CSCI 699 - ProbGen

Probabilistic and Generative Models

Willie Neiswanger

Lecture 13 - Bayesian Optimization

Today

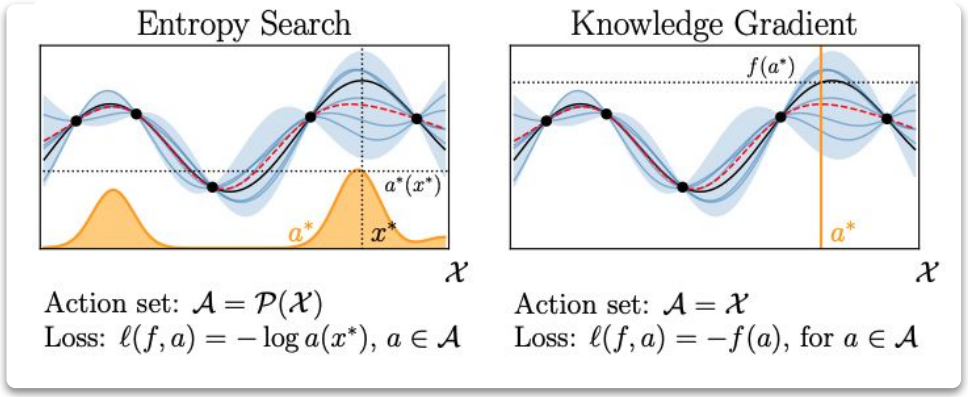
Today

Lecture: Review sequential decision making and active learning, then finish the lecture on Bayesian optimization.

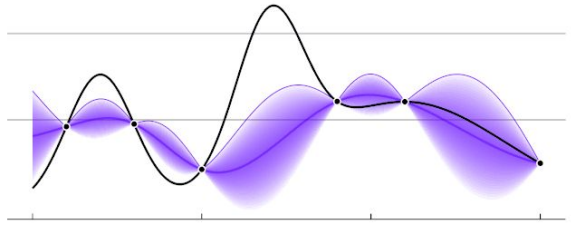
Today

Lecture: Review sequential decision making and active learning, then finish the lecture on Bayesian optimization.

- Decision making under uncertainty.
- Active learning.
- Bayesian Optimization
- *E.g.*, UCB, PI, EI, KG, ES, etc.



"Generalizing Bayesian Optimization with Decision-theoretic Entropies", 2022

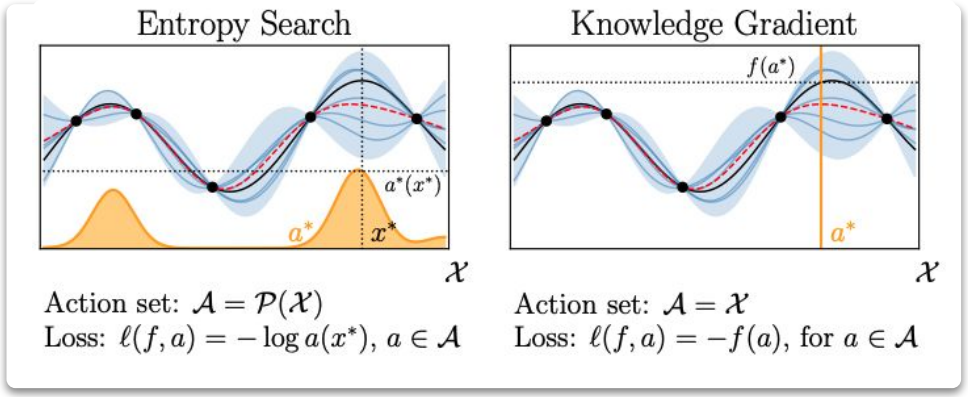


Bayesian Optimization, Wikipedia

Today

Lecture: Review sequential decision making and active learning, then finish the lecture on Bayesian optimization.

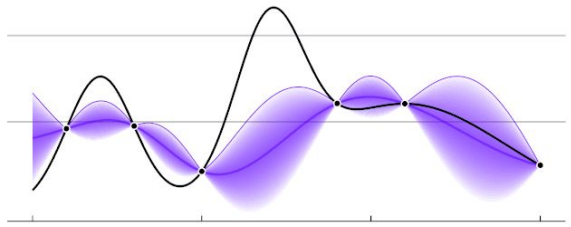
- Decision making under uncertainty.
- Active learning.
- Bayesian Optimization
- *E.g.*, UCB, PI, EI, KG, ES, etc.



"Generalizing Bayesian Optimization with Decision-theoretic Entropies", 2022

After:

- Paper presentations from students.



Bayesian Optimization, Wikipedia

Today

After: Next batch of **paper presentations.** — 5 today!

Diffusion Policy: Visuomotor Policy Learning via Action Diffusion

Cheng Chi[*1], Zhenjia Xu[*1], Siyuan Feng[2], Eric Cousineau[2], Yilun Du[3], Benjamin Burchfiel[2], Russ Tedrake[2,3], Shuran Song[1,4]

Abstract

This paper introduces Diffusion Policy, a new way of generating robot behavior by representing a robot's visuomotor policy as a conditional denoising diffusion process. We benchmark Diffusion Policy across 15 different tasks from 4 different robot manipulation benchmarks and find that it consistently outperforms existing state-of-the-art robot learning methods with an average improvement of 46.9%. Diffusion Policy learns the gradient of the action-distribution score function and iteratively optimizes with respect to this gradient field during inference via a series of stochastic Langevin dynamics steps. We find that the diffusion formulation yields powerful advantages when used for robot policies, including gracefully handling multimodal action distributions, being suitable for high-dimensional action spaces, and exhibiting impressive training stability. To fully unlock the potential of diffusion models for visuomotor policy learning on physical robots, this paper presents a set of key technical contributions including the incorporation of receding horizon control, visual conditioning, and the time-series diffusion transformer. We hope this work will help motivate a new generation of policy learning techniques that are able to leverage the powerful generative modeling capabilities of diffusion models. Code, data, and training details is available diffusion-policy.cs.columbia.edu

Keywords

Imitation learning, visuomotor policy, manipulation

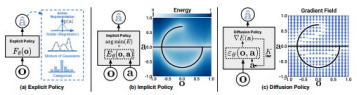


Figure 1. Policy Representations. a) Explicit policy with different types of action representations. b) Implicit policy learns an energy function conditioned on both action and observation and optimizes for actions that minimize the energy landscape c) Diffusion policy refines noise into actions via a learned gradient field. This formulation provides stable training, allows the learned policy to accurately model multimodal action distributions, and accommodates high-dimensional action sequences.

