A Novel Family of Zero-Order Optimization Algorithms derived by the Bayesian Learning Rule

Sebastian Sanokowski, Xudong Sun, Majid Khadiv
Munich Institute of Robotics and Machine Intelligence
Technical University of Munich
Munich, Germany
sebastian.sanokowski@tum.de

***Abstract*—Zero-order (ZO) optimization and evolutionary algorithms are powerful tools for scenarios where objective gradients are inaccessible. Building upon contemporary literature, which often interprets ZO methods as gradient descent on a Gaussian-smoothed objective, we extend this perspective and propose a novel framework for ZO optimization by drawing inspiration from the Bayesian Learning Rule (BLR). Rather than gradient estimation on a smoothed objective, we reformulate the task as a variational inference problem aimed at minimizing the Kullback-Leibler (KL) divergence between a Gaussian distribution and a target distribution defined by the cost of the problem. This probabilistic formulation provides a principled foundation for incorporating L-2 regularization, momentum, and temperature-dependent entropy regularization into zero-order optimization. Using this framework, we derive momentum-augmented, tempered, and L-2 regularized versions of prominent algorithms, including the Cross-Entropy Method (CEM) and Model Predictive Path Integral (MPPI) control. We evaluate our proposed variants on several challenging control benchmarks, where they demonstrate significant improvements over standard baselines in terms of both convergence rate and performance.**

***Index Terms*—Optimization, Black-Box, Zero-Order, Bayesian Learning Rule, Control**

I. Introduction

Zero-order (ZO) optimization [1] and evolutionary algorithms [2] are widely applicable in scenarios where the cost function $C(x)$ can be evaluated, but its gradient is unavailable (or expensive to compute). Such settings frequently arise in robotics [3]–[5], discrete optimization [6], [7]. More recently, ZO methods have also emerged as a highly effective alternative to Reinforcement Learning (RL) for fine-tuning Large Language Models [8], [9]. While alternative learned approaches, such as RL in the context of control [10]–[12] or diffusion-based samplers in discrete optimization [13], [14], have shown promise, they remain notoriously prone to converging toward local minima and often struggle to escape these suboptimal modes once trapped. Reflecting a growing consensus in the continuous diffusion sampler literature [15]–[18], pure simulation-based algorithms often demonstrate a superior ability to explore the entirety of a complex solution space [19]. This suggests that zero-order optimization might be a particularly compelling approach for global exploration in gradient-free settings, bypassing the manifold collapses often seen in purely learned models. Notably, globally convergent methods such as Consensus-Based Optimization (CBO) [20], [21] have further underscored the potential of simulation-based approaches for robust global exploration, even in highly non-convex landscapes.

A common perspective in recent literature is to interpret ZO algorithms as gradient descent on a smoothed objective [22]. Here, the objective is defined as the expectation of the original function with respect to a Gaussian distribution, and gradients are computed with respect to the Gaussian's parameters. Intuitively, this involves finding a region in the objective landscape where the Gaussian mass fits well.

In this work, we instead draw inspiration from the Bayesian Learning Rule (BLR) [23], a framework that has been highly successful in deriving gradient-based optimizers such as SGD, momentum-based methods, and IVON (a variational variant of Adam [24]) [25], [26]. Thus, we reformulate the zero-order optimization problem as a probability approximation problem in which we aim to find a Gaussian distribution that minimizes the Kullback-Leibler (KL) divergence to a target distribution defined by $\exp(-\beta C(x))$, where $C : \mathbb{R}^D \to \mathbb{R}$. Compared to prior approaches, this formulation naturally encourages exploration via entropy regularization via a temperature parameter $\alpha = \beta^{-1}$. Furthermore, this framework provides a principled way to incorporate L_2 regularization and momentum into the optimization process.

We leverage this framework to derive momentum-augmented, tempered, and L_2-regularized versions of prominent zero-order algorithms, such as the Cross-Entropy Method (CEM) [6] and Model Predictive Path Integral (MPPI) [27]. Through extensive evaluation on standard control benchmarks, we demonstrate that our proposed variants significantly outperform these established methods in both convergence and final performance.

II. Problem Description

We consider the problem of minimizing a cost function $C : \mathbb{R}^D \to \mathbb{R}$ in a setting where the gradient $\nabla_x C(x)$ is unavailable. Instead, we optimize the surrogate objective

$$\mathcal{L}(\theta) = \mathbb{E}_{x\sim p_\theta}\big[C(x)\big], \tag{1}$$

where p_θ is a parameterized search distribution, typically chosen as a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ with parameters $\theta = (\mu, \Sigma)$. This formulation smooths the cost surface;

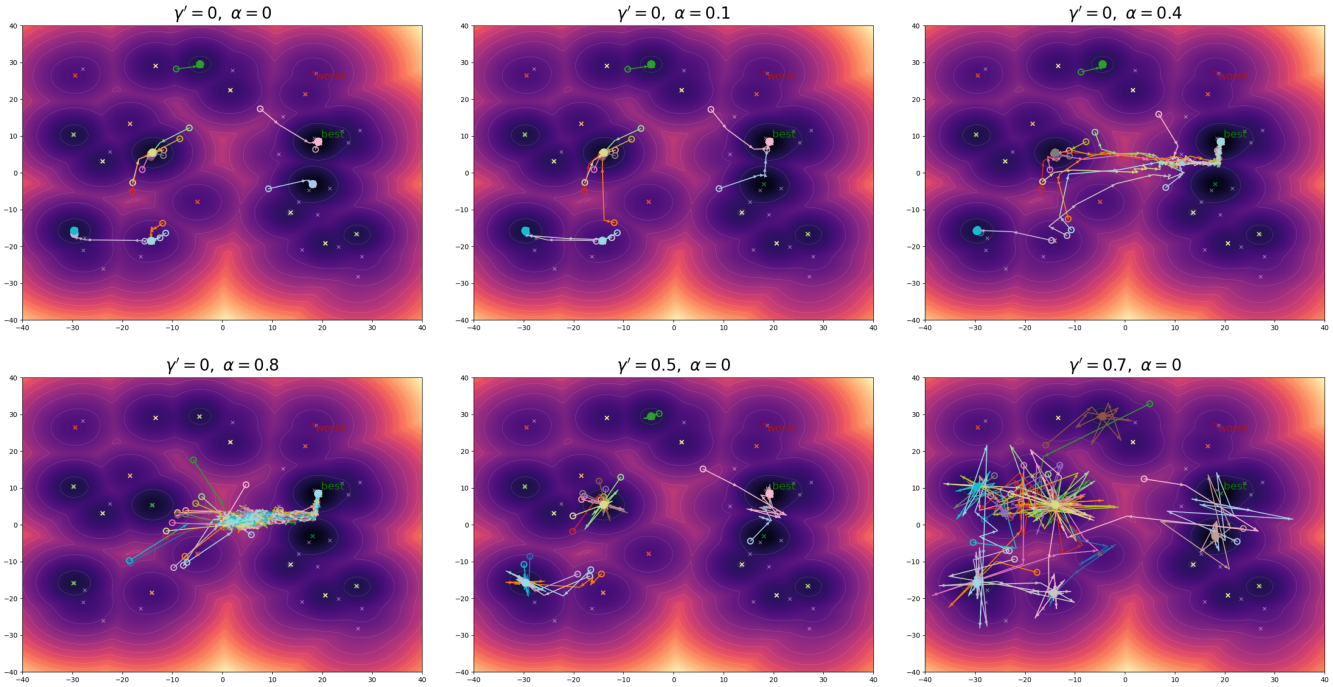

Fig. 1: Different chains of our algorithm operating on a GMM mixture optimization problem using different momentum (γ) and starting temperature (α) parameter values. For all of these plots we chose $\sigma_p = 30$.

