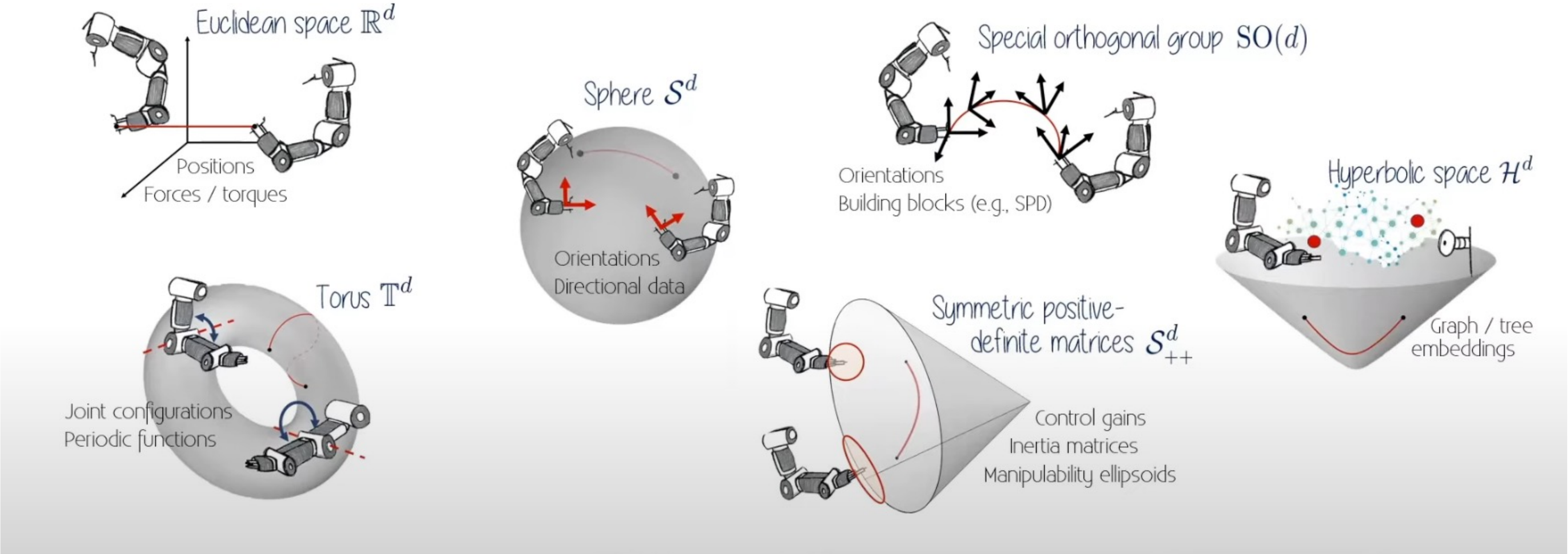
Geometry-aware Posterior Inference for High dimensional black-box optimization

Kiyoung Om

Table of contents

Motivation

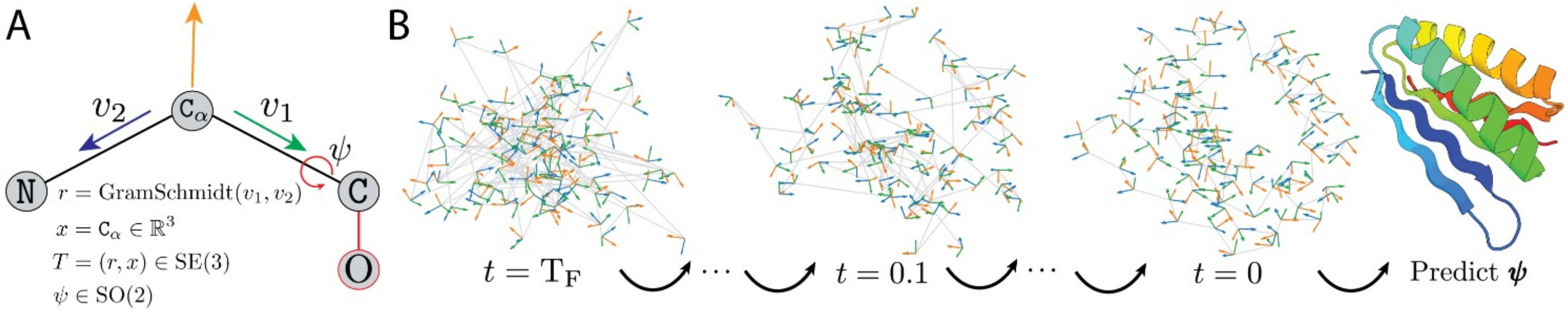
Robot manipulations with diverse manifold



- Force, torques Euclidean $\mathbb{R}^d$
- Orientations ($SO(3)$, Sphere S^d)
- Robot Poses ($SE(3)$)

Motivation

Protein Backbone Generation with SE(3) dynamics



- Starting from N amino residues.
- Rotation $SO(3)$ and Translation $\mathbb{R}^3$; $SE(3)$ group action multiple times.
- Leading to naturalistic protein structure.

Motivation

Geometric Deep learning

'cat' 'cat' 'cat'

- Example: Convolutional Neural Networks (CNNs)
- Convolution: Translation Invariance
- Max pooling: Scale Invariance
- This **'Inductive Bias'** drastically **reduce curse of dimensionality**

Motivation

Geometry-aware Black-box optimization (example)

- Consider below black-box optimization problem.

$$\text{Find } x \max_{x \in R^d} f(x)$$
$$s.t. \left|\left|x\right|\right|_2 = 1$$

- How to force x satisfy equality constraints?