1 Introduction

Policy learning from demonstration, in its simplest form, can be formulated as the supervised regression task of learning to map observations to actions. In practice however, the unique nature of predicting robot actions — such as the existence of multimodal distributions, sequential correlation, and the requirement of high precision — makes this task distinct and challenging compared to other supervised learning problems.

Prior work attempts to address this challenge by exploring different *action representations* (Fig 1 a) – using mixtures of Gaussians Mandlekar et al. (2021), categorical representations of quantized actions Shafiullah et al. (2022), or by switching the *the policy representation* (Fig 1 b) – from explicit to implicit to better capture multi-modal distributions Florence et al. (2021); Wu et al. (2020).

In this work, we seek to address this challenge by introducing a new form of robot visuomotor policy that generates behavior via a "conditional denoising diffusion process Ho et al. (2020) on robot action space", **Diffusion Policy**. In this formulation, instead of directly outputting an action, the policy infers the action-score gradient, conditioned on visual observations, for K denoising iterations (Fig. 1 c). This formulation allows robot policies to inherit several key properties from diffusion models – significantly improving performance.

- **Expressing multimodal action distributions.** By learning the gradient of the action score function Song and Ermon (2019) and performing Stochastic Langevin Dynamics sampling on this gradient field, Diffusion policy can express arbitrary normalizable distributions Neal et al. (2011), which includes multimodal action distributions, a well-known challenge for policy learning.

LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia[1*] Sadhika Malladi[1*] Suchin Gururangan[2] Sanjeev Arora[1] Danqi Chen[1]

Abstract

Instruction tuning has unlocked powerful capabilities in large language models (LLMs), using combined datasets to develop general-purpose chatbots. However, real-world applications often require a specialized suite of skills (e.g., reasoning). The challenge lies in identifying the most relevant data from these extensive datasets to effectively develop specific capabilities, a setting we frame as *targeted instruction tuning*. We propose LESS, an optimizer-aware and practically efficient algorithm to estimate data influences and perform Low-rank gradIEnt Similarity Search for instruction data selection. Crucially, LESS adapts existing influence formulations to work with the Adam optimizer and variable-length instruction data. LESS first constructs a highly reusable and transferable *gradient datastore* with low-dimensional gradient features and then selects examples based on their similarity to few-shot examples embodying a specific capability. Experiments show that training on a LESS-selected 5% of the data can often outperform training on the full dataset across diverse downstream tasks. Furthermore, the selected data is highly transferable: smaller models can be leveraged to select useful data for larger models and models from different families. Our qualitative analysis shows that our method goes beyond surface form cues to identify data that exemplifies the necessary reasoning skills for the intended downstream application. To facilitate future work, we release code and data at princeton-nlp/LESS.

1. Introduction

Instruction tuning has made large language models (LLMs) adept at following human instructions (Ouyang et al., 2022) as versatile chatbots (OpenAI, 2022; 2023; Anthropic, 2023; Google, 2023). Recent efforts curating highly diverse and wide-ranging instruction tuning datasets (Taori et al., 2023; Wang et al.; Mukherjee et al., 2023; Xu et al., 2023, *inter alia*) induce remarkably strong generalization even from a small number of examples (Zhou et al., 2023). Regardless, it remains an open problem to understand how to best utilize these various datasets.

Many real-world applications call for cultivating a specific suite of capabilities in LLMs (e.g., reasoning skills). However, training LLMs with mixed instruction tuning datasets can hinder the development of these specific capabilities. For example, Wang et al. (2023b) demonstrates that LLMs trained on a mix of instruction tuning datasets exhibit worse performance than those trained on a subset of the data. Additionally, considering the broad spectrum of user queries and the multitude of skills required to respond to them, there may not always be enough in-domain data available. Therefore, we hope to be able to effectively use the general instruction tuning data to improve specific capabilities. We frame this setting as *targeted instruction tuning*:

Given just a handful of examples embodying a specific capability, how can we effectively select relevant fine-tuning data from a large collection of instruction datasets?

We approach this problem by prioritizing training on data that directly minimizes loss on a target task instead of relying on surface form features (Gururangan et al., 2020; Xie et al., 2023b). Inspired by past works estimating the influence of individual training datapoints with gradient information (Pruthi et al., 2020; Han et al., 2023), we design an optimizer-aware approach to select such data. However, straightforward application of this influence formulation faces several challenges unique to the instruction tuning setting: (1) LLMs are traditionally fine-tuned with the Adam optimizer (Kingma & Ba, 2015) instead of the canonical SGD optimizer; (2) using sequence-level gradients of variable-length instruction data can derail the influence estimation; and (3) the large number of trainable parameters in LLMs makes the computation and storage of gradient information extremely resource-intensive.

[*]Equal contribution [1]Princeton Language and Intelligence (PLI), Princeton University, USA. [2]Department of Computer Science, University of Washington, USA. Correspondence to: Mengzhou Xia, Sadhika Malladi <{mengzhou, smalladi}@princeton.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Published as a conference paper at ICLR 2021

Denoising Diffusion Implicit Models

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

Abstract

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation without adversarial training, yet they require simulating a Markov chain for many steps in order to produce a sample. To accelerate sampling, we present denoising diffusion implicit models (DDIMs), a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPMs. In DDPMs, the generative process is defined as the reverse of a particular Markovian diffusion process. We generalize DDPMs via a class of non-Markovian diffusion processes that lead to the same training objective. These non-Markovian processes can correspond to generative processes that are deterministic, giving rise to implicit models that produce high quality samples much faster. We empirically demonstrate that DDIMs can produce high quality samples $10\times$ to $50\times$ faster in terms of wall-clock time compared to DDPMs, allow us to trade off computation for sample quality, perform semantically meaningful image interpolation directly in the latent space, and reconstruct observations with very low error.

1 Introduction

Deep generative models have demonstrated the ability to produce high quality samples in many domains (Karras et al., 2020; van den Oord et al., 2016a). In terms of image generation, generative adversarial networks (GANs, Goodfellow et al. (2014)) currently exhibits higher sample quality than likelihood-based methods such as variational autoencoders (Kingma & Welling, 2013), autoregressive models (van den Oord et al., 2016b) and normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2016). However, GANs require very specific choices in optimization and architectures in order to stabilize training (Arjovsky et al., 2017; Gulrajani et al., 2017; Karras et al., 2018; Brock et al., 2018), and could fail to cover modes of the data distribution (Zhao et al., 2018).