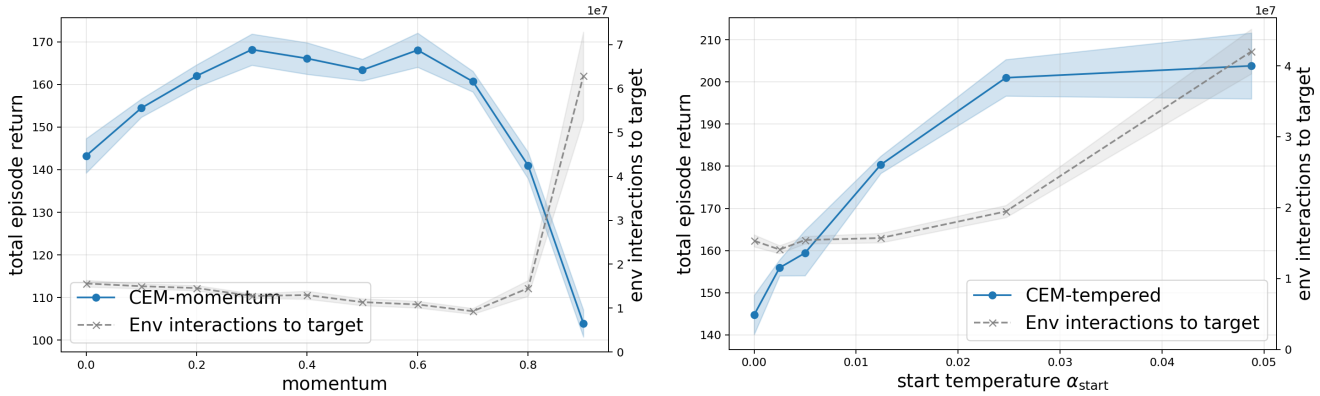

Fig. 2: **Ablation study of total episode return** on CheetahRun over 500 episode steps, varying the momentum parameter γ' (left figure) and the initial temperature parameter α_{start} (right figure). In each plot, the right axis shows the number of environment interactions required for each parameter setting to reach a target episode length of 100.

in particular, when p_θ is Gaussian, $\mathcal{L}(\theta)$ is the Gaussian convolution of $C(x)$. The gradient of the surrogate objective can be expressed using the score-function (or REINFORCE) estimator [28]:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{x\sim p_\theta}\big[C(x)\nabla_\theta \log p_\theta(x)\big]. \tag{2}$$

Let t denote the optimization step. To account for the information geometry of the parameter space, it is common to employ the proximal natural-gradient formulation as in [22]:

$$\theta_{t+1} = \operatorname*{argmin}_\theta \left[\langle \nabla_\theta \mathcal{L}(\theta_t),\, \theta - \theta_t\rangle + \frac{1}{\eta} D_{\mathrm{KL}}(p_\theta \,\|\, p_{\theta_t})\right], \tag{3}$$

where $\eta > 0$ is the step size and $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product. Intuitively, this update minimizes a first-order approximation of the objective while penalizing large changes in the search distribution through the Kullback–Leibler divergence. The resulting step is therefore small in distribution space rather than in parameter space, which yields the natural-gradient direction.

Using the second-order expansion $D_{\mathrm{KL}}(p_\theta \,\|\, p_{\theta_t}) \approx \frac{1}{2}(\theta - \theta_t)^\top F(\theta_t)(\theta - \theta_t)$, where $F(\theta_t)$ is the Fisher information matrix, one obtains the natural-gradient descent update

$$\theta_{t+1} = \theta_t - \eta F(\theta_t)^{-1}\nabla_\theta \mathcal{L}(\theta_t). \tag{4}$$

For a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, the Fisher information matrix is block diagonal with respect to μ and Σ. For the full vectorized covariance parameterization, the corresponding inverse blocks are $F_{\mu\mu}^{-1} = \Sigma$ and $F_{\Sigma\Sigma}^{-1} = 2(\Sigma \otimes \Sigma)$ [29].

Moreover, the score functions are given by

$$\nabla_\mu \log p_\theta(X) = \Sigma^{-1}(x - \mu), \tag{5}$$

$$\nabla_\Sigma \log p_\theta(X) = \frac{1}{2}\Sigma^{-1}\big[(x-\mu)(x-\mu)^\top - \Sigma\big]\Sigma^{-1}. \tag{6}$$

Substituting these expressions into (2) and applying the inverse Fisher matrix yields the following natural-gradient directions (see Ap.. A):

$$\tilde{\nabla}_\mu \mathcal{L}(\theta) = \mathbb{E}_{x\sim p_\theta}\big[C(x)(x-\mu)\big], \tag{7}$$

$$\tilde{\nabla}_\Sigma \mathcal{L}(\theta) = \mathbb{E}_{x\sim p_\theta}\big[C(x)\big((x-\mu)(x-\mu)^\top - \Sigma\big)\big]. \tag{8}$$

A. Utility Functions

As argued in [22], CEM and MPPI updates can be recovered within this natural-gradient framework by replacing the cost with an expected utility term, $C(x) \to U(C(x), \theta^*)$, where the utility function may depend on the Gaussian parameters θ^*, where $*$ denotes a stop-gradient operation, i.e. gradients are not computed with respect to the Gaussian parameters that are used within the computation of the utility function. Defining $w(x) := -U(C(x), \theta^*)$, we obtain nonnegative weights satisfying $w(x) \in [0, 1]$. For the utility functions considered below, the corresponding normalized weights satisfy $\mathbb{E}_{x\sim p_\theta}[w(x)] = 1$. Using samples $\{x_i\}_{i=1}^N \sim p_{\theta^*}$ and weights $w_i := w(x_i)$ and $\eta = 1$, Eq. 7 and Eq. 8 yield the following weighted Gaussian parameter updates

$$\mu_{t+1} = \sum_{i=1}^N w_i x_i, \quad \Sigma_{t+1} = \sum_{i=1}^N w_i (x_i - \mu_{t+1})(x_i - \mu_{t+1})^\top,$$

where $x_i \sim \mathcal{N}(\mu_t, \Sigma_t)$.

a) Cross-Entropy Method (CEM): In CEM, the utility is defined through hard thresholding based on elite samples. Let $C_{(1)} \le \cdots \le C_{(N)}$ denote the sampled costs in ascending order, and let $C_{\text{elite}} := C_{(K)}$ be the cost of the K-th best sample. Then the utility is

$$U_{\text{CEM}}(x, \theta^*) = -\frac{N}{K}\,\mathbb{I}[C(x) \le C_{\text{elite}}]\,. \tag{9}$$

By construction, $\mathbb{E}_{x\sim p_{\theta^*}}[w(x)] = 1$, and, at the sample level, the normalized weights assign equal mass to the elite set and zero mass to all remaining samples. Likewise, a utility definition for MPPI is provided in App. B.

III. Method

A. Bayesian Learning Rule for Zero-Order Optimization

Instead of minimizing the expected cost directly, we can view ZO optimization as finding a search distribution p_θ that approximates a target distribution $\pi(x)$. We define this target as a tempered posterior:

$$\pi(x) \propto \exp\left(-\beta C(x)\right) p_{\text{prior}}(x)$$

where $\beta = 1/\alpha$ is the inverse temperature and $p_{\text{prior}}(x) = \mathcal{N}(\mu_p, \tilde{\Sigma}_p)$ represents a Gaussian prior with covariance $\tilde{\Sigma}_p = \alpha\Sigma_p$ which incoporates L-2 regularization [30]. Minimizing the reverse Kullback-Leibler (KL) divergence $D_{KL}(p_\theta \,||\, \pi)$ is equivalent to minimizing the variational objective:

$$\mathcal{J}(\theta) = \beta\mathbb{E}_{x\sim p_\theta}[C(x)] + D_{KL}(p_\theta \parallel p_{\text{prior}}) - \mathbb{H}(p_\theta).$$

where we use $\mathbb{H}$ to denote the entropy of a probabilistic density function.

