overcome this problem, we introduce a form of successor representation that incorporates an uncertainty envelope over states: Gaussian Distributional Successor Features (GDSF). Rather than inferring an implicit generative process, GDSF learns to approximate the distribution of cumulative outcomes with a Gaussian through recursive moment-matching. We present this framework for efficient representation of aleatoric uncertainty, without resorting to computationally expensive existing distributional methods. We also introduce Latent Outcome Optimization, an inference procedure which finds high-reward outcomes in the data support and drives a goal-conditioned policy. This dual-mode design allows the agent to transfer to non-linear objectives in a zero-shot way while preserving closed-form efficiency for linear tasks. Experiments on the D4RL benchmark show that GDSF yields better performance in risk-sensitive tasks while achieving at least competitive results in standard linear regimes.

†Corresponding author.

1 Introduction

Developing reinforcement learning agents capable of zero-shot adaptation to arbitrary objectives, ranging from simple navigation to complex, risk-sensitive constraints, remains a persistent challenge in the quest for generalist intelligence [Touati *et al.*, 2023]. To achieve such versatility, a system must structurally decouple its knowledge of environmental dynamics from specific task incentives. While Model-Based RL (MBRL) offers a theoretical avenue for such decoupling, practical implementations often rely on autoregressive rollouts that suffer from compounding errors and prohibitive inference costs [Xu *et al.*, 2022]. Successor Features (SFs) [Barreto *et al.*, 2017; Barreto *et al.*, 2018] offer a rigorous alternative by factorizing the value function into two distinct components: a predictive successor representation that summarizes future state occupancies, and a task-specific weight vector. This structural decomposition reduces the evaluation of novel rewards from a complex planning problem to a simple linear projection, enabling near-instantaneous task inference without the instability of learned simulators.

However, the computational efficiency of standard SFs relies on a restrictive linearity assumption. Conventional approaches squash the stochastic distribution of future outcomes into a single expectation vector [Barreto *et al.*, 2017]. By doing so, they act as if the cumulative future is a deterministic quantity rather than a random one. While this reduction is sufficient for linear reward specifications, it fails to account for the higher-order statistics required to evaluate non-linear or safety-constrained objectives [Bellemare *et al.*, 2017; Tamar *et al.*, 2012]. As a second-order analysis reveals—and as is corroborated by recent theoretical findings on distributional bounds [Wang *et al.*, 2024]—the true utility under such complex rewards depends crucially not only on the mean outcome but also on its dispersion. By discarding this variance information, mean-centric representations suffer from a systematic structural deficiency: they conflate highly volatile, dangerous trajectories with stable ones whenever their expected outcomes align, rendering the agent incapable of enforcing safety constraints during zero-shot adaptation.

To bridge this gap, recent research has diverged into two primary paradigms, yet each introduces significant computational or representational trade-offs. Discrete distributional methods, such as those extending C51 [Bellemare

et al., 2017] to successor representations [Gimelfarb *et al.*, 2021], attempt to capture uncertainty via categorical approximations. However, these approaches typically suffer from the curse of dimensionality: discretizing a high-dimensional outcome space incurs prohibitive parameter growth, scaling poorly to complex environments [Reinke and Alameda-Pineda, 2023]. Conversely, generative approaches like DiSPO leverage diffusion models to represent the full outcome distribution [Zhu *et al.*, 2024]. While theoretically expressive, these models rely on iterative sampling chains that impose substantial computational overhead [Janner *et al.*, 2022]. Consequently, to maintain tractable planning speeds, such methods may use simplified reward signals during the policy improvement phase. This fallback partially undermines their distributional advantages, limiting their ability to perform true zero-shot optimization against non-linear risk metrics.

We introduce Gaussian Distributional Successor Features (GDSF) to address these limitations. Instead of relying on heavy generative priors or coarse discretization, GDSF models the successor measure using a state-conditioned Gaussian approximation learned through recursive moment matching. This explicit parameterization provides a lightweight mechanism to capture aleatoric uncertainty, avoiding the computational overhead of iterative sampling models. Crucially, this structure supports a dual-mode adaptation capability: it preserves the closed-form efficiency of standard SFs for linear reward specifications, while enabling tractable Latent Outcome Optimization for non-linear, risk-aware objectives. By conditioning a policy on these optimized latent outcomes, the agent can adapt to novel safety constraints zero-shot, ensuring precise adherence to the test-time reward landscape without additional fine-tuning.

Our main contributions are:

- **Gaussian Distributional Successor Features (GDSF):** A second-order Taylor analysis shows why mean-centric SFs are structurally misaligned with non-linear reward evaluation. GDSF augments SFs with an explicit parameterization of aleatoric uncertainty, correcting this bias without incurring the inference cost of full generative models.
- **Dual-Mode Zero-Shot Adaptation:** We combine closed-form recursive moment matching with latent outcome optimization. This mechanism traverses the uncertainty envelope to identify risk-compatible outcomes during zero-shot adaptation.
- **Experimental Validation:** Experiments on D4RL benchmarks [Younis *et al.*, 2024] show that GDSF alleviates the structural bias of typical SFs. It also achieves strong performance on non-linear reward landscapes while substantially reducing inference time relative to heavier distributional or generative baselines.