Recent works on iterative generative models (Bengio et al., 2014), such as denoising diffusion probabilistic models (DDPM, Ho et al. (2020)) and noise conditional score networks (NCSN, Song & Ermon (2019)) have demonstrated the ability to produce samples comparable to that of GANs, without having to perform adversarial training. To achieve this, many denoising autoencoding models are trained to denoise samples corrupted by various levels of Gaussian noise. Samples are then produced by a Markov chain which, starting from white noise, progressively denoises it into an image. This generative Markov Chain process is either based on Langevin dynamics (Song & Ermon, 2019) or obtained by reversing a forward *diffusion process* that progressively turns an image into noise (Sohl-Dickstein et al., 2015).

A critical drawback of these models is that they require many iterations to produce a high quality sample. For DDPMs, this is because that the generative process (from noise to data) approximates the reverse of the forward *diffusion process* (from data to noise), which could have thousands of steps; iterating over all the steps is required to produce a single sample, which is much slower compared to GANs, which only needs one pass through a network. For example, it takes around 20 hours to sample 50k images of size 32×32 from a DDPM, but less than a minute to do so from a GAN on a Nvidia 2080 Ti GPU. This becomes more problematic for larger images as sampling 50k images of size 256×256 could take nearly 1000 hours on the same GPU.

To close this efficiency gap between DDPMs and GANs, we present denoising diffusion implicit models (DDIMs). DDIMs are implicit probabilistic models (Mohamed & Lakshminarayanan, 2016) and are closely related to DDPMs, in the sense that they are trained with the same objective function.

Published as a conference paper at ICLR 2023

DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking

Gabriele Corso*, Hannes Stärk*, Bowen Jing*, Regina Barzilay & Tommi Jaakkola
CSAIL, Massachusetts Institute of Technology

Abstract

Predicting the binding structure of a small molecule ligand to a protein—a task known as *molecular docking*—is critical to drug design. Recent deep learning methods that treat docking as a regression problem have decreased runtime compared to traditional search-based methods but have yet to offer substantial improvements in accuracy. We instead frame molecular docking as a *generative* modeling problem and develop DiffDock, a diffusion generative model over the non-Euclidean manifold of ligand poses. To do so, we map this manifold to the product space of the degrees of freedom (translational, rotational, and torsional) involved in docking and develop an efficient diffusion process on this space. Empirically, DiffDock obtains a 38% top-1 success rate (RMSD<2Å) on PDBBind, significantly outperforming the previous state-of-the-art of traditional docking (23%) and deep learning (20%) methods. Moreover, while previous methods are not able to dock on computationally folded structures (maximum accuracy 10.4%), DiffDock maintains significantly higher precision (21.7%). Finally, DiffDock has fast inference times and provides confidence estimates with high selective accuracy.

1 Introduction

The biological functions of proteins can be modulated by small molecule ligands (such as drugs) binding to them. Thus, a crucial task in computational drug design is *molecular docking*—predicting the position, orientation, and conformation of a ligand when bound to a target protein—from which the effect of the ligand (if any) might be inferred. Traditional approaches for docking [Trott & Olson, 2010; Halgren et al., 2004] rely on scoring-functions that estimate the correctness of a proposed structure or pose, and an optimization algorithm that searches for the global maximum of the scoring function. However, since the search space is vast and the landscape of the scoring functions rugged, these methods tend to be too slow and inaccurate, especially for high-throughput workflows.

Recent works [Stärk et al., 2022; Lu et al., 2022] have developed deep learning models to predict the binding pose in one shot, treating docking as a regression problem. While these methods are much faster than traditional search-based methods, they have yet to demonstrate significant improvements in accuracy. We argue that this may be because the regression-based paradigm corresponds imperfectly with the objectives of molecular docking, which is reflected in the fact that standard accuracy metrics resemble the *likelihood* of the data under the predictive model rather than a regression loss. We thus frame molecular docking as a *generative modeling problem*—given a ligand and target protein structure, we learn a distribution over ligand poses.

To this end, we develop DiffDock, a diffusion generative model (DGM) over the space of ligand poses for molecular docking. We define a diffusion process over the degrees of freedom involved in docking: the position of the ligand relative to the protein (locating the binding pocket), its orientation in the pocket, and the torsion angles describing its conformation. DiffDock samples poses by running the learned (reverse) diffusion process, which iteratively transforms an uninformed, noisy prior distribution over ligand poses into the learned model distribution (Figure 1). Intuitively, this process can be viewed as the progressive refinement of random poses via updates of their translations, rotations, and torsion angles.

*Equal contribution. Correspondence to {gcorso, hstark, bjing}@mit.edu.

Compositional Visual Generation with Composable Diffusion Models

Nan Liu[1*], Shuang Li[2*], Yilun Du[2*]
Antonio Torralba[2], and Joshua B. Tenenbaum[2]

[1] University of Illinois Urbana-Champaign
[2] Massachusetts Institute of Technology
nanliu6@illinois.edu, {lishuang,yilundu,torralba,jbt}@mit.edu

Abstract. Large text-guided diffusion models, such as DALL-E 2, are able to generate stunning photorealistic images given natural language descriptions. While such models are highly flexible, they struggle to understand the composition of certain concepts, such as confusing the attributes of different objects or relations between objects. In this paper, we propose an alternative structured approach for compositional generation using diffusion models. An image is generated by composing a set of diffusion models, with each of them modeling a certain component of the image. To do this, we interpret diffusion models as energy-based models in which the data distributions defined by the energy functions may be explicitly combined. The proposed method can generate scenes at test time that are substantially more complex than those seen in training, composing sentence descriptions, object relations, human facial attributes, and even generalizing to new combinations that are rarely seen in the real world. We further illustrate how our approach may be used to compose pre-trained text-guided diffusion models and generate photorealistic images containing all the details described in the input descriptions, including the binding of certain object attributes that have been shown difficult for DALL-E 2. These results point to the effectiveness of the proposed method in promoting structured generalization for visual generation.

