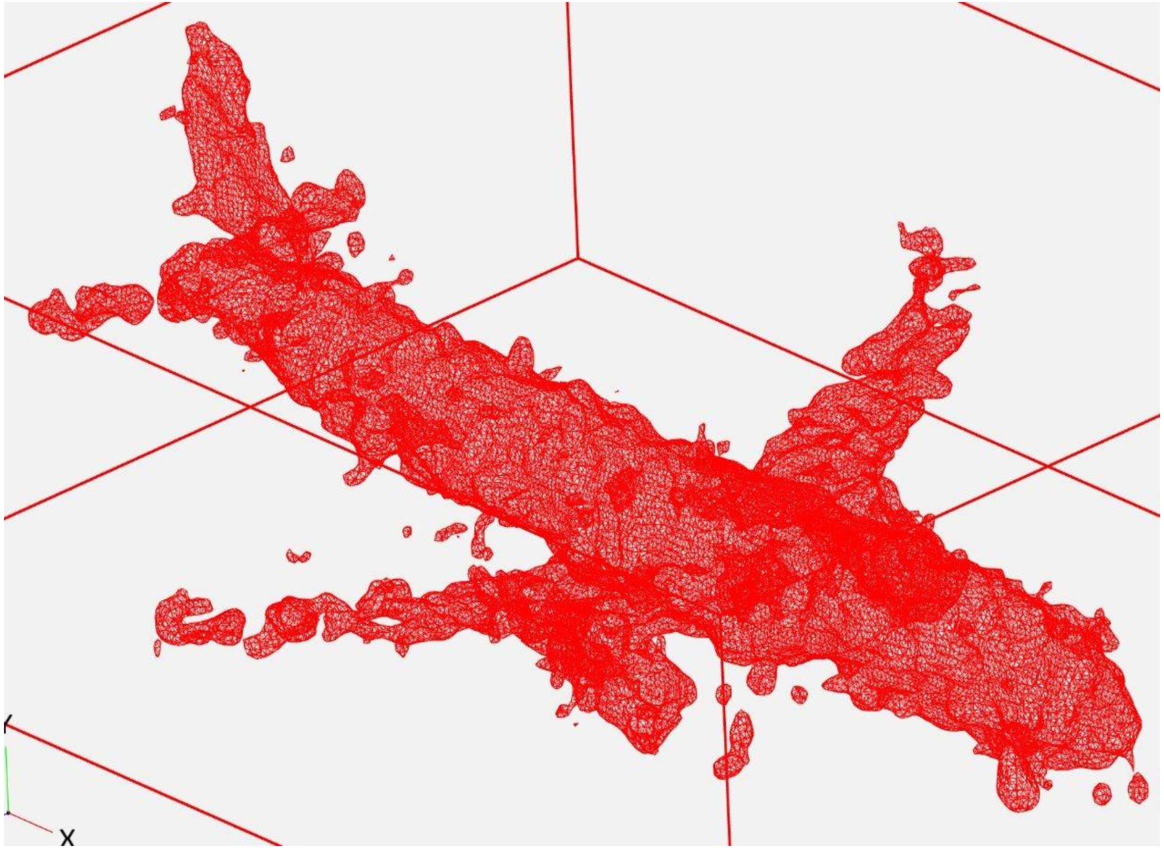
Experimental Design in the Age of Generative **AI**

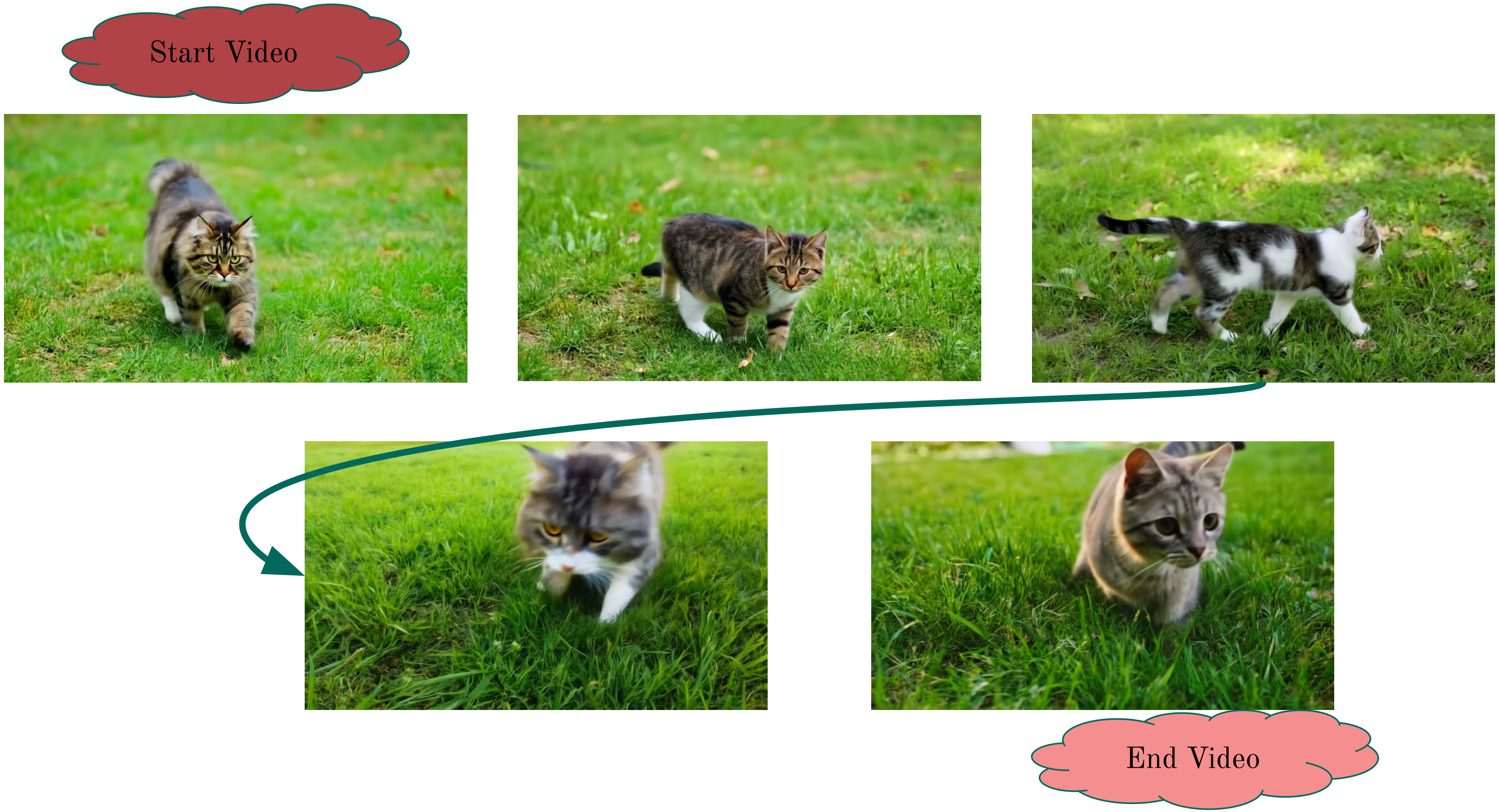
Henry Moss

AstraZeneca
Lancaster University
UNIVERSITY OF CAMBRIDGE
BOEING

DO INTERRUPT ME PLS



X


Start Video
End Video

Bayesian Optimisation

Bayesian Optimisation Recap

Goal:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}}\, f(\mathbf{x})$$

Search Space

Bayesian Optimisation Recap

Goal:

$$\mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$

Search Space

N^{th} step:

$$\mathbf{x}_N = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{X}} \alpha_N(\mathbf{x})$$

Acquisition Function

Bayesian Optimisation Recap

Goal:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} f(\mathbf{x})$$

Search Space

N[th] step:

$$\mathbf{x}_N = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \mathbb{P}(\tilde{f}(\mathbf{x}) > f^* | D_N)$$

