Design of Physics Experiment via Collision-Free Latent Space Optimization

Fengxue Zhang, Yair Altas, Louise Fan, Kaustubh Vinchure, Brian Nord, Yuxin Chen

University of Chicago

Motivating Applications

Are you having trouble optimizing your scientific experiments?

Is it because your input space is *too complex*?

Is the representation learned *losing too much information*?

Figure 1: Left: Robotic telescope; Right: Dark Energy Survey

Latent Space Optimization

$$\mathcal{X} \xrightarrow{\text{latent space mapping } g} \mathcal{Z} \xrightarrow{\text{objective mapping } h} \mathbb{R}$$

- A *latent space mapping* g to project input $\mathcal{X}$ to a latent space $\mathcal{Z}$
- An *objective mapping* $h : \mathcal{Z} \to \mathbb{R}$ such that $f(x) \approx h(g(x))$
- $h : \mathcal{Z} \to \mathbb{R}$ often modeled by *Gaussian Processes*
- [Lu+18] train a VAE during sequential optimization
- [TDHL20] periodically retrain the VAE to improve latent space

The Collision Effect

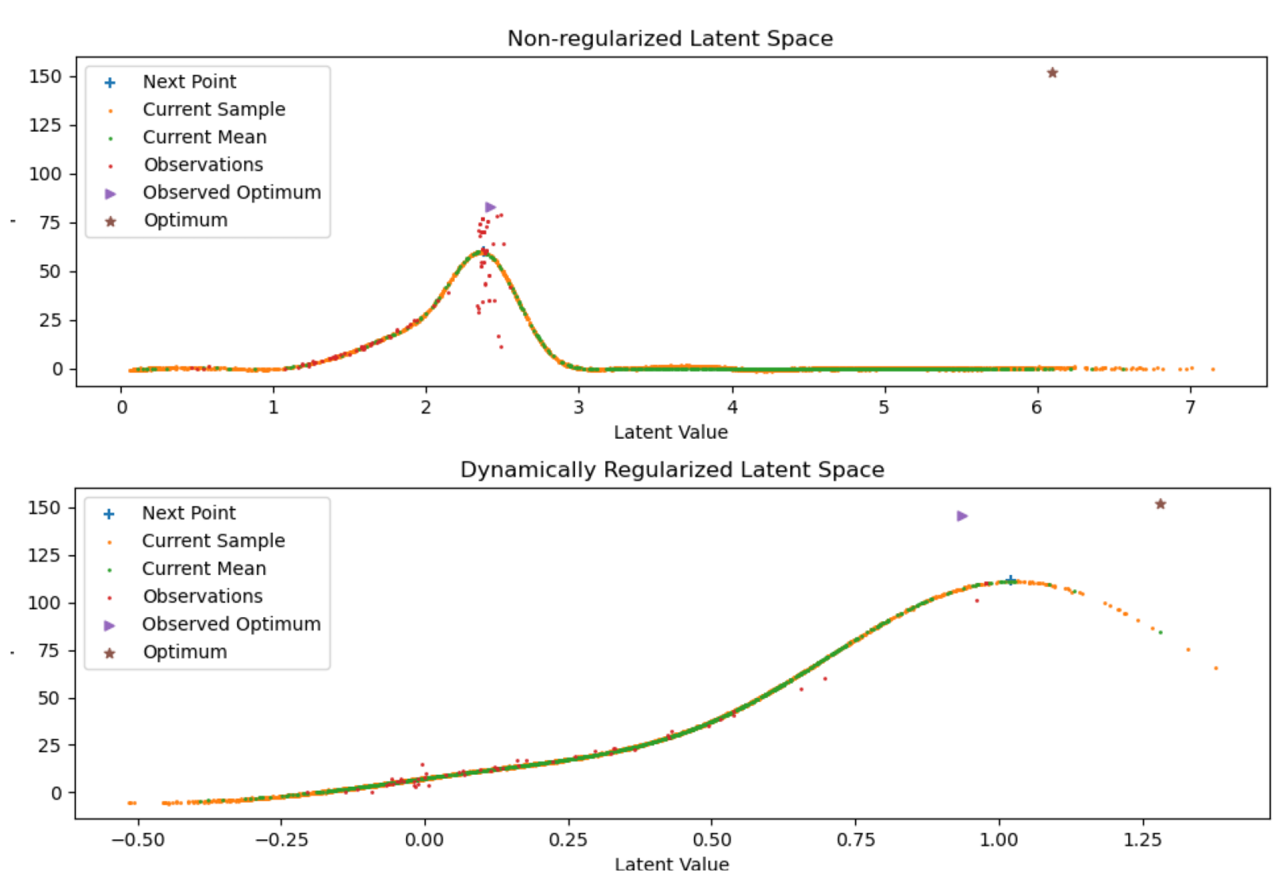


Fig. 2: Collision on Feynman Dataset

Demonstration of the *collision effect*: the regularized and non-regularized 1D latent space learned on the Feynman 6D dataset.

The Collision Penalty

For $x_i, x_j \in \mathcal{X}$, $y_i = f(x_i) + \epsilon$, $y_i = f(x_i) + \epsilon$ are the corresponding observations, and $z_i = g(x_i)$, $z_j = g(x_j)$ are the latent space representations. We define the *collision penalty* as

$$p_{ij} = \max(\lambda |y_i - y_j| - |z_i - z_j|, 0) \tag{1}$$

λ is a *penalty parameter* that controls the smoothness of the target function $h : \mathcal{Z} \to \mathbb{R}$.

Optimization-Aware Dynamic Weighting

For any pair $((x_j, z_j, y_j), (x_i, z_i, y_i))$ in a batch of observation pairs D_t we define the *importance-weighted penalty function* as

$$\tilde{p}_{ij} = p_{ij} \cdot \frac{e^{\gamma(y_i + y_j)}}{\sum_{(m,n) \in D_t} e^{\gamma(y_m + y_n)}}. \tag{2}$$

γ is *importance weight* that controls the aggressiveness of the weighting strategy.