Keywords: Compositionality, Diffusion Models, Energy-based Models, Visual Generation

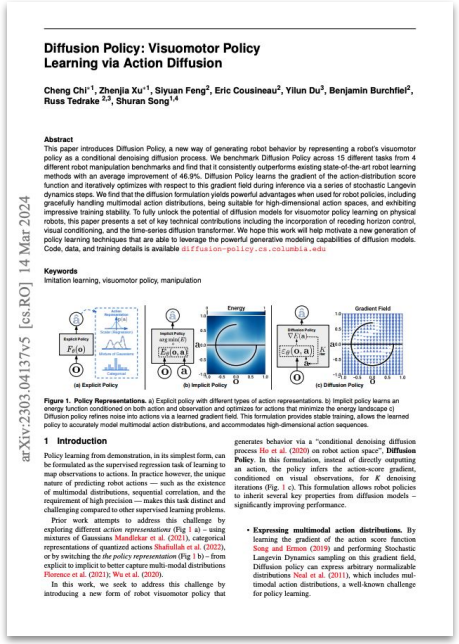
1 Introduction

Our understanding of the world is highly compositional in nature. We are able to rapidly understand new objects from their components, or compose words into complex sentences to describe the world states we encounter [26]. We are able

[*] indicates equal contribution.
Correspondence to: Shuang Li <lishuang@mit.edu>, Yilun Du <yilundu@mit.edu>
Webpage: https://energy-based-model.github.io/Compositional-Visual-Generation-with-Composable-Diffusion-Models/

Today

After: Next batch of **paper presentations**.



Diffusion Policy: Visuomotor Policy Learning via Action Diffusion

Cheng Chi[*1], Zhenjia Xu[*1], Siyuan Feng[2], Eric Cousineau[2], Yilun Du[3], Benjamin Burchfiel[2], Russ Tedrake[2,3], Shuran Song[1,4]

Abstract

This paper introduces Diffusion Policy, a new way of generating robot behavior by representing a robot's visuomotor policy as a conditional denoising diffusion process. We benchmark Diffusion Policy across 15 different tasks from 4 different robot manipulation benchmarks and find that it consistently outperforms existing state-of-the-art robot learning methods with an average improvement of 46.9%. Diffusion Policy learns the gradient of the action-distribution score function and iteratively optimizes with respect to this gradient field during inference via a series of stochastic Langevin dynamics steps. We find that the diffusion formulation yields powerful advantages when used for robot policies, including gracefully handling multimodal action distributions, being suitable for high-dimensional action spaces, and exhibiting impressive training stability. To fully unlock the potential of diffusion models for visuomotor policy learning on physical robots, this paper presents a set of key technical contributions including the incorporation of receding horizon control, visual conditioning, and the time-series diffusion transformer. We hope this work will help motivate a new generation of policy learning techniques that are able to leverage the powerful generative modeling capabilities of diffusion models. Code, data, and training details is available diffusion-policy.cs.columbia.edu

Keywords

Imitation learning, visuomotor policy, manipulation

Figure 1. Policy Representations. a) Explicit policy with different types of action representations. b) Implicit policy learns an energy function conditioned on both action and observation and optimizes for actions that minimize the energy landscape c) Diffusion policy refines noise into actions via a learned gradient field. This formulation provides stable training, allows the learned policy to accurately model multimodal action distributions, and accommodates high-dimensional action sequences.

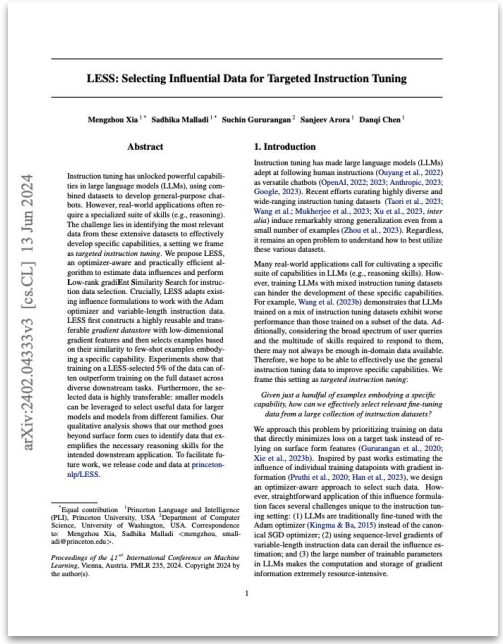
1 Introduction

Policy learning from demonstration, in its simplest form, can be formulated as the supervised regression task of learning to map observations to actions. In practice however, the unique nature of predicting robot actions — such as the existence of multimodal distributions, sequential correlation, and the requirement of high precision — makes this task distinct and challenging compared to other supervised learning problems.

Prior work attempts to address this challenge by exploring different action *representations* (Fig 1 a) – using mixtures of Gaussians Mandlekar et al. (2021), categorical representations of quantized actions Shafiullah et al. (2022), or by switching the *the policy representation* (Fig 1 b) – from explicit to implicit to better capture multi-modal distributions Florence et al. (2021); Wu et al. (2020).

In this work, we seek to address this challenge by introducing a new form of robot visuomotor policy that generates behavior via a "conditional denoising diffusion process Ho et al. (2020) on robot action space", **Diffusion Policy**. In this formulation, instead of directly outputting an action, the policy infers the action-score gradient, conditioned on visual observations, for K denoising iterations (Fig. 1 c). This formulation allows robot policies to inherit several key properties from diffusion models – significantly improving performance.

- **Expressing multimodal action distributions.** By learning the gradient of the action score function Song and Ermon (2019) and performing Stochastic Langevin Dynamics sampling on this gradient field, Diffusion policy can express arbitrary normalizable distributions Neal et al. (2011), which includes multimodal action distributions, a well-known challenge for policy learning.



LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia[1*] Sadhika Malladi[1*] Suchin Gururangan[2] Sanjeev Arora[1] Danqi Chen[1]

Abstract

Instruction tuning has unlocked powerful capabilities in large language models (LLMs), using combined datasets to develop general-purpose chatbots. However, real-world applications often require a specialized suite of skills (e.g., reasoning). The challenge lies in identifying the most relevant data from these extensive datasets to effectively develop specific capabilities, a setting we frame as *targeted instruction tuning*. We propose LESS, an optimizer-aware and practically efficient algorithm to estimate data influences and perform Low-rank gradIEnt Similarity Search for instruction data selection. Crucially, LESS adapts existing influence formulations to work with the Adam optimizer and variable-length instruction data. LESS first constructs a highly reusable and transferable *gradient datastore* with low-dimensional gradient features and then selects examples based on their similarity to few-shot examples embodying a specific capability. Experiments show that training on a LESS-selected 5% of the data can often outperform training on the full dataset across diverse downstream tasks. Furthermore, the selected data is highly transferable: smaller models can be leveraged to select useful data for larger models and models from different families. Our qualitative analysis shows that our method goes beyond surface form cues to identify data that exemplifies the necessary reasoning skills for the intended downstream application. To facilitate future work, we release code and data at princeton-nlp/LESS.