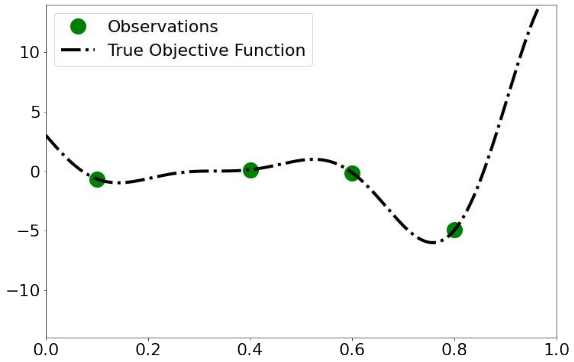
Acquisition Optimizer

Acquisition Function

Surrogate model

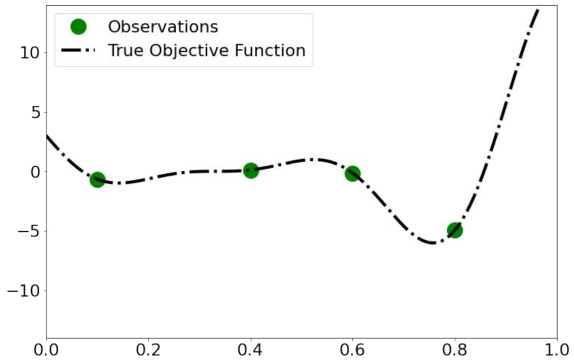
Expected Improvement

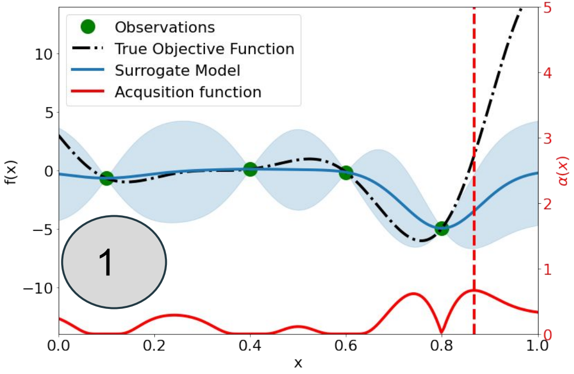
Demo BO loop



Expected Improvement

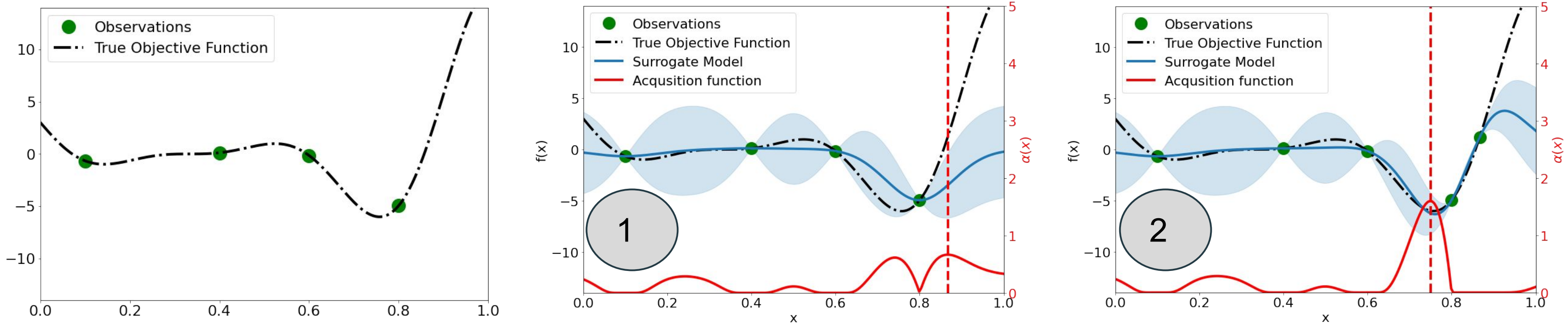
Demo BO loop





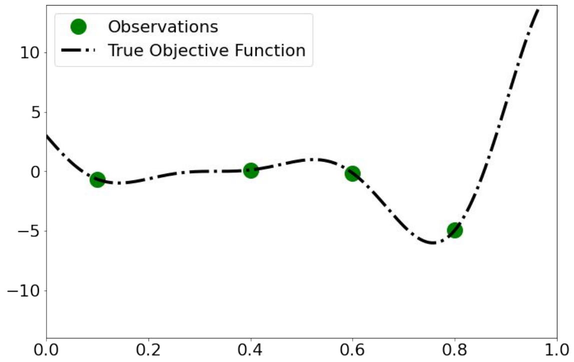
Expected Improvement

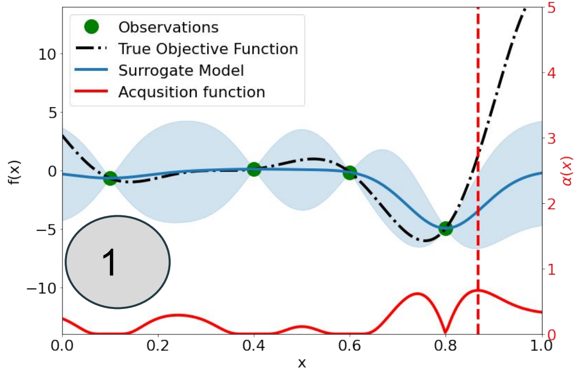
Demo BO loop

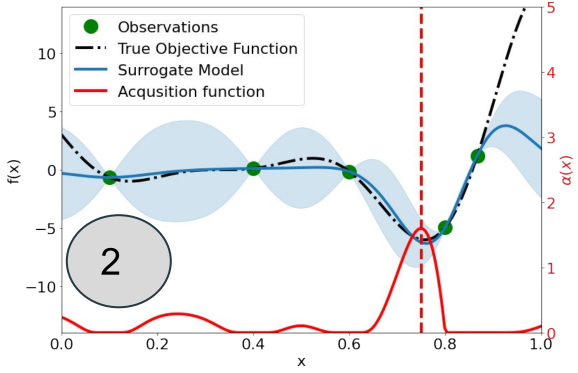


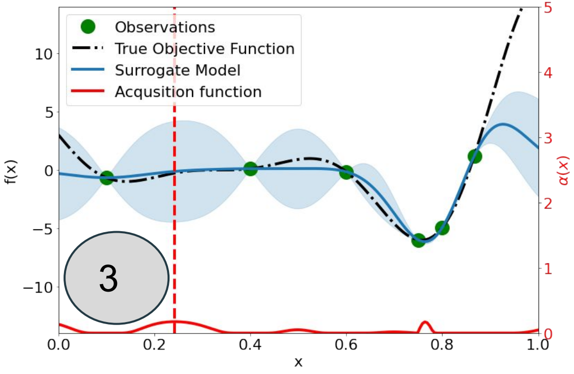
Expected Improvement

Demo BO loop



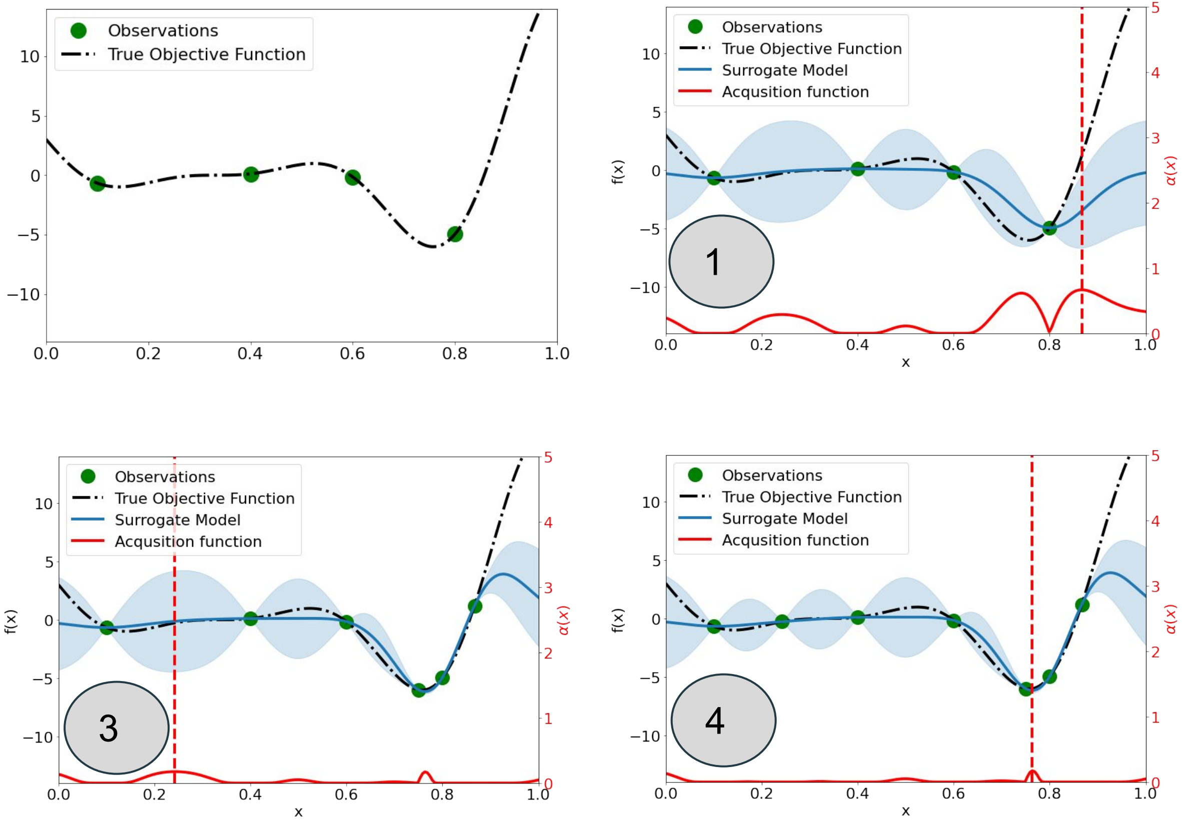






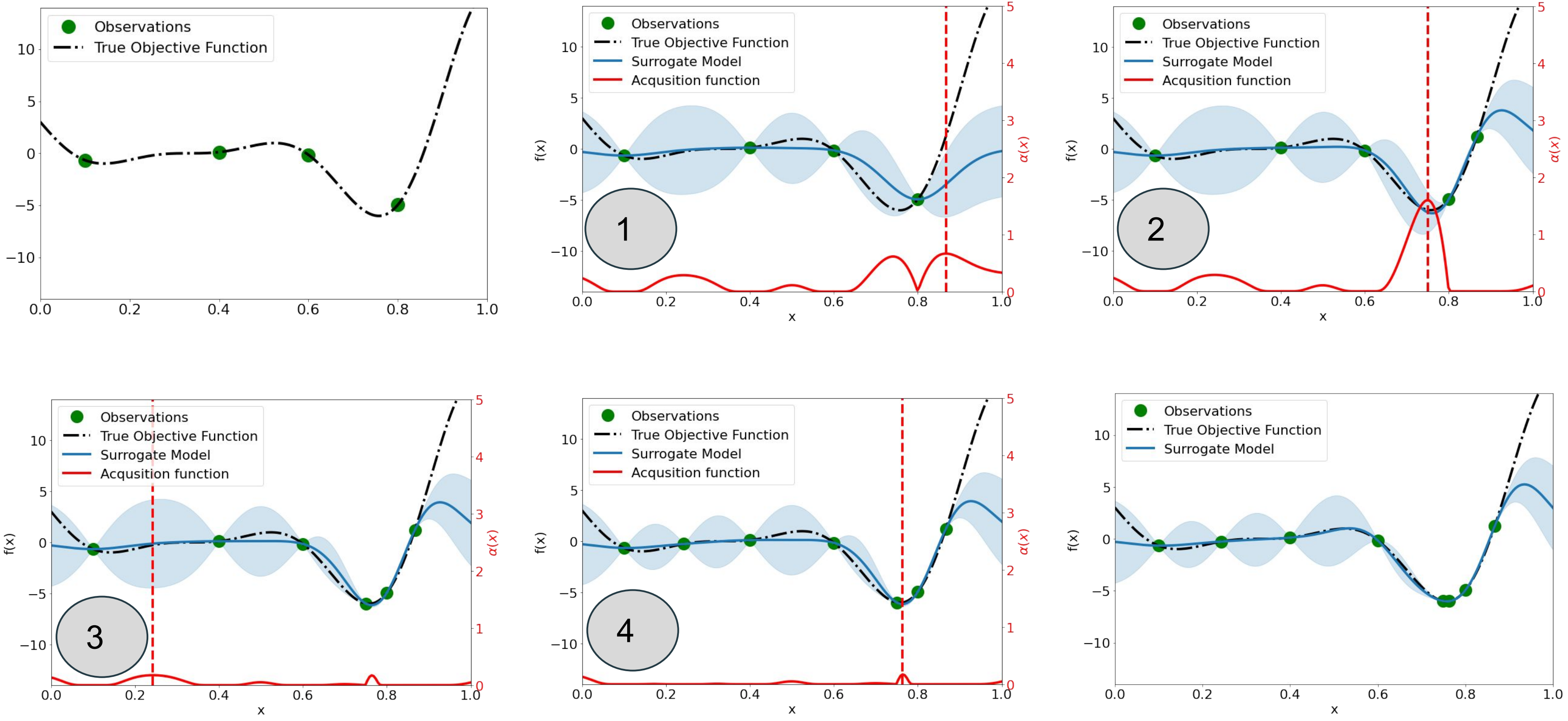
Expected Improvement

Demo BO loop

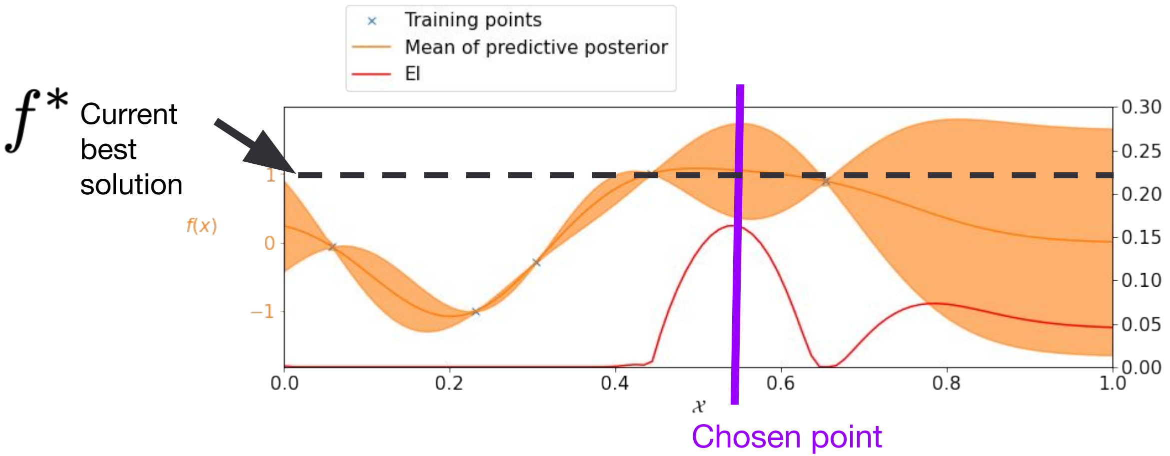


Expected Improvement

Demo BO loop



Bayesian Optimisation over Euclidean space



- Search Space: $\mathcal{X} \subset \mathbb{R}^d$
- Surrogate Model: $\tilde{f} \sim \mathcal{GP}(0, k_{\text{RBF}})$
- Acquisition optimiser: cts (+gradients)

Bayesian Optimisation of “structured” spaces

Baptista et al. 2018

Bayesian Optimisation over Categories

- Search Space: $\mathcal{X} = \{0, 1\}^d$
- Surrogate Model: $\tilde{f}(\mathbf{x}) = \alpha_0 + \sum \alpha_j x_j + \sum \alpha_{ij} x_i x_j$
- Acquisition optimiser: binary quadratic programming

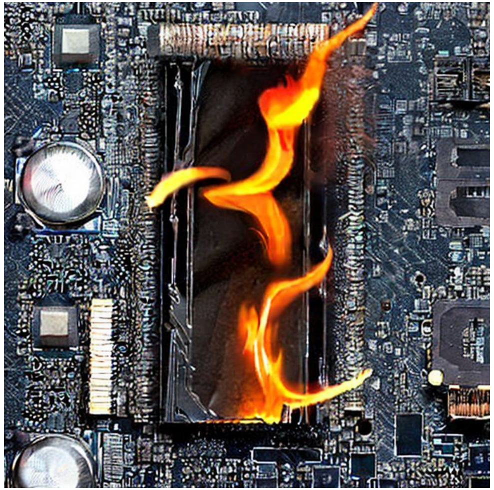
Deshwal et al. 2021

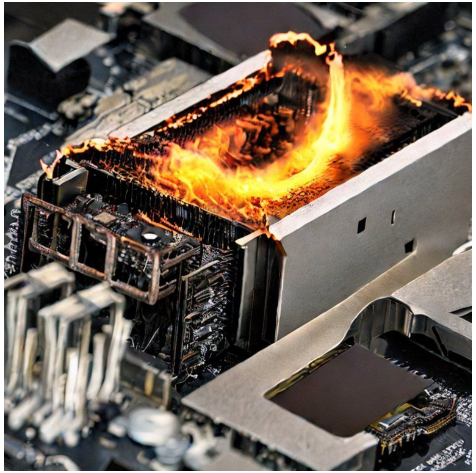
Bayesian Optimisation over Mixed Spaces

- Search Space: $\mathcal{X} = \mathbb{R}^{d_1} \cup \{0,1\}^{d_2}$
- Surrogate Model: $\tilde{f} \sim \mathcal{GP}(0, k_{\text{diffusion}})$
- Acquisition optimiser: Alternating cts + discrete

Deshwal et al. 2022

Bayesian Optimisation over Permutations





- Search Space: $\mathcal{X} = S_d$
- Surrogate Model: $\tilde{f} \sim \mathcal{GP}(0, k_{\text{Mallow}})$
- Acquisition optimiser: Evolutionary optimiser

Moss et al. 2020

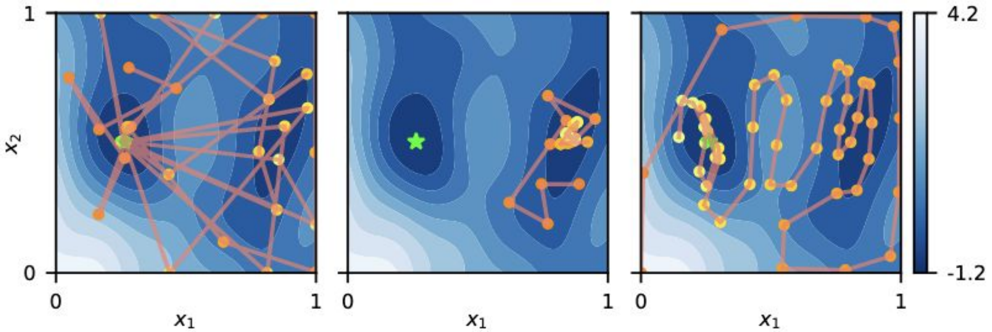
String Bayesian Optimisation

1. **Unconstrained** Any string made exclusively from characters in the alphabet Σ are allowed. S contains all these strings of any (or a fixed) length.
2. **Locally constrained** S is a collection of strings of fixed length, where the set of possible values for each character depends on its position in the string, i.e. the character s_i at location i belongs to the set $\Sigma_i \subseteq \Sigma$.
3. **Grammar constrained** S is the set of strings made from Σ that satisfy the syntactical rules specified by a context-free grammar.
4. **Candidate Set**. A space with unknown or very complex syntactical rules, but for which we have access to a large collection S of valid strings.