To incorporate momentum while respecting the information geometry of p_θ, we utilize an Accelerated Proximal Natural Gradient formulation with a momentum term $\tilde{\gamma} = \frac{\gamma}{\alpha}$ [23]:

$$\theta_{t+1} = \operatorname*{argmin}_\theta \Big[\langle \nabla_\theta \mathcal{J}(\theta_t), \theta - \theta_t\rangle + \frac{(1+\tilde{\gamma})}{\eta} D_{KL}(p_\theta || p_{\theta_t}) - \frac{\tilde{\gamma}}{\eta} D_{KL}(p_\theta || p_{\theta_{t-1}})\Big].$$

Thus, increasing $\tilde{\gamma}$ strengthens both the extrapolative effect away from $p_{\theta_{t-1}}$ and the proximal pull toward p_{θ_t}, thereby amplifying momentum while maintaining stability around the current iterate. For the Gaussian case, where $p_\theta = \mathcal{N}(\mu, \Sigma)$ and assuming only small perturbations in μ and Σ, we propose to solve the equations for the stationary condition, where $\Sigma_{t+1} = \Sigma_t$ and $\mu_{t+1} = \mu_t$, which is equivalent to setting the natural gradient to zero. With the approximations outlined in App. C, we get the following equations for the mean and variance:

$$0 = -\eta\left[\beta(\mu_t - \bar{x}) + \Sigma_t\tilde{\Sigma}_p^{-1}(\mu_t - \mu_p)\right] + \tilde{\gamma}\,(\mu_t - \mu_{t-1}) \tag{10}$$

$$0 = -\eta\left[\Sigma_t\tilde{\Sigma}_p^{-1}\Sigma_t - (2-\beta)\Sigma_t - \beta S\right] + \tilde{\gamma}\,(\Sigma_t - \Sigma_{t-1}) \tag{11}$$

where $\bar{x} := \sum_{i=1}^N w_i x_i$ and $S := \sum_{i=1}^N w_i (x_i - \mu_t)(x_i - \mu_t)^T$ with $x_i \sim \mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$ and w_i is a weighting as introduced in Sec. II-A. As Eq. 11 does not have a closed form solution in general, we will simplify it in the following section to the case where Σ and $\tilde{\Sigma}_p$ are diagonal.

1) Parameter Reconfiguration: To ensure the prior and momentum remain influential in the deterministic limit ($\alpha \to 0$), we rescale the parameters as $\beta = 1/\alpha$, $\tilde{\gamma} = \gamma/\alpha$, $\tilde{\sigma}_p^2 = \alpha\,\sigma_p^2$ and by introducing $\gamma' = \frac{\gamma}{\eta}$. Substituting these into Eq. 10 and Eq. 11 and solving for the current parameters yields the reconfigured updates (see App. C2):

$$\boxed{\mu_t = \bar{x} + \frac{1}{1 + \frac{\sigma_{t-1}^2}{\sigma_p^2} - \gamma'}\left(\frac{\sigma_{t-1}^2}{\sigma_p^2}(\mu_p - \bar{x}) + \gamma'(\bar{x} - \mu_{t-1})\right)} \tag{12}$$

and for the variance (see App. C1):

$$\boxed{\sigma_t^2 = \frac{2\left(\gamma'\sigma_{t-1}^2 + s\right)}{(\gamma' + 1 - 2\alpha) + \sqrt{(\gamma' + 1 - 2\alpha)^2 + 4\frac{(\gamma'\sigma_{t-1}^2 + s)}{\sigma_p^2}}}}. \tag{13}$$

In these equations s refers to the diagonal version of S. In order to prevent singularities in Eq. 12 $\gamma' \in]2\alpha - 1, 1 + \frac{\sigma_t^2}{\sigma_p^2}[$ must hold and in Eq. 13 singularities can be savely prevented by choosing $\alpha < \frac{1}{2} + \gamma'$.

2) Limiting Cases: The following limiting cases demonstrate how the framework recovers and extends standard algorithms:

Uninformative Prior:

As $\sigma_p^2 \to \infty$, the updates simplify to:

$$\mu_t = \bar{x} + \frac{\gamma'}{1-\gamma'}(\bar{x} - \mu_{t-1})$$
$$\sigma_t^2 = s + \frac{1}{\gamma' + 1 - 2\alpha}\left(\gamma'(\sigma_{t-1}^2 - s) + 2\alpha s\right)$$

where $\alpha < \frac{\gamma'+1}{2}$ to prevents singularities.

No Momentum:

Next, setting $\gamma' = 0$ recovers the standard weighted updates:

$$\mu_t = \bar{x}, \quad \sigma_t^2 = \frac{s}{1-2\alpha} \tag{14}$$

where $\alpha < \frac{1}{2}$. Note that when $\alpha = 0$, depending on the choice of weights w_i, this recovers the CEM for elite-set indicators or MPPI for softmax cost weighting.

a) Comparison to Conventional Momentum-based CEM Updates: While prior work has employed momentum to stabilize CEM iterations, our approach differs fundamentally in its update mechanism. Conventional momentum-based updates, such as those in [31], typically follow a standard exponential moving average (EMA) structure:

$$\mu_t = \bar{x} + \lambda_1(\mu_{t-1} - \bar{x}), \quad \sigma_t^2 = s + \lambda_2(\sigma_{t-1}^2 - s)$$

where $\lambda_1, \lambda_2 \in [0, 1]$ are smoothing factors that interpolate between the previous state and the current sample statistics. This yields a convex combination, ensuring that the updated parameters remain within the interval spanned by the previous iterate and the new observations.

In contrast, our proposed updates (for $\alpha = 0$) can be written in the innovation-based form

$$\mu_t = \bar{x} + \frac{\gamma'}{1-\gamma'}(\bar{x} - \mu_{t-1}), \quad \sigma_t^2 = s + \frac{\gamma'}{1+\gamma'}(\sigma_{t-1}^2 - s),$$

where $\gamma' \in]-1, 1[$. This representation reveals a regime-switching behavior. For $\gamma' \in]-1, 0]$, the updates behave similarly to an EMA and therefore act as a damping mechanism. In this regime, however, care must be taken since the variance update σ_t^2 is not automatically guaranteed to remain positive, which is a necessary condition for a valid variance parameterization. By contrast, for $\gamma' \in]0, 1[$, the system enters a momentum regime characterized by extrapolation, where the update is pushed in the direction of the innovation $(\bar{x} - \mu_{t-1})$.

Furthermore, unlike previous approaches in which λ_1 and λ_2 are typically tuned independently, our formulation intrinsically couples the updates of μ_t and σ_t through the single parameter γ'. Finally, our derivation exposes an asymmetric scaling: the mean update is weighted by $(1-\gamma')^{-1}$, whereas the variance update is weighted by $(1+\gamma')^{-1}$. As a result, for $\gamma' > 0$ the mean can respond more aggressively to directional trends, while the variance evolves in a comparatively more controlled and stable manner.