1. Introduction

Instruction tuning has made large language models (LLMs) adept at following human instructions (Ouyang et al., 2022) as versatile chatbots (OpenAI, 2022; 2023; Anthropic, 2023; Google, 2023). Recent efforts curating highly diverse and wide-ranging instruction tuning datasets (Taori et al., 2023; Wang et al.; Mukherjee et al., 2023; Xu et al., 2023, *inter alia*) induce remarkably strong generalization even from a small number of examples (Zhou et al., 2023). Regardless, it remains an open problem to understand how to best utilize these various datasets.

Many real-world applications call for cultivating a specific suite of capabilities in LLMs (e.g., reasoning skills). However, training LLMs with mixed instruction tuning datasets can hinder the development of these specific capabilities. For example, Wang et al. (2023b) demonstrates that LLMs trained on a mix of instruction tuning datasets exhibit worse performance than those trained on a subset of the data. Additionally, considering the broad spectrum of user queries and the multitude of skills required to respond to them, there may not always be enough in-domain data available. Therefore, we hope to be able to effectively use the general instruction tuning data to improve specific capabilities. We frame this setting as *targeted instruction tuning*:

Given just a handful of examples embodying a specific capability, how can we effectively select relevant fine-tuning data from a large collection of instruction datasets?

We approach this problem by prioritizing training on data that directly minimizes loss on a target task instead of relying on surface form features (Gururangan et al., 2020; Xie et al., 2023b). Inspired by past works estimating the influence of individual training datapoints with gradient information (Pruthi et al., 2020; Han et al., 2023), we design an optimizer-aware approach to select such data. However, straightforward application of this influence formulation faces several challenges unique to the instruction tuning setting: (1) LLMs are traditionally fine-tuned with the Adam optimizer (Kingma & Ba, 2015) instead of the canonical SGD optimizer; (2) using sequence-level gradients of variable-length instruction data can derail the influence estimation; and (3) the large number of trainable parameters in LLMs makes the computation and storage of gradient information extremely resource-intensive.

[*]Equal contribution [1]Princeton Language and Intelligence (PLI), Princeton University, USA. [2]Department of Computer Science, University of Washington, USA. Correspondence to: Mengzhou Xia, Sadhika Malladi <mengzhou, smalladi@princeton.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Published as a conference paper at ICLR 2021

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation without adversarial training, yet they require simulating a Markov chain for many steps in order to produce a sample. To accelerate sampling, we present denoising diffusion implicit models (DDIMs), a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPMs. In DDPMs, the generative process is defined as the reverse of a particular Markovian diffusion process. We generalize DDPMs via a class of non-Markovian diffusion processes that lead to the same training objective. These non-Markovian processes can correspond to generative processes that are deterministic, giving rise to implicit models that produce high quality samples much faster. We empirically demonstrate that DDIMs can produce high quality samples $10\times$ to $50\times$ faster in terms of wall-clock time compared to DDPMs, allow us to trade off computation for sample quality, perform semantically meaningful image interpolation directly in the latent space, and reconstruct observations with very low error.

1 INTRODUCTION

Deep generative models have demonstrated the ability to produce high quality samples in many domains (Karras et al., 2020; van den Oord et al., 2016a). In terms of image generation, generative adversarial networks (GANs, Goodfellow et al. (2014)) currently exhibits higher sample quality than likelihood-based methods such as variational autoencoders (Kingma & Welling, 2013), autoregressive models (van den Oord et al., 2016b) and normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2016). However, GANs require very specific choices in optimization and architectures in order to stabilize training (Arjovsky et al., 2017; Gulrajani et al., 2017; Karras et al., 2018; Brock et al., 2018), and could fail to cover modes of the data distribution (Zhao et al., 2018).

Recent works on iterative generative models (Bengio et al., 2014), such as denoising diffusion probabilistic models (DDPM, Ho et al. (2020)) and noise conditional score networks (NCSN, Song & Ermon (2019)) have demonstrated the ability to produce samples comparable to that of GANs, without having to perform adversarial training. To achieve this, many denoising autoencoding models are trained to denoise samples corrupted by various levels of Gaussian noise. Samples are then produced by a Markov chain which, starting from white noise, progressively denoises it into an image. This generative Markov Chain process is either based on Langevin dynamics (Song & Ermon, 2019) or obtained by reversing a forward *diffusion process* that progressively turns an image into noise (Sohl-Dickstein et al., 2015).

A critical drawback of these models is that they require many iterations to produce a high quality sample. For DDPMs, this is because that the generative process (from noise to data) approximates the reverse of the forward *diffusion process* (from data to noise), which could have thousands of steps; iterating over all the steps is required to produce a single sample, which is much slower compared to GANs, which only needs one pass through a network. For example, it takes around 20 hours to sample 50k images of size 32×32 from a DDPM, but less than a minute to do so from a GAN on a Nvidia 2080 Ti GPU. This becomes more problematic for larger images as sampling 50k images of size 256×256 could take nearly 1000 hours on the same GPU.

To close this efficiency gap between DDPMs and GANs, we present denoising diffusion implicit models (DDIMs). DDIMs are implicit probabilistic models (Mohamed & Lakshminarayanan, 2016) and are closely related to DDPMs, in the sense that they are trained with the same objective function.

Published as a conference paper at ICLR 2023

DIFFDOCK: DIFFUSION STEPS, TWISTS, AND TURNS FOR MOLECULAR DOCKING

Gabriele Corso*, Hannes Stärk*, Bowen Jing*, Regina Barzilay & Tommi Jaakkola
CSAIL, Massachusetts Institute of Technology

ABSTRACT

Predicting the binding structure of a small molecule ligand to a protein—a task known as *molecular docking*—is critical to drug design. Recent deep learning methods that treat docking as a regression problem have decreased runtime compared to traditional search-based methods but have yet to offer substantial improvements in accuracy. We instead frame molecular docking as a *generative* modeling problem and develop DIFFDOCK, a diffusion generative model over the non-Euclidean manifold of ligand poses. To do so, we map this manifold to the product space of the degrees of freedom (translational, rotational, and torsional) involved in docking and develop an efficient diffusion process on this space. Empirically, DIFFDOCK obtains a 38% top-1 success rate (RMSD<2Å) on PDBBind, significantly outperforming the previous state-of-the-art of traditional docking (23%) and deep learning (20%) methods. Moreover, while previous methods are not able to dock on computationally folded structures (maximum accuracy 10.4%), DIFFDOCK maintains significantly higher precision (21.7%). Finally, DIFFDOCK has fast inference times and provides confidence estimates with high selective accuracy.