- Search Space: $\mathcal{X} = S$
- Surrogate Model: $\tilde{f} \sim \mathcal{GP}(0, k_{\mathrm{SSK}})$
- Acquisition optimiser: Genetic algorithms

Folch et al. 2022
Yang et al. 2023

Bayesian Optimisation over Paths

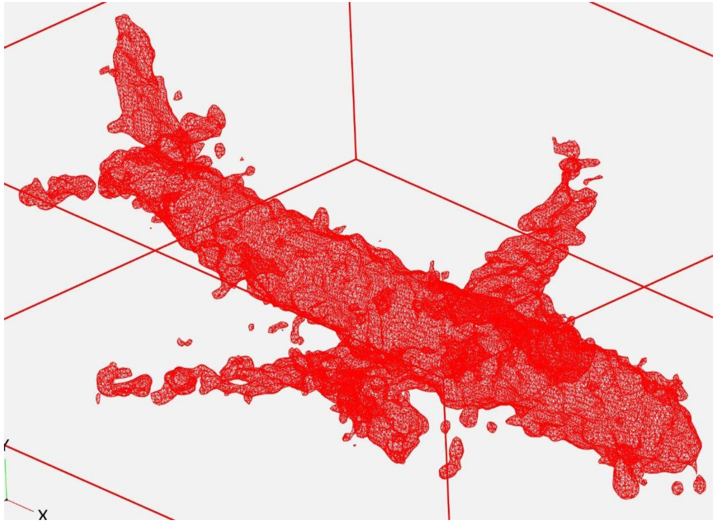


(a) EI (b) EI per unit cost (c) MONGOOSE

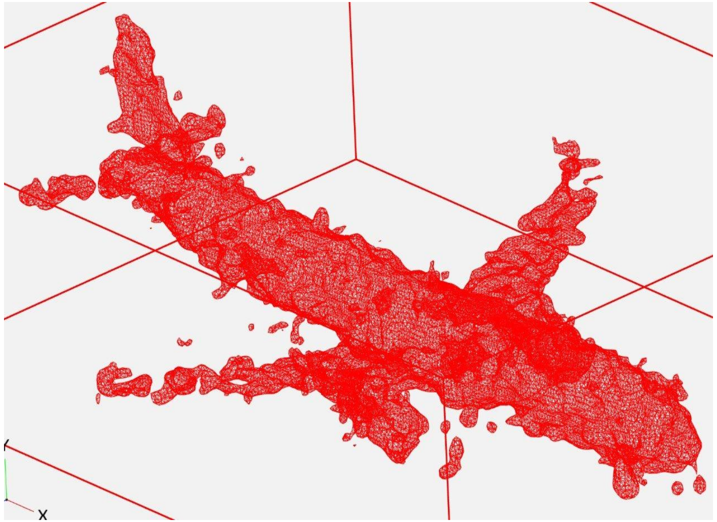
- Search Space: $\mathcal{X} = \{\text{smooth paths} \in \mathbb{R}^d\}$
- Surrogate Model: $\tilde{f} \sim \mathcal{GP}(0, k_{\mathrm{RBF}})$
- Acquisition optimiser: Travelling Salesman / RL

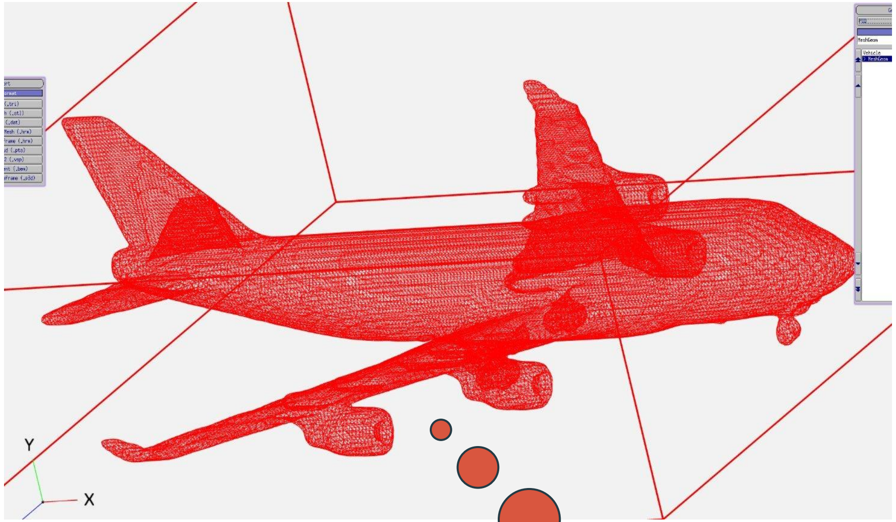
Bayesian Optimisation of “highly structured” spaces

You know it when you see it



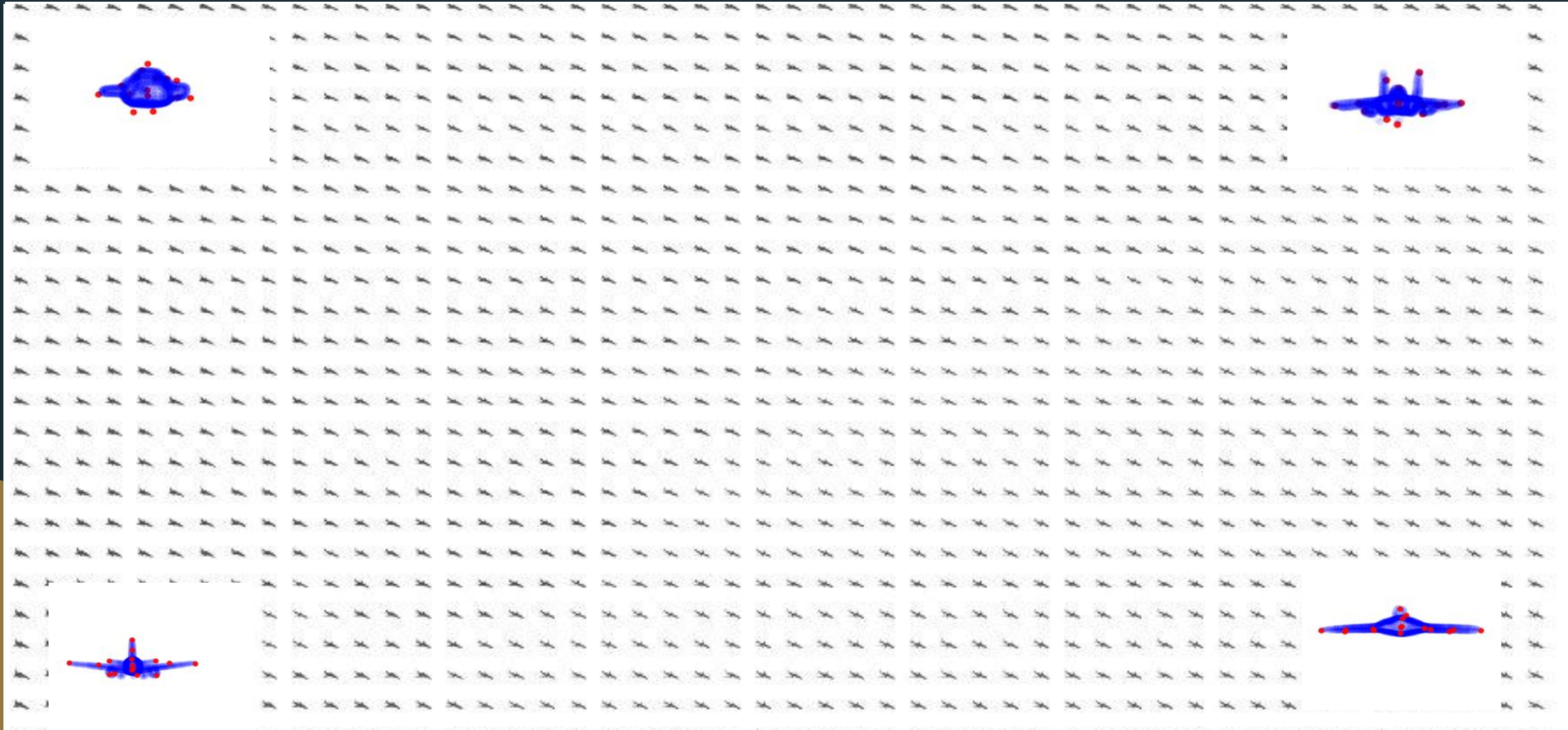
You know it when you see it





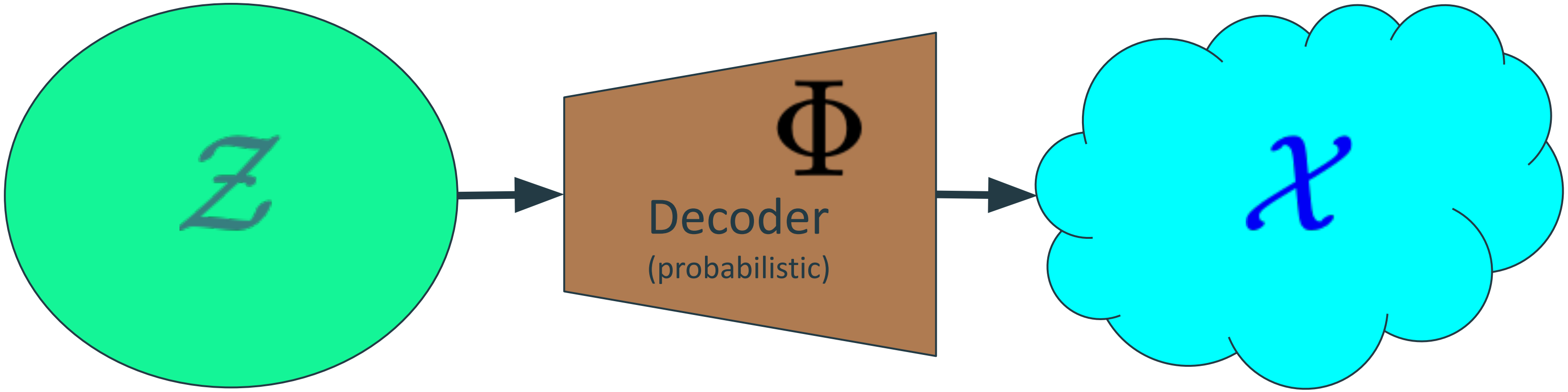
- Vanishingly small support
- Can’t learn *is_plane* function

You know it when you see it



Generative AI is cool right?

Priors over "**Highly Structured**" spaces



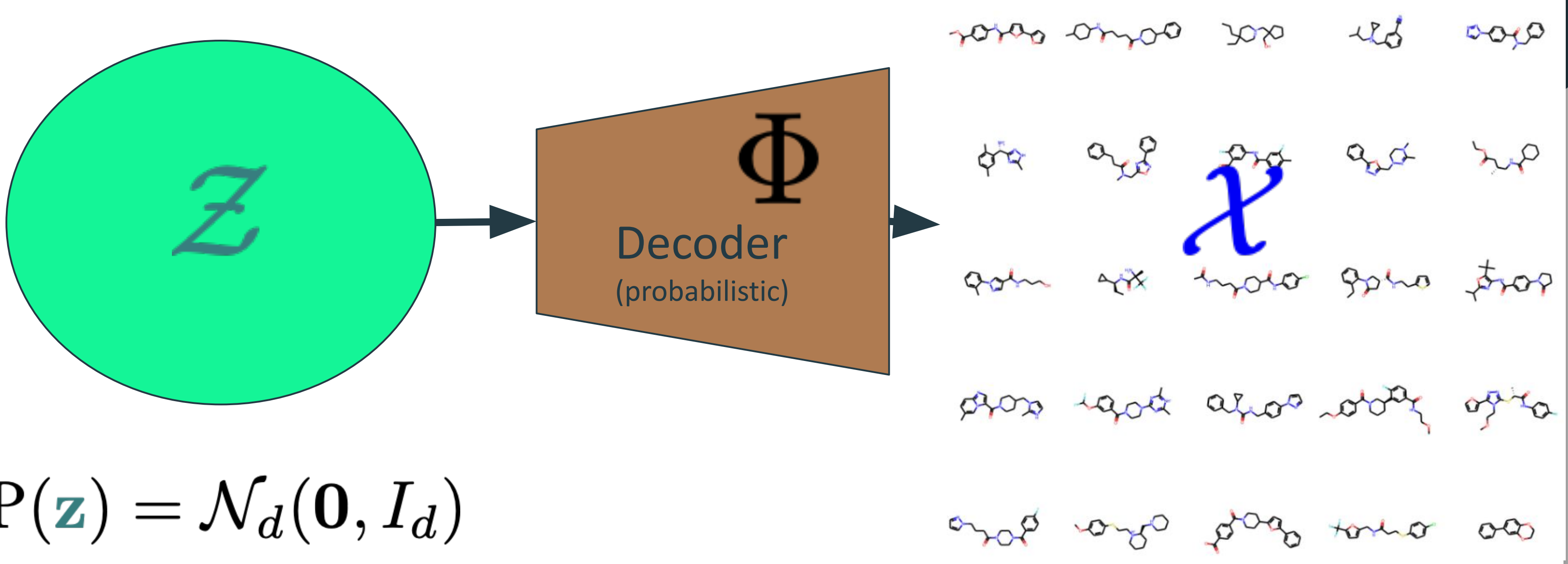
$$\mathbb{P}(\mathbf{z}) = \mathcal{N}_d(\mathbf{0}, I_d)$$

$$\mathbb{P}(\mathbf{x}) = ???$$

VAE / Diffusion / Normalising Flows e.t.c on unlabeled data

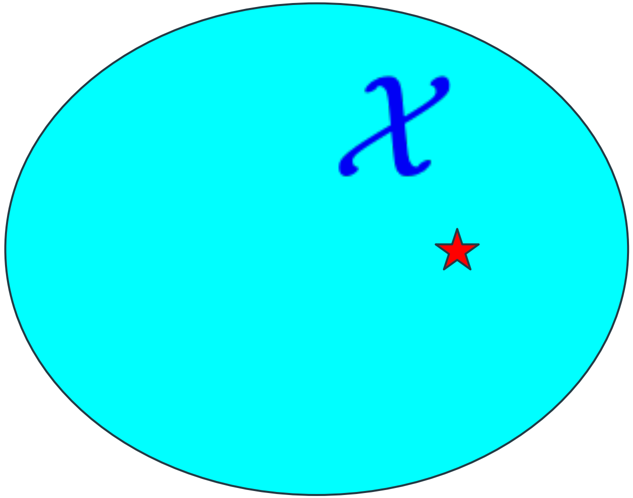
Richards et al. 2022

Priors over “**Highly Structured**” molecule spaces



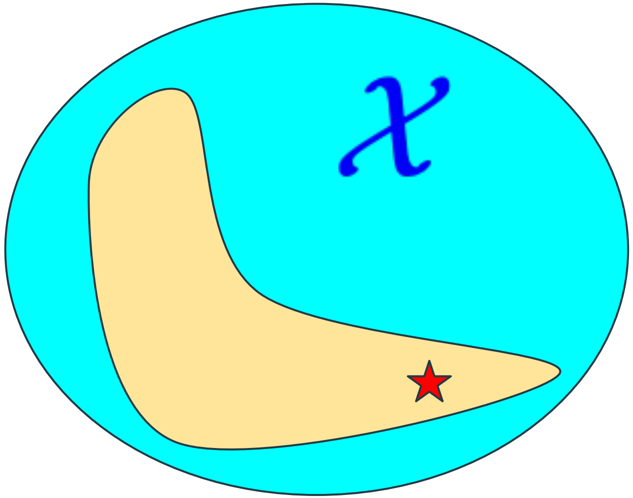
$$\mathbb{P}(\mathbf{z}) = \mathcal{N}_d(\mathbf{0}, I_d)$$