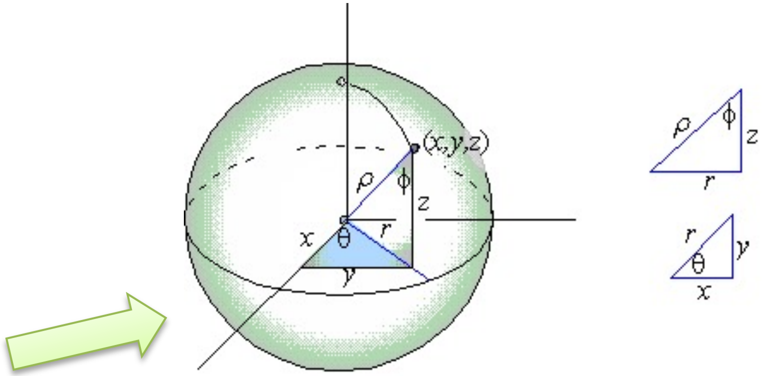
- IDEA:

1. Optimize on $x \in R^d$; then project x to $||x|| = 1$

2. Optimize on $x \in R^d$; with constraint penalty

3. **Optimize on $\boldsymbol{x} \in \boldsymbol{S}^{\mathbf{d-1}}$** : Sphere

Original Euclidean space $\{x, y, z\}$ → Spherical Coordinates $\{\phi, \theta\}$; $\rho = 1$



Motivation

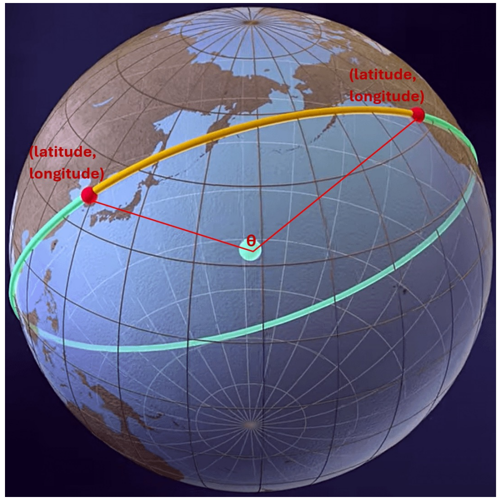
Geometry-aware Black-box optimization (example)

- Now, original constrained optimization problem

$$\text{Find } x \max_{x \in R^d} f(x)$$
$$s.t. \|x\|_2 = 1$$

- Transformed to **unconstrained** optimization problem.

$$\text{Find } x \max_{x \in S^{d-1}} f(x)$$



- Here, the point on the sphere S^{d-1} **is not following Euclidean**

- **Hint: Distance between two points on Earth?**

It is not an Euclidean Distance $\|x - y\|_2$

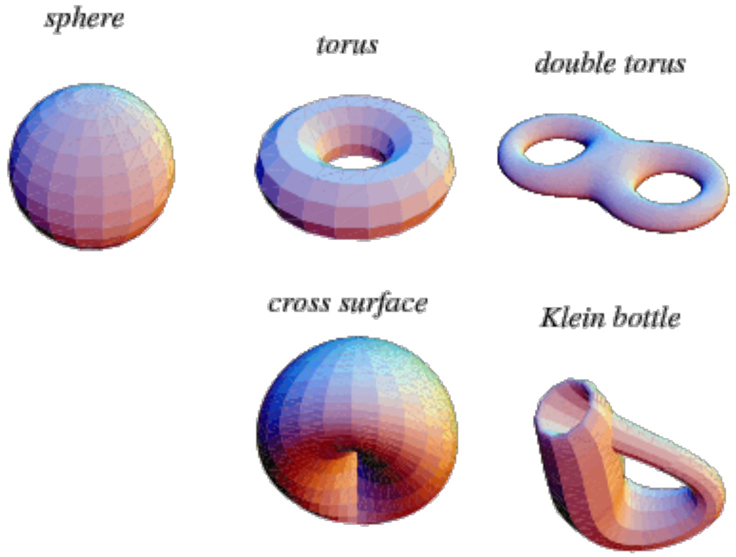
To deal with these non-Euclidean geometry, define **Riemannian Manifold**

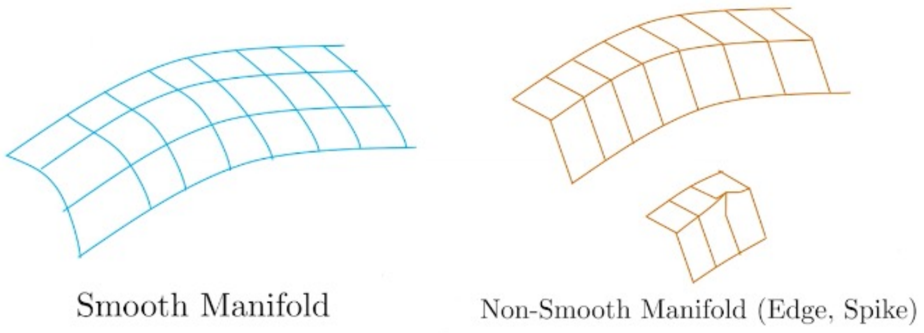
Preliminaries

Riemannian Manifold

- Informal: Smooth manifold (topological) that does **not follows Euclidean** geometry
  But **follows Euclidean** when **look closer to certain point $p \in M$. (Locally Resembles)**