1 INTRODUCTION

The biological functions of proteins can be modulated by small molecule ligands (such as drugs) binding to them. Thus, a crucial task in computational drug design is *molecular docking*—predicting the position, orientation, and conformation of a ligand when bound to a target protein—from which the effect of the ligand (if any) might be inferred. Traditional approaches for docking [Trott & Olson, 2010; Halgren et al., 2004] rely on scoring-functions that estimate the correctness of a proposed structure or pose, and an optimization algorithm that searches for the global maximum of the scoring function. However, since the search space is vast and the landscape of the scoring functions rugged, these methods tend to be too slow and inaccurate, especially for high-throughput workflows.

Recent works [Stärk et al., 2022; Lu et al., 2022] have developed deep learning models to predict the binding pose in one shot, treating docking as a regression problem. While these methods are much faster than traditional search-based methods, they have yet to demonstrate significant improvements in accuracy. We argue that this may be because the regression-based paradigm corresponds imperfectly with the objectives of molecular docking, which is reflected in the fact that standard accuracy metrics resemble the *likelihood* of the data under the predictive model rather than a regression loss. We thus frame molecular docking as a *generative modeling problem*—given a ligand and target protein structure, we learn a distribution over ligand poses.

To this end, we develop DIFFDOCK, a diffusion generative model (DGM) over the space of ligand poses for molecular docking. We define a diffusion process over the degrees of freedom involved in docking: the position of the ligand relative to the protein (locating the binding pocket), its orientation in the pocket, and the torsion angles describing its conformation. DIFFDOCK samples poses by running the learned (reverse) diffusion process, which iteratively transforms an uninformed, noisy prior distribution over ligand poses into the learned model distribution (Figure 1). Intuitively, this process can be viewed as the progressive refinement of random poses via updates of their translations, rotations, and torsion angles.

*Equal contribution. Correspondence to {gcorso, hstark, bjing}@mit.edu.

Compositional Visual Generation with Composable Diffusion Models

Nan Liu[1*], Shuang Li[2*], Yilun Du[2*]
Antonio Torralba[2], and Joshua B. Tenenbaum[2]

[1] University of Illinois Urbana-Champaign
[2] Massachusetts Institute of Technology
nanliu6@illinois.edu, {lishuang,yilundu,torralba,jbt}@mit.edu

Abstract. Large text-guided diffusion models, such as DALL-E 2, are able to generate stunning photorealistic images given natural language descriptions. While such models are highly flexible, they struggle to understand the composition of certain concepts, such as confusing the attributes of different objects or relations between objects. In this paper, we propose an alternative structured approach for compositional generation using diffusion models. An image is generated by composing a set of diffusion models, with each of them modeling a certain component of the image. To do this, we interpret diffusion models as energy-based models in which the data distributions defined by the energy functions may be explicitly combined. The proposed method can generate scenes at test time that are substantially more complex than those seen in training, composing sentence descriptions, object relations, human facial attributes, and even generalizing to new combinations that are rarely seen in the real world. We further illustrate how our approach may be used to compose pre-trained text-guided diffusion models and generate photorealistic images containing all the details described in the input descriptions, including the binding of certain object attributes that have been shown difficult for DALL-E 2. These results point to the effectiveness of the proposed method in promoting structured generalization for visual generation.

Keywords: Compositionality, Diffusion Models, Energy-based Models, Visual Generation

1 Introduction

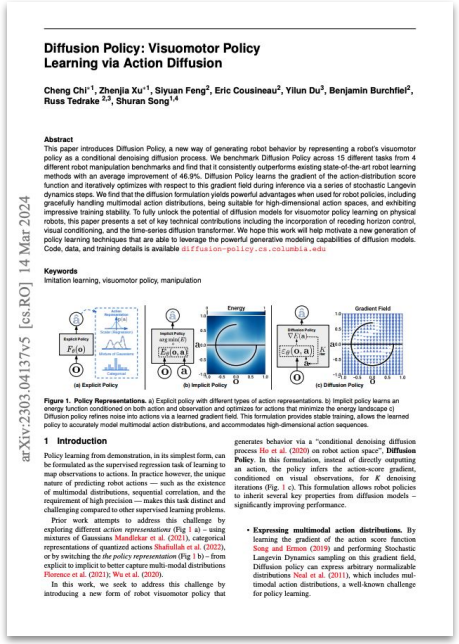
Our understanding of the world is highly compositional in nature. We are able to rapidly understand new objects from their components, or compose words into complex sentences to describe the world states we encounter [26]. We are able

[*] indicates equal contribution.
Correspondence to: Shuang Li <lishuang@mit.edu>, Yilun Du <yilundu@mit.edu>
Webpage: https://energy-based-model.github.io/Compositional-Visual-Generation-with-Composable-Diffusion-Models/

Diffusion Policy — denoising diffusion for robotics

Today

After: Next batch of **paper presentations**.



Diffusion Policy: Visuomotor Policy Learning via Action Diffusion

Cheng Chi[*1], Zhenjia Xu[*1], Siyuan Feng[2], Eric Cousineau[2], Yilun Du[3], Benjamin Burchfiel[2], Russ Tedrake[2,3], Shuran Song[1,4]

Abstract

This paper introduces Diffusion Policy, a new way of generating robot behavior by representing a robot's visuomotor policy as a conditional denoising diffusion process. We benchmark Diffusion Policy across 15 different tasks from 4 different robot manipulation benchmarks and find that it consistently outperforms existing state-of-the-art robot learning methods with an average improvement of 46.9%. Diffusion Policy learns the gradient of the action-distribution score function and iteratively optimizes with respect to this gradient field during inference via a series of stochastic Langevin dynamics steps. We find that the diffusion formulation yields powerful advantages when used for robot policies, including gracefully handling multimodal action distributions, being suitable for high-dimensional action spaces, and exhibiting impressive training stability. To fully unlock the potential of diffusion models for visuomotor policy learning on physical robots, this paper presents a set of key technical contributions including the incorporation of receding horizon control, visual conditioning, and the time-series diffusion transformer. We hope this work will help motivate a new generation of policy learning techniques that are able to leverage the powerful generative modeling capabilities of diffusion models. Code, data, and training details is available diffusion-policy.cs.columbia.edu

Keywords

Imitation learning, visuomotor policy, manipulation

Figure 1. Policy Representations. a) Explicit policy with different types of action representations. b) Implicit policy learns an energy function conditioned on both action and observation and optimizes for actions that minimize the energy landscape c) Diffusion policy refines noise into actions via a learned gradient field. This formulation provides stable training, allows the learned policy to accurately model multimodal action distributions, and accommodates high-dimensional action sequences.

