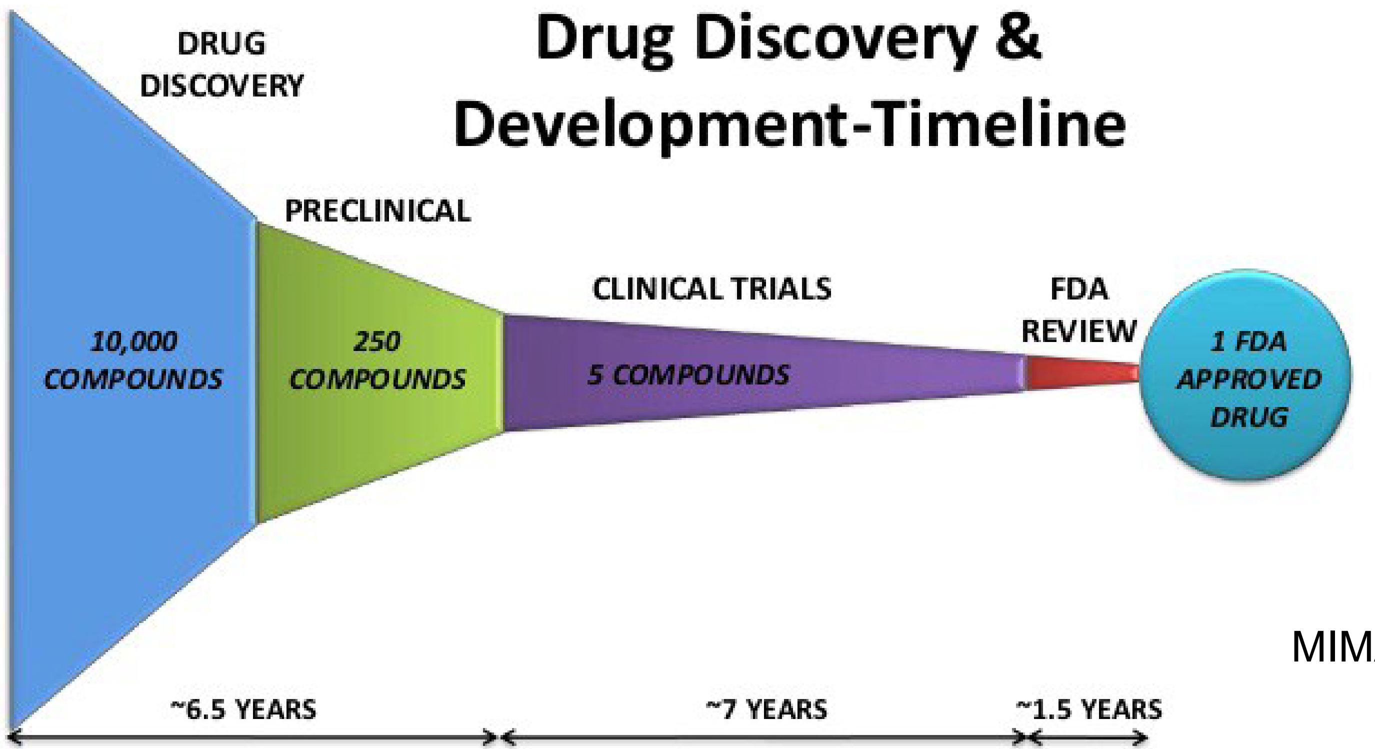
Bayesian Optimization over Combinatorial Spaces

Application #1: Drug/Vaccine Design

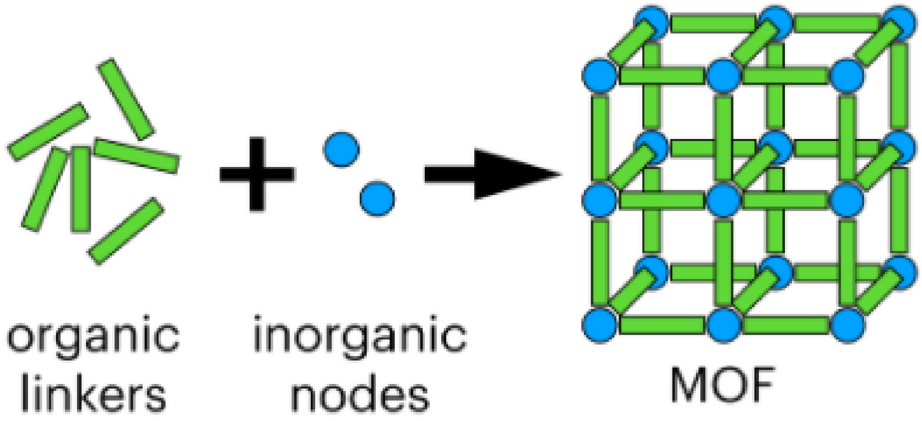


Credit:

MIMA healthcare

- Accelerate the discovery of promising designs

Application #2: Nanoporous Materials Design

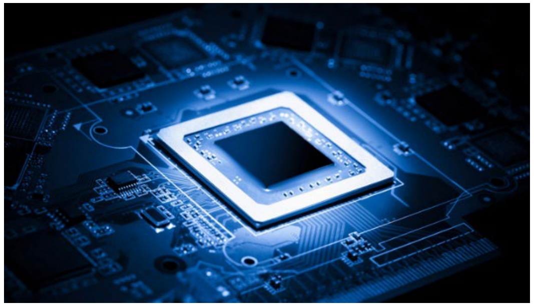


- **Sustainability applications**
  - Storing gases (e.g., hydrogen powered cars)
  - Separating gases (e.g., carbon dioxide from flue gas of coalfired power plants)
  - Detecting gases (e.g., detecting pollutants in outdoor air)

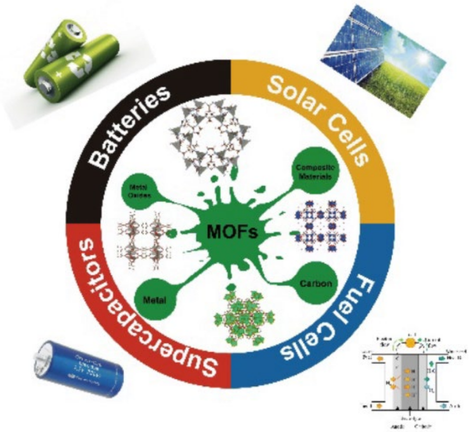
Combinatorial BO: The Problem

- **Goal:** find optimized combinatorial structures

Drug design



Hardware design



Material design

- Many other science and engineering applications

Combinatorial BO: The Problem

- **Given:** a combinatorial space of structures X (e.g., sequences, graphs) and an expensive black-box function $f(x \in X)$ to evaluate each structure $x \in X$

- **Find:** optimized combinatorial structure x^*

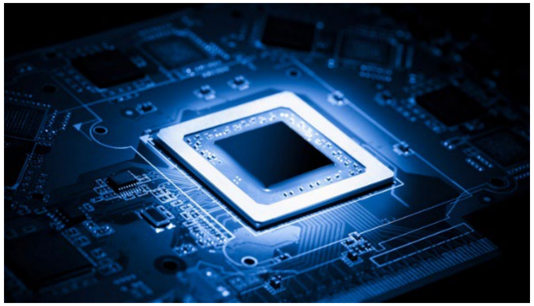
$$\boxed{x^* = arg \max_{x \in X} f(x)}$$

- **Evaluation:** number of function evaluations to (approximately) optimize $f(x)$

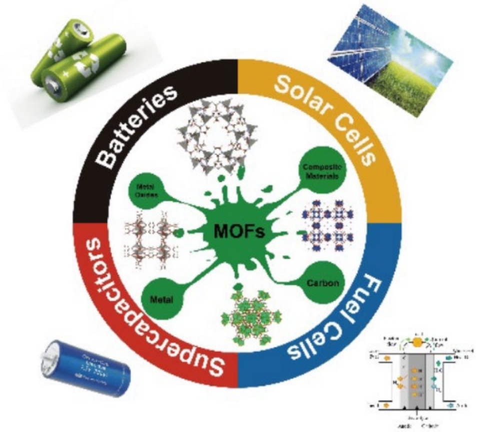
Combinatorial BO: Challenges

- **Goal:** find optimized combinatorial structures

Drug design



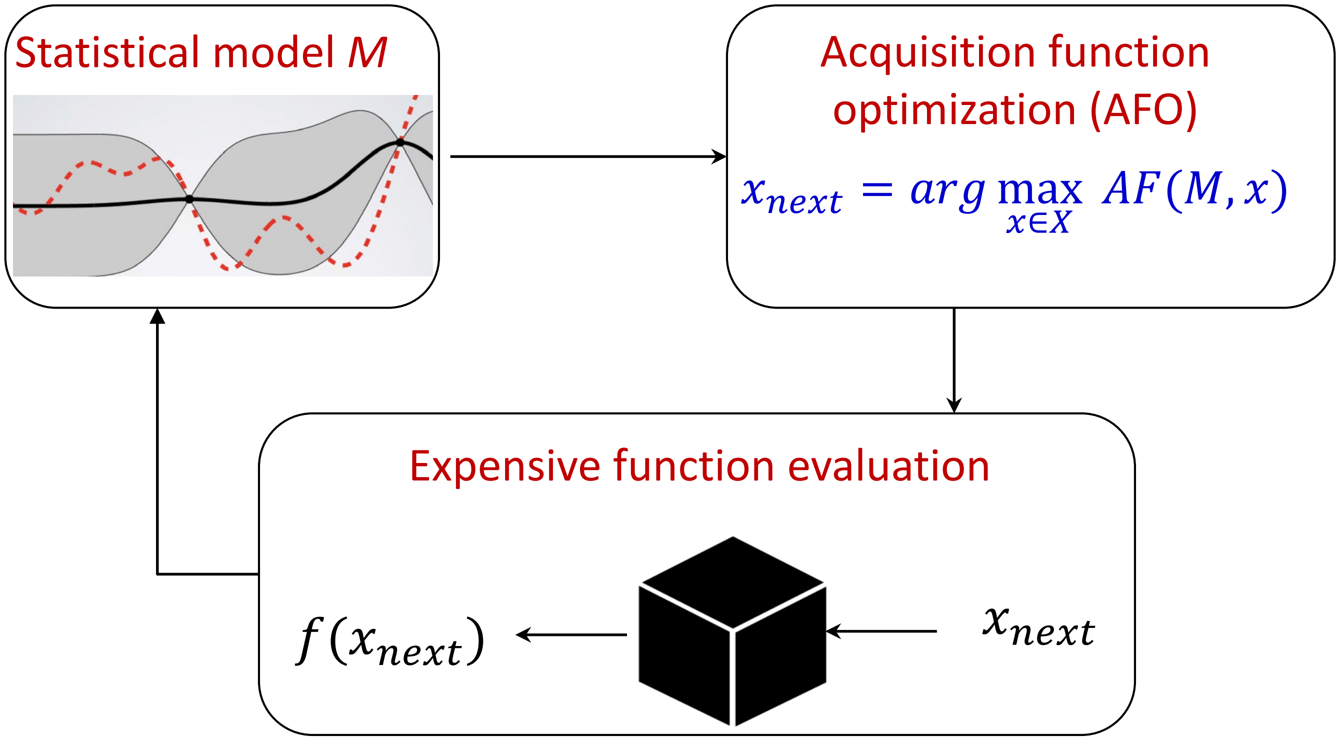
Hardware design



Material design

- **Challenges**
  - Evaluating each candidate design is expensive
  - Large combinatorial space of designs (e.g., sequences, graphs)

Combinatorial BO: Technical Challenges



- Effective modeling over combinatorial structures (e.g., sequences, graphs)
- Solving hard combinatorial optimization problem to select next structure

Definition of Combinatorial Space

- **Space of binary structures $X = \{0,1\}^n$**
  - Each structure $x \in X$ be represented using n binary variables $x_1, x_2, \dots, x_n$

- **Categorical variables**
  - x_i can take more than two candidate values

- **How to deal with categorical variables?**
  - Option 1: Encode them as binary variables (a common practice)
  - Option 2: Modeling and reasoning over categorical variables