IV. Experiments

A. 2-D Toy Problem

We begin with a simple 2-D toy problem comprising 40 Gaussian mixtures. The position of each Gaussian is sampled uniformly within $[-40, 40]$, with a fixed variance of 3.0 and randomly sampled mixture weights. Figure 1 illustrates the performance of our algorithm on this problem for varying values of the starting temperature α_{start} and momentum γ. We employ temperature annealing [32]–[34], gradually reducing the initial temperature to zero over successive iterations. Each independent chain of the algorithm is initialized with the same standard deviation $\sigma_{\text{init}} = 6$, and the means are sampled independently from a normal distribution with zero mean and a standard deviation of 18. The results show how the momentum parameter γ and the starting temperature α_{start} affect the optimization trajectories. While both parameters encourage exploration, the temperature parameter helps the optimizer to reliably find the global optimum of the toy example.

B. Robot Control Problems

We address the problem of robot control, where the fitness function is defined as $c(A) = -\sum_{t=1}^{T} R(s_t, a_t)$, where the cost depends on a sequence of actions $A = \{a_1, \ldots, a_T\}$ and s_t representing the state at time step t.

We conduct experiments on the CheetahRun environment on the Deep Mind Control Suite [35] with an episode length of 500. The control for each time step is initialized at zero with a standard deviation of 2.2. A $\tanh$ transformation is applied before the action is fed into the environment. Results are presented in Figure 2, demonstrating that a wide range of momentum parameters in the interval $]0, 0.7]$ enhances both final performance and convergence speed (as measured by environment interactions to reach the target). For temperature, we also apply annealing and observe that increasing the starting temperature generally improves performance. However, excessively high temperatures lead to an increase in the number of environment interactions required to achieve the performance target of 100, highlighting an exploration-exploitation trade-off.

V. Conclusion and Future Work

In this work, we introduced a probabilistic framework for zero-order optimization, reformulating the problem as variational inference to minimize the KL divergence between a Gaussian and a cost-defined target distribution. This approach enables principled integration of momentum, temperature, and L-2 regularization, recovering established methods like CEM and MPPI as special cases. Our experiments demonstrate significant improvements in convergence and performance on high-dimensional control tasks. For future work, we plan to extend empirical validation on simulated and real-world robotic control tasks.

ACKNOWLEDGEMENT

This work was partially supported by the Huawei-TUM joint laboratory. This research used resources provided by the EuroHPC Joint Undertaking and hosted by IT4Innovations (Karolina), IZUM (Vega), and LuxProvide (Meluxina).

REFERENCES

[1] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*. SIAM, 2009.

[2] H.-P. P. Schwefel, *Evolution and optimum seeking: the sixth generation*. John Wiley & Sons, Inc., 1993.

[3] A. Jordana, J. Zhang, J. Amigo, and L. Righetti, "An introduction to zero-order optimization techniques for robotics," *arXiv preprint arXiv:2506.22087*, 2025.

[4] V. Dhedin, I. Taouil, S. Omar, D. Yu, K. Tao, A. Dai, and M. Khadiv, "Dynaretarget: Dynamically-feasible retargeting using sampling-based trajectory optimization," *arXiv preprint arXiv:2602.06827*, 2026.

[5] X. Sun, A. Jordana, M. Fornasier, J. Etesami, and M. Khadiv, "Consensus-based optimization (cbo): Towards global optimality in robotics," *arXiv preprint arXiv:2602.06868*, 2026.

[6] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodology and computing in applied probability*, vol. 1, no. 2, pp. 127–190, 1999.

[7] S. Reifenstein and T. Leleu, "Neural ising machines via unrolling and zeroth-order training," *arXiv preprint arXiv:2602.00302*, 2026.

[8] Y. Gan and P. Isola, "Neural thickets: Diverse task experts are dense around pretrained weights," *arXiv preprint arXiv:2603.12228*, 2026.

[9] X. Qiu, Y. Gan, C. F. Hayes, Q. Liang, Y. Xu, R. Dailey, E. Meyerson, B. Hodjat, and R. Miikkulainen, "Evolution strategies at scale: Llm fine-tuning beyond reinforcement learning," *arXiv preprint arXiv:2509.24372*, 2025.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[11] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[12] C. Voelcker, A. Brunnbauer, M. Hussing, M. Nauman, P. Abbeel, E. Eaton, R. Grosu, A.-m. Farahmand, and I. Gilitschenski, "Relative entropy pathwise policy optimization," *arXiv preprint arXiv:2507.11019*, 2025.

[13] S. Sanokowski, S. Hochreiter, and S. Lehner, "A diffusion model framework for unsupervised neural combinatorial optimization," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 21–27 Jul 2024, pp. 43 346–43 367. [Online]. Available: https://proceedings.mlr.press/v235/sanokowski24a.html

[14] S. Sanokowski, W. Berghammer, M. Ennemoser, H. P. Wang, S. Hochreiter, and S. Lehner, "Scalable discrete diffusion samplers: Combinatorial optimization and statistical physics," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=peNgxpbdxB

[15] Q. Zhang and Y. Chen, "Path integral sampler: A stochastic control approach for sampling," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://openreview.net/forum?id=_uCb2ynRu7Y

[16] F. Vargas, W. Grathwohl, and A. Doucet, "Denoising diffusion samplers," *arXiv preprint arXiv:2302.13834*, 2023.

[17] F. Vargas, S. Padhy, D. Blessing, and N. Nüsken, "Transport meets variational inference: Controlled monte carlo diffusions," in *The Twelfth International Conference on Learning Representations*, 2024.

[18] L. Richter and J. Berner, "Improved sampling via learned diffusions," in *International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=fmPpbbPjQb

[19] J. He, Y. Du, F. Vargas, D. Zhang, S. Padhy, R. OuYang, C. Gomes, and J. M. Hernández-Lobato, "No trick, no treat: Pursuits and challenges towards simulation-free training of neural samplers," *arXiv preprint arXiv:2502.06685*, 2025.

[20] J. A. Carrillo, S. Jin, L. Li, and Y. Zhu, "A consensus-based global optimization method for high dimensional machine learning problems," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 27, p. S5, 2021.

[21] M. Fornasier, H. Huang, J. Klemenc, and G. Malaspina, "From consensus-based optimization to evolution strategies: Proof of global convergence," *arXiv preprint arXiv:2602.11677*, 2026.

[22] N. Wagener, C.-A. Cheng, J. Sacks, and B. Boots, "An online learning approach to model predictive control," *arXiv preprint arXiv:1902.08967*, 2019.

[23] M. E. Khan and H. Rue, "The bayesian learning rule," *Journal of Machine Learning Research*, vol. 24, no. 281, pp. 1–46, 2023.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] W. Lin, M. Schmidt, and M. E. Khan, "Handling the positive-definite constraint in the bayesian learning rule," in *International conference on machine learning*. PMLR, 2020, pp. 6116–6126.

[26] Y. Shen, N. Daheim, B. Cong, P. Nickl, G. M. Marconi, C. Bazan, R. Yokota, I. Gurevych, D. Cremers, M. E. Khan *et al.*, "Variational learning is effective for large deep networks, 2024," *URL https://arxiv.org/abs/2402.17641*.

[27] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1433–1440.

[28] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, pp. 229–256, 1992. [Online]. Available: https://doi.org/10.1007/BF00992696

[29] T. D. Barfoot, "Multivariate gaussian variational inference by natural gradient descent," *stat*, vol. 1050, p. 27, 2020.

[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*.

[31] C. Pinneri, S. Sawant, S. Blaes, J. Achterhold, J. Stueckler, M. Rolinek, and G. Martius, "Sample-efficient cross-entropy method for real-time planning," in *Conference on Robot Learning*. PMLR, 2021, pp. 1049–1065.