2 Related Work

2.1 Zero-Shot Transfer with Successor Features

SFs allow for zero-shot adaptation by splitting environment dynamics from rewards [Barreto *et al.*, 2017]. Yet, the standard toolkit—including Universal SFs [Borsa *et al.*, 2019] and Forward-Backward representations [Touati and Ollivier, 2021]—suffers from a mean-centric bias. By betting on reward linearity, these methods effectively blind the agent to aleatoric uncertainty, leading to errors whenever risk matters. Newer distributional variants (like SFR [Reinke and Alameda-Pineda, 2023] or generative approaches [Zhu *et al.*, 2024]) do try to map the full outcome landscape. But they come with a heavy price tag: computational drag from complex parameterizations like diffusion models. This leaves a specific gap: we need a method that captures just enough uncertainty to handle risk, without carrying the computational baggage of full distributional modeling.

2.2 Distributional Reinforcement Learning

Distributional RL approximates the full return distribution rather than just its expectation [Bellemare *et al.*, 2017], leveraging algorithms like QR-DQN [Dabney *et al.*, 2018] to improve performance. Crucially, recent theory proves that learning distributions enables tighter second-order bounds that scale with the variance of the return [Wang *et al.*, 2024]. This insight strongly motivates capturing outcome dispersion. While methods like RaSF [Gimelfarb *et al.*, 2021] and DiSPO [Zhu *et al.*, 2024] extend this to successor features, they often suffer from restrictive assumptions or high sampling latency due to diffusion models. In contrast, GDSF leverages the benefits of second-order information via a lightweight Gaussian approximation, ensuring analytic efficiency for zero-shot adaptation.

3 Preliminaries

Environmental and Outcome Indicators. We view the controlled Markov process as an MDP without rewards, described by the tuple $(\mathcal{S}, \mathcal{A}, P, \gamma)$. It is natural to associate a vector-valued signal $\phi(s,a) \in \mathbb{R}^d$ with each transition. To this end, we define the cumulative discounted outcome of a policy π when starting in state s as the following random variable:

$$\mathbf{Z}^\pi(s) = \sum_{t\geq 0} \gamma^t \phi(s_t, a_t), \quad \text{where } s_0 = s,\ a_t \sim \pi. \tag{1}$$

The most straightforward approach to represent SFs is to interpret the expectation of this random variable, denoted as $\boldsymbol{\psi}^\pi(s) \triangleq \mathbb{E}[\mathbf{Z}^\pi(s)]$.

Zero-Shot Reward Specification. In zero-shot transfer, the goal is to generalize to tasks that are unseen during the training phase. At evaluation time, a task is typically specified through a utility function $f : \mathbb{R}^d \to \mathbb{R}$. The objective then becomes maximizing the expected utility:

$$J^\pi(s) = \mathbb{E}\left[f(\mathbf{Z}^\pi(s))\right]. \tag{2}$$

However, the majority of previous SF methods require that f needs to be linear (that is, $f(\mathbf{z}) = \mathbf{w}^\top \mathbf{z}$). Under this linearity assumption, the evaluation reduces to a simple dot product $J^\pi(s) = \mathbf{w}^\top \boldsymbol{\psi}^\pi(s)$. This kind of representation is typically not applicable when you have non-linear reward specifications, because it relies only on expectations and ignores the distributional information that comes from cumulative outcomes.

4 Methodology

Our goal in this work is to achieve zero-shot adaptation to non-linear rewards: once the policy $\pi(a|s, \mathbf{z})$ is trained offline, no further retraining or finetuning is required when a new task is specified at test time. We avoid modeling full distributional dynamics; instead, we preserve only the *reward-relevant* variance needed for risk-aware selection at inference time. Injecting this lightweight signal into standard successor features yields a representation that is both risk-attentive and computationally tractable.

Overview of GDSF. GDSF works in two stages. During offline learning, we train networks to predict the mean $\boldsymbol{\mu}_\theta(s)$ and variance $\boldsymbol{\sigma}^2_\theta(s)$ of cumulative outcomes using the offline dataset. At test time, when given a new reward function f, we sample outcomes from the learned Gaussian and select the one that maximizes f while staying in high-density regions. A goal-conditioned policy then executes this selected outcome. Figure 1 shows this process.

4.1 Motivation

Mean-centric SFs work well when the environment has linear rewards, but they become inadequate once the reward function f at test time is non-linear. Rather than reasoning about the full outcome distribution, they simply apply the reward function to the expected cumulative outcome:

$$J^\pi(s) \approx \hat{J}_{\text{mean}}(s) = f(\mathbb{E}[\mathbf{Z}^\pi(s)]). \tag{3}$$

This kind of simplification has drawbacks. It forces the stochastic cumulative outcome $\mathbf{Z}^\pi(s)$ to be mapped, in effect, to a deterministic point—which is just its mean. To tackle this bias properly, we need a more honest representation of which outcomes are actually attained.