Combinatorial BO: Summary of Approaches

- Trade-off complexity of model and tractability of AFO

- Simple statistical models and tractable search for AFO
  - BOCS [Baptista et al., 2018]

- Complex statistical models and heuristic search for AFO
  - SMAC [Hutter et al., 2011] and COMBO [Oh et al., 2019]

- Complex statistical models and tractable/accurate AFO
  - L2S-DISCO [Deshwal et al., 2020] and MerCBO [Deshwal et al., 2021]
  - Reduction to continuous BO [Gómez-Bombarelli et al., 2018]…

Aside: Combinatorial BO vs. Structured Prediction

- **Structured prediction (SP)** [Lafferty et al., 2001] [Bakir et al., 2007]
  - Generalization of classification to structured outputs (e.g., sequences, trees, and graphs)
    - POS tagging, parsing, information extraction, image segmentation
  - CRFs, Structured Perceptron, Structured SVM

- Complexity of cost function vs. tractability of inference
  - Simple cost functions (e.g., first-order) and tractable inference
  - Complex cost functions (e.g., higher-order) and heuristic inference
  - Learning to search for SP [Daume' et al., 2009] [Doppa et al., 2014]

- **Key Difference:** Small data vs. big data setting

Combinatorial BO: Summary of Approaches

- Trade-off complexity of model and tractability of AFO

- Simple statistical models and tractable search for AFO
  - BOCS [Baptista et al., 2018]

- Complex statistical models and heuristic search for AFO
  - SMAC [Hutter et al., 2011] and COMBO [Oh et al., 2019]

- Complex statistical models and tractable/accurate AFO
  - L2S-DISCO [Deshwal et al., 2020] and MerCBO [Deshwal et al., 2021]
  - Reduction to continuous BO [Gómez-Bombarelli et al., 2018]…

BOCS Algorithm [Baptista et al., 2018]

- Linear surrogate model over binary structures
  - $f(x \in X) = \theta^T . \phi(x)$
  - $\phi(x)$ consists of up to Quadratic (second-order) terms
  - $\phi(x) = [x_1, x_2, \dots, x_d, x_1 . x_2, x_1 . x_3, \dots, x_{d-1} . x_d]$
- Thompson sampling as acquisition function
- Acquisition function optimization
  - Binary quadratic program

$$x_{next} = arg \max_{x \in \{0,1\}^d} b^T x + x^T A\, x$$

BOCS Algorithm [Baptista et al., 2018]

- Linear surrogate model over binary structures
  - $f(x \in X) = \theta^T . \phi(x)$
  - $\phi(x)$ consists of up to Quadratic (second-order) terms
  - $\phi(x) = [x_1, x_2, \ldots, x_d, x_1 . x_2, x_1 . x_3, \ldots, x_{d-1} . x_d]$

May not be sufficient to capture desired dependencies

- Thompson sampling as acquisition function

- Acquisition function optimization
  - Binary quadratic program

$$x_{next} = arg \max_{x \in \{0,1\}^d} b^T x + x^T A\, x$$

BOCS Algorithm [Baptista et al., 2018]

- Linear surrogate model over binary structures
  - $f(x \in X) = \theta^T . \phi(x)$
  - $\phi(x)$ consists of up to Quadratic (second-order) terms
  - $\phi(x) = [x_1, x_2, \ldots, x_d, x_1 . x_2, x_1 . x_3, \ldots, x_{d-1} . x_d]$
- Thompson sampling as acquisition function
- Acquisition function optimization
  - Binary quadratic program

Cannot handle declarative constraints for valid structures

$$x_{next} = arg \max_{x \in \{0,1\}^d} \; b^T x + x^T A\, x$$

Combinatorial BO: Summary of Approaches

- Trade-off complexity of model and tractability of AFO

- Simple statistical models and tractable search for AFO
  - BOCS [Baptista et al., 2018]

- Complex statistical models and heuristic search for AFO
  - SMAC [Hutter et al., 2011] and COMBO [Oh et al., 2019]

- Complex statistical models and tractable/accurate AFO
  - L2S-DISCO [Deshwal et al., 2020] and MerCBO [Deshwal et al., 2021]
  - Reduction to continuous BO [Gómez-Bombarelli et al., 2018]…

SMAC Algorithm [Hutter et al., 2010, 2011]

- Random forest as surrogate model
  - works naturally for categorical variables
  - Prediction/Uncertainty (= empirical mean/variance over trees)

Uncertainty estimates can be poor

- Expected impr… n function

- Hand-designed local search with restarts for AFO

SMAC Algorithm [Hutter et al., 2010, 2011]

- Random forest as surrogate model
  - works naturally for categorical variables
  - Prediction/Uncertainty (= empirical mean/variance over trees)

- Expected improvement as acquisition function

- Hand-designed local search with restarts for AFO

Can potentially get stuck in local optima

COMBO Algorithm [Oh et al., 2019]

- GP with diffusion kernel [Kondor and Lafferty 2002]
  - Requires a graph representation of the input space X

$$\boxed{K(V,V) = exp(-\beta L(G))}$$

- Expected improvement as acquisition function

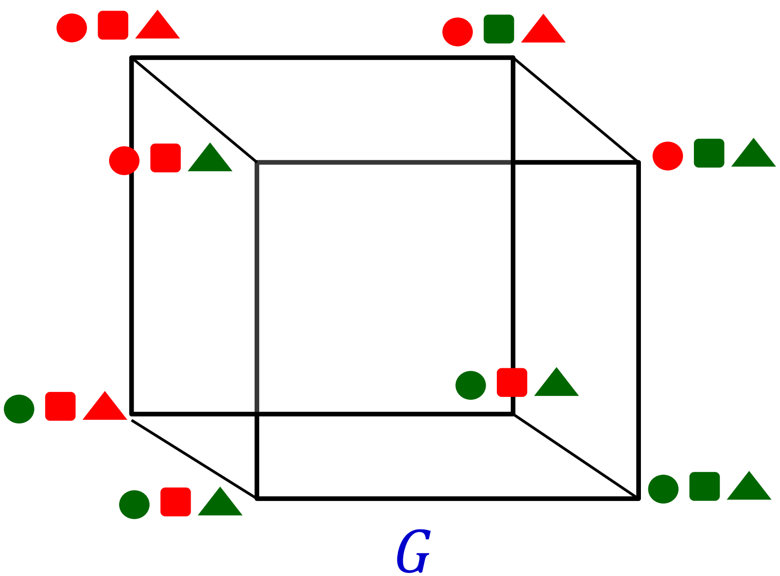
- Local search with random restarts for AFO

COMBO Algorithm [Oh et al., 2019]

- GP with diffusion kernel [Kondor and Lafferty 2002]
  - Requires a graph representation of the input space X

$$K(V, V) = exp(-\beta L(G))$$

- **Combinatorial graph representation** [Oh et al., 2019]



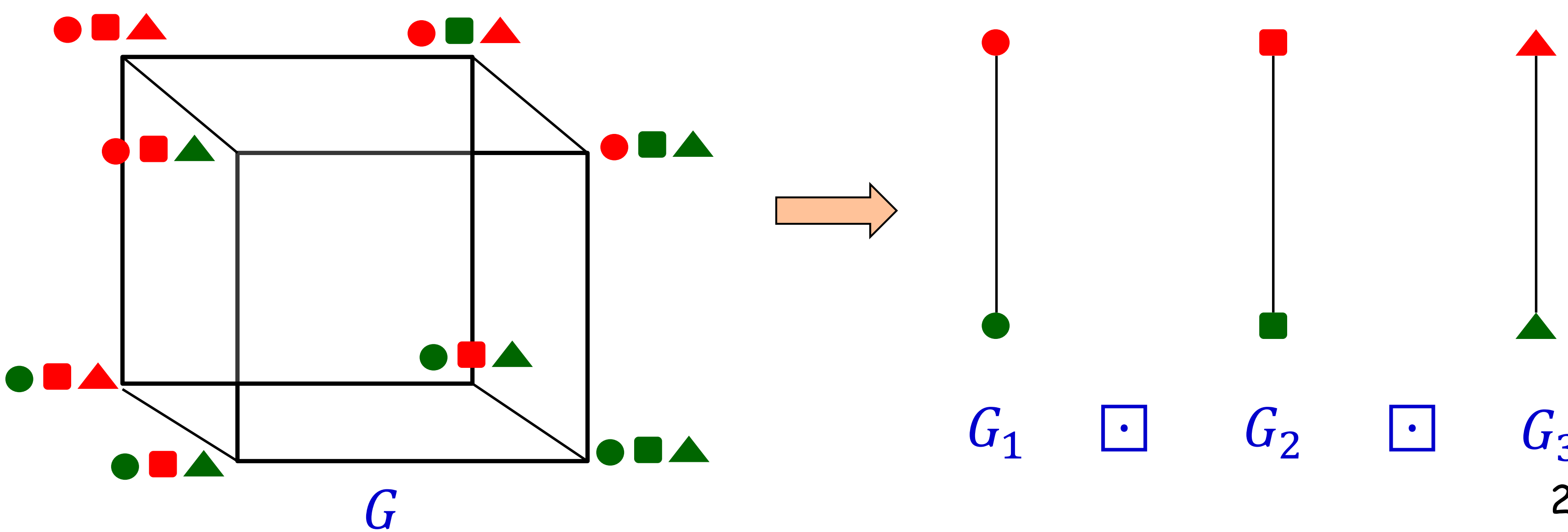
Each vertex is a candidate structure $x \in X$

COMBO Algorithm [Oh et al., 2019]

- GP with diffusion kernel [Kondor and Lafferty 2002]
  - Requires a graph representation of the input space X

$$K(V,V) = exp(-\beta L(G))$$

- **Combinatorial graph representation** [Oh et al., 2019]
  - Graph Cartesian product of subgraphs



COMBO Algorithm [Oh et al., 2019]

- GP with diffusion kernel [Kondor and Lafferty 2002]
  - Requires a graph representation of the input space X

Cannot use SOTA acquisition functions if we cannot sample functions from GP posterior

- Expected improvement as acquisition function

- Local search with random restarts for AFO

COMBO Algorithm [Oh et al., 2019]

- GP with diffusion kernel [Kondor and Lafferty 2002]
  - Requires a graph representation of the input space X

$$\boxed{K(V,V) = exp(-\beta L(G))}$$

- Expected improvement as acquisition function

- Local search with random restarts for AFO

Can potentially get stuck in local optima

Combinatorial BO: Summary of Approaches

- Trade-off complexity of model and tractability of AFO

- Simple statistical models and tractable search for AFO
  - BOCS [Baptista et al., 2018]

- Complex statistical models and heuristic search for AFO
  - SMAC [Hutter et al., 2011] and COMBO [Oh et al., 2019]

- Complex statistical models and tractable/accurate AFO
  - L2S-DISCO [Deshwal et al., 2020] and MerCBO [Deshwal et al., 2021]
  - Reduction to continuous BO [Gómez-Bombarelli et al., 2018]…

MerCBO Algorithm [Deshwal et al., 2021]

- Same surrogate model as COMBO
  - GP with discrete diffusion kernel and graph representation
- Thompson sampling as acquisition function
  - Mercer features allow sampling functions from GP posterior
- Acquisition function optimization
  - Binary quadratic program
  - Parametrized submodular relaxation (PSR) solver

$$\boxed{x_{next} = arg \max_{x \in \{0,1\}^d} b^T x + x^T A\, x}$$

MerCBO Algorithm [Deshwal et al., 2021]

- Same surrogate model as COMBO
  - GP with discrete diffusion kernel and graph representation
- Thompson sampling as acquisition function
  - Mercer features allow sampling functions from GP posterior
- Acquisition function optimization
  - Binary quadratic program
  - Parametrized submodular relaxation (PSR) solver