[32] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.220.4598.671

[33] M. Hibat-Allah, E. M. Inack, R. Wiersema, R. G. Melko, and J. Carrasquilla, "Variational neural annealing," *Nat. Mach. Intell.*, vol. 3, no. 11, pp. 952–961, 2021. [Online]. Available: https://doi.org/10.1038/s42256-021-00401-3

[34] S. Sanokowski, W. Berghammer, S. Hochreiter, and S. Lehner, "Variational annealing on graphs for combinatorial optimization," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/c9c54ac0dd5e942b99b2b51c297544fd-Abstract-Conference.html

[35] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq *et al.*, "Deepmind control suite," *arXiv preprint arXiv:1801.00690*, 2018.

[36] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in c++," *The art of scientific computing*, vol. 2, p. 1002, 2007.

APPENDIX

A. Proximal Natural Gradient Formulation

The proximal natural gradient update is formulated as:

$$\theta_{t+1} = \operatorname{argmin} \theta \left[\langle \nabla_\theta \mathcal{L}(\theta), \theta - \theta_t \rangle + \frac{1}{\eta} D_{KL}(p_\theta \parallel p_{\theta_t}) \right], \tag{15}$$

where:

- η is the learning rate,
- $\langle \cdot, \cdot \rangle$ is the dot product,
- $D_{KL}(p_\theta \parallel p_{\theta_t})$ is the Kullback-Leibler (KL) divergence between the distributions parameterized by θ and θ_t.

This update minimizes a **first-order approximation** of the loss $\mathcal{L}(\theta)$ while penalizing changes in the model distribution through the KL divergence. The KL divergence acts as a **trust-region constraint** in distribution space, ensuring that updates remain small in an information-geometric sense.

Taylor Expansion of the KL Divergence

The KL divergence can be approximated using a second-order Taylor expansion around θ_t:

$$D_{KL}(p_\theta \parallel p_{\theta_t}) \approx \frac{1}{2}(\theta - \theta_t)^T F(\theta_t)(\theta - \theta_t), \tag{16}$$

where $F(\theta_t)$ is the **Fisher Information Matrix (FIM)** evaluated at θ_t. The FIM is defined as:

$$F(\theta) = \mathbb{E}_{x \sim p_\theta} \left[\nabla_\theta \log p_\theta(x) \nabla_\theta \log p_\theta(x)^T \right]. \tag{17}$$

Derivation of the Natural Gradient Update

Substituting the Taylor expansion into the proximal update and solving for θ yields the natural gradient update:

$$\theta_{t+1} = \theta_t - \eta F^{-1}(\theta_t) \nabla_\theta \mathcal{L}(\theta_t). \tag{18}$$

Here, $F^{-1}(\theta_t)$ acts as a **preconditioner**, transforming the gradient $\nabla_\theta \mathcal{L}(\theta_t)$ into the **natural gradient direction**. This direction accounts for the curvature of the parameter space induced by the statistical model.

Fisher Information Matrix for a Gaussian Distribution

For a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, the FIM is **block-diagonal** with respect to the mean μ and covariance Σ. The inverse blocks of the FIM are [29]:

$$F_{\mu\mu}^{-1} = \Sigma, \quad F_{\Sigma\Sigma}^{-1} = 2(\Sigma \otimes \Sigma). \tag{19}$$

- $F_{\mu\mu}^{-1} = \Sigma$ reflects the covariance of the mean parameters.
- $F_{\Sigma\Sigma}^{-1} = 2(\Sigma \otimes \Sigma)$ accounts for the covariance of the covariance parameters, where $\otimes$ is the **Kronecker product**.

Kronecker Product and Natural Gradients

The Kronecker product $\Sigma \otimes \Sigma$ is used to represent the **covariance structure** of the covariance matrix Σ. For a matrix A, the operation $(\Sigma \otimes \Sigma) \cdot \text{vec}(A)$ is equivalent to:

$$\text{vec}(\Sigma A \Sigma^T), \tag{20}$$

where $\text{vec}(A)$ is the vectorized form of A. This operation is crucial for computing the natural gradient direction for the covariance parameters.

Natural Gradient Directions

The natural gradient directions for the mean μ and covariance Σ are obtained by applying the inverse FIM to the score function gradients:

$$\nabla_\mu \log p_\theta = \Sigma^{-1}(x - \mu), \tag{21}$$

$$\nabla_\Sigma \log p_\theta = \frac{1}{2}\Sigma^{-1}[(x - \mu)(x - \mu)^T - \Sigma]\Sigma^{-1}. \tag{22}$$

The natural gradient directions are:

$$\tilde{\nabla}_\mu \mathcal{L}(\theta) = \mathbb{E}_{x \sim p_\theta}\big[C(x)(x - \mu)\big], \tag{23}$$

$$\tilde{\nabla}_\Sigma \mathcal{L}(\theta) = \mathbb{E}_{X \sim p_\theta}\big[C(x)((x - \mu)(x - \mu)^T - \Sigma)\big]. \tag{24}$$

B. Utility-weighted Gaussian update

Let

$$w(x) := -U(C(x), \theta^*),$$

so that $w(x) \geq 0$. Replacing the cost by the utility in the natural-gradient directions from (7)–(8) yields

$$\tilde{\nabla}_\mu = -\mathbb{E}_{x \sim p_{\theta^*}}\big[w(x)(x - \mu_t)\big], \tag{25}$$

$$\tilde{\nabla}_\Sigma = -\mathbb{E}_{x \sim p_{\theta^*}}\big[w(x)\big((x - \mu_t)(x - \mu_t)^\top - \Sigma_t\big)\big]. \tag{26}$$

By setting the learning rate to one, therefore arrives at

$$\mu_{t+1} = \mu_t + \mathbb{E}\big[w(x)(x)\big], \tag{27}$$

$$\Sigma_{t+1} = \Sigma_t + \mathbb{E}\big[w(x)\big((x - \mu_t)(x - \mu_t)^\top - \Sigma_t\big)\big]. \tag{28}$$

If, in addition, the weights satisfy $\mathbb{E}[w(x)] = 1$, then the mean update simplifies to

$$\mu_{t+1} = \mathbb{E}[w(x)x]. \tag{29}$$

Approximating expectations by Monte Carlo averages over samples $x_1, \ldots, x_N \sim p_{\theta^*}$ gives

$$\mu_{t+1} = \frac{1}{N}\sum_{i=1}^{N} w_i x_i, \tag{30}$$

$$\Sigma_{t+1} = \frac{1}{N}\sum_{i=1}^{N} w_i (x_i - \mu_{t+1})(x_i - \mu_{t+1})^\top, \tag{31}$$

where $w_i := w(x_i)$. Thus, if $\frac{1}{N}\sum_{i=1}^{N} w_i = 1$, the update coincides with the standard weighted empirical mean and covariance.

Note: Alternatively, these updates can also be derived as in App. C, where we solve for the stationary point, where $\Sigma_{t+1} = \Sigma_t$ and $\mu_{t+1} = \mu_t$ with the additional asssumtion that $\Sigma_t \approx \Sigma_{t-1}$. This is then equivalent to setting the gradient in Eq. 25 and in Eq. 26 to zero and solving for μ_t and Σ_t.

1) Cross-Entropy Method: For CEM, the utility is

$$U_{\text{CEM}}(x, \theta^*) = -\frac{N}{K}\,\mathbb{I}[C(x) \leq C_{\text{elite}}], \tag{32}$$

so that

$$w_i = \frac{N}{K}\,\mathbb{I}[C(x_i) \leq C_{\text{elite}}]. \tag{33}$$

Since exactly K samples belong to the elite set,

$$\frac{1}{N}\sum_{i=1}^{N} w_i = \frac{1}{N}\cdot\frac{N}{K}\cdot K = 1. \tag{34}$$

Substituting into (30)–(31) yields

$$\mu_{t+1} = \frac{1}{K}\sum_{i:\, C(x_i)\le C_{\text{elite}}} x_i, \tag{35}$$

$$\Sigma_{t+1} = \frac{1}{K}\sum_{i:\, C(x_i)\le C_{\text{elite}}} (x_i-\mu_{t+1})(x_i-\mu_{t+1})^\top. \tag{36}$$

Hence CEM recovers the empirical mean and covariance of the elite set.