Second-Order Taylor Analysis. The specific cause of this error can be identified by looking at the second-order Taylor expansion of f around the mean outcome $\boldsymbol{\mu} = \mathbb{E}[\mathbf{Z}^\pi(s)]$:

$$\mathbb{E}[f(\mathbf{Z})] \approx f(\boldsymbol{\mu}) + \underbrace{\frac{1}{2}\text{Tr}\left(\nabla^2 f(\boldsymbol{\mu})\text{Cov}(\mathbf{Z})\right)}_{\text{Missing Dispersion Term}}. \tag{4}$$

$\text{Tr}(\cdot)$ denotes the trace operator. Here, $\nabla^2 f(\boldsymbol{\mu})$ represents the Hessian of the reward function, which captures its curvature and $\text{Cov}(\mathbf{Z})$ is the covariance in cumulative outcomes. This expansion serves as a diagnostic tool for understanding locally where mean-based evaluation fails, rather than being a theoretical guarantee that holds globally.

Implications of Outcome Dispersion. Eq. (4) reveals the weakness of mean-only representations. In the linear setting ($\nabla^2 f = 0$), there is no second-order term and mean-based evaluation is exact, explaining why standard SFs perform well under linear reward specifications. More generally, for non-linear reward functions, the expected utility also depends on the variation across outcomes through the covariance term. Mean-centric approaches ignore this dependence, leading to misaligned evaluation when outcome variability is high. This motivates us to augment successor representations with lightweight uncertainty information for improved zero-shot evaluation.

4.2 Gaussian Distributional Successor Features

A simple generalization of Standard Successor Features in order to account for the dispersion in non-linear reward evaluation is provided as follows: We approximate the uncertainty over outcomes with a state-conditioned diagonal Gaussian:

$$\xi_\theta(\mathbf{z}|s) \approx \mathcal{N}\big(\boldsymbol{\mu}_\theta(s), \text{diag}(\boldsymbol{\sigma}^2_\theta(s))\big). \tag{5}$$

This Gaussian captures the uncertainty needed for risk-aware evaluation without modeling full multimodal dynamics. We use a state-conditioned distribution to separate reward evaluation from action-level control—it acts as a sufficient statistic for outcome dispersion while remaining lightweight.

Control is handled by a separate goal-conditioned policy $\pi(a|s, \mathbf{z})$, which we train on the offline dataset using standard hindsight relabeling techniques. This separation means our uncertainty model only needs to capture risk-relevant distinctions for evaluation, not generative control information.

4.3 Learning via Moment-Matching Regression

The model is trained so that it fits the first two moments of the cumulative outcome distribution. This is done via temporal-difference learning.

First Moment (Mean). The mean $\boldsymbol{\mu}_\theta(s)$ is trained by minimizing the squared Bellman error:

$$\mathcal{L}_\mu(\theta) = \mathbb{E}_\mathcal{D}\left[\|\boldsymbol{\mu}_\theta(s) - (\phi(s,a) + \gamma\boldsymbol{\mu}_{\bar{\theta}}(s'))\|^2\right]. \tag{6}$$

Second Moment (Outcome Dispersion). We focus on dispersion along the sampled trajectory rather than branching across next states, since the latter is orthogonal to our risk-aware selection objective and tends to underestimate uncertainty at poor states. Variance is therefore propagated recursively along observed transitions.

The way we implement this is by forcing the variance network $\boldsymbol{\sigma}^2_\theta(s)$ to align with a bootstrapped target distribution:

$$\mathcal{L}_{\sigma^2}(\theta) = \mathbb{E}_\mathcal{D}\left[\text{KL}\big(\mathcal{N}(\boldsymbol{y}_\mu, \boldsymbol{y}_{\sigma^2}) \,\|\, \mathcal{N}(\boldsymbol{\mu}_\theta(s), \boldsymbol{\sigma}^2_\theta(s))\big)\right], \tag{7}$$

Here, $\text{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence. where $\boldsymbol{y}_\mu = \phi(s,a) + \gamma\boldsymbol{\mu}_{\bar{\theta}}(s')$ and

$$\boldsymbol{y}_{\sigma^2} = \gamma^2\boldsymbol{\sigma}^2_{\bar{\theta}}(s') + \epsilon. \tag{8}$$

Here ϵ absorbs the irreducible transition noise. The objective yields a conservative proxy for long-horizon dispersion that is sufficient for identifying reward-sensitive regions and more stable than exhaustive distributional modeling.

4.4 Inference-Time Evaluation and Latent Outcome Optimization

When we get to evaluation time and are given a non-linear reward function f, we estimate the expected utility by sampling plausible cumulative outcomes. These outcomes need to be consistent with the uncertainty representation that we learned:

$$\hat{R}(s) = \frac{1}{K}\sum_{k=1}^{K} f(\mathbf{z}^{(k)}), \quad \mathbf{z}^{(k)} \sim \xi_\theta(\cdot|s). \tag{9}$$

The way we view inference here is as a selection problem over feasible outcomes. This is different from treating it