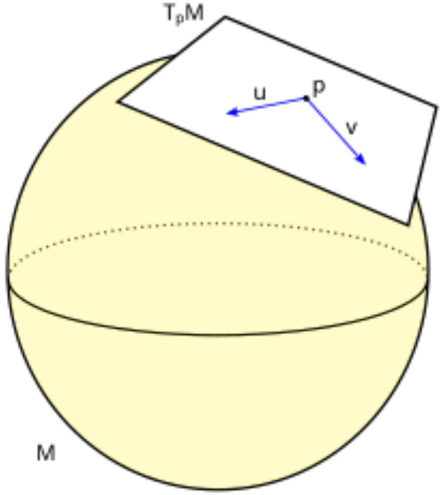
- EX): Earth
  Distance between New York and Seoul (long distance): $2\pi r \times \frac{\theta}{360^0}$
  Distance between two people (short distance): calculate with ruler; $||x - y||$





Preliminaries

Riemannian Manifold



Formal description:

- M be a smooth manifold C^{∞}, (No edges, spikes)
- For each point $p \in M$, there is an associated vector space T_pM called tangent space of M at p.
- We define metric g to 'measure' in M. $g_p : T_pM \times T_pM \to \mathbb{R}$
- We write $\langle u, v \rangle_p = g_p(u, v)$ on the tangent space.
- The norm is defined as: $\|v\|_p = \sqrt{g_p(v, v)}$
- Smooth manifold M + Riemannian metric g = Riemannian Manifold (M, g)

Preliminaries

Riemannian Manifold

Geodesics: Distance between New York and Seoul

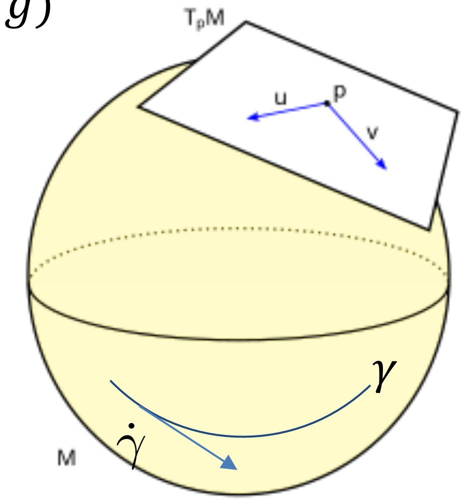
- Smooth manifold M + Riemannian metric g = Riemannian Manifold (M, g)
- Now we can calculate **length of the curve** defined on M as:

$$L(\gamma) = \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}\, dt.$$

- Where $\gamma: [a, b] \to M$ continuously differentiable curve with boundary condition: $\gamma(a) = p, \gamma(b) = q$
- By taking infimum of this leads to **distance** called **geodesic distance**

$$d_M(p, q) = \inf_{\gamma:[a,b]\to M,\ \gamma(a)=p, \gamma(b)=q} \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}\, dt.$$

And corresponding curve γ is called **Geodesics.**



Preliminaries

Riemannian Manifold

Mapping function between Manifold and Tangent Space:

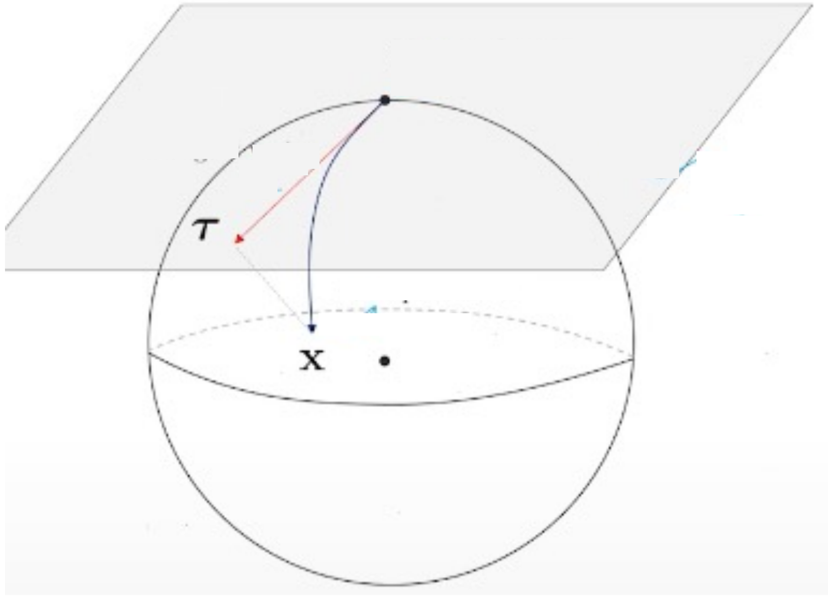
- Exponential map:
  From tangent space to M