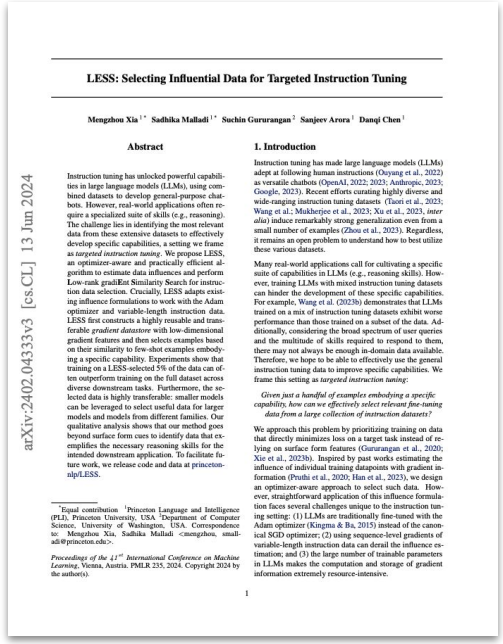
1 Introduction

Policy learning from demonstration, in its simplest form, can be formulated as the supervised regression task of learning to map observations to actions. In practice however, the unique nature of predicting robot actions — such as the existence of multimodal distributions, sequential correlation, and the requirement of high precision — makes this task distinct and challenging compared to other supervised learning problems.

Prior work attempts to address this challenge by exploring different *action representations* (Fig 1 a) – using mixtures of Gaussians Mandlekar et al. (2021), categorical representations of quantized actions Shafiullah et al. (2022), or by switching the *the policy representation* (Fig 1 b) – from explicit to implicit to better capture multi-modal distributions Florence et al. (2021); Wu et al. (2020).

In this work, we seek to address this challenge by introducing a new form of robot visuomotor policy that generates behavior via a "conditional denoising diffusion process Ho et al. (2020) on robot action space", **Diffusion Policy**. In this formulation, instead of directly outputting an action, the policy infers the action-score gradient, conditioned on visual observations, for K denoising iterations (Fig. 1 c). This formulation allows robot policies to inherit several key properties from diffusion models – significantly improving performance.

- **Expressing multimodal action distributions.** By learning the gradient of the action score function Song and Ermon (2019) and performing Stochastic Langevin Dynamics sampling on this gradient field, Diffusion policy can express arbitrary normalizable distributions Neal et al. (2011), which includes multimodal action distributions, a well-known challenge for policy learning.



LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia[1*] Sadhika Malladi[1*] Suchin Gururangan[2] Sanjeev Arora[1] Danqi Chen[1]

Abstract

Instruction tuning has unlocked powerful capabilities in large language models (LLMs), using combined datasets to develop general-purpose chatbots. However, real-world applications often require a specialized suite of skills (e.g., reasoning). The challenge lies in identifying the most relevant data from these extensive datasets to effectively develop specific capabilities, a setting we frame as *targeted instruction tuning*. We propose LESS, an optimizer-aware and practically efficient algorithm to estimate data influences and perform Low-rank gradIEnt Similarity Search for instruction data selection. Crucially, LESS adapts existing influence formulations to work with the Adam optimizer and variable-length instruction data. LESS first constructs a highly reusable and transferable *gradient datastore* with low-dimensional gradient features and then selects examples based on their similarity to few-shot examples embodying a specific capability. Experiments show that training on a LESS-selected 5% of the data can often outperform training on the full dataset across diverse downstream tasks. Furthermore, the selected data is highly transferable: smaller models can be leveraged to select useful data for larger models and models from different families. Our qualitative analysis shows that our method goes beyond surface form cues to identify data that exemplifies the necessary reasoning skills for the intended downstream application. To facilitate future work, we release code and data at princeton-nlp/LESS.

1. Introduction

Instruction tuning has made large language models (LLMs) adept at following human instructions (Ouyang et al., 2022) as versatile chatbots (OpenAI, 2022; 2023; Anthropic, 2023; Google, 2023). Recent efforts curating highly diverse and wide-ranging instruction tuning datasets (Taori et al., 2023; Wang et al.; Mukherjee et al., 2023; Xu et al., 2023, *inter alia*) induce remarkably strong generalization even from a small number of examples (Zhou et al., 2023). Regardless, it remains an open problem to understand how to best utilize these various datasets.

Many real-world applications call for cultivating a specific suite of capabilities in LLMs (e.g., reasoning skills). However, training LLMs with mixed instruction tuning datasets can hinder the development of these specific capabilities. For example, Wang et al. (2023b) demonstrates that LLMs trained on a mix of instruction tuning datasets exhibit worse performance than those trained on a subset of the data. Additionally, considering the broad spectrum of user queries and the multitude of skills required to respond to them, there may not always be enough in-domain data available. Therefore, we hope to be able to effectively use the general instruction tuning data to improve specific capabilities. We frame this setting as *targeted instruction tuning*:

Given just a handful of examples embodying a specific capability, how can we effectively select relevant fine-tuning data from a large collection of instruction datasets?

We approach this problem by prioritizing training on data that directly minimizes loss on a target task instead of relying on surface form features (Gururangan et al., 2020; Xie et al., 2023b). Inspired by past works estimating the influence of individual training datapoints with gradient information (Pruthi et al., 2020; Han et al., 2023), we design an optimizer-aware approach to select such data. However, straightforward application of this influence formulation faces several challenges unique to the instruction tuning setting: (1) LLMs are traditionally fine-tuned with the Adam optimizer (Kingma & Ba, 2015) instead of the canonical SGD optimizer; (2) using sequence-level gradients of variable-length instruction data can derail the influence estimation; and (3) the large number of trainable parameters in LLMs makes the computation and storage of gradient information extremely resource-intensive.

[1]Equal contribution [1]Princeton Language and Intelligence (PLI), Princeton University, USA. [2]Department of Computer Science, University of Washington, USA. Correspondence to: Mengzhou Xia, Sadhika Malladi <mengzhou, smalladi@princeton.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Published as a conference paper at ICLR 2021

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation without adversarial training, yet they require simulating a Markov chain for many steps in order to produce a sample. To accelerate sampling, we present denoising diffusion implicit models (DDIMs), a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPMs. In DDPMs, the generative process is defined as the reverse of a particular Markovian diffusion process. We generalize DDPMs via a class of non-Markovian diffusion processes that lead to the same training objective. These non-Markovian processes can correspond to generative processes that are deterministic, giving rise to implicit models that produce high quality samples much faster. We empirically demonstrate that DDIMs can produce high quality samples $10\times$ to $50\times$ faster in terms of wall-clock time compared to DDPMs, allow us to trade off computation for sample quality, perform semantically meaningful image interpolation directly in the latent space, and reconstruct observations with very low error.