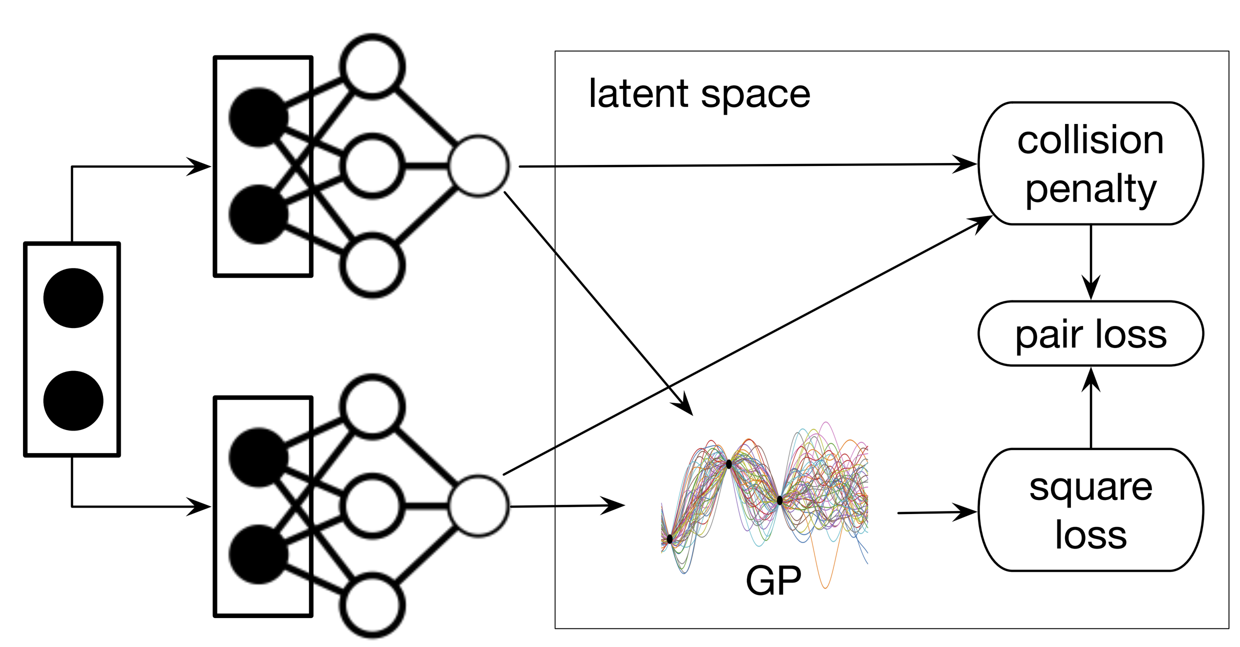
Pair Loss for Kernel Learning

Combining the *collision penalty* and *regression loss* of GP, we define the *pair loss* function L as

$$L_{\rho,\lambda,\gamma}(M_t, K_t, D_t) = \frac{1}{||D_t||^2} \sum_{i,j \in D_t} (GP_{K_t}(M_t(x_i)) - y_i)^2 + (GP_{K_t}(M_t(x_j)) - y_j)^2 + \rho \tilde{p}_{ij}, \tag{3}$$

- $GP_{Kt}(M_t(x_i))$: GP's posterior mean on x_i with kernel K_t and neural network M_t at timestep t.