Standard BO



$$\max_{x \in \mathcal{X}} f(x)$$

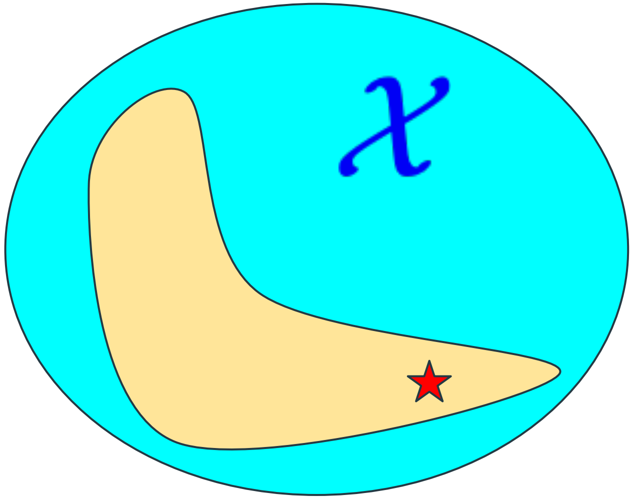
Standard BO



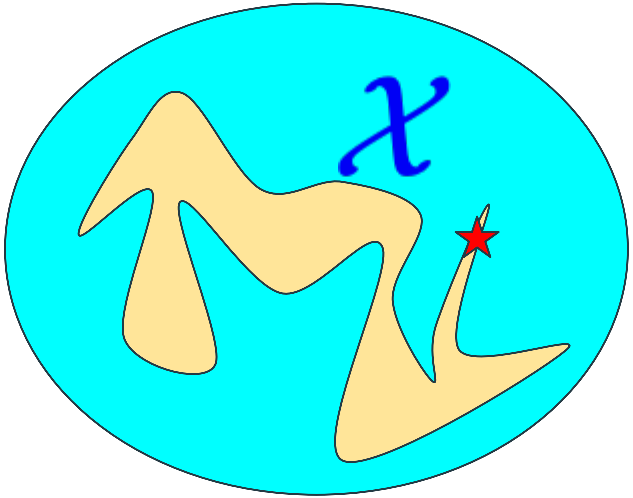
$$\max_{x \in \mathcal{X}} f(x)$$

$$: P(C(x) = 0) > t$$

Standard BO v.s. BO with GenAI prior



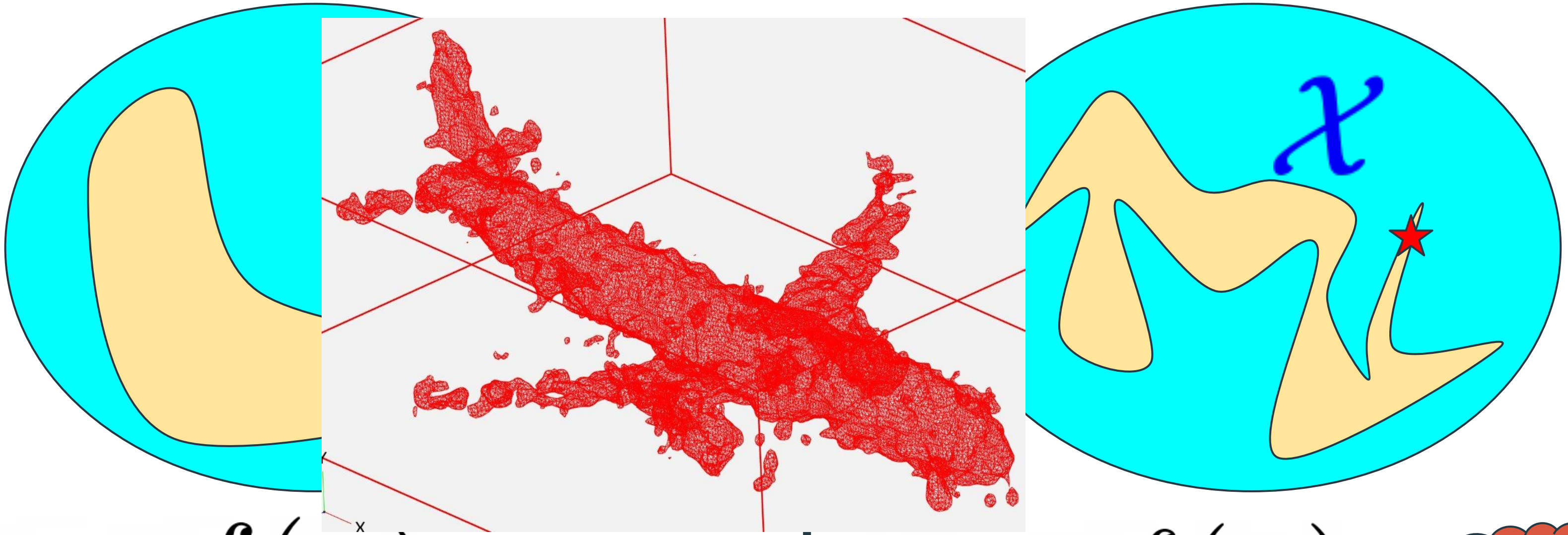
$$\max_{x \in \mathcal{X}} f(x) \; : P(C(x) = 0) > t$$



$$\max_{x \in \mathcal{X}} f(x)$$

$$: P(x) > t$$

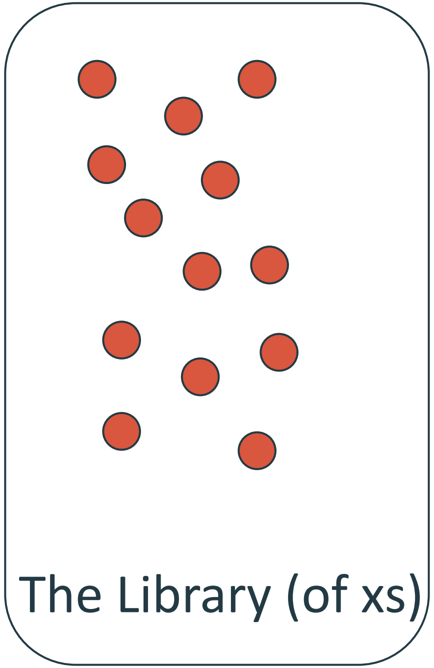
Standard BO v.s. BO with GenAI prior



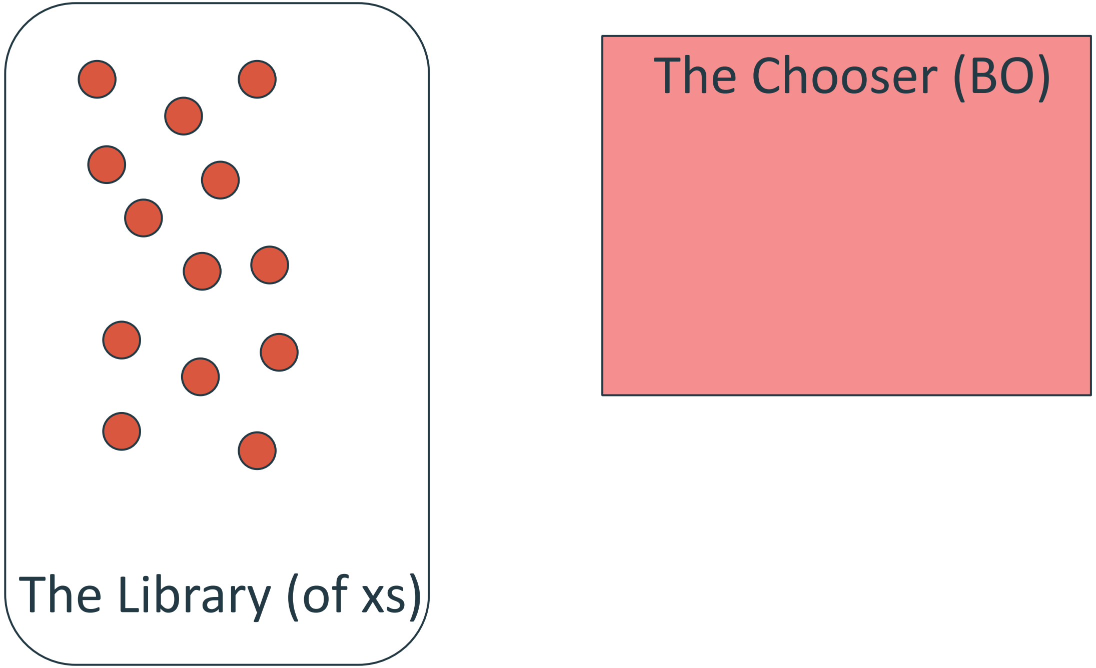
$$\max_{x\in\mathcal{X}} f(x) \quad : P(C(x)=0) > t$$

$$\max_{x\in\mathcal{X}} f(x) \quad : P(x) > t$$

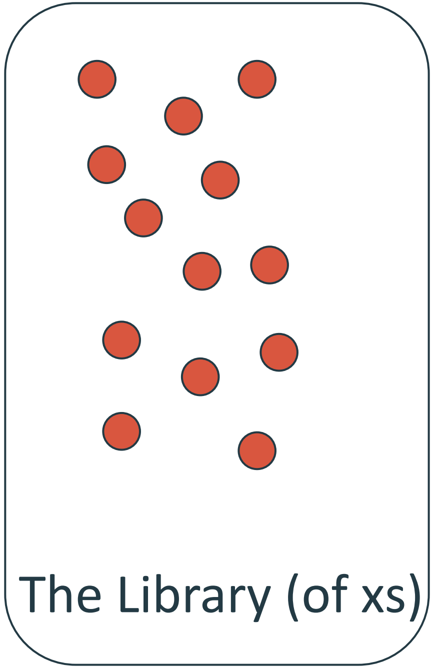
BO over **libraries of structures**



BO over **libraries of structures**



BO over **libraries of structures**

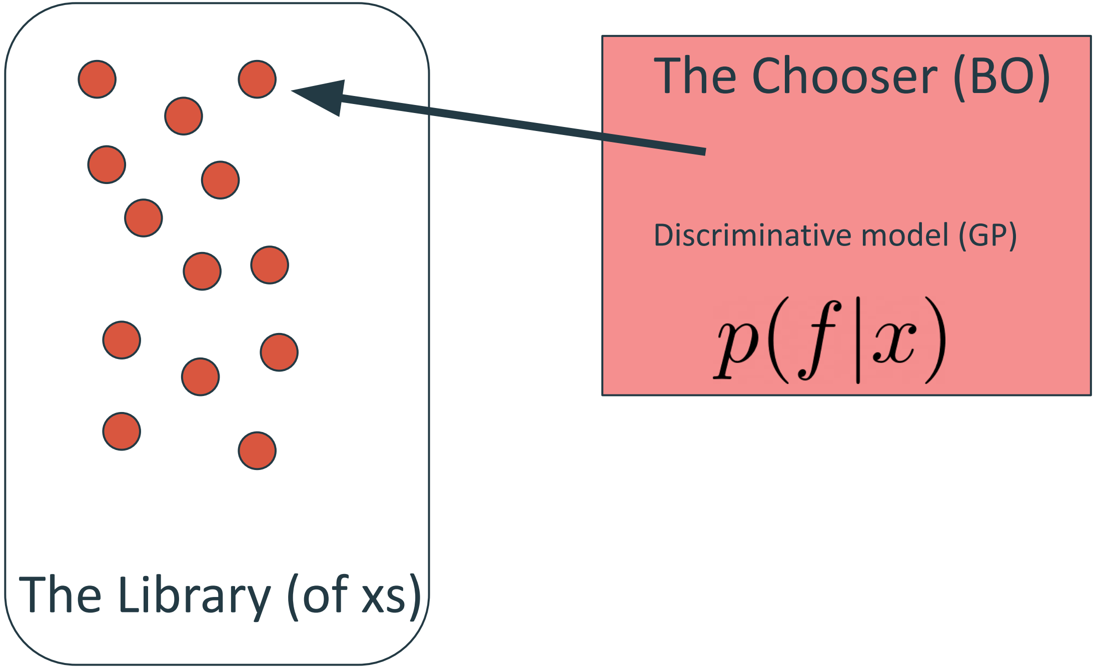


The Chooser (BO)