2) MPPI: For MPPI, the utility is

$$U_{\text{MPPI}}(x,\theta^*) = -\frac{\exp\!\big(-\frac{1}{\lambda}C(x)\big)}{\mathbb{E}_{X\sim p_{\theta^*}}\big[\exp\!\big(-\frac{1}{\lambda}C(x)\big)\big]}, \qquad \lambda>0, \tag{37}$$

which gives the weight

$$w(x) = \frac{\exp\!\big(-\frac{1}{\lambda}C(x)\big)}{\mathbb{E}_{X\sim p_{\theta^*}}\big[\exp\!\big(-\frac{1}{\lambda}C(x)\big)\big]}. \tag{38}$$

Its expectation is

$$\mathbb{E}[w(X)] = \frac{\mathbb{E}\big[\exp\!\big(-\frac{1}{\lambda}C(x)\big)\big]}{\mathbb{E}\big[\exp\!\big(-\frac{1}{\lambda}C(x)\big)\big]} = 1. \tag{39}$$

In practice, one uses the Monte Carlo approximation

$$w_i := \frac{\exp\!\big(-\frac{1}{\lambda}C(x_i)\big)}{\frac{1}{N}\sum_{j=1}^{N}\exp\!\big(-\frac{1}{\lambda}C(x_j)\big)}, \tag{40}$$

for which $\frac{1}{N}\sum_{i=1}^{N} w_i = 1$ holds exactly. Substituting into (30)–(31) gives

$$\mu_{t+1} = \frac{\sum_{i=1}^{N}\exp\!\big(-\frac{1}{\lambda}C(x_i)\big)\,x_i}{\sum_{j=1}^{N}\exp\!\big(-\frac{1}{\lambda}C(x_j)\big)}, \tag{41}$$

$$\Sigma_{t+1} = \frac{\sum_{i=1}^{N}\exp\!\big(-\frac{1}{\lambda}C(x_i)\big)\,(x_i-\mu_{t+1})(x_i-\mu_{t+1})^\top}{\sum_{j=1}^{N}\exp\!\big(-\frac{1}{\lambda}C(x_j)\big)}. \tag{42}$$

Thus, MPPI corresponds to a soft, exponentially weighted update of the Gaussian mean and covariance.

C. Derivation of Zero-Order Accelerated Natural Gradient Updates

We aim to minimize the composite loss function:

$$\mathcal{L} = D_{KL}(p\|q) + D_{KL}(p\|\pi)$$

where the target distribution is $\pi(x)\propto\exp(-\beta C(x))$, and the variational distributions are Gaussian:

$$p(x) = \mathcal{N}(\mu_t,\Sigma_t), \quad q(x) = \mathcal{N}(\mu_p,\tilde{\Sigma}_p),$$

where $\tilde{\Sigma}_p = \alpha\Sigma_p$ We define the weighted empirical statistics as:

$$\bar{x} = \sum_{i=1}^{N} w_i x_i, \quad S = \sum_{i=1}^{N} w_i (x_i-\mu_t)(x_i-\mu_t)^T$$

where w_i is as in App. B and we sample $x_i\sim\mathcal{N}(\mu_{t-1},\Sigma_{t-1})$ by using the following approximations:

Instead of using samples from the current Gaussian search distribution p_{θ_t}, we use samples from the previous iterate $p_{\theta_{t-1}}$ instead by introducing importance weights. Formally, for any $f(x)$,

$$\mathbb{E}_{x\sim p_{\theta_t}}[f(x)] = \mathbb{E}_{x\sim p_{\theta_{t-1}}}\left[\frac{p_{\theta_t}(x)}{p_{\theta_{t-1}}(x)}f(x)\right].$$

Assuming consecutive parameter updates are small, $p_{\theta_t}\approx p_{\theta_{t-1}}$, so the importance ratio is close to one. We therefore approximate

$$\mathbb{E}_{x\sim p_{\theta_t}}[f(x)] \approx \mathbb{E}_{x\sim p_{\theta_{t-1}}}[f(x)],$$

and treat the empirical moments $\bar{x}$ and S as fixed when solving for (μ_t,Σ_t). This gives a one-step-lag approximation and leads to explicit plug-in updates.

1) Gradient with Respect to Σ_t: The Kullback-Leibler divergence between two Gaussians is given by:

$$D_{KL}(p\|q) = \frac{1}{2}\Big[\log\frac{\det\tilde{\Sigma}_p}{\det\Sigma_t} - d + \operatorname{tr}(\tilde{\Sigma}_p^{-1}\Sigma_t) + (\mu_p-\mu_t)^\top\tilde{\Sigma}_p^{-1}(\mu_p-\mu_t)\Big].$$

Using standard matrix derivatives for symmetric matrices, $\frac{\partial\log\det X}{\partial X} = X^{-1}$ and $\frac{\partial\operatorname{tr}(AX)}{\partial X} = A$, we obtain:

$$\frac{\partial D_{KL}(p\|q)}{\partial\Sigma} = \frac{1}{2}\big(\tilde{\Sigma}_p^{-1} - \Sigma_t^{-1}\big).$$

The natural gradient of the second term, $D_{KL}(p\|\pi)$, is:

$$\tilde{\nabla}_\Sigma D_{KL}(p\|\pi) = \beta\left(\mathbb{E}[C(x)(x-\mu_t)(x-\mu_t)^T] - \mathbb{E}[C(x)]\Sigma_t\right) - \Sigma_t.$$

Combining these and applying the Fisher Information Matrix (FIM) transformation, the total natural gradient w.r.t. Σ is:

$$\tilde{\nabla}_\Sigma\mathcal{L} = \Sigma_t\tilde{\Sigma}_p^{-1}\Sigma_t + \beta\mathbb{E}\left[C(x)(x-\mu_t)(x-\mu_t)^T - \mathbb{E}[C(x)]\Sigma_t\right] - 2\Sigma_t.$$

For the utility functions as in App. B we have $\mathbb{E}[C(x)] = -1$ and substituting the sample covariance $S\approx -\mathbb{E}[C(x)(x-\mu_t)(x-\mu_t)^T]$, we have:

$$\tilde{\nabla}_\Sigma\mathcal{L} = \Sigma_t\tilde{\Sigma}_p^{-1}\Sigma_t + \beta S + (\beta-2)\Sigma_t.$$

Incorporating momentum with learning rate η and momentum coefficient γ, the update equation at the stationary point (i.e $\mu_t=\mu_{t+1}$ and $\Sigma_t=\Sigma_{t+1}$) is (see App. D for more details on the momentum updates):

$$0 = -\eta\left(\Sigma_t\tilde{\Sigma}_p^{-1}\Sigma_t + (\beta-2)\Sigma_t - \beta S\right) + \gamma(\Sigma_t-\Sigma_{t-1}).$$

In the diagonal case ($\Sigma_t\to\sigma_t^2$, $\tilde{\Sigma}_p\to\tilde{\sigma}_p^2$), this simplifies to a quadratic form:

$$0 = -\eta\frac{\sigma_t^4}{\tilde{\sigma}_p^2} + \eta(2-\beta)\sigma_t^2 + \eta\beta s^2 + \gamma(\sigma_t^2-\sigma_{t-1}^2).$$