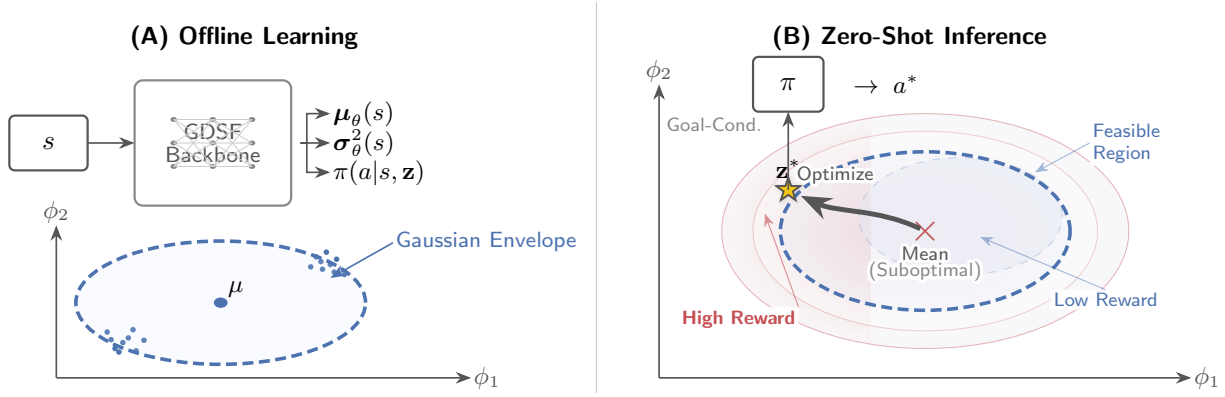


Figure 1: **Conceptual Overview of GDSF. (A) Offline Learning:** The GDSF backbone learns a state-marginal Gaussian (blue ellipse) that acts as a conservative uncertainty envelope covering the multi-modal outcomes (scattered points) from the offline dataset. Note that the mean $\boldsymbol{\mu}$ falls in a low-density region between modes, rendering it uninformative. **(B) Zero-Shot Inference:** Under a novel non-linear reward landscape (offset red rings), the mean outcome yields only mediocre value. GDSF performs latent outcome optimization to identify a high-reward outcome $\mathbf{z}^*$ within the feasible uncertainty envelope, which is then executed by the goal-conditioned policy π.

as a planning procedure or as policy optimization. To make zero-shot adaptation possible, what we do is perform latent outcome optimization. This filters for the best outcomes that are feasible:

$$\mathbf{z}^* = \arg\max_{\mathbf{z}\in\mathcal{Z}(s)} f(\mathbf{z}), \quad \mathcal{Z}(s) = \{\mathbf{z} : \log \xi_\theta(\mathbf{z}|s) \geq \delta\}. \tag{10}$$

What the constraint does is guarantee that the outcomes we choose are restricted to regions that were commonly reached in the offline dataset. This way, we avoid out-of-distribution hallucinations when doing inference. What we're essentially doing is swapping test-time policy optimization for a policy that is constrained over outcome selection instead.

This enables instantaneous zero-shot adaptation to new reward specifications without environment interaction.

4.5 Theoretical Analysis

In this section, we analyze the approximations inherent in GDSF. Our goal is to justify why the proposed uncertainty envelope, although simplified, can serve as a sufficient and conservative statistic for zero-shot reward evaluation.

Definition 1 (Conservative Outcome Envelope)**.** Let's consider the learned Gaussian approximation $\xi_\theta(\mathbf{z}|s) \triangleq \mathcal{N}(\boldsymbol{\mu}_\theta(s), \mathrm{diag}(\boldsymbol{\sigma}^2_\theta(s)))$. What we define as the *Conservative Outcome Envelope* is the set of outcomes that reside within k standard deviations of the mean:

$$\mathcal{E}_k(s) \triangleq \{\mathbf{z} : |z_i - \mu_{\theta,i}(s)| \leq k\,\sigma_{\theta,i}(s),\ \forall i\}\,. \tag{11}$$

In practical terms, this envelope delimits the specific region of the outcome space that we consider to be reliably supported by the offline dataset that we have. This element-wise constraint arises naturally from the diagonal Gaussian structure: since each outcome dimension is modeled independently, each dimension i is bounded by its own $\sigma_{\theta,i}(s)$ without conflating dimensions of different scales. In our implementation, the latent outcome optimization in Sec. 4.4 restricts $\mathbf{z}^*$ to $\mathcal{E}_k(s)$ as an axis-aligned, tractable surrogate for the density level set $\mathcal{Z}(s)$; both characterize the high-density support of ξ_θ.

Proposition 1 (Variance Lower Bound Property)**.** It's worth recalling that for the true cumulative outcome variable $\mathbf{Z}^\pi(s)$, the variance decomposes in a natural way via the Law of Total Variance:

$$\mathrm{Var}(\mathbf{Z}^\pi(s)) = \underbrace{\mathbb{E}_{s'\sim P^\pi(\cdot|s)}[\mathrm{Var}(\mathbf{Z}^\pi(s')|s')]}_{\text{Accumulated Execution Noise}} + \underbrace{\mathrm{Var}_{s'\sim P^\pi(\cdot|s)}(\mathbb{E}[\mathbf{Z}^\pi(s')|s'])}_{\text{Cross-State Branching Variance}}. \tag{12}$$

Here, $P^\pi(s'|s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[P(s'|s,a)]$ denotes the policy-induced transition kernel. Our recursive objective for $\boldsymbol{\sigma}^2_\theta$ (Eq. 7) has been designed to explicitly capture the first term, which is the accumulated dispersion. At the same time, it intentionally omits the second term, which is the cross-state branching. What this means is that the learned variance constitutes a theoretical lower bound on the total outcome variance: $\mathbb{E}[\boldsymbol{\sigma}^2_\theta(s)] \leq \mathrm{Var}(\mathbf{Z}^\pi(s))$.