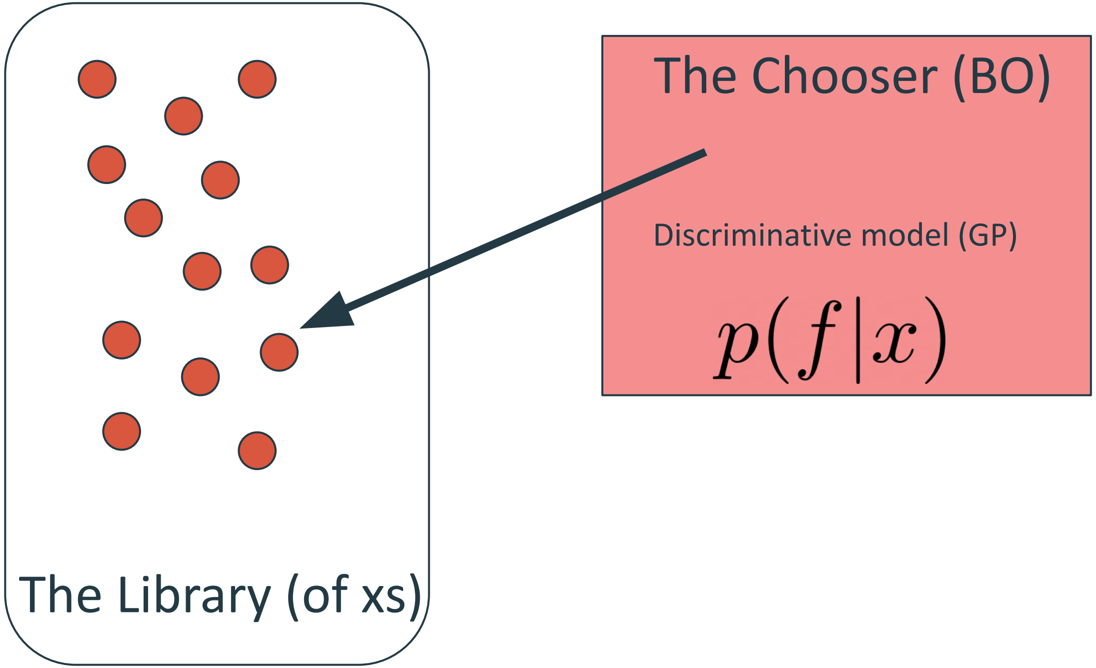
Discriminative model (GP)

$$p(f|x)$$

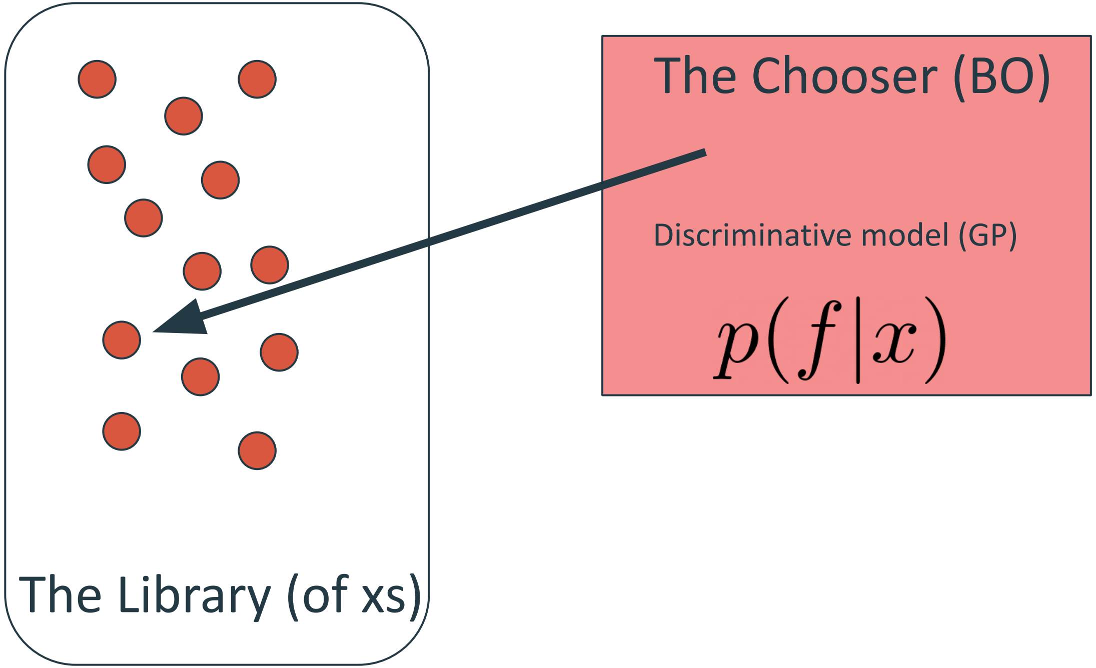
BO over **libraries of structures**



BO over **libraries of structures**



BO over **libraries of structures**



BO over "**highly structured**" spaces

Can we use $\mathbb{P}(\mathbf{x})$???

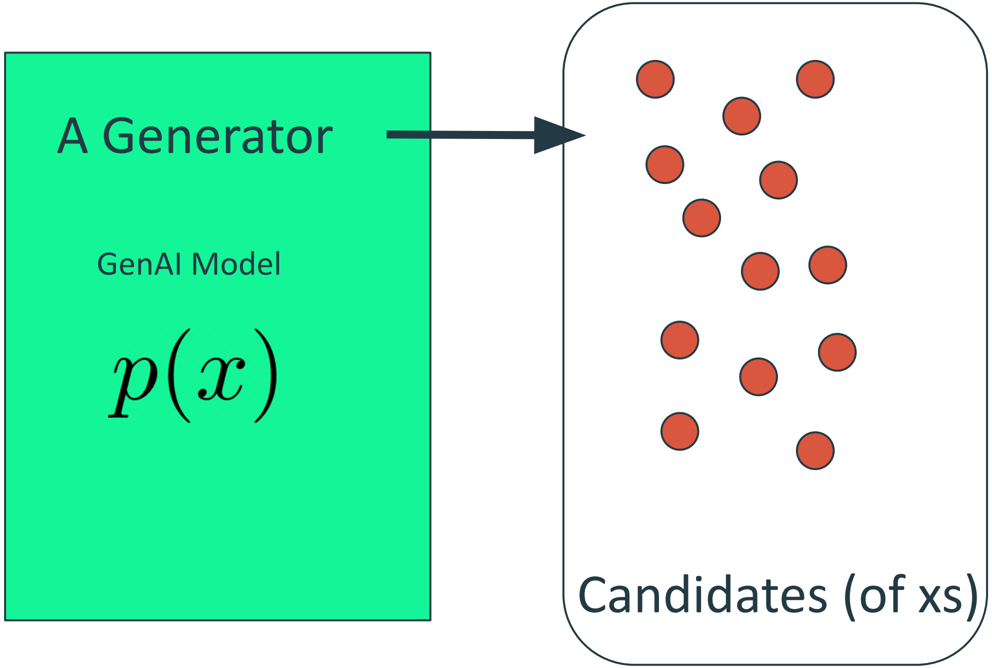
A Generator

GenAI Model

$p(x)$

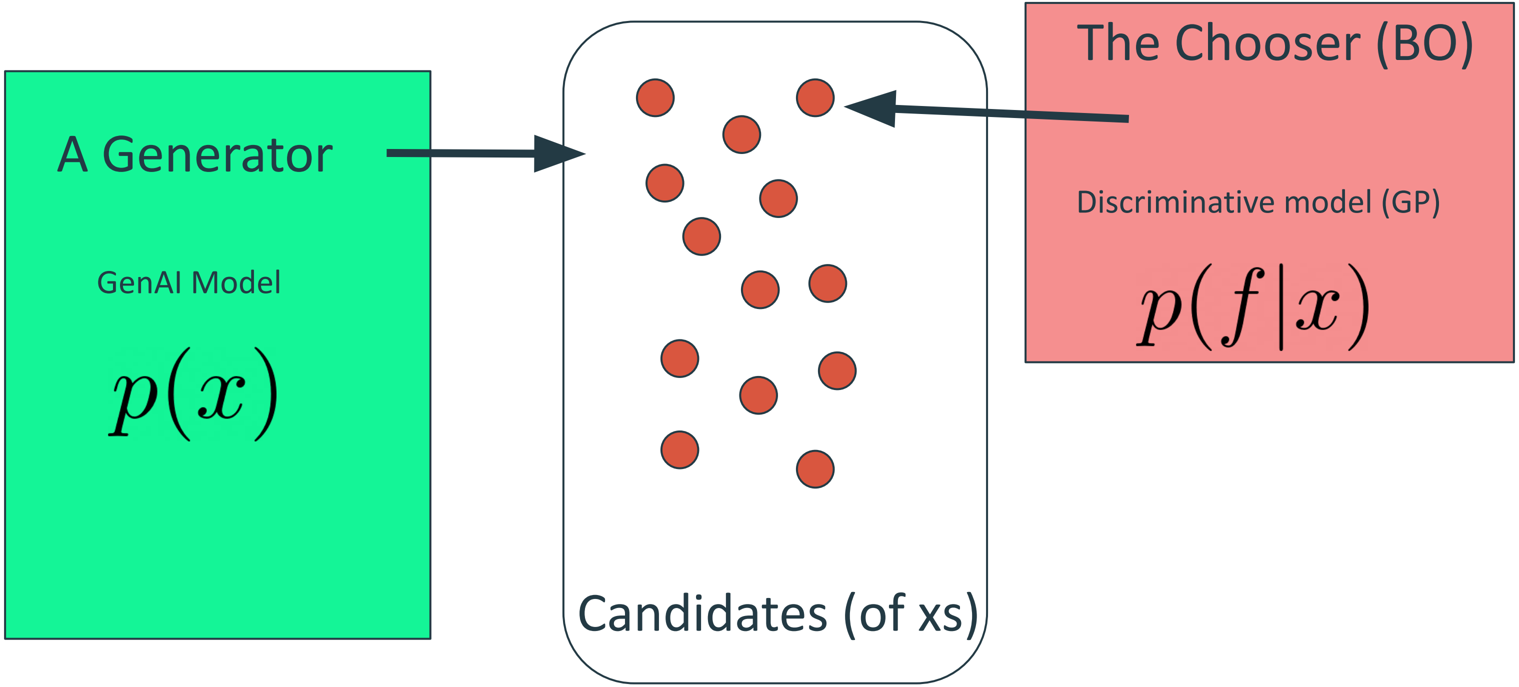
BO over "**highly structured**" spaces

Can we use $\mathbb{P}(\mathbf{x})$???



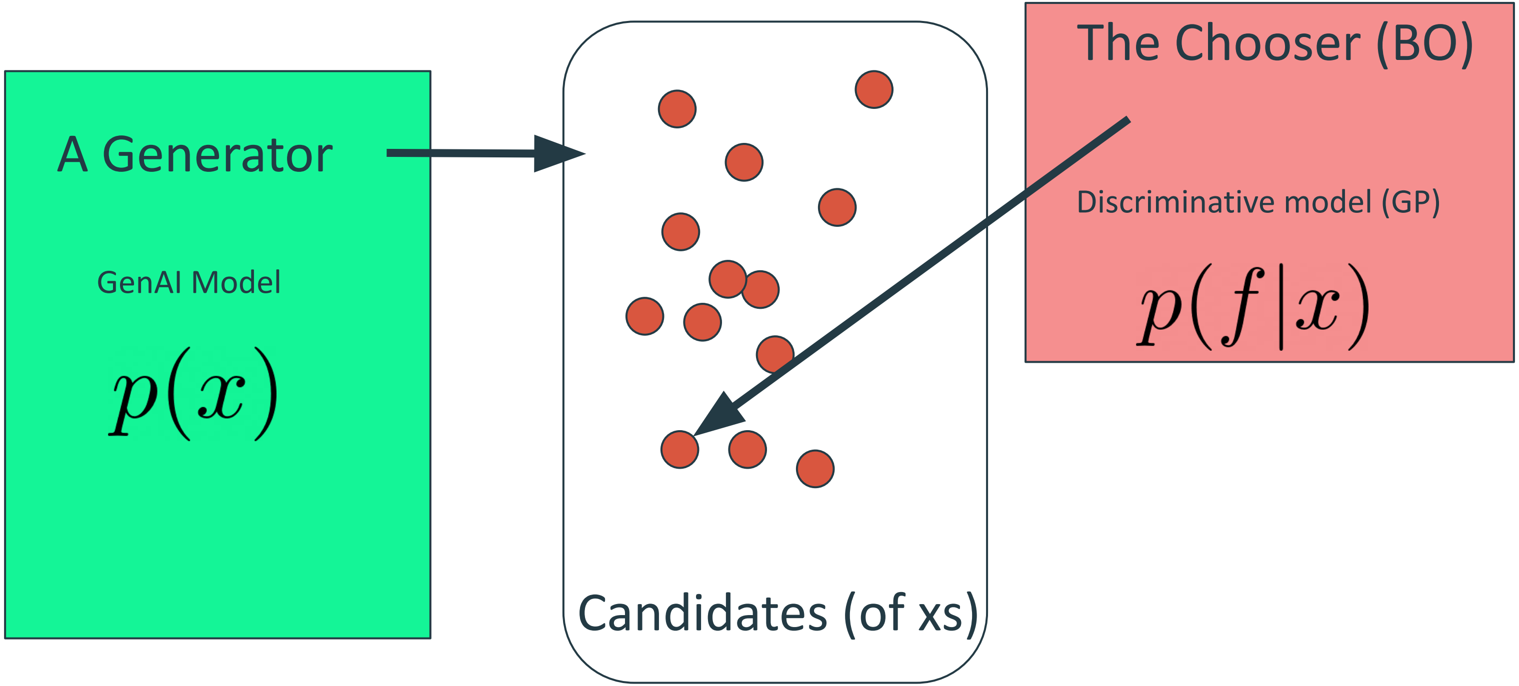
BO over “**highly structured**” spaces

Can we use $\mathbb{P}(\mathbf{x})$???