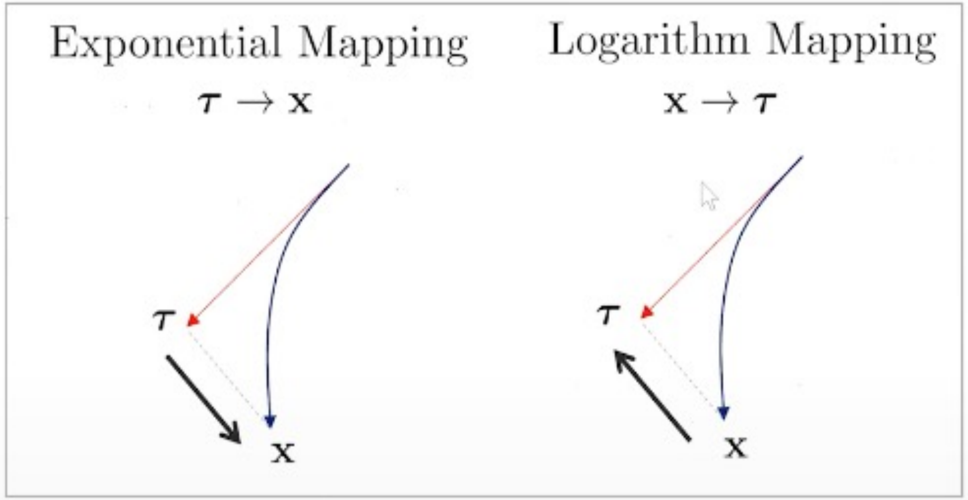
  $$Exp_p : T_pM \to M$$

- Logarithmic map:
  From M to tangent space

  $$\mathrm{Log}_p : U \subset M \to T_pM$$

  Where U is locally defined near p





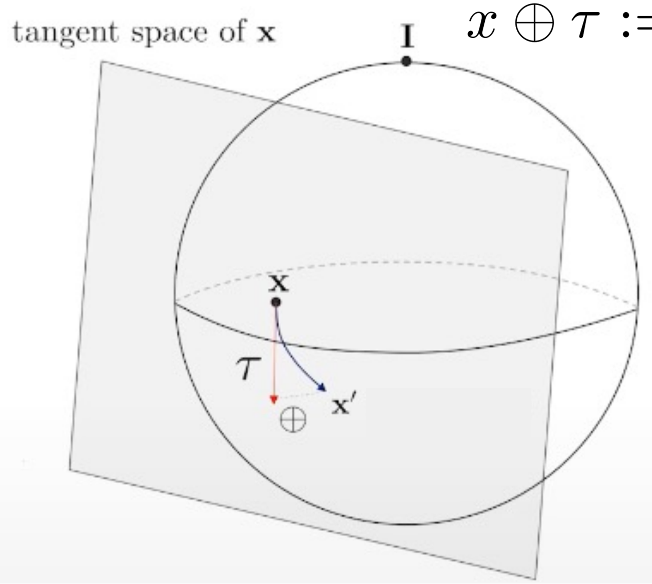
Preliminaries

Riemannian Manifold

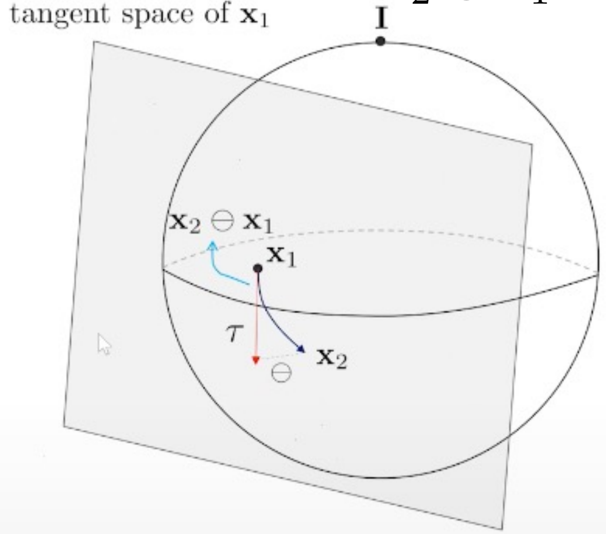
Plus and Minus operator in Riemannian manifold:

- We can define $\oplus$ and $\ominus$ operator as:

$$x \oplus \tau := \mathrm{Exp}_x(\tau),\ \tau \in T_x M$$



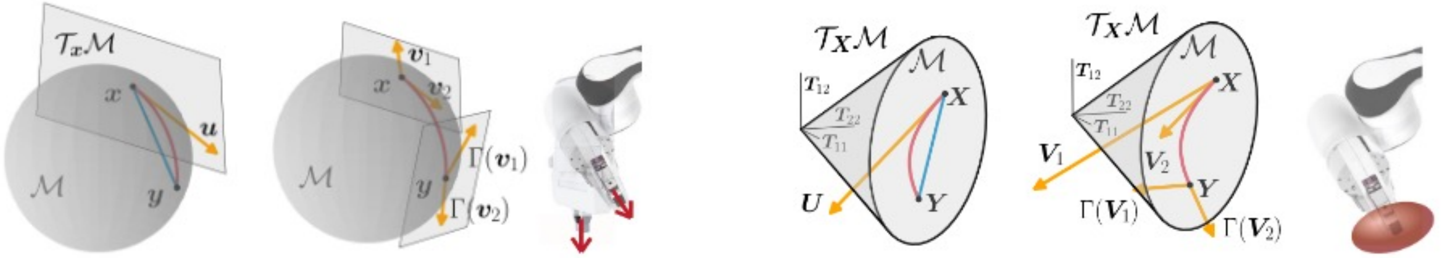
$$x_2 \ominus x_1 := \mathrm{Log}_{x_1}(x_2) \in T_{x_1} M$$



Preliminaries

Riemannian Manifold

Examples of Geodesics, Mapping function on Sphere, SPD manifold



(a) Sphere manifold $\mathcal{S}^2$ (incl. e.g. orientations). (b) SPD manifold $\mathcal{S}^2_{++}$ (incl. e.g. stiffness ellipsoids)

Table 1: Principal operations on the sphere $\mathcal{S}^d$ and SPD manifold $\mathcal{S}^D_{++}$ (see [25, 26, 27] for details).