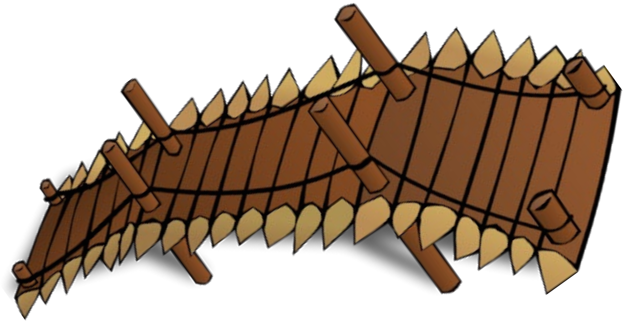
$$x_{next} = arg \max_{x \in \{0,1\}^d} b^T x + x^T A\, x$$

MerCBO: Acquisition Function

- Mercer features allow sampling functions from GP posterior

- Missing puzzle to leverage prior acquisition functions
  - Thompson Sampling (TS)
  - Predictive Entropy Search (PES)
  - Max-value Entropy Search (MES)
  - …

BO for continuous spaces



BO for discrete spaces

MerCBO: Mercer Features

- **Key Idea:** exploit the structure of combinatorial graph G to compute its eigenspace in closed-form

MerCBO: Mercer Features

- **Key Idea:** exploit the structure of combinatorial graph G to compute its eigenspace in closed-form

- Graph Laplacian $L(G)$ decomposes over those of sub-graphs

$$L(G) = L(G_1) \oplus L(G_2) \oplus L(G_3)$$

$\oplus$ is Kronecker sum operator

MerCBO: Mercer Features

- **Key Idea:** exploit the structure of combinatorial graph G to compute its eigenspace in closed-form

- Graph Laplacian $L(G)$ decomposes over those of sub-graphs

$$L(G) = L(G_1) \oplus L(G_2) \oplus L(G_3)$$

$\oplus$ is Kronecker sum operator

- [Hammack et al., 2011] Given two graphs G_1 and G_2 with the eigenspace of their Laplacians being $\{\lambda_1, U_1\}$ and $\{\lambda_2, U_2\}$ respectively, the eigenspace of $L(G_1 \boxdot G_2)$ is given by $\{\lambda_1 \bowtie \lambda_2, U_1 \otimes U_2\}$.

MerCBO: Mercer Features

- **Key Idea:** exploit the structure of combinatorial graph G to compute its eigenspace in closed-form

- Graph Laplacian $L(G)$ decomposes over those of sub-graphs

$$L(G) = L(G_1) \oplus L(G_2) \oplus L(G_3)$$

$\oplus$ is Kronecker sum operator

- [Hammack et al., 2011] Given two graphs G_1 and G_2 with the eigenspace of their Laplacians being $\{\lambda_1, U_1\}$ and $\{\lambda_2, U_2\}$ respectively, the eigenspace of $L(G_1 \boxdot G_2)$ is given by $\{\lambda_1 \bowtie \lambda_2, U_1 \otimes U_2\}$.

- Each G_i has eigenvalue {0,2} and eigenvectors {[1, 1], [1, -1]}

MerCBO: Mercer Features

- **Key Idea:** exploit the structure of combinatorial graph G to compute its eigenspace in closed-form

- **Eigenvalue set:** $\{0, 2, \ldots, 2n\}$
  - j^{th} eigenvalue occurs with $\binom{n}{j}$ multiplicity

- **Eigenvector set:** Hadamard matrix (H) of order 2^n

$$H_{ij} = (-1)^{\langle r_i, r_j \rangle}$$

MerCBO: Mercer Features

$$K(x_1, x_2) = \sum_{i=0}^{2^n-1} e^{-\beta\lambda_i} u_i([x_1])\, \mathbf{u_j}([\mathbf{x_2}])$$

$$K(x_1, x_2) = \sum_{i=0}^{2^n-1} e^{-\beta\lambda_i} \; -1^{<r_{i,} x_1>} \; -1^{<r_{i,} x_2>}$$

$$K(x_1, x_2) = \phi(x_1)^T \phi(x_2)$$

$$\boxed{\phi(x)_i = \{\sqrt{e^{-\beta\lambda_i}} \; -1^{<r_{i,} x>}\}}$$

MerCBO: Mercer Features

$$K(x_1, x_2) = \sum_{i=0}^{2^n-1} e^{-\beta\lambda_i} \, -1^{<r_i, x_1>} \; -1^{<r_i, x_2>}$$

$$K(x_1, x_2) = \phi(x_1)^T \phi(x_2)$$

$$\phi(x)_i = \{\sqrt{e^{-\beta\lambda_i}} \, -1^{<r_i, x>}\}$$

j^{th} order Mercer features: first j distinct eigenvalues

MerCBO Algorithm [Deshwal et al., 2021]

- Same surrogate model as COMBO
  - GP with discrete diffusion kernel and graph representation

- Thompson sampling as acquisition function
  - Mercer features allow sampling functions from GP posterior

- Acquisition function optimization
  - Binary quadratic program
  - Parametrized submodular relaxation (PSR) solver

$$x_{next} = arg \max_{x \in \{0,1\}^d} b^T x + x^T A\, x$$

MerCBO: Acquisition Function Optimization

$$x_{next} = arg \max_{x \in \{0,1\}^n} \; b^T x + x^T A \, x$$

- **Parametrized Submodular Relaxation (PSR) solver**
  - Construct a Λ-parametrized submodular relaxation

$$\boxed{h_\Lambda(x) + x^T A^- x} \leq \boxed{x^T A x + b^T x}$$

Solve using min. graph cut algorithms

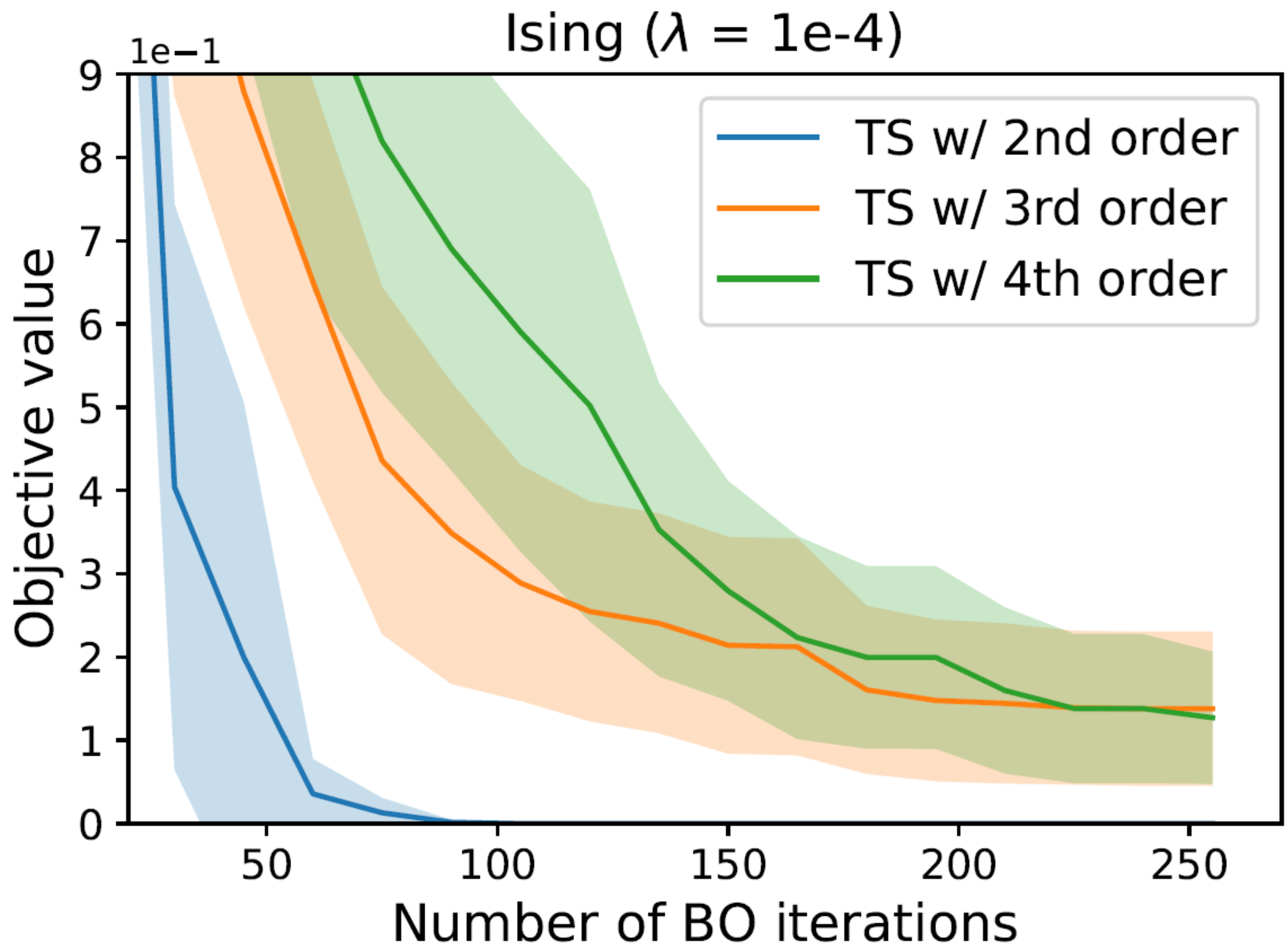
  - Optimize the relaxation over Λ

$$\boxed{h_{\Lambda_1}(x) + x^T A^- x} \leq \boxed{h_{\Lambda_2}(x) + x^T A^- x} \leq \; \dots . \quad \boxed{x^T A x + b^T x}$$

Inspired by work on prescriptive price optimization [Ito and Fujimaki, 2016]

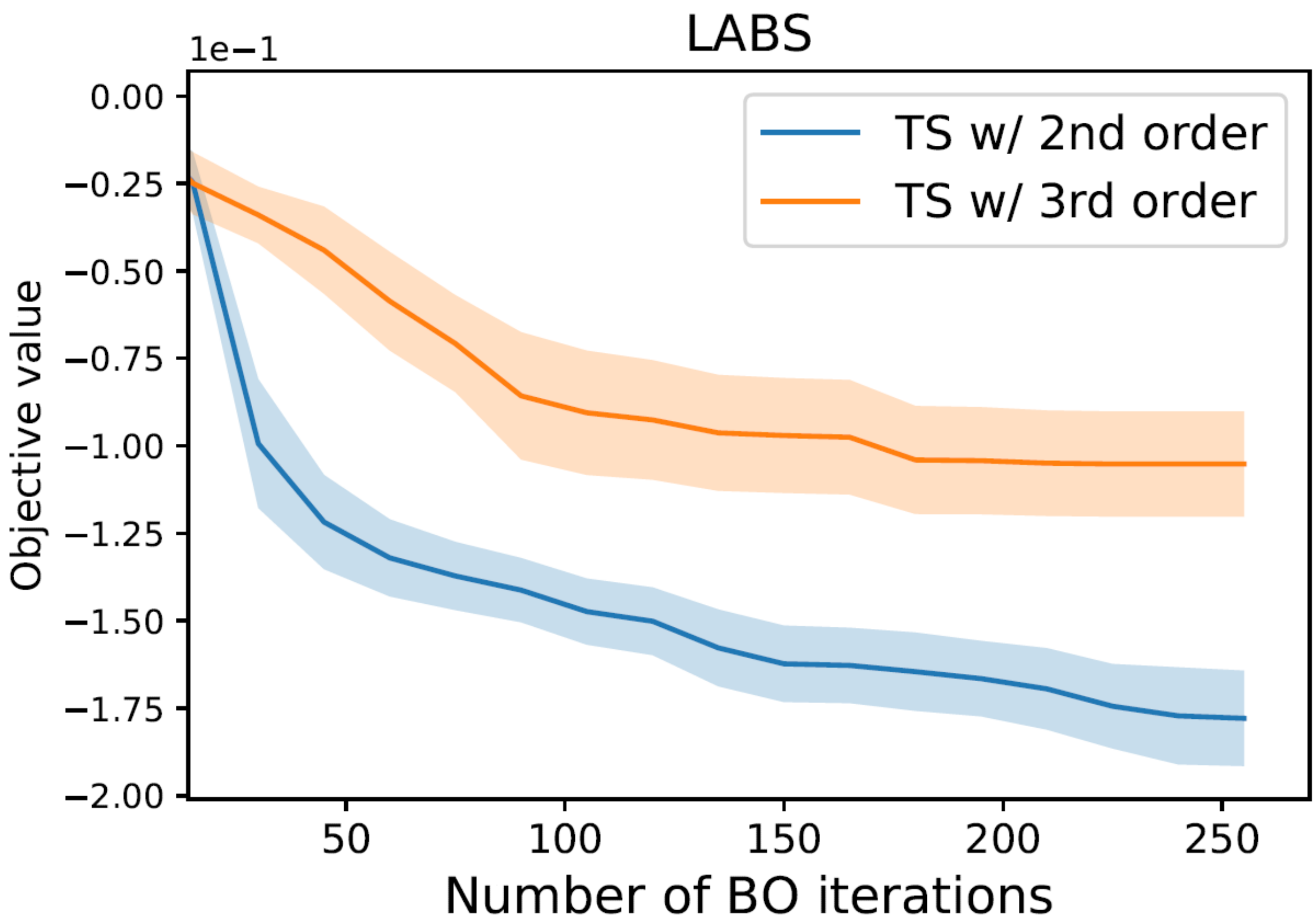
MerCBO Results #1: Order of Features

- Second-order features provide the best trade-off
  - Tractability and good overall BO performance