This property is important because it shows that our Gaussian envelope provides a conservative abstraction of outcome uncertainty under non-linear rewards. Here "conservative" refers to anchoring inference to the high-density support (Remark 2), rather than inflating the envelope with branching variance.

To unpack this decomposition intuitively: the *first term* captures the variance attributable to stochastic execution conditional on reaching each next state—this irreducible noise is what our recursive objective (Eq. 7) is designed to track. The *second term* captures additional variance from stochastic branching across different next states—this cross-state variability is intentionally omitted by our model. Because we retain only the first term, the learned $\boldsymbol{\sigma}^2_\theta(s)$ constitutes a lower bound on total outcome variance under the true model, making our envelope conservative by construction.

Remark 1 (Implications for Evaluation)**.** What this lower bound property implies is that GDSF constructs an envelope that is tight around the anticipated trajectory. It isolates the irreducible execution noise from the multi-modal branching dynamics that can occur. When it comes to zero-shot evaluation, this is actually a feature rather than a bug. It mitigates the "average-case" bias that you get from mean-only methods. At the same time, it circumvents the excessive pessimism that would arise if we were to mistake unique branching routes for chaos that has high variance.

Remark 2 (Safeguard Against Hallucination)**.** An important point is that the latent outcome optimization $\mathbf{z}^* = \arg\max f(\mathbf{z})$ gets strictly constrained to $\mathbf{z} \in \mathcal{E}_k(s)$. What $\mathcal{E}_k(s)$ does is anchor the search to the high-density region that comes from the behavior policy. Because of this, the selected target $\mathbf{z}^*$ is in-distribution by definition. If we assume that the goal-conditioned policy $\pi(a|s, \mathbf{z})$ generalizes reasonably well within the data support (which is a standard assumption in offline RL), then this constraint effectively prevents the agent from pursuing "hallucinated" high-reward outcomes. These would be outcomes that are physically unreachable in practice.

5 Experiments

We assess GDSF on zero-shot adaptation for non-linear rewards on the D4RL benchmark. Our experiments examine the behavior of different successor representations under an evaluation-time reward shift, particularly for non-linear rewards. We consider:

(RQ1) In comparison to other methods that are mainly designed for linear reward specifications, how well can GDSF generalize to holdout non-linear rewards without retraining?

(RQ2) To what extent does adding a Gaussian outcome envelope over the successor representation improve policy evaluation when the reward function is sensitive to outcome dispersion?

(RQ3) How does GDSF perform with zero-shot reward shift, compared to offline RL and successor feature baselines?

5.1 Experimental Setup

Environments and Outcome Space. We use two continuous-control domains: Navigation (AntMaze `umaze`, `medium`, `large`) and Locomotion (MuJoCo `HalfCheetah`, `Hopper`, `Walker2d`). For the Navigation domain, we define the outcome $\phi(s, a)$ as the (x, y) coordinates that specify position. For Locomotion, the outcome includes both the forward velocity and the control costs that are incurred. Thus, $\phi(s, a) \in \mathbb{R}^2$ in both domains. Figure 2 gives a visual overview of these environments.

Test-Time Reward Specifications. To evaluate how well zero-shot adaptation works under reward transformations that happen at evaluation time, we look at three different families of test-time objectives. These families differ in what their functional form is and how they depend on cumulative outcomes:

1. Linear Reward (Base): $f(\mathbf{z}) = \mathbf{w}^\top \mathbf{z}$. This serves as a reference case. In this case, linear composition of successor features is exact, and this approach is commonly adopted when doing standard SF-based evaluation.
2. Threshold Reward (Const): $f(\mathbf{z}) = \mathbf{w}^\top \mathbf{z} \cdot \mathbb{1}[\|\mathbf{z}\|_\infty < \delta]$. Thus, the reward becomes zero whenever $\|\mathbf{z}\|_\infty \geq \delta$. Here $\mathbb{1}[\cdot]$ denotes the indicator function, equal to 1 when its argument is true and 0 otherwise. What these objectives do is impose hard constraints on the cumulative outcomes. This happens via a discontinuous feasibility region.
3. Gaussian Reward (Shape): $f(\mathbf{z}) = \exp(-\|\mathbf{z} - \mathbf{g}\|^2/\beta)$. These objectives are for precision-sensitive tasks that emphasize how close you are to a target outcome $\mathbf{g}$. The decay is smooth and is governed by the parameter β.

Baselines. We compare GDSF against baselines that fall into three categories. This comparison lets us rigorously evaluate whether modeling outcome dispersion is necessary and whether our Gaussian representation is efficient:

- **Mean-Centric SFs:** USFA [Borsa *et al.*, 2019] and FB [Touati and Ollivier, 2021], which compress the outcome distribution into a single expectation $\mathbb{E}[\phi]$, isolating the gain from explicitly capturing *outcome dispersion*.
- **Distributional & Generative SFs:** SFR [Reinke and Alameda-Pineda, 2023] (categorical discretization) and DiSPO [Zhu *et al.*, 2024] (diffusion-based planning); these test whether our lightweight Gaussian matches their expressivity at lower compute cost.
- **Standard Offline RL:** IQL [Kostrikov *et al.*, 2022], MOPO [Yu *et al.*, 2020], and COMBO [Yu *et al.*, 2021]. These model-free baselines reference standard offline performance and highlight SF decomposition flexibility.