To solve for σ_t^2, we use the *Citardauq Formula* [36] (a numerically stable variant of the quadratic formula) $x = \frac{2c}{-b\mp\sqrt{b^2-4ac}}$. Mapping our terms to $ax^2+bx+c=0$ with $x=\sigma_t^2$:

$$a = \frac{1}{\tilde{\sigma}_p^2}, \quad b = \frac{\gamma}{\eta}+\beta-2, \quad c = -\left(\frac{\gamma}{\eta}\sigma_{t-1}^2 + \beta s^2\right).$$

Selecting the positive root to ensure a valid variance, we arrive at:

$$\sigma_t^2 = \frac{2\left(\frac{\gamma}{\eta}\sigma_{t-1}^2 + \beta s^2\right)}{\left(\frac{\gamma}{\eta} + \beta - 2\right) + \sqrt{\left(\frac{\gamma}{\eta} + \beta - 2\right)^2 + \frac{4}{\tilde{\sigma}_p^2}\left(\frac{\gamma}{\eta}\sigma_{t-1}^2 + \beta s^2\right)}} \tag{43}$$

With substituting back $\tilde{\gamma} = \frac{\gamma}{\alpha}$, $\tilde{\sigma}_p^2 = \alpha\,\sigma_p^2$ and substituting $\gamma' = \frac{\gamma}{\eta}$ we have:

$$\boxed{\sigma_t^2 = \frac{2\left(\gamma'\sigma_{t-1}^2 + s\right)}{(\gamma' + 1 - 2\alpha) + \sqrt{(\gamma' + 1 - 2\alpha)^2 + 4\frac{(\gamma'\sigma_{t-1}^2 + s)}{\sigma_p^2}}}}. \tag{44}$$

When $\sigma_p \to \infty$ this formula further simplifies in the following way:

$$\begin{aligned}\sigma_t^2 &= \frac{\gamma'\sigma_{t-1}^2 + s}{\gamma' + 1 - 2\alpha} \\ &= \frac{(\gamma' + 1 - 2\alpha)\,s - (\gamma' - 2\alpha)\,s + \gamma'\sigma_{t-1}^2}{\gamma' + 1 - 2\alpha}\end{aligned}$$

Which leads to the final update formula:

$$\boxed{\sigma_t^2 = s + \frac{1}{\gamma' + 1 - 2\alpha}\left(\gamma'(\sigma_{t-1}^2 - s) + 2\alpha s\right)}$$

2) Gradient with Respect to μ: The natural gradient of the loss with respect to μ is:

$$\tilde{\nabla}_\mu \mathcal{L} = \beta\mathbb{E}[C(x)(x - \mu_t)] - \Sigma_t\tilde{\Sigma}_p^{-1}(\mu_p - \mu_t).$$

Including momentum, replacing Σ_t with Σ_{t-1}, as $\Sigma_t \approx \Sigma_{t-1}$ and setting the update to zero for the equilibrium point (see App. D for more details on the momentum updates):

$$-\eta\left[\beta(\mu_t - \bar{x}) - \Sigma_{t-1}\tilde{\Sigma}_p^{-1}(\mu_p - \mu_t)\right] + \gamma(\mu_t - \mu_{t-1}) = 0$$

where we used $\bar{x} \approx -\mathbb{E}[C(x)x]$ and $\mathbb{E}[C(x)] = -1$. Solving for μ_t:

$$\mu_t = \left[(\gamma - \eta\beta)I - \eta\Sigma_{t-1}\tilde{\Sigma}_p^{-1}\right]^{-1}\left(\gamma\mu_{t-1} - \eta\beta\bar{x} - \eta\Sigma_{t-1}\tilde{\Sigma}_p^{-1}\mu_p\right). \tag{45}$$

In the diagonal case, this simplifies to the following element-wise update:

$$\mu_t = \frac{\frac{\gamma}{\eta}\mu_{t-1} - \beta\bar{x} - \frac{\sigma_{t-1}^2}{\sigma_p^2}\mu_p}{\frac{\gamma}{\eta} - \left(\beta + \frac{\sigma_{t-1}^2}{\sigma_p^2}\right)}. \tag{46}$$

Then substituting back $\tilde{\gamma} = \frac{\gamma}{\alpha}$, $\tilde{\sigma}_p^2 = \alpha\,\sigma_p^2$ and $\gamma' = \frac{\gamma}{\eta}$ we get:

$$\mu_t = \frac{\gamma'\mu_{t-1} - \bar{x} - \frac{\sigma_t^2}{\sigma_p^2}\mu_p}{\gamma' - \left(1 + \frac{\sigma_t^2}{\sigma_p^2}\right)} \tag{47}$$

This equation can also be written as:

$$\begin{aligned}\mu_t &= \frac{\bar{x} + \frac{\sigma_{t-1}^2}{\sigma_p^2}\mu_p - \gamma'\mu_{t-1}}{1 + \frac{\sigma_{t-1}^2}{\sigma_p^2} - \gamma'} \\ &= \frac{(1 + \frac{\sigma_{t-1}^2}{\sigma_p^2} - \gamma')\bar{x} - (\frac{\sigma_{t-1}^2}{\sigma_p^2} - \gamma')\bar{x} + \frac{\sigma_{t-1}^2}{\sigma_p^2}\mu_p - \gamma'\mu_{t-1}}{1 + \frac{\sigma_{t-1}^2}{\sigma_p^2} - \gamma'}\end{aligned}$$

Which leads to the final update formula:

$$\boxed{\mu_t = \bar{x} + \frac{1}{1 + \frac{\sigma_{t-1}^2}{\sigma_p^2} - \gamma'}\left(\frac{\sigma_{t-1}^2}{\sigma_p^2}(\mu_p - \bar{x}) + \gamma'(\bar{x} - \mu_{t-1})\right)}$$

D. Derivation of the Gaussian momentum updates from the KL-proximal objective.

We derive the momentum terms directly from the Gaussian KL divergence, without using exponential-family duality. Let

$$q(\mu, \Sigma) = \mathcal{N}(\mu, \Sigma), \quad q_t = \mathcal{N}(\mu_t, \Sigma_t), \quad q_{t-1} = \mathcal{N}(\mu_{t-1}, \Sigma_{t-1}).$$

The accelerated proximal update is defined by

$$\theta_{t+1} = \arg\min_{\mu,\Sigma}\Big[\langle\nabla\mathcal{J}(\theta_t), \theta - \theta_t\rangle + \frac{1 + \tilde{\gamma}}{\eta}D_{\mathrm{KL}}(q(\theta)\|q_t) \tag{48}$$

$$- \frac{\tilde{\gamma}}{\eta}D_{\mathrm{KL}}(q(\theta)\|q_{t-1})\Big]. \tag{49}$$

For two Gaussian densities

$$q = \mathcal{N}(\mu, \Sigma), \qquad q' = \mathcal{N}(\mu', \Sigma'),$$

the reverse KL divergence is

$$D_{\mathrm{KL}}(q\|q') = \frac{1}{2}\Big[\log\frac{|\Sigma'|}{|\Sigma|} - d + \mathrm{tr}\big((\Sigma')^{-1}\Sigma\big) \tag{50}$$

$$+ (\mu' - \mu)^T(\Sigma')^{-1}(\mu' - \mu)\Big]. \tag{51}$$

a) Mean update.: From (51), the terms depending on μ are

$$\frac{1 + \tilde{\gamma}}{2\eta}(\mu_t - \mu)^T\Sigma_t^{-1}(\mu_t - \mu) - \frac{\tilde{\gamma}}{2\eta}(\mu_{t-1} - \mu)^T\Sigma_{t-1}^{-1}(\mu_{t-1} - \mu).$$