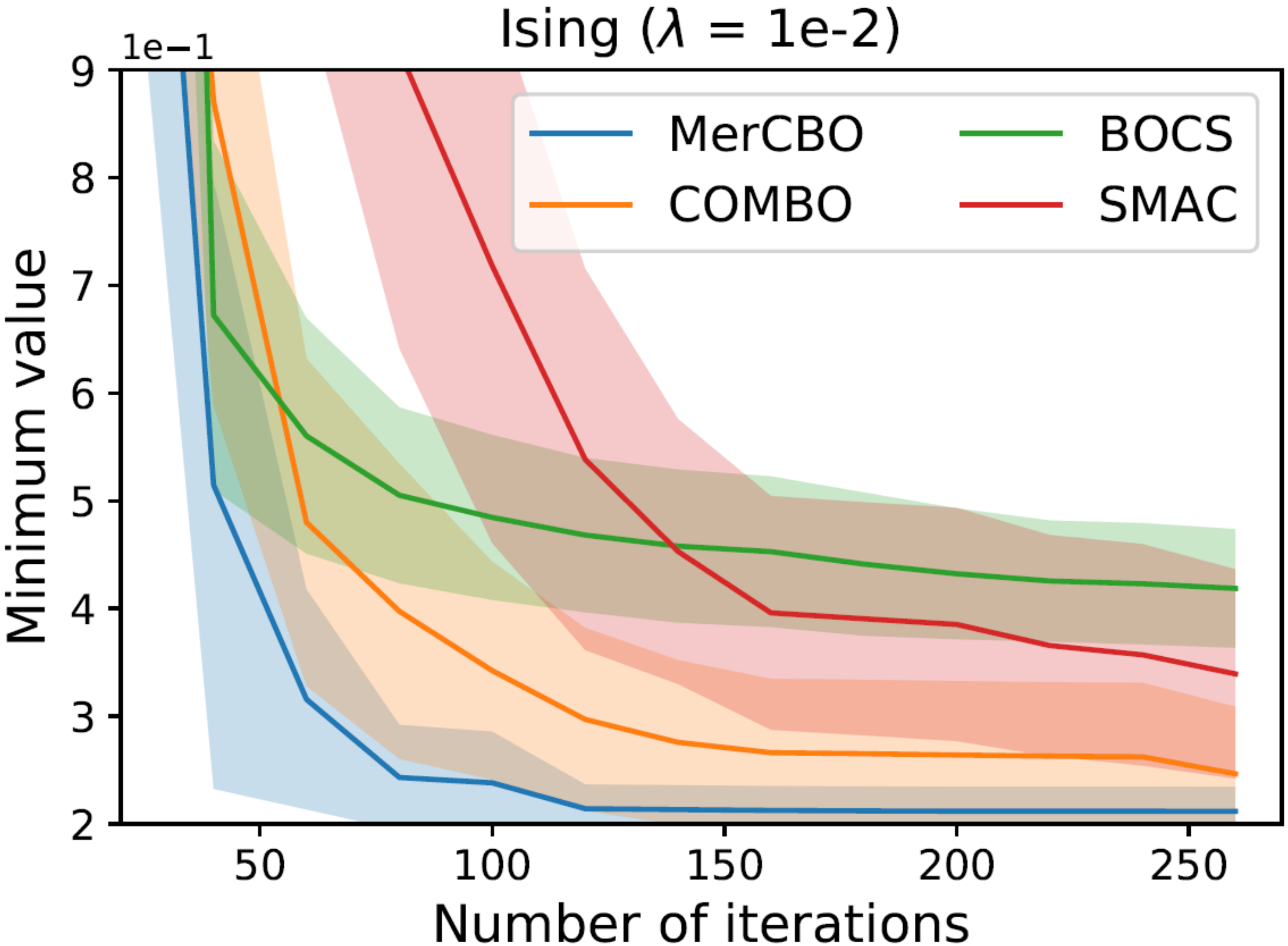
MerCBO Results #1: Order of Features

- Second-order features provide the best trade-off
  - Tractability and good overall BO performance



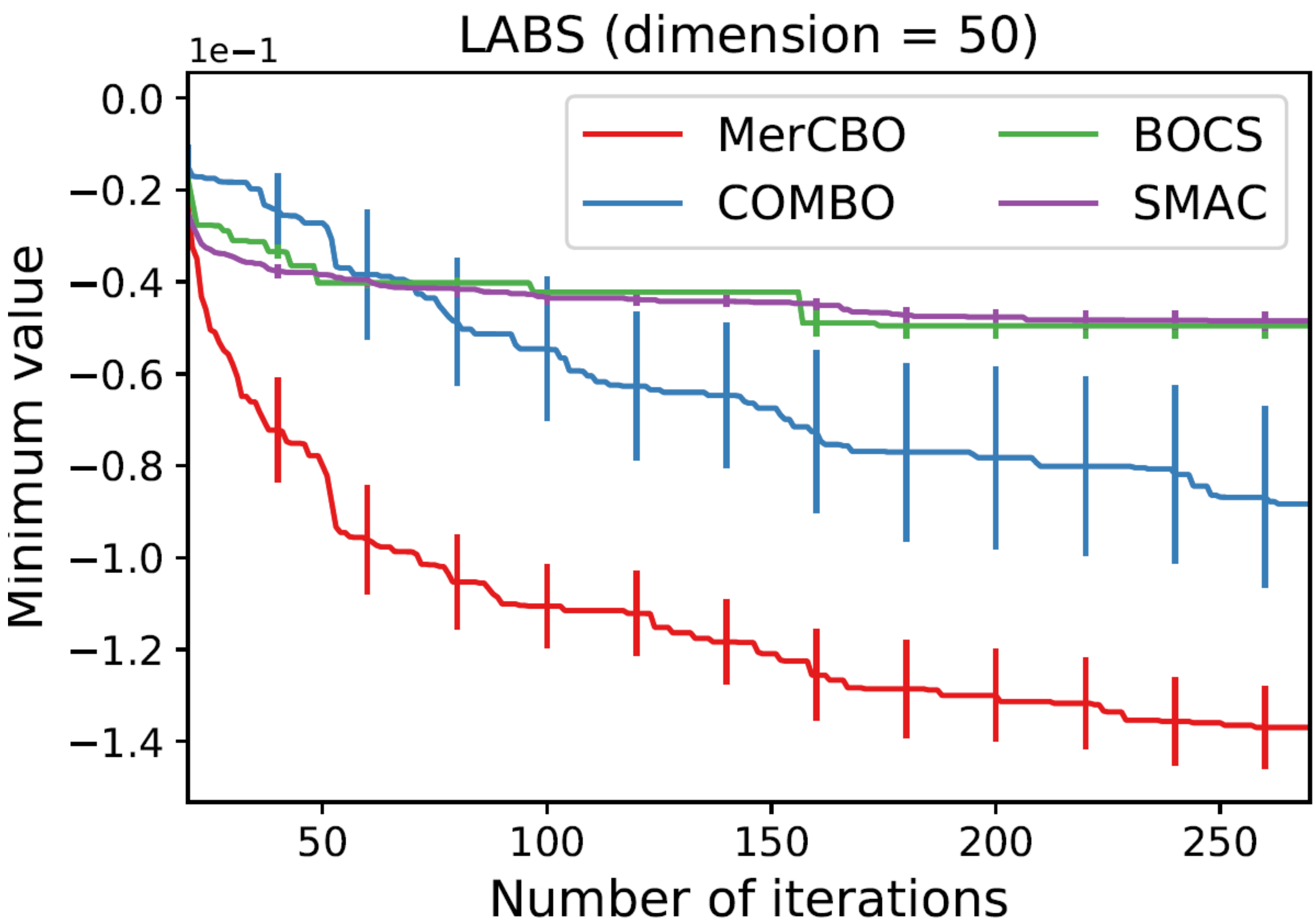
MerCBO Results #2: Comparison with State-of-the-art

- MerCBO outperforms prior methods



MerCBO Results #2: Comparison with State-of-the-art

- MerCBO outperforms prior methods



MerCBO for Biological Sequence Design

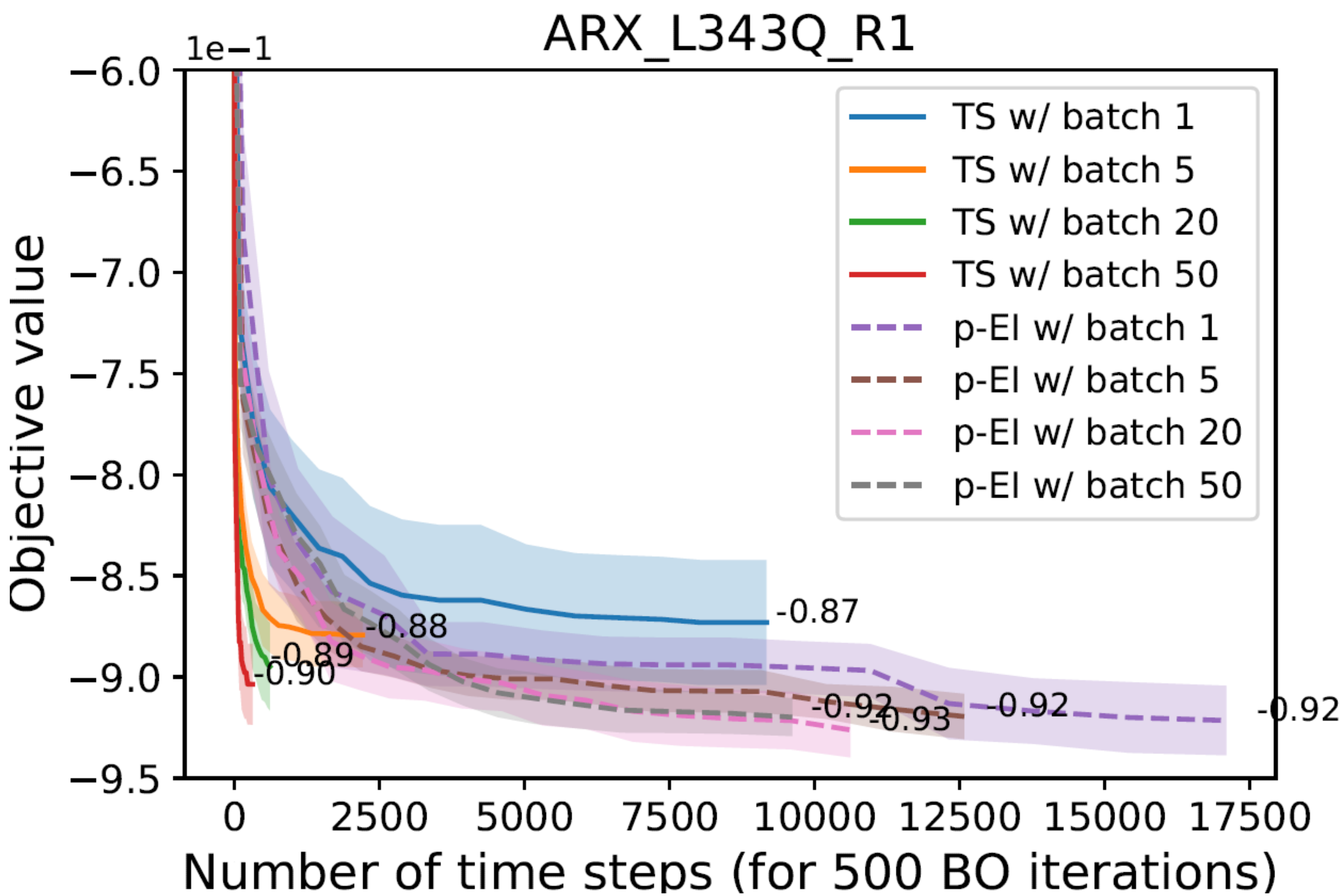
- Design of optimized biological structures such as DNA and proteins have many medical applications