Manifold	$d_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{y})$	$\mathrm{Exp}_{\boldsymbol{x}}(\boldsymbol{u})$	$\mathrm{Log}_{\boldsymbol{x}}(\boldsymbol{y})$
$\mathcal{S}^d$	$\arccos(\boldsymbol{x}^\mathsf{T}\boldsymbol{y})$	$\boldsymbol{x}\cos(\|\boldsymbol{u}\|) + \overline{\boldsymbol{u}}\sin(\boldsymbol{u})$	$d(\boldsymbol{x}, \boldsymbol{y})\,\frac{\boldsymbol{y}-\boldsymbol{x}^\mathsf{T}\boldsymbol{y}\,\boldsymbol{x}}{\|\boldsymbol{y}-\boldsymbol{x}^\mathsf{T}\boldsymbol{y}\,\boldsymbol{x}\|}$
$\mathcal{S}^D_{++}$	$\|\log(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{Y}\boldsymbol{X}^{-\frac{1}{2}})\|_\mathrm{F}$	$\boldsymbol{X}^{\frac{1}{2}}\exp(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{X}^{-\frac{1}{2}})\boldsymbol{X}^{\frac{1}{2}}$	$\boldsymbol{X}^{\frac{1}{2}}\log(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{Y}\boldsymbol{X}^{-\frac{1}{2}})\boldsymbol{X}^{\frac{1}{2}}$

Preliminaries

Riemannian Manifold

Euclidean space is a special case of a Riemannian Manifold

- For each point in $p \in \mathbb{R}^d$, associated tangent space: $T_p\mathbb{R}^d \cong \mathbb{R}^d$

Manifold	$p \in M$	$p \in \mathbb{R}^d$
Metric:	$g_p : T_pM \times T_pM \to \mathbb{R}$	$g_p : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
Inner Product:	$\langle u, v \rangle_p = g_p(u, v)$	$g(u, v) = u^\top v = \sum_{i=1}^{d} u_i v_i$
Norm	$\lVert v \rVert_p = \sqrt{g_p(v, v)}$	$\lvert\lvert v \rvert\rvert_p = \lvert\lvert v \rvert\rvert = \sqrt{v \cdot v}$
Geodesic distance:	$d_M(p, q) = \inf \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}\, dt.$	$d_{\mathbb{R}^d}(p, q) = \lvert\lvert p - q \rvert\rvert_2$

Prior works

Geometry Aware Bayesian Optimization (GaBO)

$$\text{find } x^* = \text{argmax}_{x \in X} f(x)$$

within R rounds, B batch of queries

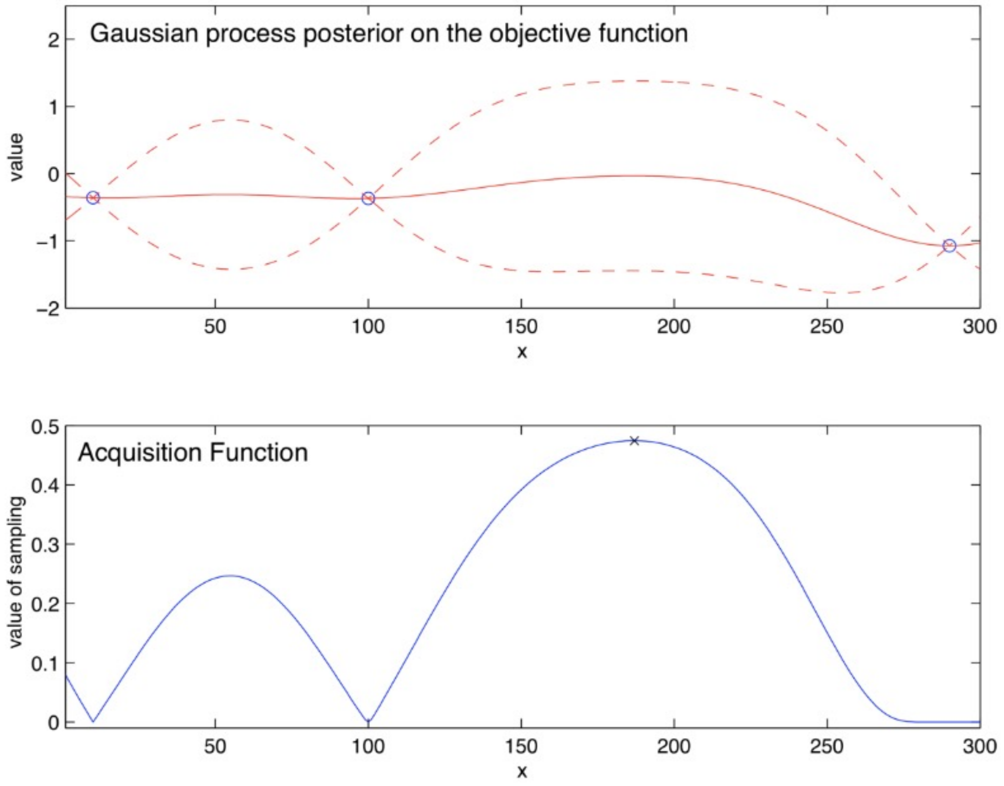
$$\text{find } x^* = \text{argmax}_{x \in M} f(x)$$

within R rounds, B batch of queries

Prior works

Geometry Aware Bayesian Optimization (GaBO)

Vanilla Bayesian optimization framework:



1. Train Gaussian Process (GP) Based **surrogate models** for estimating candidate posterior
$$f(x) \sim GP(\mu(x), K(x, x'))$$

2. Use **acquisition function** (EI, UCB) to select next promising point
$$x_{t+1} = \mathrm{argmax}_x\, \alpha(x)$$

3. Evaluate expensive objective function at selected point
$$y_{t+1} = f(x_{t+1})$$

4. Update GP model with new observation data

Repeat 1~4 until convergence or budget exhausted

Prior works

Geometry Aware Bayesian Optimization (GaBO)

Adding Inductive Bias for Geometry:

1. Train Gaussian Process (GP) Based **surrogate models** for estimating candidate posterior

$$\boxed{f(x) \sim GP(\mu(x), K(x, x'))}$$

$$k(x_i, x_j) = \theta \exp(-\beta d(x_i, x_j)^2)$$

$$k(x_i, x_j) = \theta \exp(-\beta d_M(x_i, x_j)^2)$$

From distance based kernels to geodesics.