Implementation and Inference Protocol. GDSF models the outcome distribution using a dual-head MLP that outputs the mean $\boldsymbol{\mu}_\theta(s)$ and variance $\boldsymbol{\sigma}^2_\theta(s)$ for each state.

At inference, we adapt the evaluation to the task type:

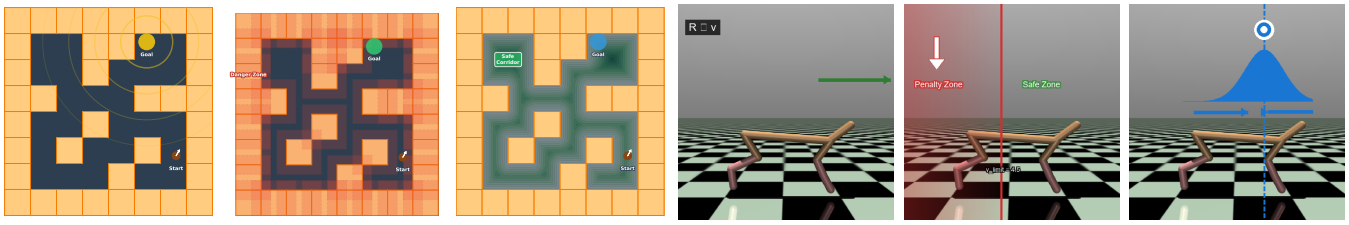
- For linear rewards, we perform standard analytic evaluation using the predicted mean.
- For non-linear rewards, we apply latent outcome optimization: we sample $K = 64$ candidate outcomes from the Gaussian envelope $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ and select the outcome $\mathbf{z}^* = \arg\max f(\mathbf{z})$. This explicitly leverages the learned uncertainty to filter out risky trajectories.

We report mean episodic returns over 10 episodes per seed, averaged across 5 random seeds.

5.2 Comparative Experiment

Table 1 reports quantitative results on the navigation and locomotion domains. We structure the analysis according to the three test-time reward families (Sec. 5.1), ordered by increasing sensitivity to outcome dispersion.

Performance on Linear Rewards (Mean-Sufficient Baseline). We first consider linear rewards, where the reward function is flat ($\nabla^2 f = 0$) and mean-based evaluation is exact. This setting verifies that GDSF introduces no unnecessary conservatism when uncertainty is irrelevant.



(a) Ant: Base (b) Ant: Const (c) Ant: Shape (d) Cheetah: Base (e) Cheetah: Const (f) Cheetah: Shape

Figure 2: **Unified visualization of all six tasks.** (a-c) AntMaze navigation tasks. (d-f) HalfCheetah locomotion tasks.

Environment	USFA†	FB†	SFR†	MOPO	IQL	COMBO	DiSPO	GDSF (Ours)
Linear Reward (sanity check)								
large-play	264±42	298±38	312±40	196±28	342±48	358±42	**402±45**	394±43
medium-play	578±62	618±54	635±51	456±42	634±58	**672±48**	658±55	645±50
umaze	612±38	638±35	648±37	508±46	652±41	**695±39**	684±43	678±40
halfcheetah-med	4876±138	5024±125	5118±132	4298±142	5087±108	**5638±94**	5612±118	5584±112
hopper-med	2954±285	3142±248	3187±265	2085±258	2812±208	**3187±145**	3165±162	3138±158
walker2d-med	3087±215	3245±188	3318±195	2894±218	**4456±172**	2987±165	4218±158	4184±152
Threshold Non-linearity (safety constraints)								
large-play	98±35	124±41	265±52	64±24	92±31	87±36	248±58	**312±52**
medium-play	248±52	287±48	512±68	145±38	215±48	194±54	486±72	**572±62**
umaze	348±48	382±42	558±62	234±43	268±42	245±50	534±68	**608±54**
halfcheetah-med	3284±285	3587±248	4587±235	3128±268	3685±252	4187±218	4512±242	**4826±228**
hopper-med	634±318	892±295	2284±412	168±148	287±235	412±208	2158±438	**2564±365**
walker2d-med	982±378	1215±325	2918±338	738±318	1124±412	865±288	2784±348	**3265±312**
Gaussian Non-linearity (precision targets)								
large-play	128±36	156±38	282±48	78±26	115±39	102±35	268±51	**308±44**
medium-play	292±55	326±50	487±62	182±45	246±56	218±43	465±65	**528±54**
umaze	378±42	418±38	518±55	264±40	294±50	268±46	498±58	**556±48**
halfcheetah-med	3645±248	3924±218	4165±228	3298±245	3872±205	**4368±182**	4052±215	4315±198
hopper-med	958±342	1284±315	2298±385	195±175	324±268	458±248	2186±405	**2418±348**
walker2d-med	1265±312	1538±285	2954±295	824±258	1218±342	912±252	2865±285	**3084±268**

Table 1: **Zero-Shot Adaptation Performance on Non-Linear Objectives.** We report the average episodic return (± std. dev.) on D4RL benchmarks across three distinct reward families defined in Sec. 5.1: Linear (sanity check), Threshold (safety constraints), and Gaussian (precision targets). While mean-centric baselines (USFA, FB) rely on linear approximations that fail to capture risk, and generative methods (DiSPO) suffer from sampling inefficiencies, GDSF achieves superior adaptation by leveraging the learned Gaussian envelope for tractable latent outcome optimization. All results are averaged over 5 random seeds.