Biological Sequence Design: Three Desiderata

- **Diversity**
  - uncover a diverse set of structures
- **Parallel experiments**
  - Select a batch of structures for evaluation in each round
- **Real-time accelerated design**
  - Use parallel experimental resources to accelerate optimization

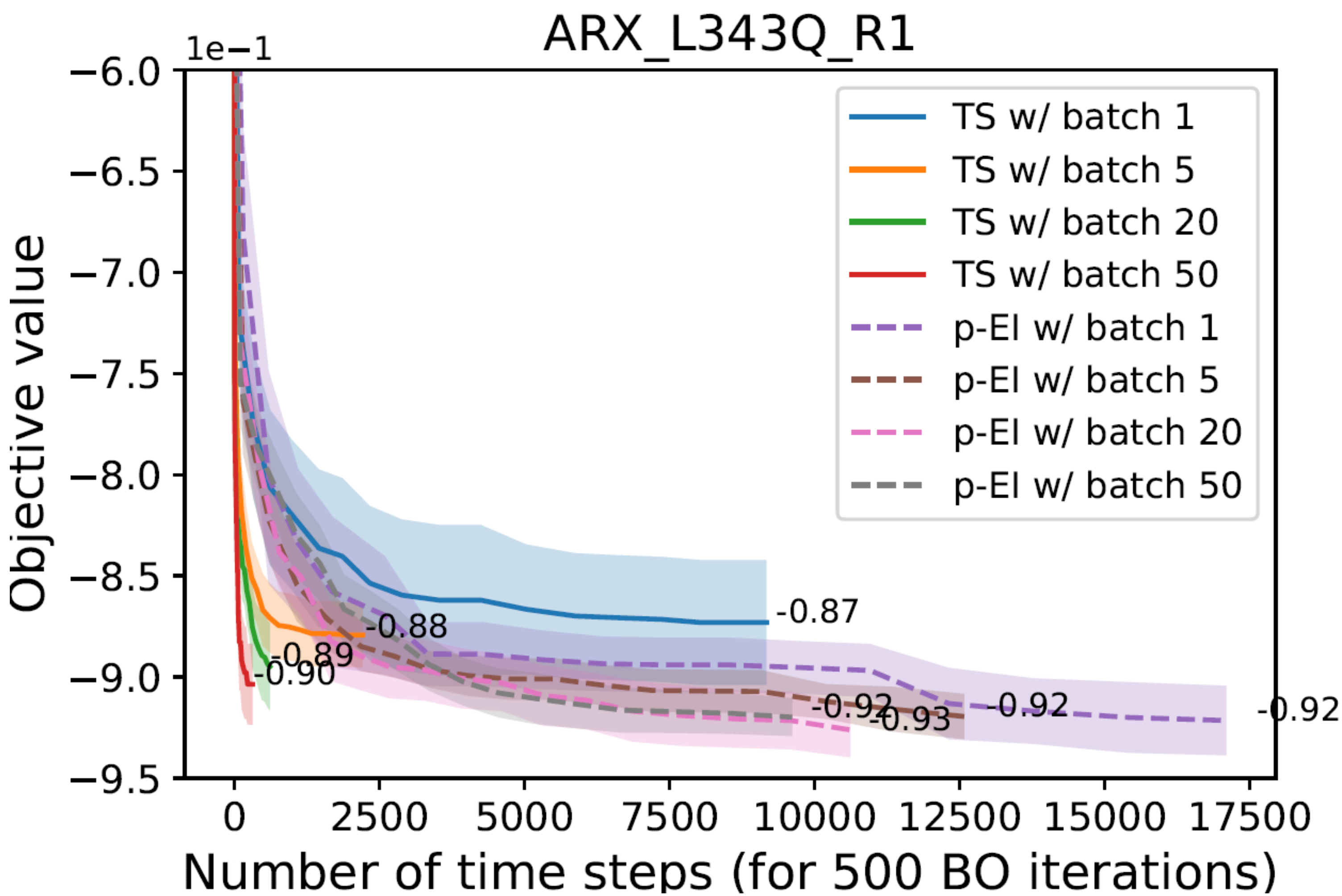
MerCBO Results #3: Real-time acceleration

- TS is better than EI for real-time accelerated design



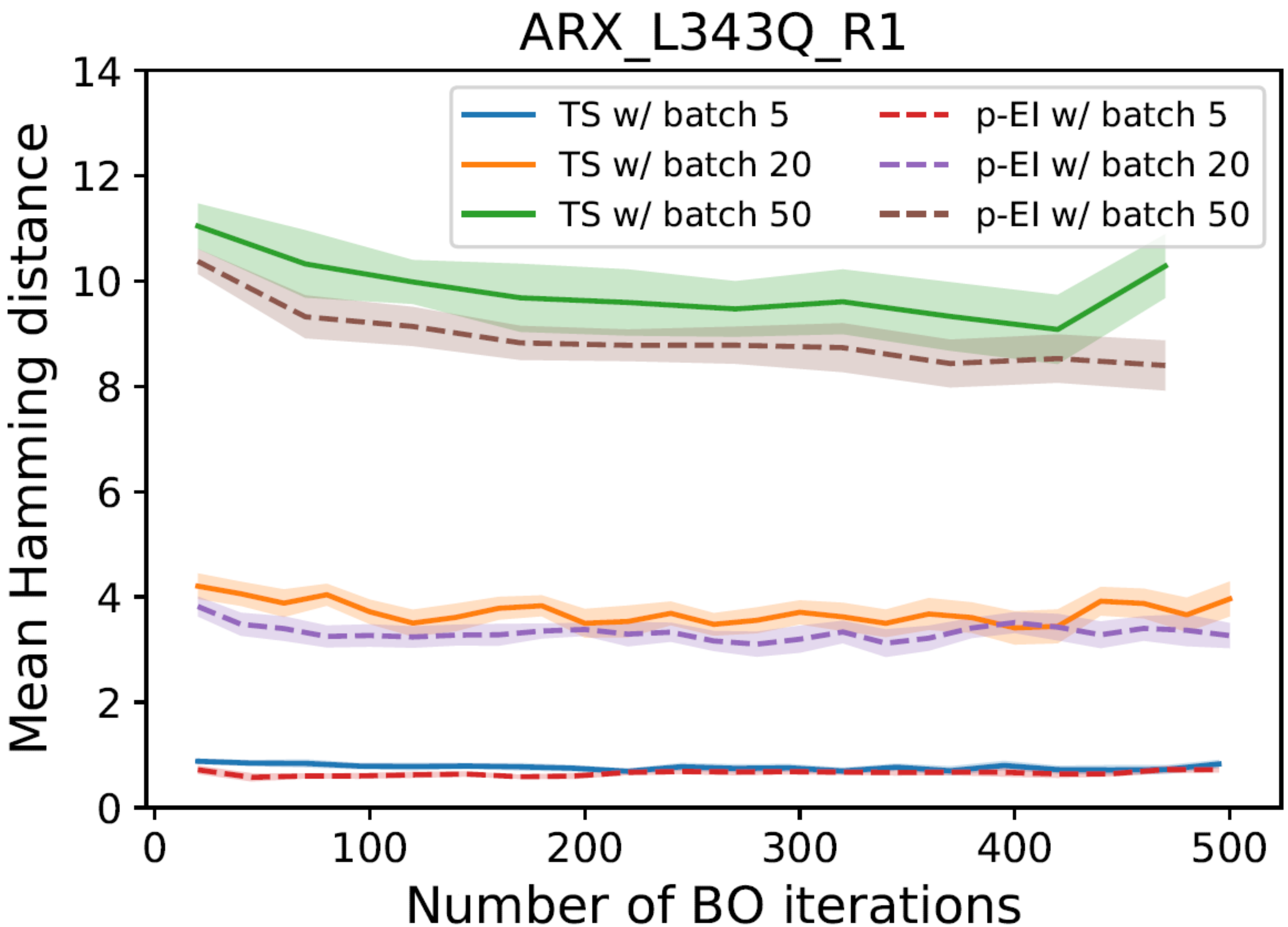
MerCBO Results #3: Real-time acceleration

- TS improvement over EI increases with batch size



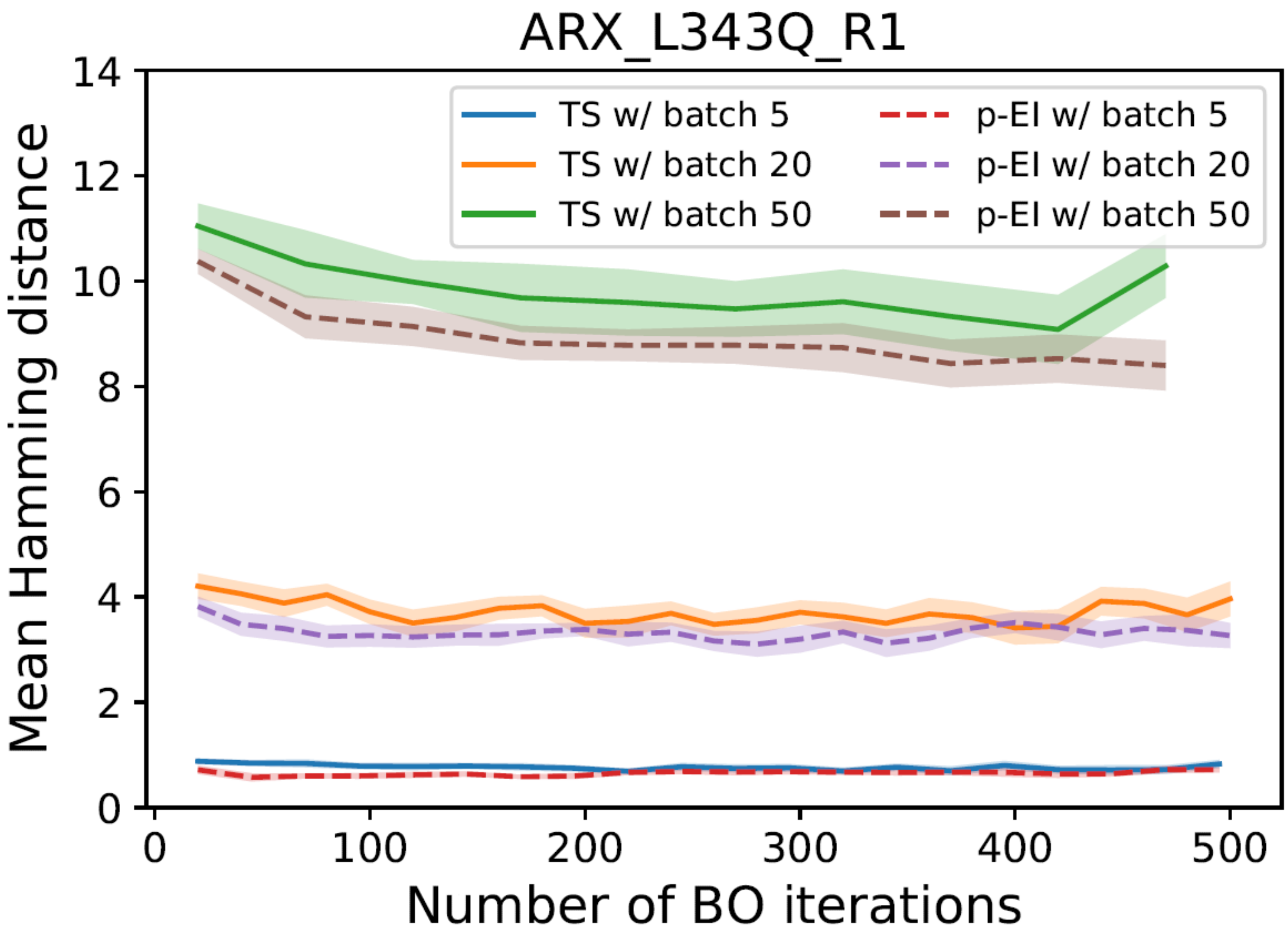
MerCBO Results #4: Diversity of sequences