2. Use **acquisition function** (EI, UCB) to select next promising point

$$\boxed{x_{t+1} = \text{argmax}_x \, \alpha(x)}$$

Conjugate Gradient on Riemannian manifolds

3. Evaluate expensive objective function at selected point

$$y_{t+1} = f(x_{t+1})$$

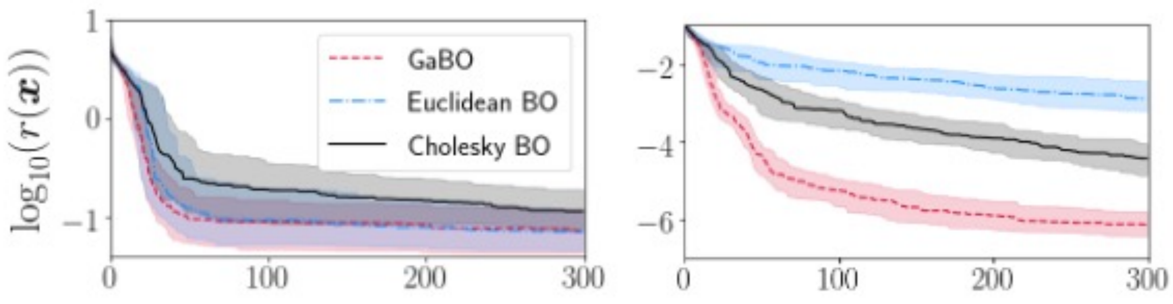
4. Update GP model with new observation data

Repeat until convergence or budget exhausted

Prior works

Geometry Aware Bayesian Optimization (GaBO)

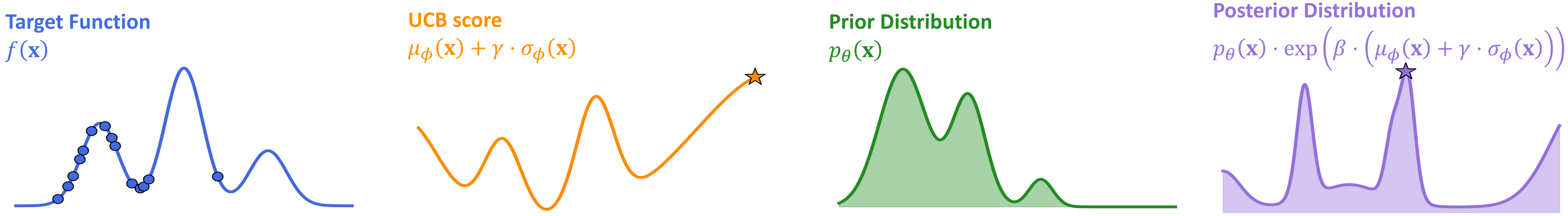
Results:



- Much better than Euclidean BO with constraint penalty.
- Revealing that Geometry-aware (**inductive bias**) helpful for the optimization.
- **Still suffer from high-dimensional settings (BO's fundamental problem)**

Prior works

Posterior Inference with Diffusion Models for High-dimensional Black-box optimization



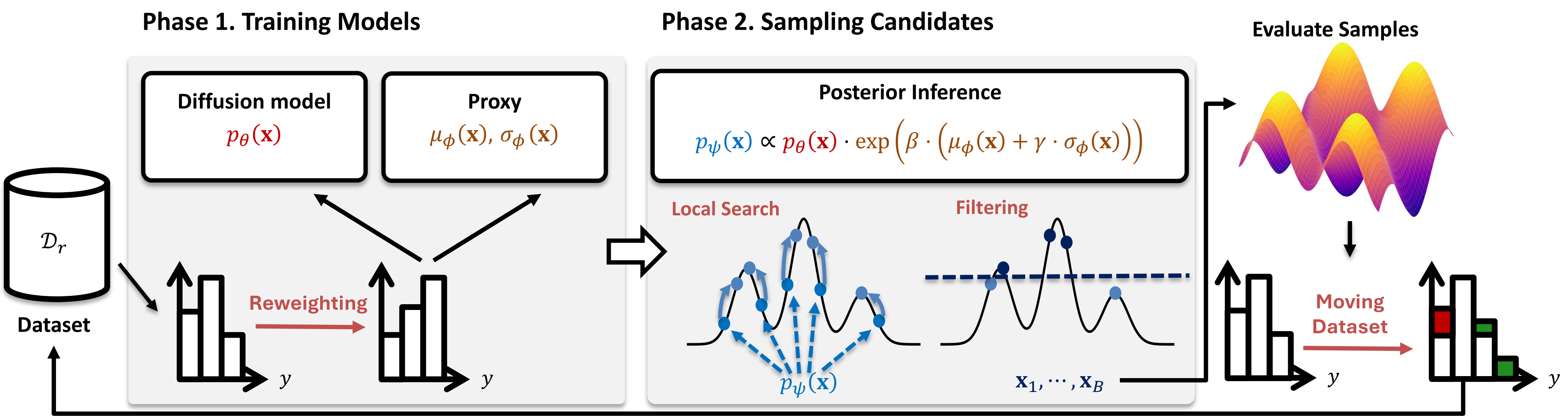
Motivating figure of our methods

- In high-dimensional space, directly searching $\mathrm{argmax}_{\mathbf{x}}\, \mu_\phi(\mathbf{x}) + \gamma \cdot \sigma_\phi(\mathbf{x})$ may lead to suboptimal results.
- Sampling from the posterior distribution prevents overemphasized exploration of boundary.

$$p_{\mathrm{tar}}(\mathbf{x}) = \frac{1}{Z} \cdot p_\theta(\mathbf{x}) \exp\left(\beta \cdot r_\phi(\mathbf{x})\right)$$

Prior works

Posterior Inference with Diffusion Models for High-dimensional Black-box optimization



- Train diffusion prior $p_\theta(x)$, and network ensembles $\mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x})$.
- Fine-tune diffusion models with **Relative Trajectory Balance (RTB)** loss.
- Sampling candidates, evaluate, and repeat.