Feature	Return	W_1
Fourier (RFF)	**85.8**	**0.13**
Learnable MLP	81.4	0.13
Random Proj.	72.3	0.15

Repr.	Return	W_1	Size
Gaussian (**Ours**)	**85.8**	**0.13**	$2d$
C51 (Cat.)	79.8	0.15	$51d$
QR (Quantile)	81.5	0.16	$51d$

Table 2: Ablation results on feature encoders and distribution representations. Left: feature encoder ablation. Right: distribution representation ablation.

As expected, GDSF achieves performance comparable to strong offline RL baselines (IQL, COMBO) and mean-centric SF methods (USFA). For instance, on `AntMaze-Large`, GDSF obtains 394 ± 43, closely matching DiSPO (402 ± 45). This confirms that the dual-mode planner correctly defaults to mean-based evaluation in linear regimes, introducing neither bias nor instability.

Threshold Non-Linearity (Safety Constraints). Threshold rewards impose sharp discontinuities whenever a constraint is violated. This makes the setting highly sensitive to outcome dispersion: two policies with the same expected outcome can yield widely different utility. We find that mean-centric SFs (USFA, FB) collapse in this setting because their structural evaluation bias makes them blind to the risk of threshold violations. For example, on `AntMaze-Large`, the performance of USFA plummets from 264 ± 42 to 98 ± 35. In contrast, GDSF achieves significantly higher returns (312 ± 52). It even scores higher than the generative baseline DiSPO (608 vs. 534 on `AntMaze-Umaze`, Threshold regime). We attribute this gain to our explicit Gaussian parameterization. While DiSPO relies on costly iterative sampling to estimate constraint violation probabilities, GDSF directly leverages the predicted variance $\boldsymbol{\sigma}_\theta^2$ in latent outcome

Method Variant	Return	Δ	Role in Framework
GDSF (Full)	**85.84** $\pm$ 1.02	-	*GDSF (Full Method)*
w/o Variance (Mean-Only)	70.12 $\pm$ 1.32	-18.3%	*Validates Structural Bias (Prop. 1)*
w/o Latent Outcome Optimization	77.81 $\pm$ 1.11	-9.4%	*Validates Active Outcome Selection*

Table 3: **Ablation of GDSF Mechanisms.** We isolate the impact of explicit variance modeling and latent outcome optimization. Removing the variance constraint (reverting to a Mean-Only baseline) causes catastrophic failure on risk-sensitive tasks, validating the structural bias analysis in Sec. 3.

optimization, enforcing constraints with greater precision and computational efficiency.

Gaussian Non-Linearity (Precision-Sensitive Objectives). Gaussian-shaped rewards impose smooth, exponential penalties for deviations from a target outcome, testing the agent's ability to minimize outcome dispersion. This setting highlights the challenges generative SFs face in high-precision, zero-shot optimization. While DiSPO models the full distribution, its sampling-based inference struggles to identify high-precision outcomes. In contrast, GDSF explicitly separates mean and variance, allowing latent outcome optimization to actively select the outcome closest to the target $\mathbf{g}$ within the learned envelope $\boldsymbol{\sigma}_\theta^2$, thereby enabling precision-aware adaptation. This results in higher returns, e.g., 2418 ± 348 on `Hopper-Medium` versus 2186 ± 405 for DiSPO.

Comparison with Fixed-Policy Baselines. Standard offline RL algorithms (IQL, MOPO) optimize policies with respect to the original training reward, and their performance under linear shifts reflects only passive re-evaluation of the original behavior (non-adaptation) under the new reward - such algorithms cannot achieve success on a new objective that requires an adapted policy. In contrast, GDSF performs active inference-time adaptation: it exploits the learned dynamics and selects the lowest cost outcome according to the Gaussian envelope, directly steering behavior to achieve higher expected rewards.

5.3 Ablation Study

We perform ablation studies to quantify the contribution of each key GDSF component. Specifically, we examine the impact of explicit Gaussian parameterization (supported by Prop. 1) and latent outcome optimization (enabling the dual-mode inference mechanism). Results (Tables 2 and 3) are obtained on `PointMaze-UMaze-v3` under a composite non-linear reward (ϕ-quadratic plus cross-dimension interaction terms) designed to stress-test the mean and variance channels of the architecture.

Feature Encoder Architecture (left side of Table 2). We analyze the effect of the state encoder. Consistent with prior work in kernel-based RL, Random Fourier Features (RFF) provide the best balance between performance (85.8) and training stability (W_1 error 0.13). Learnable MLPs achieve a lower mean return (81.4 vs. 85.8 for RFF), possibly due to spectral bias when fitting high-frequency reward landscapes. Hence, RFF is used as the default encoder.

Efficiency of Distributional Representation (right side of Table 2). A core hypothesis of GDSF is that a lightweight Gaussian approximation suffices for control, making high-dimensional categorical parameterizations unnecessary. The right side of Table 2 provides clear empirical support: the 2-parameter Gaussian representation (μ, σ^2) outperforms C51 $(51d)$ and Quantile Regression $(51d)$ baselines (85.8 vs. $79.8/81.5$). This suggests that, in continuous control, the first two moments constitute a sufficient statistic for risk-sensitive decision-making, whereas high-dimensional categorical distributions may overfit to sampling noise and reduce generalization.