- TS is better than EI for diversity of sequences



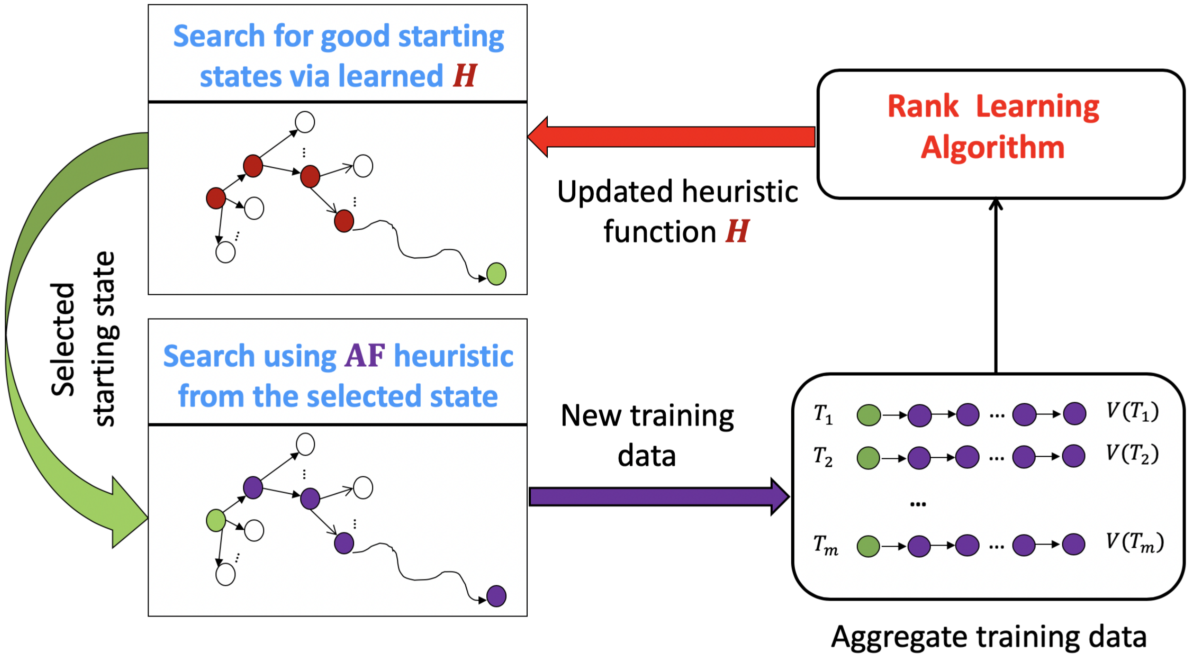
MerCBO Results #4: Diversity of sequences

- TS improvement over EI increases with batch size



Learning to Search Framework [Deshwal et al., 2021]

- Use machine learning to improve the accuracy of search
  - Continuously update the search control knowledge using the training data generated from the previous search experience



Learning to Search Framework [Deshwal et al., 2021]

- Defines a new family of search-style BO approaches

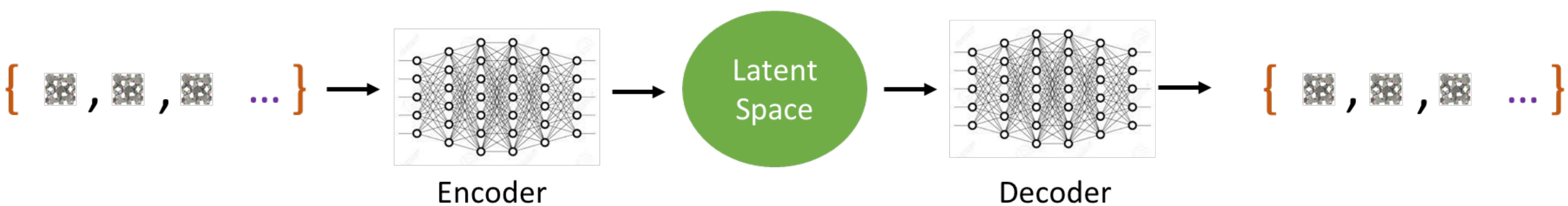
- Can work with any complex statistical model and acquisition function

- Can handle complex domain constraints to select ``valid'' structures for evaluation

Reduction to Continuous BO [Gómez-Bombarelli et al., 2018]...

- **Key Idea:** Convert discrete space into continuous space

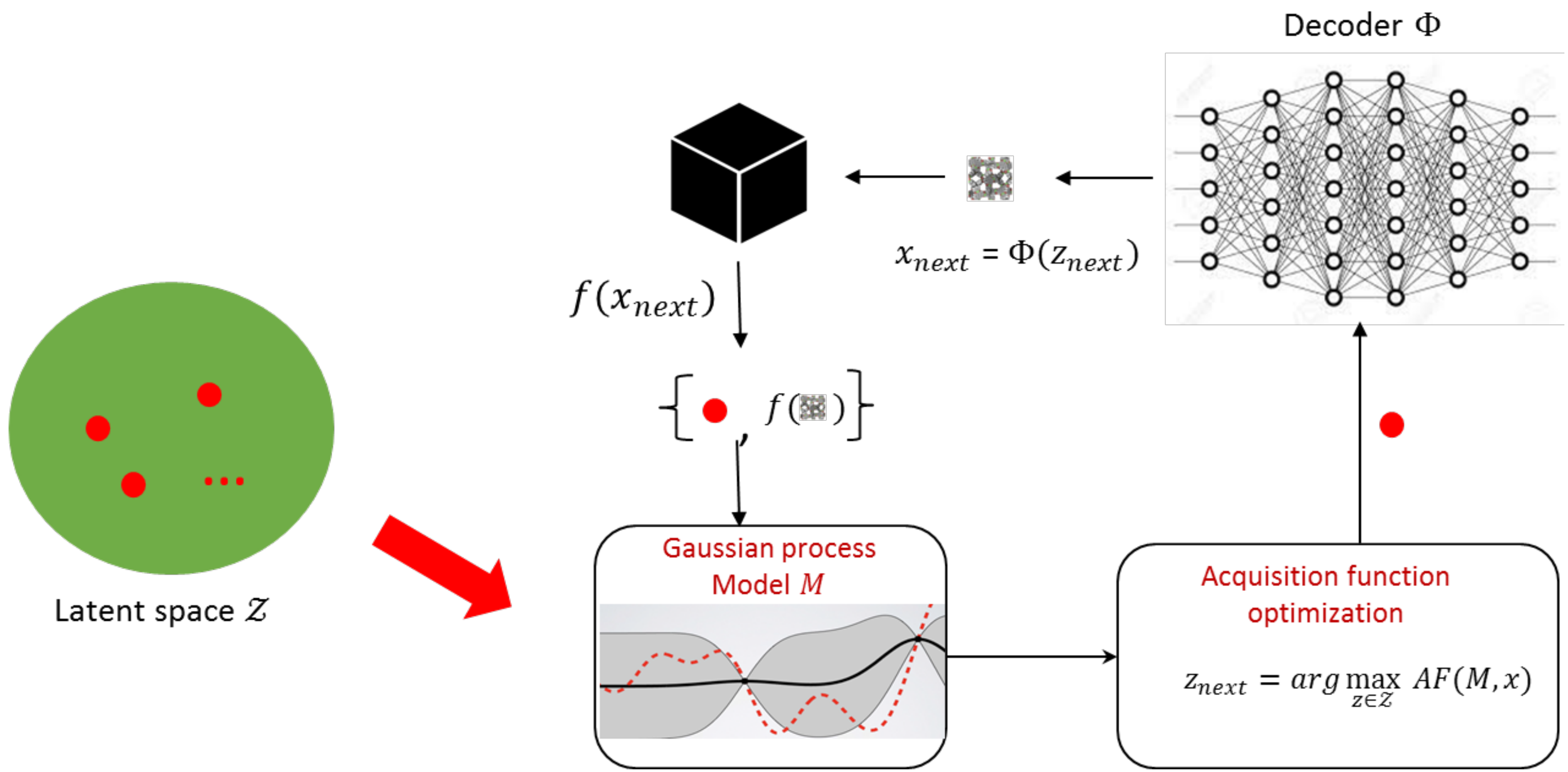
- Train a deep generative model (VAE) using unsupervised structures



- Perform BO in the learned continuous latent space
  - Surrogate modeling and acquisition function optimization in latent space (vs. combinatorial space)

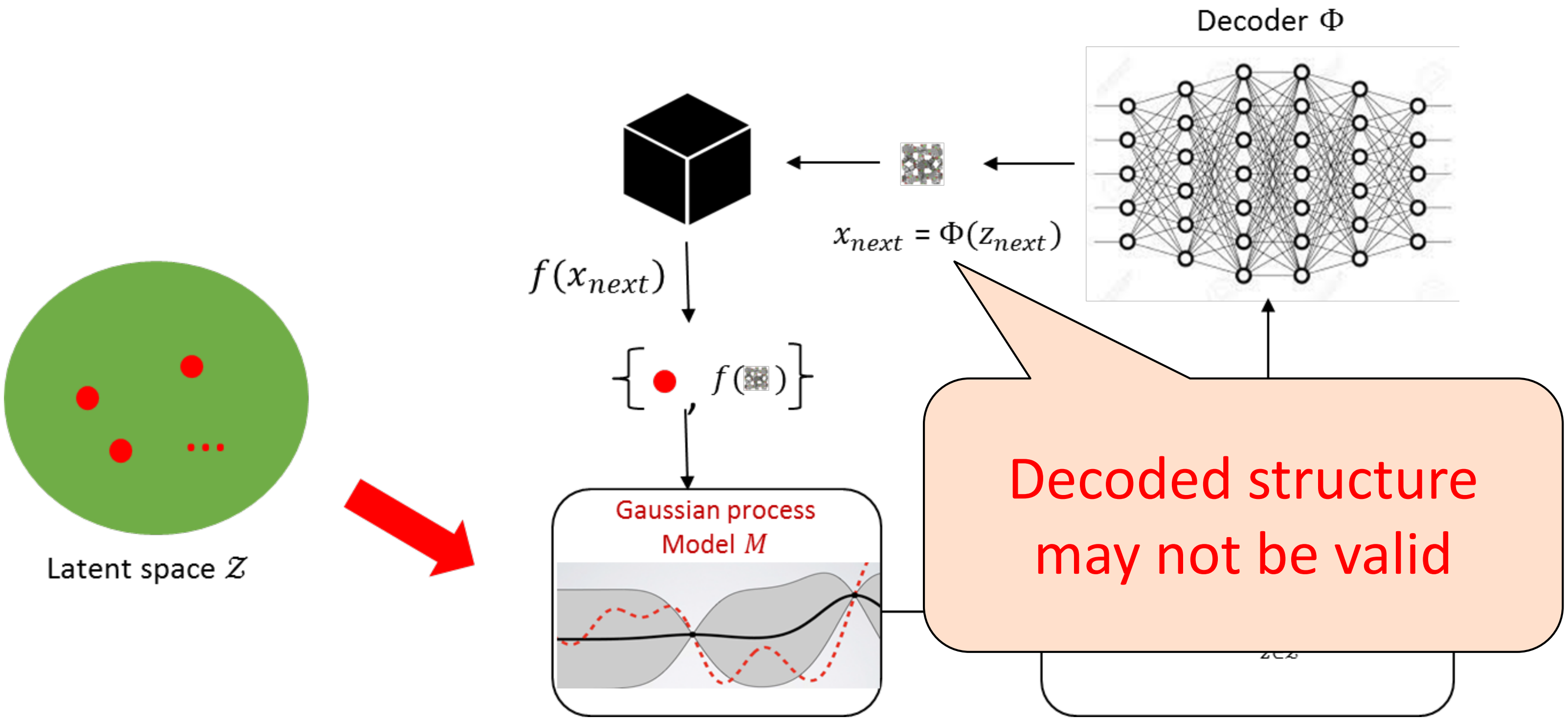
Reduction to Continuous BO [Gómez-Bombarelli et al., 2018]…