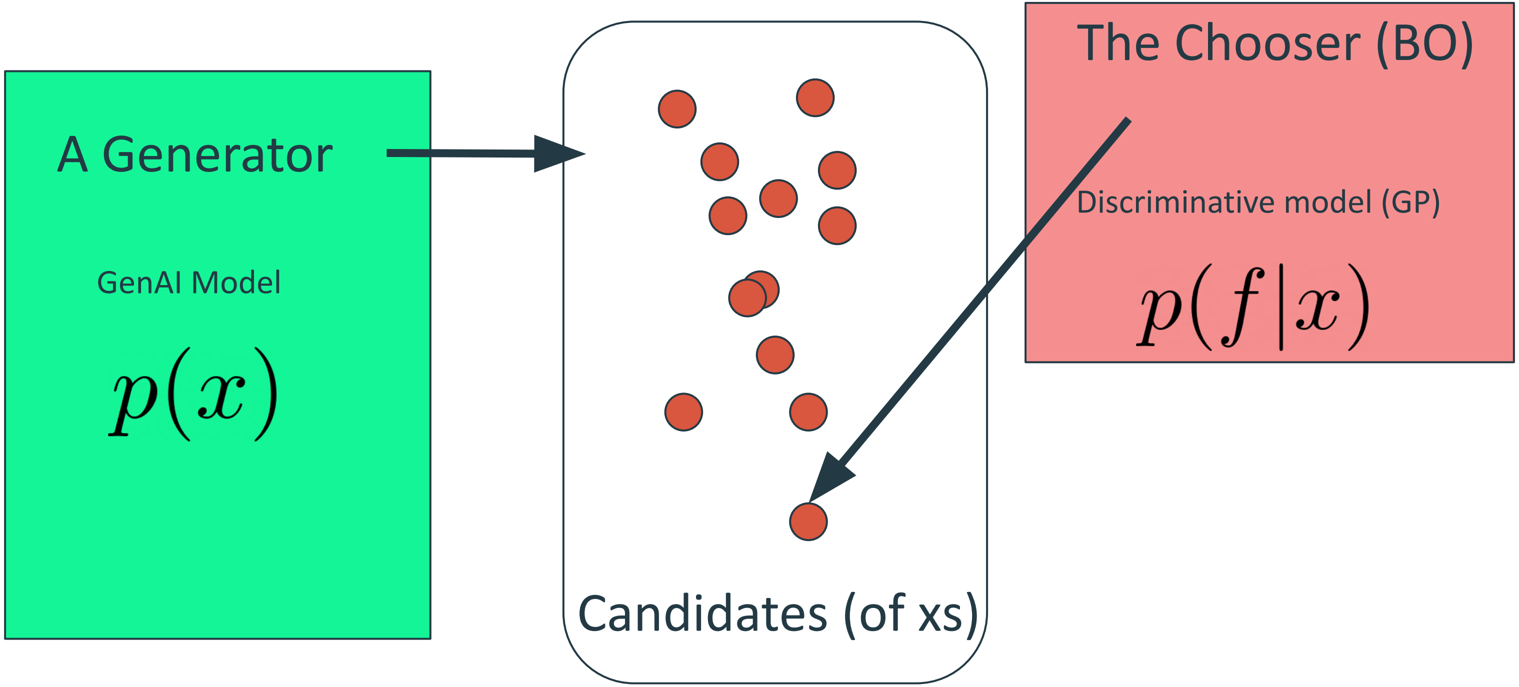
BO over "**highly structured**" spaces

Can we use $\mathbb{P}(\mathbf{x})$???



BO over “**highly structured**” spaces

Can we use $\mathbb{P}(\mathbf{x})$???



GenAI is built for sampling

Initial design: $$\mathbf{x}_0 \sim p(\mathbf{x})$$

GenAI is built for sampling: BO?

Initial design: $$\mathbf{x}_0 \sim p(\mathbf{x})$$

BO steps: $$\mathbf{x}_N \sim p(\mathbf{x} | f_{\mathbf{x}} > f^*, D_N)$$

PoI from GP

GenAI is built for sampling: BO?

Initial design: $$\mathbf{x}_0 \sim p(\mathbf{x})$$

BO steps: $$\mathbf{x}_N \sim p(\mathbf{x} | f_{\mathbf{x}} > f^*, D_N)$$

PoI from GP

- Sampling not optimising
- Model over $\mathcal{X}$ (or $\mathcal{Z}$)

How do we “condition” GenAI models?

- Train conditional model (expensive, requires specificity)
- Fine-tuning on the fly (still expensive / fragile)
- Model-specific (e.g. guidance for flow-matching)
- **MCMC (???)**
- **Mess around with latents (exploit locality)**

Jamiroquai

文A 45 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Jamiroquai (/dʒəˈmɪrəkwaɪ/ *jə-MIRR-ə-kwy*) are an English acid jazz and funk band from London. Formed in 1992, they are fronted by vocalist Jay Kay, and were prominent in the London-based funk and acid jazz movement of the 1990s. They built on their acid jazz sound in their early releases and later drew from rock, disco, electronic and Latin music genres. Lyrically, the group have addressed social and environmental justice. Kay has remained the only constant member through several line-up changes.

Jamiroquai

Jamiroquai performing at the Coachella Music Festival in 2018

The band made their debut under Acid Jazz Records but subsequently found mainstream success under Sony. While under this label, three of their albums have

The Return of the Space Cowboy

Article Talk

From Wikipedia, the free encyclopedia

The Return of the Space Cowboy is the second album by English funk and acid jazz band Jamiroquai. The album was released on 17 October 1994 under Sony Soho

Anyone?

Return of the **latent space** COWBOYS

Categorical Optimisation With Belief Of underlYing Smoothness

Henry Moss,
Sebastian Ober,
Tom Diethe, Head of the Centre for AI, Data Science & AI, Biopharma R&D, AstraZeneca, Cambridge, UK

AstraZeneca
Lancaster University
UNIVERSITY OF CAMBRIDGE

VAEs are built for sampling

Initial design: $\mathbf{x}_0 \sim p(\mathbf{x})$

VAEs are built for sampling

Initial design: $\mathbf{x}_0 \sim p(\mathbf{x})$

COWBOYS chooses: $\mathbf{x}_N \sim p(\mathbf{x} | f_{\mathbf{x}} > f^*, D_N)$

VAEs are built for sampling

Initial design: $$\mathbf{x}_0 \sim p(\mathbf{x})$$

COWBOYS chooses: $$\mathbf{x}_N \sim p(\mathbf{x} | f_{\mathbf{x}} > f^*, D_N)$$

- Sampling not optimising
- Tanimoto Model over $\mathcal{X}$

VAEs are built for sampling

Initial design: $\mathbf{x}_0 \sim p(\mathbf{x})$

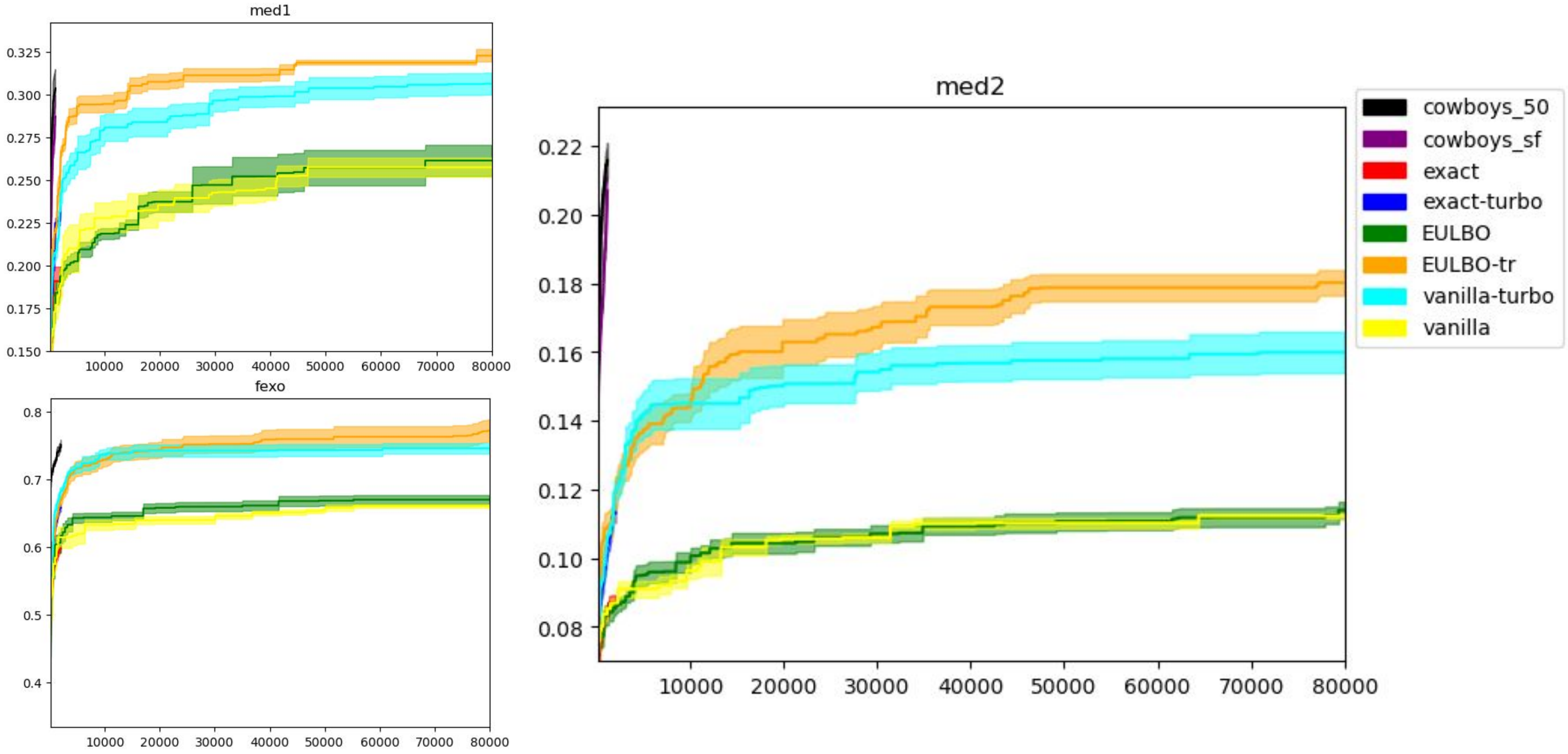
COWBOYS chooses: $\mathbf{x}_N \sim p(\mathbf{x} | f_{\mathbf{x}} > f^*, D_N)$

- Sampling not optimising
- Tanimoto Model over $\mathcal{X}$

So use MCMC:

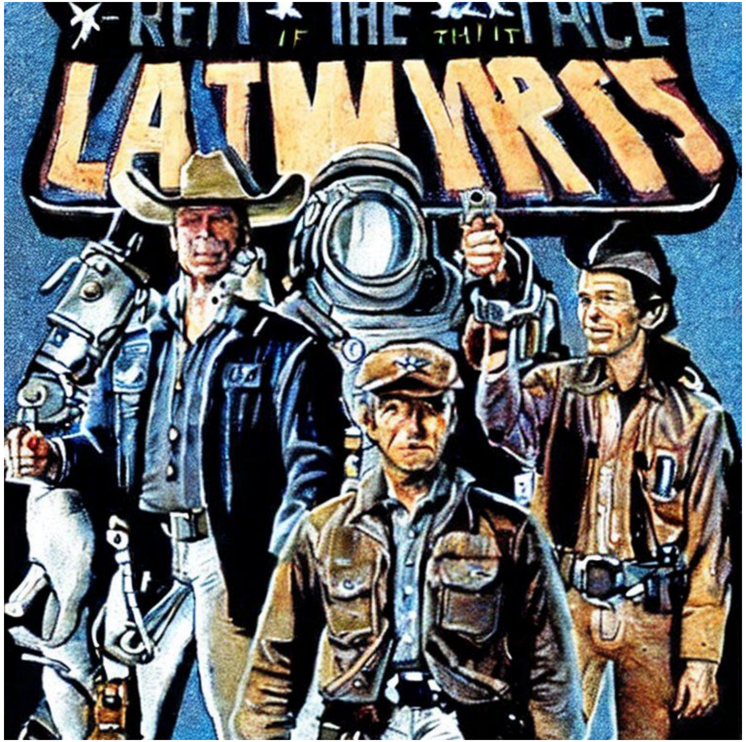
1. Pre-conditioned Crank Nicolson
2. Parallel Tempering

Toy Problem: Guacamole Molecule Opt



ICML 2025

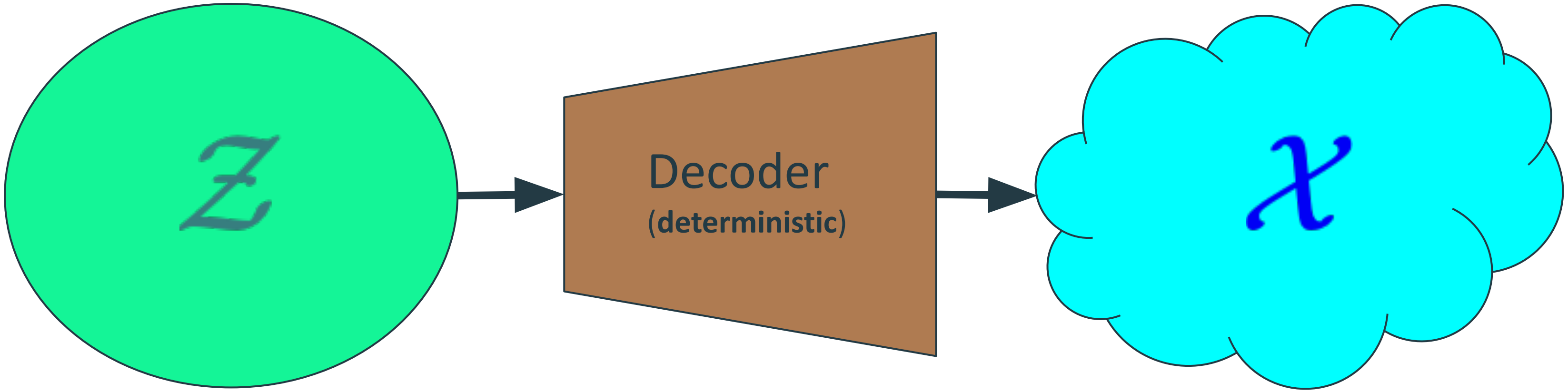
- Sample, don't optimize
- Beware of the geometry of latent spaces
- Still loads to do here!