Impact of Variance and Optimization Mechanisms. Table 3 presents an ablation study quantifying the specific contributions of GDSF components. We first examine the necessity of variance modeling. Removing the uncertainty envelope (*w/o Variance*) causes returns to precipitate from $\mathbf{85.84}$ to 70.12, a drop of 18.3%. This result empirically validates the structural bias analysis in Sec. 3, demonstrating that without second-moment correction, the agent fails to distinguish between stable and volatile trajectories, leading to suboptimal risk assessment.

We further analyze the benefit of latent outcome selection. Disabling this inference-time optimization (*w/o Latent Opt.*) and relying on passive sampling results in a 9.4% performance decline (77.81). This confirms that the learned Gaussian envelope functions as a *navigable feasible region*, enabling the agent to actively identify and select high-utility outcomes within the support, rather than settling for the average outcome of the underlying behavior policy.

6 Conclusion

We addressed the structural bias of mean-centric successor features by showing that they misvalue policies under non-linear reward specifications. To resolve this, we introduced GDSF, which augments standard representations with a lightweight, state-conditioned uncertainty envelope via recursive moment matching, formalized as a lower bound on total variance that isolates accumulated execution noise from branching dynamics.

This formulation supports a dual-mode inference mechanism: it preserves closed-form efficiency for linear rewards while enabling tractable Latent Outcome Optimization for risk-sensitive objectives, bypassing the computational overhead of iterative planning or generative sampling.

Future directions include extending the explicit parameterization to Mixture-of-Gaussians for branching dynamics, and leveraging the learned envelope for active risk-aware exploration in online settings.

Acknowledgements

This work was supported in part by RC Grant No. EP/Y027787/1, in part by the National Natural Science Foundation of China under Grant 62272391 and Grant 62402391, in part by the Fundamental Research Funds for the Central Universities under Grant No. D5000250275, and in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM116.

References

[Barreto *et al.*, 2017] André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, David Silver, and Hado van Hasselt. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 4055–4065, Long Beach, CA, USA, 2017. Curran Associates, Inc.

[Barreto *et al.*, 2018] André Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin Zidek, and Rémi Munos. Transfer in deep reinforcement learning using successor features and generalised policy improvement. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 501–510, Stockholm, Sweden, 2018. PMLR.

[Bellemare *et al.*, 2017] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 449–458, Sydney, Australia, 2017. PMLR.

[Borsa *et al.*, 2019] Diana Borsa, André Barreto, John Quan, Daniel Mankowitz, Hado van Hasselt, Rémi Munos, David Silver, and Tom Schaul. Universal successor features approximators. In *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.

[Dabney *et al.*, 2018] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 2892–2901, New Orleans, Louisiana, USA, 2018. AAAI Press.

[Gimelfarb *et al.*, 2021] Michael Gimelfarb, André Barreto, Scott Sanner, and Chi-Guhn Lee. Risk-aware transfer in reinforcement learning using successor features. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 17298–17310, Virtual, 2021. Curran Associates, Inc.

[Janner *et al.*, 2022] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 9902–9915, Baltimore, MD, USA, 2022. PMLR.

[Kostrikov *et al.*, 2022] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2022.

[Reinke and Alameda-Pineda, 2023] Chris Reinke and Xavier Alameda-Pineda. Successor feature representations. *Transactions on Machine Learning Research (TMLR)*, 2023.

[Tamar *et al.*, 2012] Aviv Tamar, Dotan Di Castro, and Shie Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 387–396, Edinburgh, Scotland, 2012. Omnipress.

[Touati and Ollivier, 2021] Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 13–23, Virtual, 2021. Curran Associates, Inc.

[Touati *et al.*, 2023] Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *Proceedings of the International Conference on Learning Representations (ICLR)*, Kigali, Rwanda, 2023.

[Wang *et al.*, 2024] Kaiwen Wang, Owen Oertell, Alekh Agarwal, Nathan Kallus, and Wen Sun. More benefits of being distributional: Second-order bounds for reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Vienna, Austria, 2024. PMLR.

[Xu *et al.*, 2022] Yingchen Xu, Jack Parker-Holder, Aldo Pacchiano, Philip J. Ball, Oleh Rybkin, Stephen Roberts, Tim Rocktäschel, and Edward Grefenstette. Learning general world models in a handful of reward-free deployments. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 26249–26263, New Orleans, LA, USA, 2022. Curran Associates, Inc.

[Younis *et al.*, 2024] Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K Terry. Minari. https://doi.org/10.5281/zenodo.13767625, 2024.

[Yu *et al.*, 2020] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 14129–14142, Virtual, 2020. Curran Associates, Inc.

[Yu *et al.*, 2021] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: Conservative offline model-based policy optimization. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 28954–28967, Virtual, 2021. Curran Associates, Inc.

[Zhu *et al.*, 2024] Chuning Zhu, Xinqi Wang, Tyler Han, Simon S. Du, and Abhishek Gupta. Distributional successor features enable zero-shot policy optimization. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, Vancouver, Canada, 2024. Curran Associates, Inc.