- BO in the learned latent space



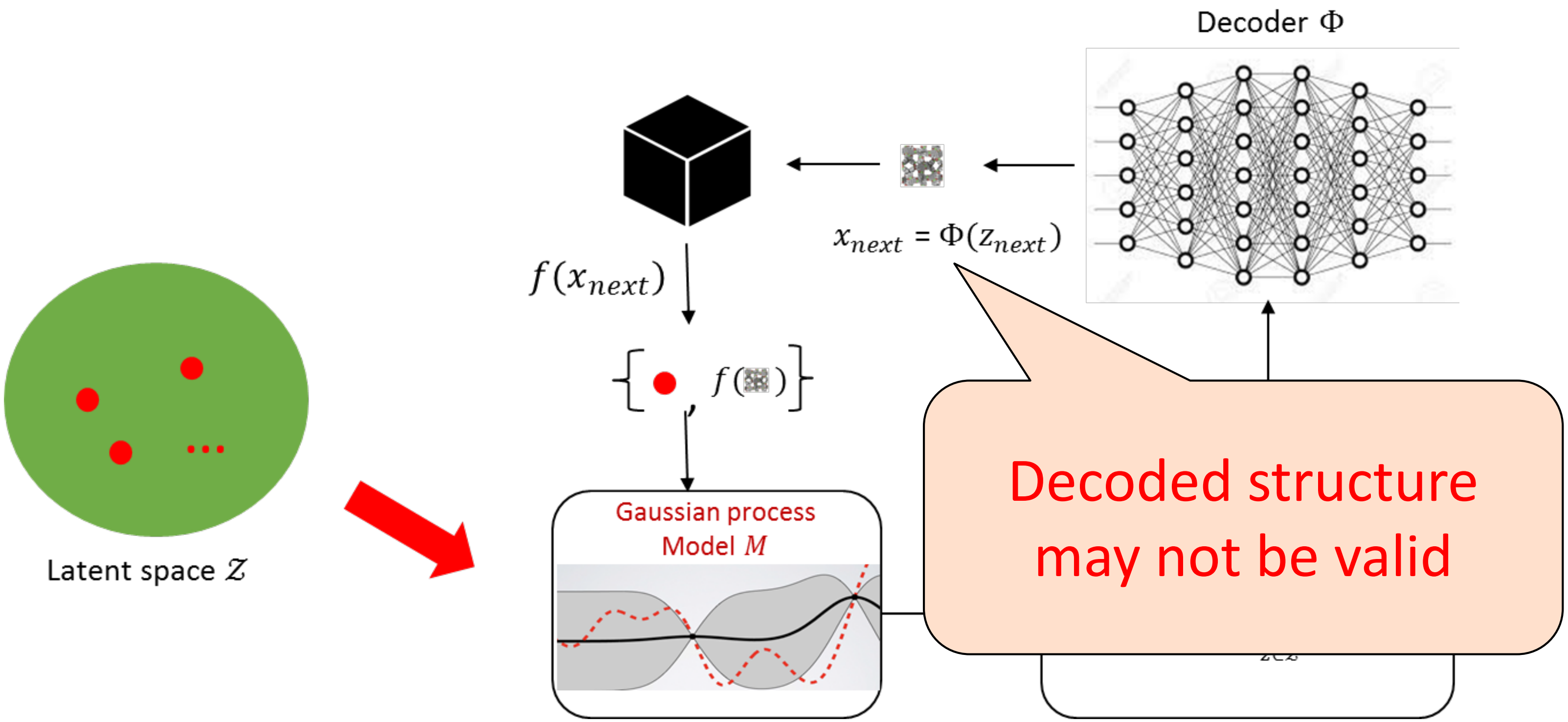
Reduction to Continuous BO [Gómez-Bombarelli et al., 2018]...

- BO in the learned latent space



Reduction to Continuous BO [Gómez-Bombarelli et al., 2018]...

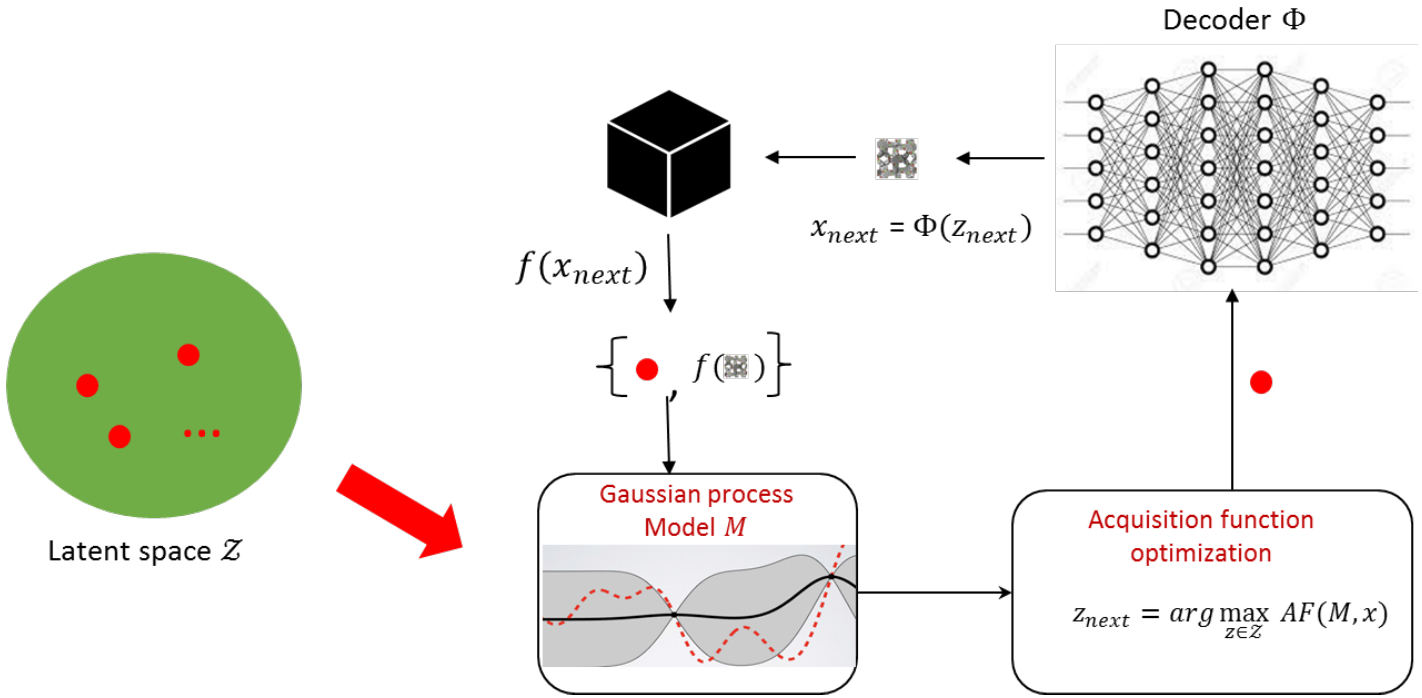
- BO in the learned latent space



- Some recent work to address this challenge
  - Griffiths R.-R. and Hernández-Lobato J. M.: Constrained Bayesian optimization for Automatic Chemical Design Using Variational Autoencoders, Chemical Science, 2019

Reduction to Continuous BO [Gómez-Bombarelli et al., 2018]...

- BO in the learned latent space



- Challenges
  - Doesn't (explicitly) incorporate information about decoded structures
  - Surrogate model may not generalize well for small data setting

Improve Latent Space via Weighted Retraining [Tripp et al., 2020]

- Periodically retrain the deep generative model
- Assign importance weights to training data proportional to their objective function value

Improve Latent Space via Weighted Retraining [Tripp et al., 2020]

- Periodically retrain the deep generative model

- Assign importa ng data proportional to their object

Computationally expensive

Improve Latent Space via Weighted Retraining [Tripp et al., 2020]

- Periodically retrain the deep generative model
- Assign importance weights to training data proportional to their objective function value

Overall approach is not effective for small-data setting

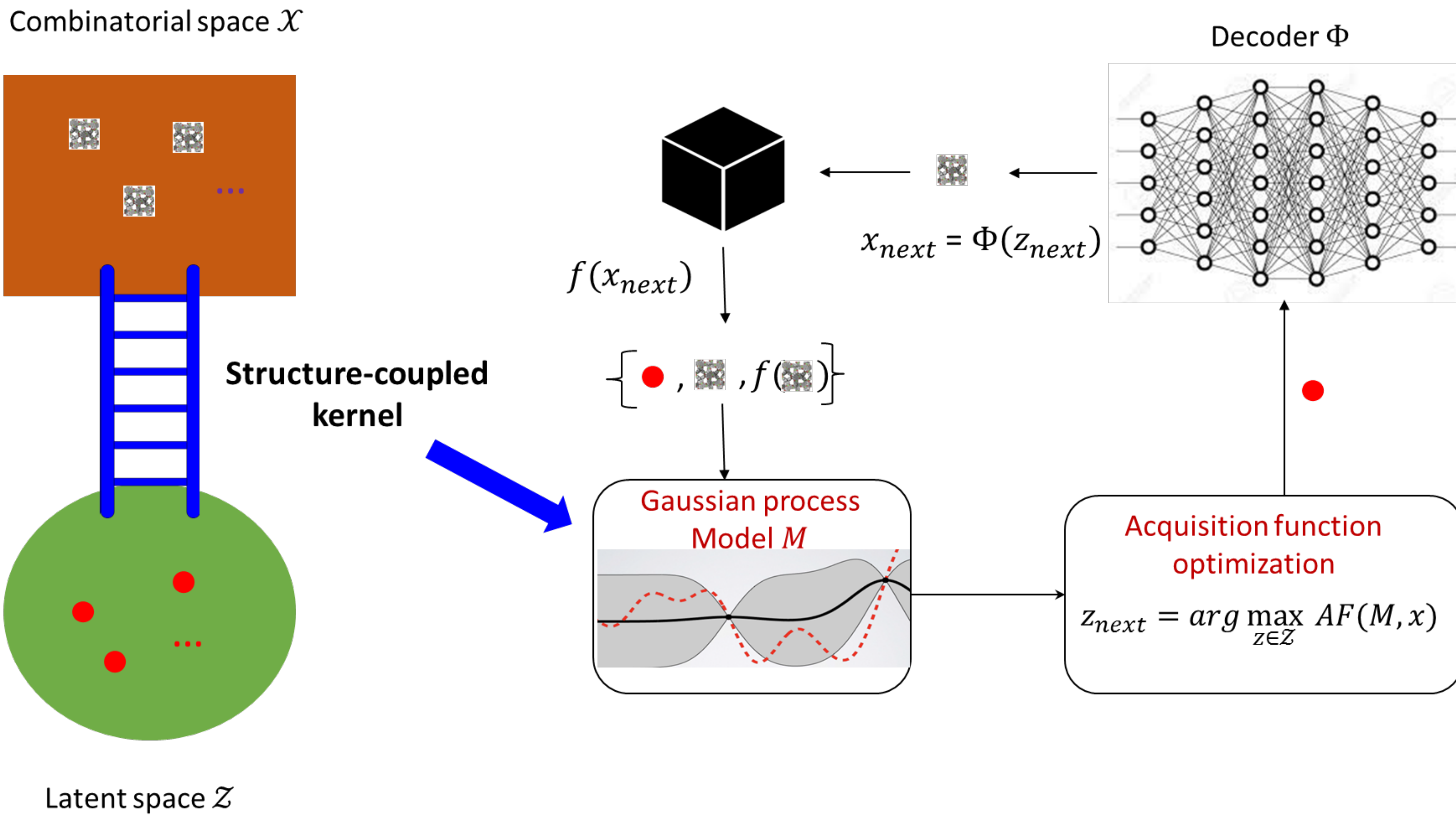
Uncertainty-guided Latent Space BO [Notin et al., 2021]