Linear combinations of Latents in Generative Models: Subspaces and Beyond

Erik Bodin
Carl Henrik Ek
Henry Moss

Priors over "**Highly Structured**" spaces

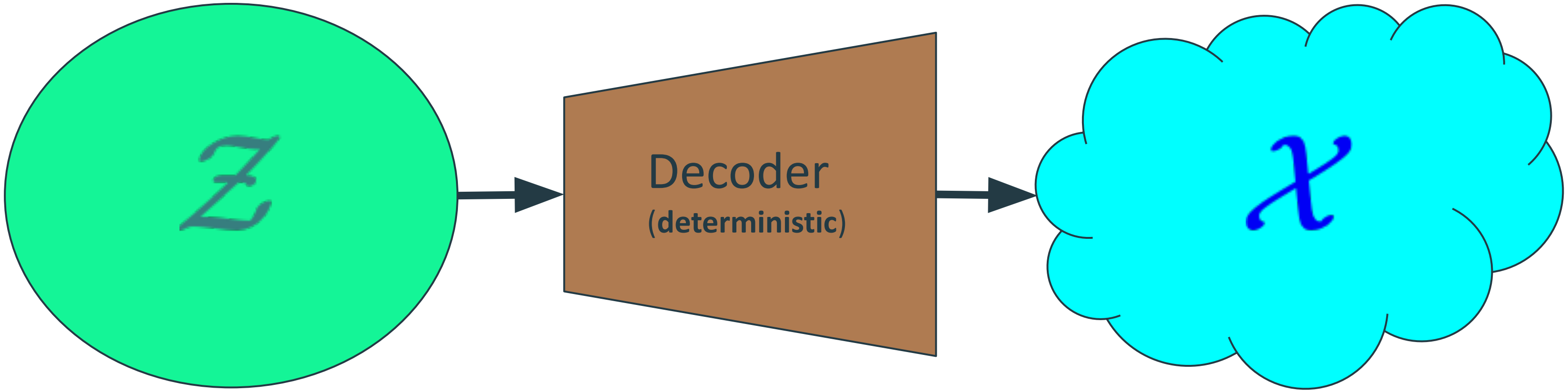


$$\mathbb{P}(\mathbf{z}) = \mathcal{N}_d(\mathbf{0}, I_d)$$

$$\mathbb{P}(\mathbf{x}) = ???$$

Diffusion / Flow matching e.t.c (d>>10_000)

Priors over "**Highly Structured**" spaces



$\mathbb{P}(\mathbf{z}) = \mathcal{N}_d(\mathbf{0}, I_d)$

$\mathbb{P}(\mathbf{x}) = ???$

Only give the model what it is trained to expect!

https://www.inf.fu-berlin.de/lehre/WS17/SemAlg/notes/02_highdim2.pdf

An aside: **typical** Gaussian samples

Theorem 2 (Gaussian Annulus Theorem). *The Gaussian Annulus Theroem states, that nearly all the probability of a spherical Gaussian with unit variance is concentrated in a thin annulus of width* $O(1)$ *at radius* $\sqrt{d}$.

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I_d) \Rightarrow |\mathbf{z}|^2 \sim \chi^2(d)$$

Common “latent space manipulations”

- Interpolation
- Addition
- Centroids (averaging)

Common “latent space manipulations”

- Interpolation
- Addition
- Centroids (averaging)
- **Subspaces!**

Common “latent space manipulations”

- Interpolation
- Addition
- Centroids (averaging)
- **Subspaces!**

Only give the model what it is trained to expect!

Common "latent space manipulations"

- Interpolation
- Addition
- Centroids (averaging)
- **Subspaces!**

Only give the model what it is trained to expect!

$$\mathbf{z} \sim \mathcal{N}(0, I_d)$$

Common "latent space manipulations"

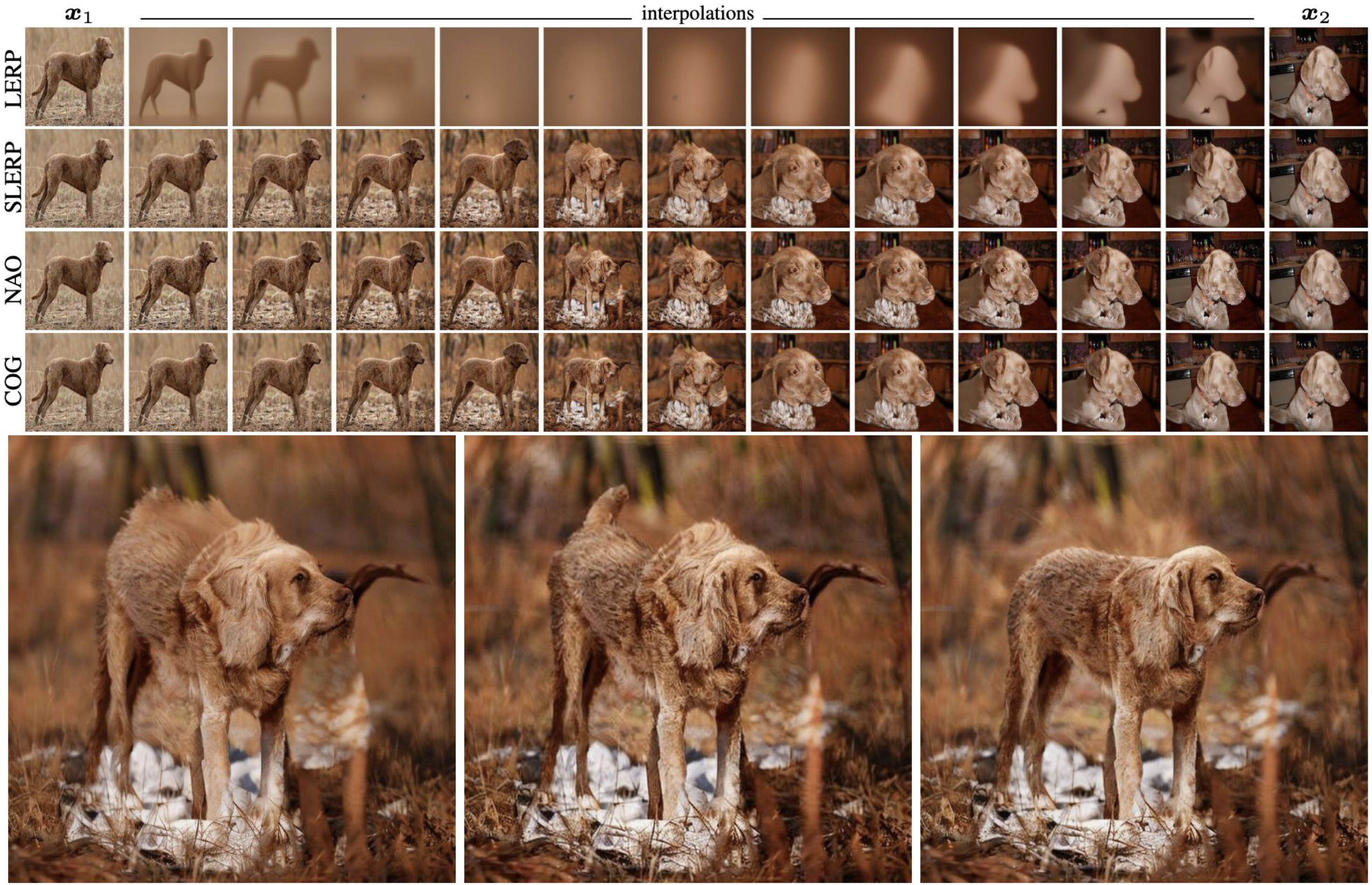
- Interpolation
- Addition
- Centroids (averaging)
- **Subspaces!**

Only give the model what it is trained to expect!

$$\mathbf{z} \sim \mathcal{N}(0, I_d)$$

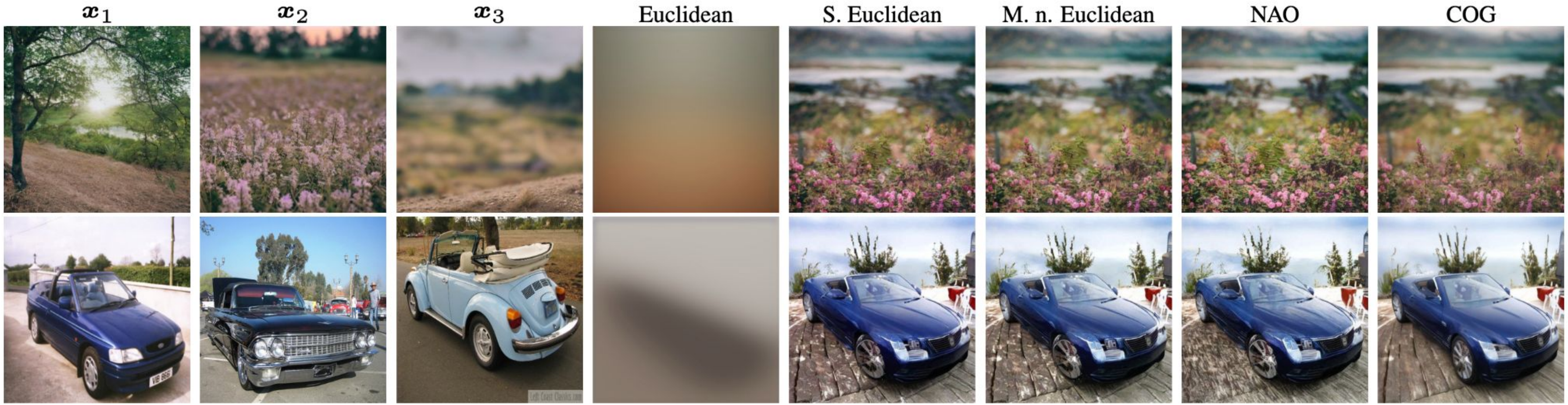
COG: LINEAR COMBINATIONS OF GAUSSIAN LATENTS

Interpolation



Interpolation

Centroid determination



Centroid determination

Common “latent space manipulations”

$$\boldsymbol{y} := \sum_{k=1}^{K} w_k \boldsymbol{x}_k = \boldsymbol{w}^T \boldsymbol{X},$$

Common “latent space manipulations”

$$\boldsymbol{y} := \sum_{k=1}^{K} w_k \boldsymbol{x}_k = \boldsymbol{w}^T \boldsymbol{X},$$

- **Interpolation:** $\boldsymbol{y} = \boldsymbol{w}^T \boldsymbol{X}$, where $\boldsymbol{w} = [w_1, 1 - w_1]$, $w_1 \in [0, 1]$, and $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$.

Common “latent space manipulations”

$$\boldsymbol{y} := \sum_{k=1}^{K} w_k \boldsymbol{x}_k = \boldsymbol{w}^T \boldsymbol{X},$$

- **Interpolation:** $\boldsymbol{y} = \boldsymbol{w}^T \boldsymbol{X}$, where $\boldsymbol{w} = [w_1, 1 - w_1]$, $w_1 \in [0, 1]$, and $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$.
- **Centroids:** $\boldsymbol{y} = \boldsymbol{w}^T \boldsymbol{X}$, where $\boldsymbol{w} = [\frac{1}{K}]^K$, and $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K]$.

Common "latent space manipulations"

$$\boldsymbol{y} := \sum_{k=1}^{K} w_k \boldsymbol{x}_k = \boldsymbol{w}^T \boldsymbol{X},$$

- **Interpolation:** $\boldsymbol{y} = \boldsymbol{w}^T \boldsymbol{X}$, where $\boldsymbol{w} = [w_1, 1 - w_1]$, $w_1 \in [0, 1]$, and $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$.
- **Centroids:** $\boldsymbol{y} = \boldsymbol{w}^T \boldsymbol{X}$, where $\boldsymbol{w} = [\frac{1}{K}]^K$, and $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K]$.
- **Subspace projections:**

$$\boldsymbol{y} = s(\boldsymbol{x}) = \boldsymbol{U}\boldsymbol{U}^T \boldsymbol{x} = \boldsymbol{U}\boldsymbol{h}$$

LOL: Linear combinations Of Latents

You have to give the model what it is trained to expect!