Prior works

Posterior Inference with Diffusion Models for High-dimensional Black-box optimization

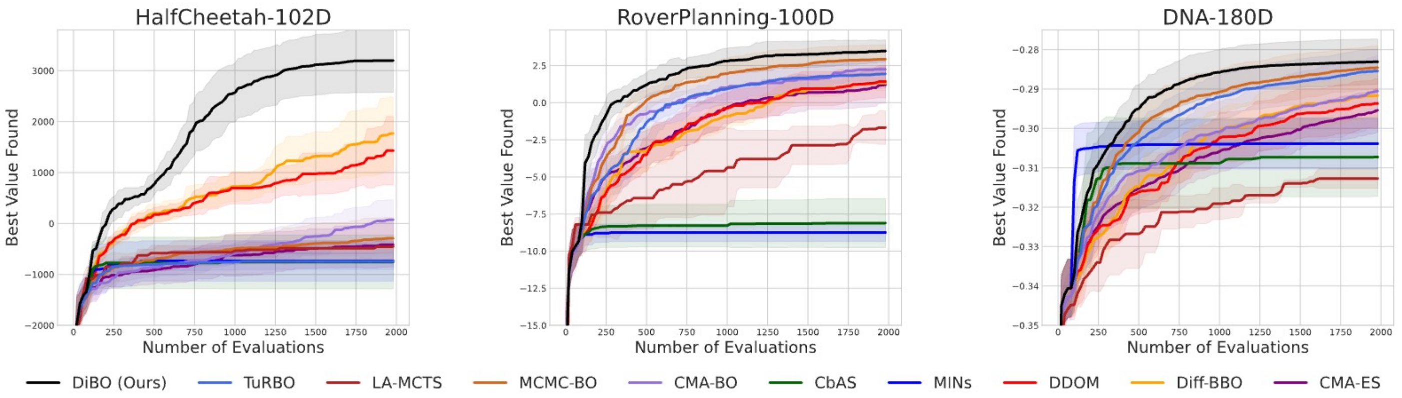


Table 5. Average time (in seconds) for each round in each method.

	Rastrigin-200D	Rastrigin-400D	Ackley-200D	Ackley-400D	Levy-200D	Levy-400D	Rosenbrock-200D	Rosenbrock-400D
TuRBO	244.39 ± 0.21	1089.09 ± 0.04	33.91 ± 0.24	41.50 ± 0.06	167.58 ± 0.21	59.70 ± 0.02	452.21 ± 0.15	45.05 ± 0.01
LA-MCTS	222.95 ± 6.59	256.82 ± 8.89	150.27 ± 3.97	184.14 ± 1.67	90.47 ± 1.69	229.80 ± 11.99	154.56 ± 4.08	223.59 ± 7.52
MCMC-BO	341.98 ± 3.54	429.02 ± 4.61	370.03 ± 4.17	345.60 ± 3.42	337.87 ± 3.65	429.02 ± 4.61	419.07 ± 5.12	448.56 ± 5.13
CMA-BO	643.14 ± 4.01	833.97 ± 2.12	661.15 ± 4.52	854.23 ± 2.77	694.67 ± 4.59	871.33 ± 3.66	645.65 ± 4.95	857.39 ± 2.53
CbAS	212.69 ± 5.27	213.64 ± 5.79	207.35 ± 4.01	213.67 ± 6.62	212.61 ± 4.05	212.69 ± 9.83	203.87 ± 1.43	221.68 ± 5.01
MINs	28.36 ± 0.45	32.20 ± 0.12	28.83 ± 0.41	29.14 ± 0.60	29.91 ± 0.17	29.95 ± 0.37	30.22 ± 1.02	28.81 ± 0.67
DDOM	23.74 ± 0.28	26.57 ± 0.11	23.53 ± 0.04	26.62 ± 0.16	23.40 ± 0.15	26.46 ± 0.15	23.19 ± 0.12	26.55 ± 0.15
Diff-BBO	128.75 ± 0.89	143.80 ± 1.16	131.16 ± 0.86	143.96 ± 1.68	130.07 ± 0.74	143.97 ± 1.79	128.33 ± 1.74	144.03 ± 1.58
CMA-ES	0.03 ± 0.01	0.05 ± 0.00	0.03 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.05 ± 0.00	0.03 ± 0.00	0.04 ± 0.00
DiBO	42.74 ± 0.26	79.63 ± 1.07	39.72 ± 0.18	71.99 ± 0.10	39.55 ± 0.10	72.21 ± 0.53	39.57 ± 0.28	72.88 ± 0.34

- Did well on high-dimensional settings
- Much more **efficient**, compared to BO based methods.

Method

Geometry-aware Posterior Inference for High dimensional black-box optimization

Problem is given as:

$$\text{Find } x^* = \underset{x \in M}{argmax}\, f(x)$$

- For R rounds B batch of Query
- Where function $f(x) : M \to \mathbb{R}$ is black-box.
- Manifold M is known and assumed to have high dimensionality.