- Leverage the epistemic uncertainty of the decoder to guide the optimization process

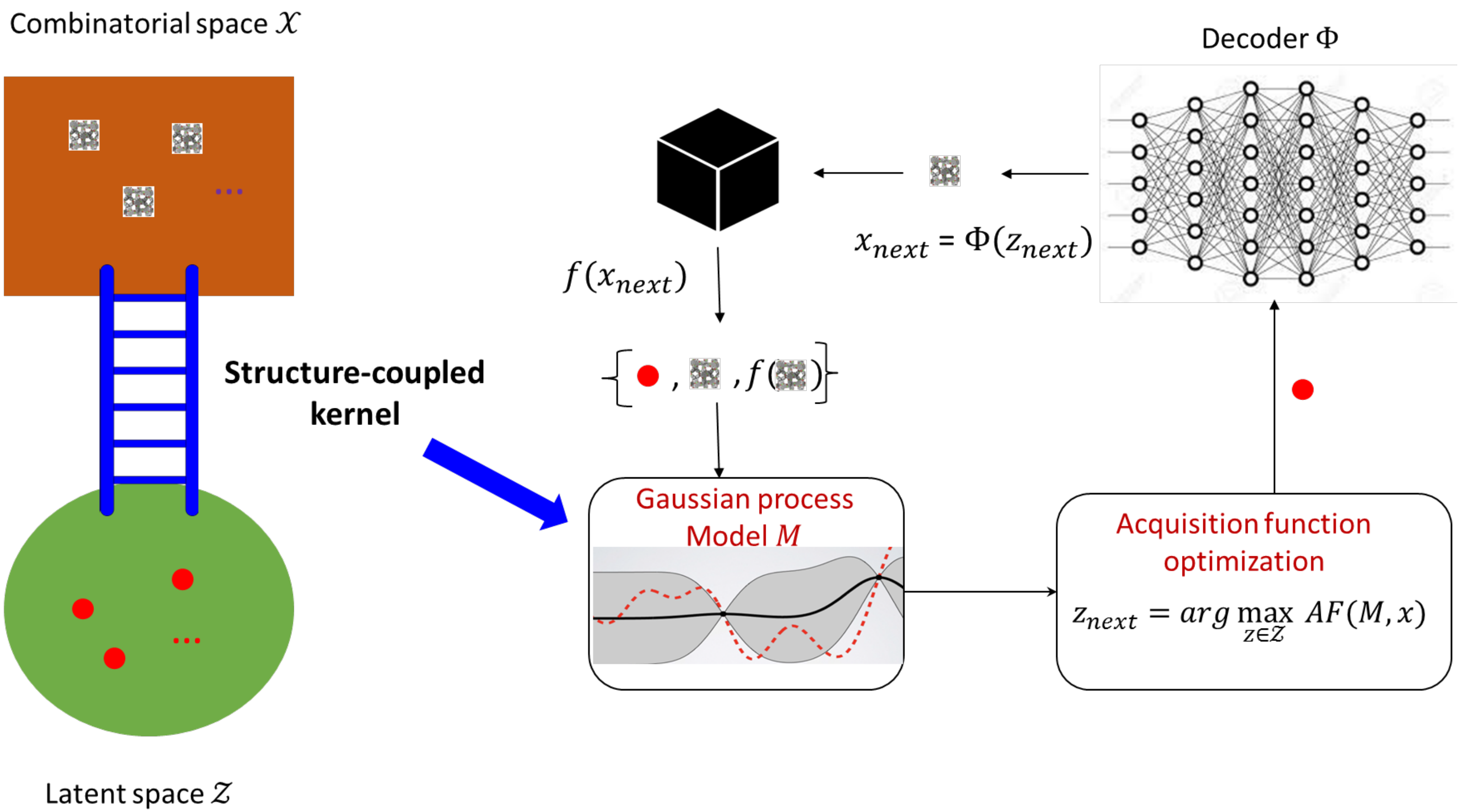
- Importance sampling-based estimator for uncertainty quantification over high-dimensional discrete structures

- No retraining of deep generative model is needed

LADDER Algorithm [Deshwal and Doppa, 2021]



LADDER Algorithm [Deshwal and Doppa, 2021]



- Key Idea: Combines the complementary strengths of deep generative models and structured kernels for better surrogate modeling

Structure-Coupled Kernel

- Structure-coupled kernel (c) combines
  - Continuous kernels over latent space $\mathcal{Z}$ (e.g., Matern)
  - Structured kernels (e.g., generic/hand-designed strings, graphs)

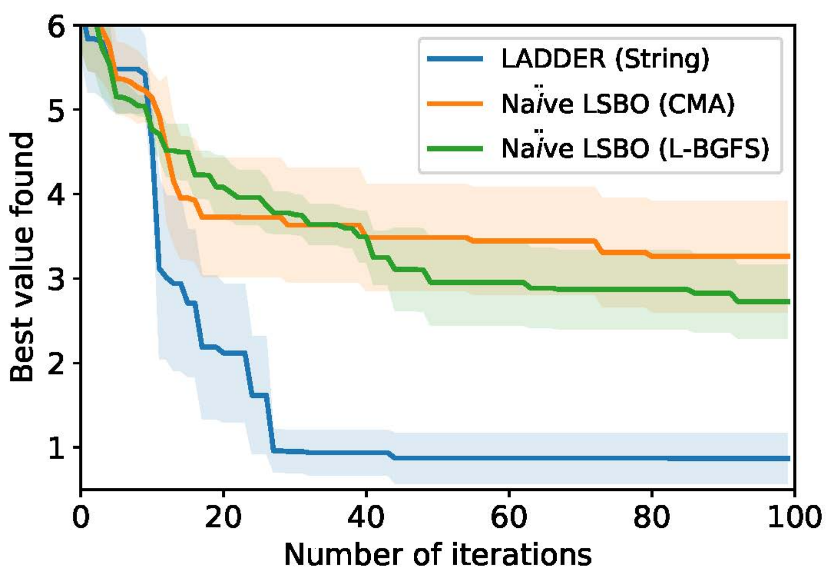
- Key Idea
  - Extrapolate eigenfunctions of the latent space kernel matrix **L** with basis functions from the structured kernel k

$$\boxed{c(z, z') = k_z^T K^{-1} L K^{-1} k_{z'}}$$

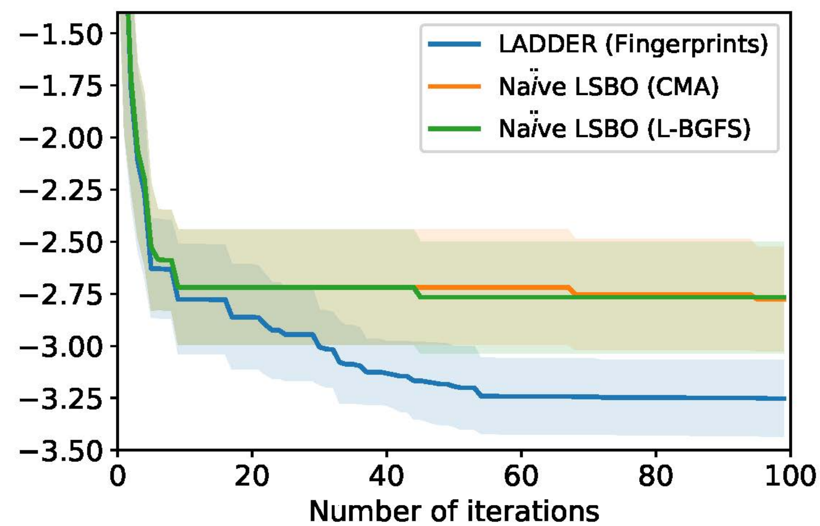
- Generalized Nystrom Extension [Ref]
  - k acts like a smooth extrapolating kernel

Latent Space BO Results #1

- LADDER outperforms latent space BO real benchmarks



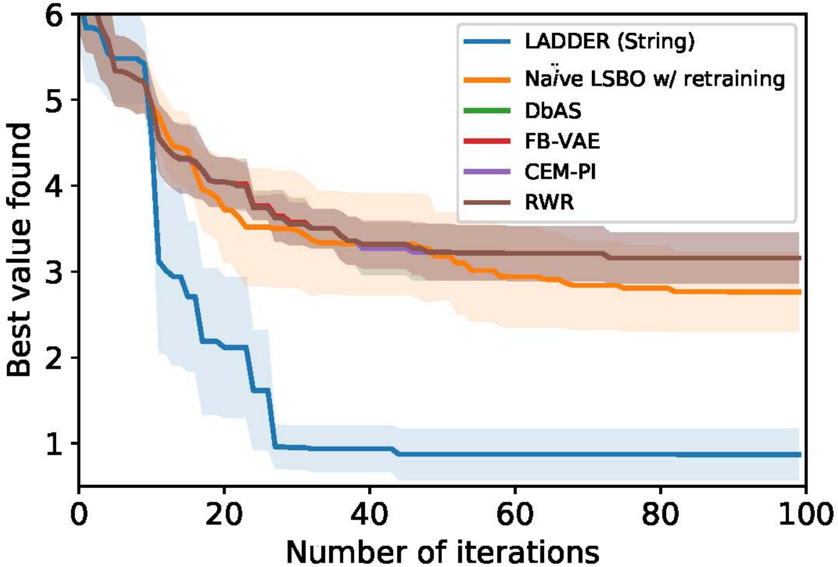
Arithmetic expression task



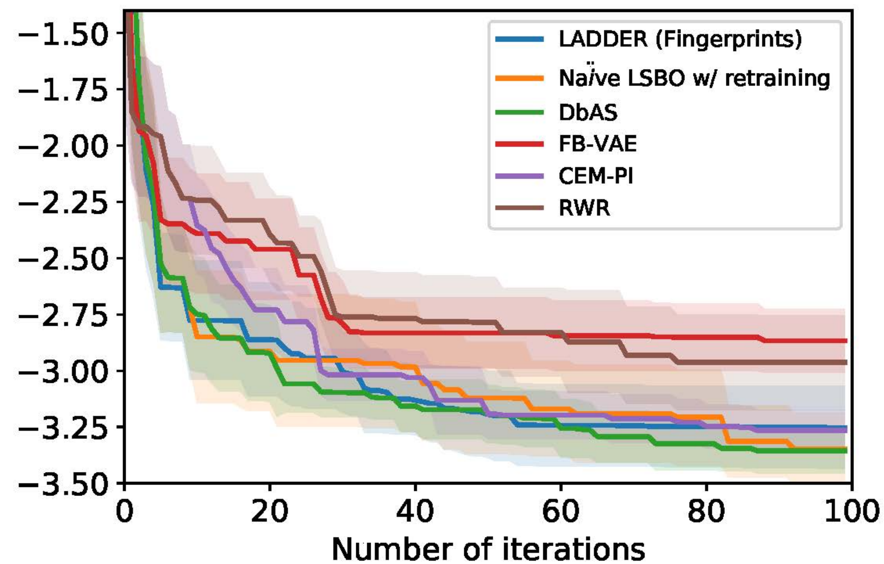
Chemical design task

Latent Space BO Results #2

- LADDER is competitive or better than state-of-the-art methods



Arithmetic expression task



Chemical design task

Code and Software

- MerCBO: https://github.com/aryandeshwal/MerCBO
- LADDER: https://github.com/aryandeshwal/LADDER
- BOPS: https://github.com/aryandeshwal/BOPS
- COMBO: https://github.com/QUVA-Lab/COMBO
- SMAC: https://github.com/automl/SMAC3

Questions ?