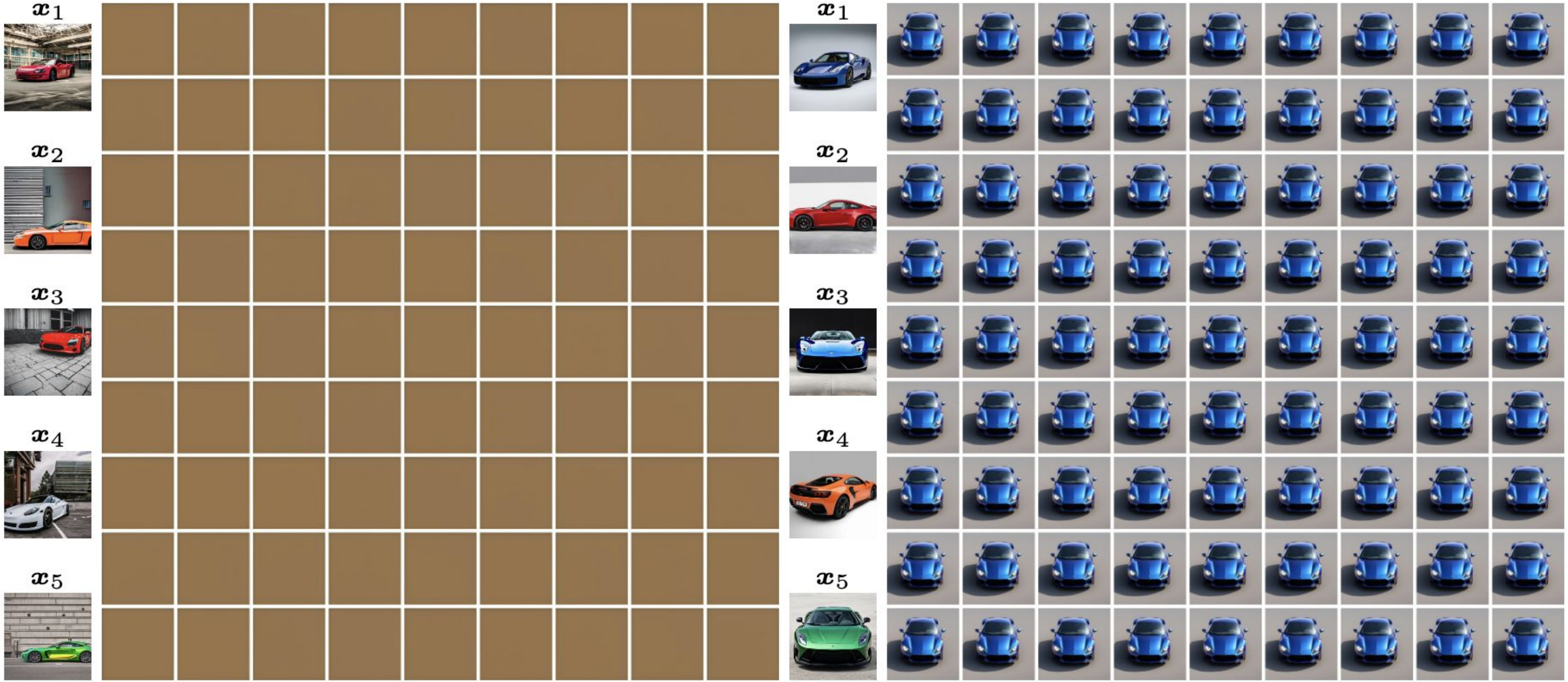
LOL: Linear combinations Of Latents

$$\boldsymbol{y} := \sum_{k=1}^{K} w_k \boldsymbol{x}_k = \boldsymbol{w}^T \boldsymbol{X},$$

You have to give the model what it is trained to expect!

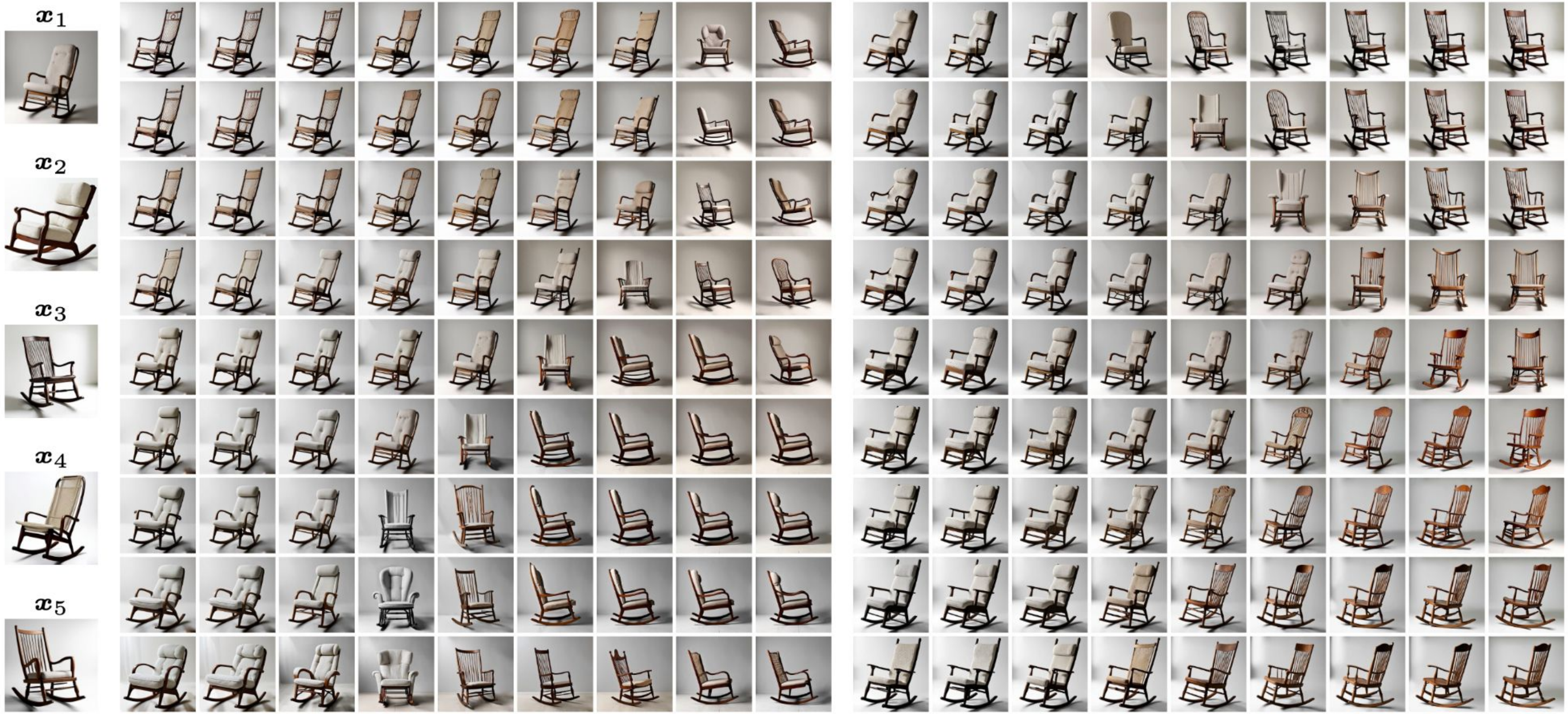
Instead decode $\mathbf{z} = \frac{\mathbf{y}}{\sqrt{(\beta)}} \sim \mathcal{N}(0, I)$ $\quad \beta = \sum_{k=1}^{K} w_k^2.$

Low-dimensional (searchable) subspaces without LOL

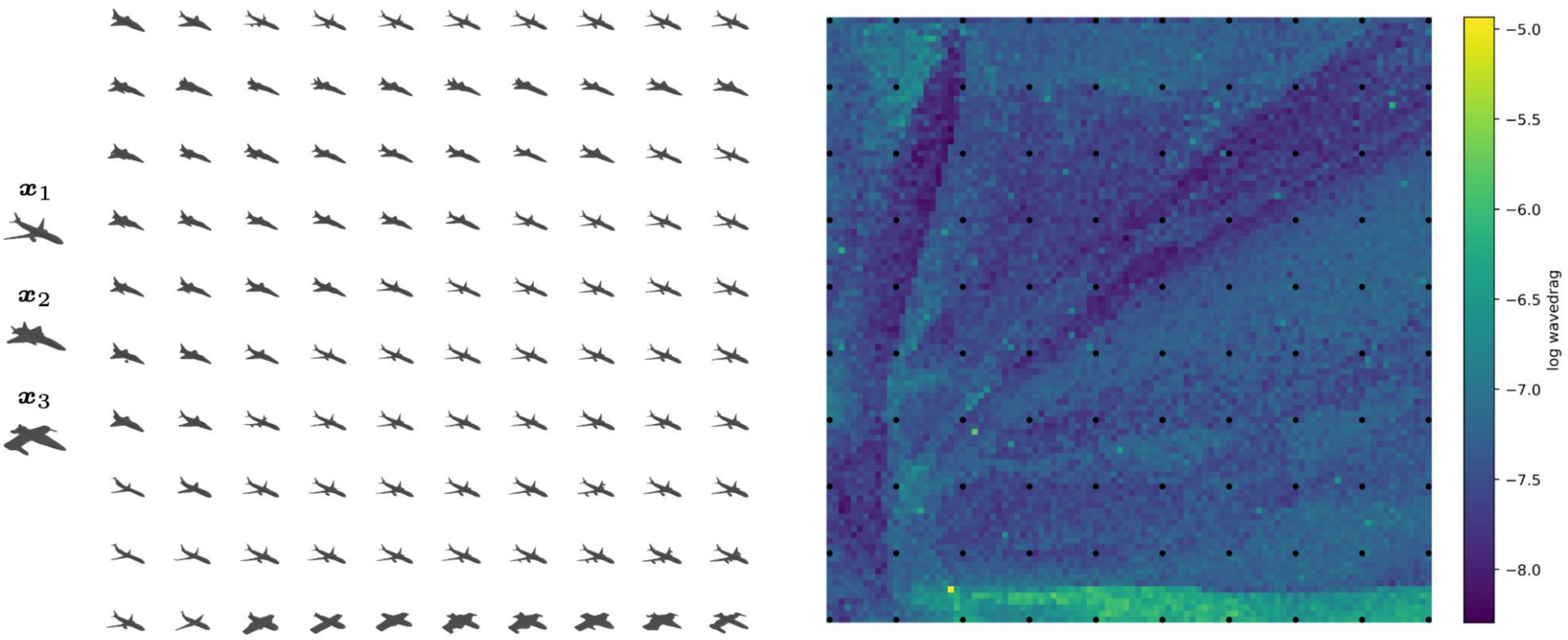


Low-dimensional (searchable) subspaces with LOL

More Low-dimensional (searchable) subspaces

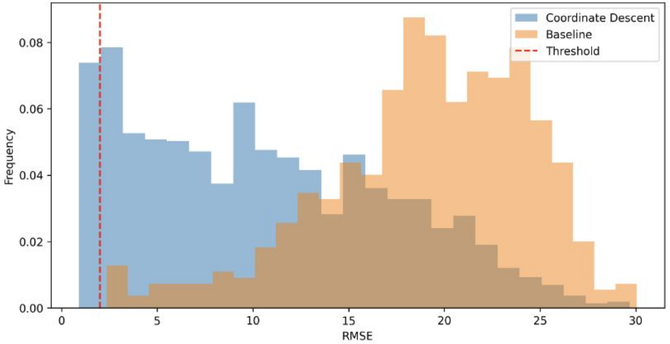


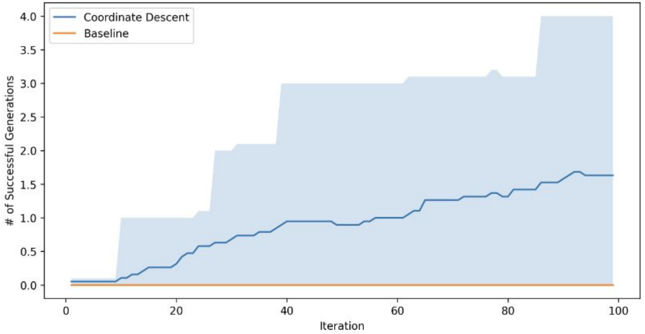
Even more Low-dimensional (searchable) subspaces



Find long “realistic protein”, using RFDiffusion

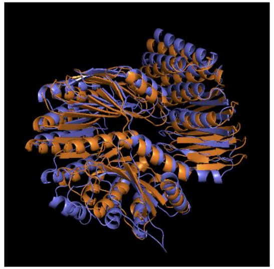
RFDiffusion -> ProteinMPNN -> Alphafold



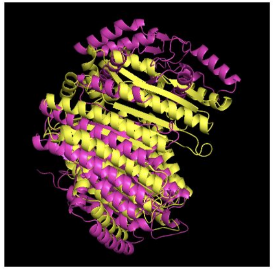


Work in progress: optimisation in latent space of RFdiffusion

Best random design

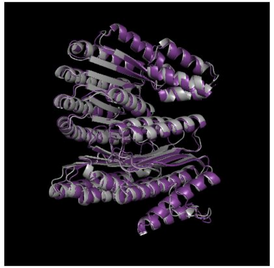


5th best random design

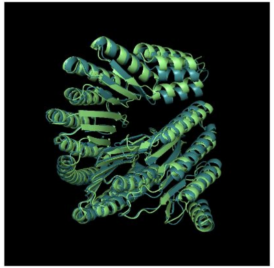


10th best random design

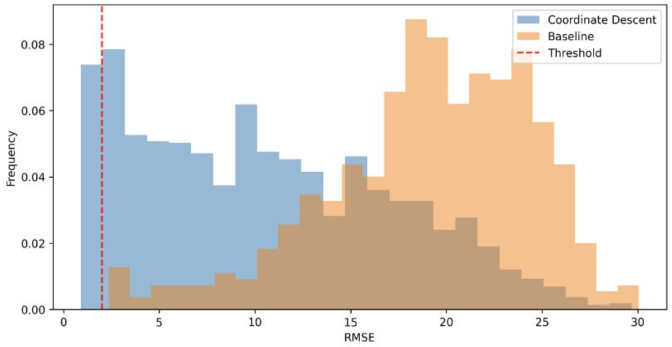
best from coordinate descent

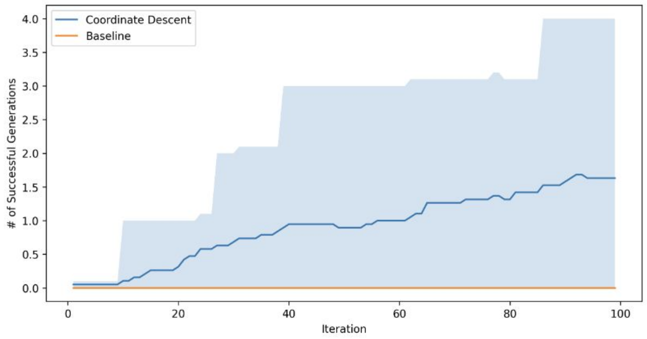


5th best from coordinate descent



10th best from coordinate descent





Work in progress: optimisation in latent space of RFDiffusion

ICLR 2025:

- Perform any linear combination of latents
- More general distributions
- Statistical tests to assert validity of inversions

- Now doing optimisation in these subspaces!

More interpolation (audio and video)