Using

$$\nabla_\mu\frac{1}{2}(\mu - a)^T M(\mu - a) = M(\mu - a) \quad \text{for symmetric } M,$$

the first-order condition of (49) with respect to μ becomes

$$0 = \nabla_\mu\mathcal{J}(\theta_t) + \frac{1 + \tilde{\gamma}}{\eta}\Sigma_t^{-1}(\mu_{t+1} - \mu_t) - \frac{\tilde{\gamma}}{\eta}\Sigma_{t-1}^{-1}(\mu_{t+1} - \mu_{t-1}). \tag{52}$$

This equation is exact. To obtain a simple momentum form, we use the local proximal approximation that consecutive covariances are close,

$$\Sigma_{t+1} \approx \Sigma_t \approx \Sigma_{t-1}, \tag{53}$$

so that in particular

$$\Sigma_t \Sigma_{t-1}^{-1} \approx I.$$

Multiplying (52) by Σ_t yields

$$0 = \Sigma_t \nabla_\mu \mathcal{J}(\theta_t) + \frac{1+\tilde{\gamma}}{\eta}(\mu_{t+1} - \mu_t) - \frac{\tilde{\gamma}}{\eta}\Sigma_t \Sigma_{t-1}^{-1}(\mu_{t+1} - \mu_{t-1}).$$

Using (53) and the decomposition

$$\mu_{t+1} - \mu_{t-1} = (\mu_{t+1} - \mu_t) + (\mu_t - \mu_{t-1}),$$

we obtain

$$0 \approx \Sigma_t \nabla_\mu \mathcal{J}(\theta_t) + \frac{1+\tilde{\gamma}}{\eta}(\mu_{t+1} - \mu_t) - \frac{\tilde{\gamma}}{\eta}\Big[(\mu_{t+1} - \mu_t) + (\mu_t - \mu_{t-1})\Big].$$

Combining terms gives

$$0 \approx \Sigma_t \nabla_\mu \mathcal{J}(\theta_t) + \frac{1}{\eta}(\mu_{t+1} - \mu_t) - \frac{\tilde{\gamma}}{\eta}(\mu_t - \mu_{t-1}),$$

or equivalently

$$\mu_{t+1} - \mu_t \approx -\eta\, \Sigma_t \nabla_\mu \mathcal{J}(\theta_t) + \tilde{\gamma}(\mu_t - \mu_{t-1}). \tag{54}$$

Recognizing the Gaussian natural gradient for the mean block,

$$\tilde{\nabla}_\mu \mathcal{J}(\theta_t) = \Sigma_t \nabla_\mu \mathcal{J}(\theta_t),$$

we can rewrite (54) as

$$\mu_{t+1} - \mu_t \approx -\eta\, \tilde{\nabla}_\mu \mathcal{J}(\theta_t) + \tilde{\gamma}(\mu_t - \mu_{t-1}). \tag{55}$$

b) Covariance update.: From (51), the terms depending on Σ are

$$\frac{1+\tilde{\gamma}}{2\eta}\Big[-\log|\Sigma| + \mathrm{tr}(\Sigma_t^{-1}\Sigma)\Big] - \frac{\tilde{\gamma}}{2\eta}\Big[-\log|\Sigma| + \mathrm{tr}(\Sigma_{t-1}^{-1}\Sigma)\Big].$$

Using the standard identities

$$\nabla_\Sigma \log|\Sigma| = \Sigma^{-1}, \qquad \nabla_\Sigma \mathrm{tr}(A\Sigma) = A^T,$$

and the symmetry of Σ_t^{-1} and Σ_{t-1}^{-1}, the first-order condition of (49) with respect to Σ becomes

$$0 = \nabla_\Sigma \mathcal{J}(\theta_t) + \frac{1+\tilde{\gamma}}{2\eta}\big(\Sigma_t^{-1} - \Sigma_{t+1}^{-1}\big) - \frac{\tilde{\gamma}}{2\eta}\big(\Sigma_{t-1}^{-1} - \Sigma_{t+1}^{-1}\big). \tag{56}$$

Again, this equation is exact. To obtain the momentum form, we linearize the inverse locally around Σ_t. Under the proximal approximation (53), one may use the first-order relation

$$\Sigma_{t+1}^{-1} - \Sigma_t^{-1} \approx -\Sigma_t^{-1}(\Sigma_{t+1} - \Sigma_t)\Sigma_t^{-1},$$

and similarly

$$\Sigma_{t-1}^{-1} - \Sigma_t^{-1} \approx -\Sigma_t^{-1}(\Sigma_{t-1} - \Sigma_t)\Sigma_t^{-1}.$$

Substituting these approximations into (56) and multiplying from the left and right by Σ_t gives

$$0 \approx \Sigma_t \nabla_\Sigma \mathcal{J}(\theta_t)\Sigma_t + \frac{1+\tilde{\gamma}}{2\eta}(\Sigma_{t+1} - \Sigma_t) - \frac{\tilde{\gamma}}{2\eta}(\Sigma_{t+1} - \Sigma_{t-1}).$$

Using

$$\Sigma_{t+1} - \Sigma_{t-1} = (\Sigma_{t+1} - \Sigma_t) + (\Sigma_t - \Sigma_{t-1}),$$

we obtain

$$0 \approx \Sigma_t \nabla_\Sigma \mathcal{J}(\theta_t)\Sigma_t + \frac{1+\tilde{\gamma}}{2\eta}(\Sigma_{t+1} - \Sigma_t) - \frac{\tilde{\gamma}}{2\eta}\Big[(\Sigma_{t+1} - \Sigma_t) + (\Sigma_t - \Sigma_{t-1})\Big].$$

Combining terms yields

$$0 \approx \Sigma_t \nabla_\Sigma \mathcal{J}(\theta_t)\Sigma_t + \frac{1}{2\eta}(\Sigma_{t+1} - \Sigma_t) - \frac{\tilde{\gamma}}{2\eta}(\Sigma_t - \Sigma_{t-1}),$$

and therefore

$$\Sigma_{t+1} - \Sigma_t \approx -2\eta\, \Sigma_t \nabla_\Sigma \mathcal{J}(\theta_t)\Sigma_t + \tilde{\gamma}(\Sigma_t - \Sigma_{t-1}). \tag{57}$$

Recognizing the Gaussian natural gradient for the covariance block,

$$\tilde{\nabla}_\Sigma \mathcal{J}(\theta_t) = 2\Sigma_t \nabla_\Sigma \mathcal{J}(\theta_t)\Sigma_t,$$

this becomes

$$\Sigma_{t+1} - \Sigma_t \approx -\eta\, \tilde{\nabla}_\Sigma \mathcal{J}(\theta_t) + \tilde{\gamma}(\Sigma_t - \Sigma_{t-1}). \tag{58}$$

c) Result.: Thus, by differentiating the Gaussian KL terms directly and using the local approximation $\Sigma_{t+1} \approx \Sigma_t \approx \Sigma_{t-1}$, the accelerated KL-proximal objective yields the approximate momentum updates

$$\mu_{t+1} - \mu_t \approx -\eta\, \tilde{\nabla}_\mu \mathcal{J}(\theta_t) + \tilde{\gamma}(\mu_t - \mu_{t-1}), \tag{59}$$

$$\Sigma_{t+1} - \Sigma_t \approx -\eta\, \tilde{\nabla}_\Sigma \mathcal{J}(\theta_t) + \tilde{\gamma}(\Sigma_t - \Sigma_{t-1}). \tag{60}$$

Hence, the momentum terms arise from the difference of the two Gaussian KL proximal terms: the first term pulls the new iterate toward the current distribution q_t, while the second term pushes it away from the previous distribution q_{t-1}.