We want to sample from: $\underline{p_{\text{tar}}(\mathbf{x})} = \frac{1}{Z} \cdot \underline{p_\theta(\mathbf{x})} \exp\left(\beta \cdot \underline{r_\phi(\mathbf{x})}\right)$

1. Expressive Prior 2. Surrogate function

3. Posterior Sampling method

Method

Geometry-aware Posterior Inference for High dimensional black-box optimization

Expressive Prior $p_\theta(x)$

Possible choices:

- Riemannian Diffusion Models
- Riemannian Score-Based Generative Modeling
- Riemannian Flow-matching

Surrogate function $r_\phi(x): M \to R$

Possible choices:

- Geometry Aware CNN

Posterior Sampling method

Possible choices:

- Riemannian Classifier Guidance
- Twisted Diffusion Sampler
- **Fine-tuning method? (Not developed yet)**

Method (Not fixed)

Geometry-aware Posterior Inference for High dimensional black-box optimization

- Task:

  Synthetic function with S^d, SPD^d domain.

  Real world tasks: Robot manipulation task.

Algorithm:

1. Train Prior $p_\theta(x)$ ← Riemannian Score-Based Generative Modeling
2. Train Surrogate function $r_\phi(x): M \to R$ ← Geometry Aware CNN
3. Posterior Sampling method ← Twisted Diffusion Sampler
4. Sample from posterior $\{x_i\}_{i=1}^N \sim p_\theta(x) \exp r_\phi(x)$
5. Query samples to the function $\{y_i\}_{i=1}^N = \{f(x_i)\}_{i=1}^N$

Repeat 1~5 iteratively

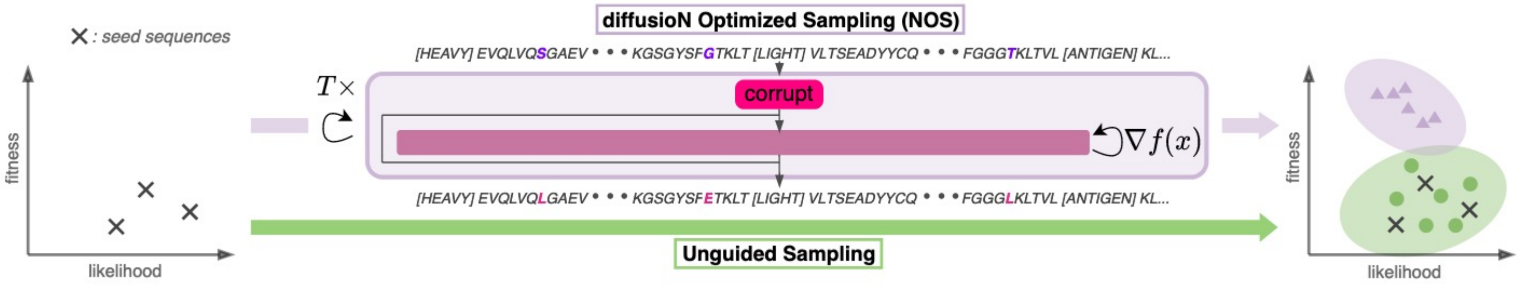
Another Task

Protein Structure Design with Riemannian Posterior Inference

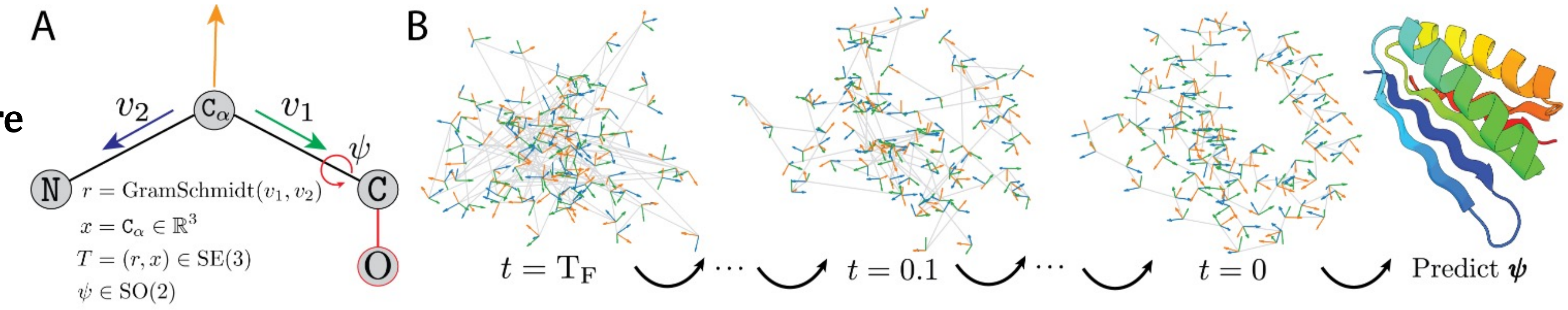
- If we can develop **Riemannian Fine-tuning method** $p_{\mathrm{tar}}(\mathbf{x}) = \frac{1}{Z} \cdot p_\theta(\mathbf{x}) \exp(\beta \cdot r_\phi(\mathbf{x})) \quad \mathbf{x} \in M$

  Task: Protein design with high binding affinity.

Protein **Sequence**



Protein **Structure**



- Design protein structure iteratively optimizing arbitrary function

Future Plan

Geometric Deep Learning + Generative models

- Designing best way to incorporate inductive bias in Generative models.
- Application of above methods to downstream tasks (Robot manipulation & Material/Scientific Design)

Reference

- Posterior Inference with Diffusion Models for High-dimensional Black-box Optimization (ours)
- SE(3) diffusion model with application to protein backbone generation
- Fast protein backbone generation with SE(3) flow matching
- Protein Design with Guided Discrete Diffusion
- Flow Matching on General Geometries
- Riemannian Diffusion Models
- Riemannian Score-Based Generative Modelling
- Bayesian Optimization Meets Riemannian Manifolds in Robot Learning
- High-Dimensional Bayesian Optimization via Nested Riemannian Manifolds

Generative Protein Design

Riemannian Generative Models

Geometry Aware BO

EOD

Q&A