- ρ: the regularization weight.

CoFLO: Collision-Free Latent Space Optimization



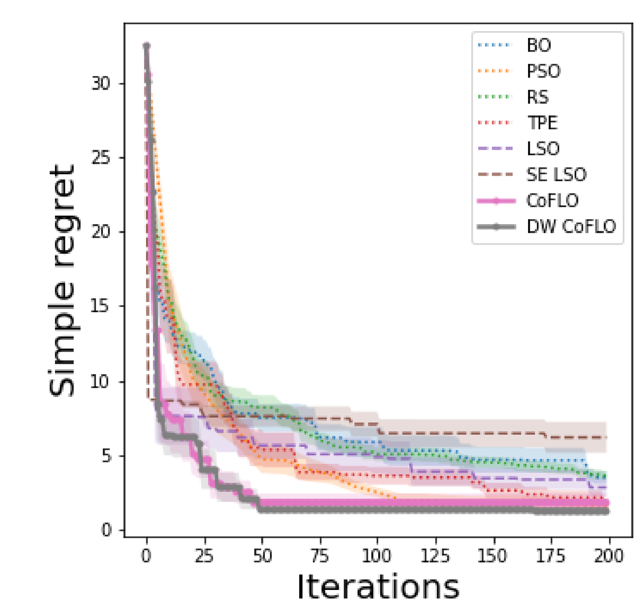
CoFLO concurrently feeds the *pair-wise input* into the same network to calculate the collision penalty, then combines it with the square loss of GP to calculate the *pair loss* function.

Algorithmic Details

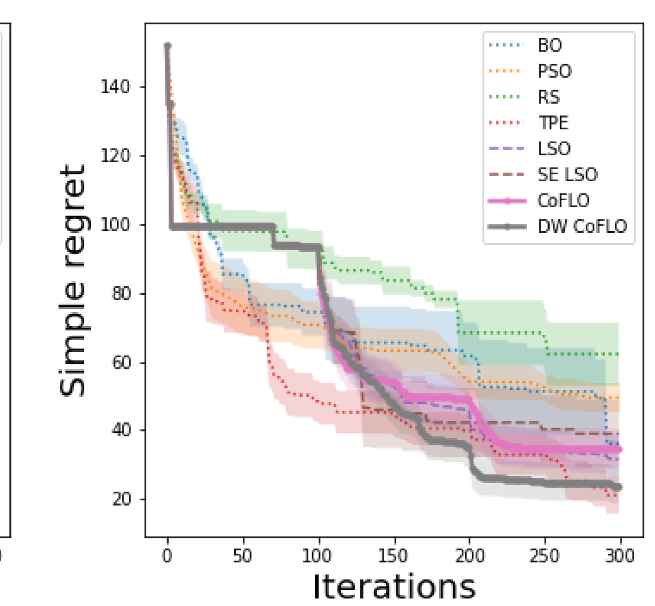
Algorithm 1 Collision-Regularized Latent Space Optimization (CoFLO)

Input: Regularization weight ρ (cf. Equation 3), penalty parameter λ (cf. Equation 1), retrain interval $\tilde{T}$, importance weight parameter γ (cf. Equation 2), neural network M_0, base kernel K_0, prior mean μ_0, total time steps T;
for $t = 1$ *to* T **do**
  $x_t \leftarrow \arg\max_{x \in D} \alpha(M_t(x))$ ▷ *maximize acquisition function*
  $y_t \leftarrow$ evaluation on x_t ▷ *update observation*
  if $t \equiv 0 \pmod{\tilde{T}}$ **then**
    $M_{t+1}, K_{t+1} \leftarrow$ retrain M_t and K_t with the pair loss function $L_{\rho,\lambda,\gamma}(M_t, K_t, D_t)$ as defined in equation 3 ▷ *periodical retrain*
  end if
end for
Output: $\max_t y_t$

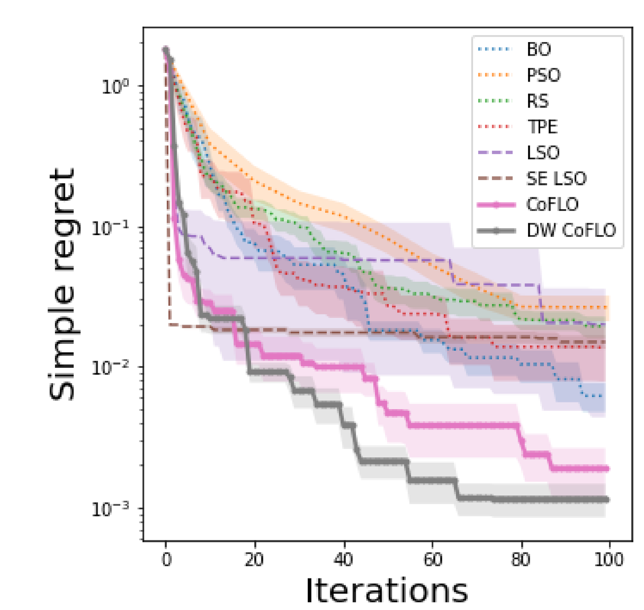
Experiment Results



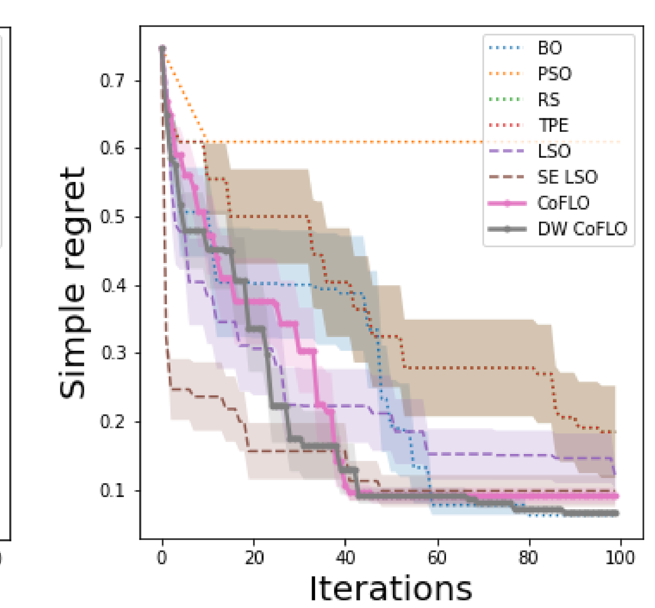
Rastrigin 2D



Feynman 6D



Supernova 3D



Cosmic Survey 14D [Nor+16]

Remarks

- A simple plugin amendment to *penalize collisions* in the latent space, with *optimization-aware dynamic weighting* for adjusting the collision penalty.
- Extensive empirical study on four synthetic and real-world datasets.

* *This work was supported by the Center for Data and Computing at the University of Chicago via a JTFI AI + Science Grant.*