1 INTRODUCTION

Deep generative models have demonstrated the ability to produce high quality samples in many domains (Karras et al., 2020; van den Oord et al., 2016a). In terms of image generation, generative adversarial networks (GANs, Goodfellow et al. (2014)) currently exhibits higher sample quality than likelihood-based methods such as variational autoencoders (Kingma & Welling, 2013), autoregressive models (van den Oord et al., 2016b) and normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2016). However, GANs require very specific choices in optimization and architectures in order to stabilize training (Arjovsky et al., 2017; Gulrajani et al., 2017; Karras et al., 2018; Brock et al., 2018), and could fail to cover modes of the data distribution (Zhao et al., 2018).

Recent works on iterative generative models (Bengio et al., 2014), such as denoising diffusion probabilistic models (DDPM, Ho et al. (2020)) and noise conditional score networks (NCSN, Song & Ermon (2019)) have demonstrated the ability to produce samples comparable to that of GANs, without having to perform adversarial training. To achieve this, many denoising autoencoding models are trained to denoise samples corrupted by various levels of Gaussian noise. Samples are then produced by a Markov chain which, starting from white noise, progressively denoises it into an image. This generative Markov Chain process is either based on Langevin dynamics (Song & Ermon, 2019) or obtained by reversing a forward *diffusion process* that progressively turns an image into noise (Sohl-Dickstein et al., 2015).

A critical drawback of these models is that they require many iterations to produce a high quality sample. For DDPMs, this is because that the generative process (from noise to data) approximates the reverse of the forward *diffusion process* (from data to noise), which could have thousands of steps; iterating over all the steps is required to produce a single sample, which is much slower compared to GANs, which only needs one pass through a network. For example, it takes around 20 hours to sample 50k images of size 32×32 from a DDPM, but less than a minute to do so from a GAN on a Nvidia 2080 Ti GPU. This becomes more problematic for larger images as sampling 50k images of size 256×256 could take nearly 1000 hours on the same GPU.

To close this efficiency gap between DDPMs and GANs, we present denoising diffusion implicit models (DDIMs). DDIMs are implicit probabilistic models (Mohamed & Lakshminarayanan, 2016) and are closely related to DDPMs, in the sense that they are trained with the same objective function.

Published as a conference paper at ICLR 2023

DIFFDOCK: DIFFUSION STEPS, TWISTS, AND TURNS FOR MOLECULAR DOCKING

Gabriele Corso*, Hannes Stärk*, Bowen Jing*, Regina Barzilay & Tommi Jaakkola
CSAIL, Massachusetts Institute of Technology

ABSTRACT

Predicting the binding structure of a small molecule ligand to a protein—a task known as *molecular docking*—is critical to drug design. Recent deep learning methods that treat docking as a regression problem have decreased runtime compared to traditional search-based methods but have yet to offer substantial improvements in accuracy. We instead frame molecular docking as a *generative* modeling problem and develop DIFFDOCK, a diffusion generative model over the non-Euclidean manifold of ligand poses. To do so, we map this manifold to the product space of the degrees of freedom (translational, rotational, and torsional) involved in docking and develop an efficient diffusion process on this space. Empirically, DIFFDOCK obtains a 38% top-1 success rate (RMSD<2Å) on PDBBind, significantly outperforming the previous state-of-the-art of traditional docking (23%) and deep learning (20%) methods. Moreover, while previous methods are not able to dock on computationally folded structures (maximum accuracy 10.4%), DIFFDOCK maintains significantly higher precision (21.7%). Finally, DIFFDOCK has fast inference times and provides confidence estimates with high selective accuracy.

1 INTRODUCTION

The biological functions of proteins can be modulated by small molecule ligands (such as drugs) binding to them. Thus, a crucial task in computational drug design is *molecular docking*—predicting the position, orientation, and conformation of a ligand when bound to a target protein—from which the effect of the ligand (if any) might be inferred. Traditional approaches for docking [Trott & Olson, 2010; Halgren et al., 2004] rely on scoring-functions that estimate the correctness of a proposed structure or pose, and an optimization algorithm that searches for the global maximum of the scoring function. However, since the search space is vast and the landscape of the scoring functions rugged, these methods tend to be too slow and inaccurate, especially for high-throughput workflows.

Recent works [Stärk et al., 2022; Lu et al., 2022] have developed deep learning models to predict the binding pose in one shot, treating docking as a regression problem. While these methods are much faster than traditional search-based methods, they have yet to demonstrate significant improvements in accuracy. We argue that this may be because the regression-based paradigm corresponds imperfectly with the objectives of molecular docking, which is reflected in the fact that standard accuracy metrics resemble the *likelihood* of the data under the predictive model rather than a regression loss. We thus frame molecular docking as a *generative modeling problem*—given a ligand and target protein structure, we learn a distribution over ligand poses.

To this end, we develop DIFFDOCK, a diffusion generative model (DGM) over the space of ligand poses for molecular docking. We define a diffusion process over the degrees of freedom involved in docking: the position of the ligand relative to the protein (locating the binding pocket), its orientation in the pocket, and the torsion angles describing its conformation. DIFFDOCK samples poses by running the learned (reverse) diffusion process, which iteratively transforms an uninformed, noisy prior distribution over ligand poses into the learned model distribution (Figure 1). Intuitively, this process can be viewed as the progressive refinement of random poses via updates of their translations, rotations, and torsion angles.

*Equal contribution. Correspondence to {gcorso, hstark, bjing}@mit.edu.

Compositional Visual Generation with Composable Diffusion Models

Nan Liu[1*], Shuang Li[2*], Yilun Du[2*]
Antonio Torralba[2], and Joshua B. Tenenbaum[2]

[1] University of Illinois Urbana-Champaign
[2] Massachusetts Institute of Technology
nanliu6@illinois.edu, {lishuang,yilundu,torralba,jbt}@mit.edu

Abstract. Large text-guided diffusion models, such as DALL-E 2, are able to generate stunning photorealistic images given natural language descriptions. While such models are highly flexible, they struggle to understand the composition of certain concepts, such as confusing the attributes of different objects or relations between objects. In this paper, we propose an alternative structured approach for compositional generation using diffusion models. An image is generated by composing a set of diffusion models, with each of them modeling a certain component of the image. To do this, we interpret diffusion models as energy-based models in which the data distributions defined by the energy functions may be explicitly combined. The proposed method can generate scenes at test time that are substantially more complex than those seen in training, composing sentence descriptions, object relations, human facial attributes, and even generalizing to new combinations that are rarely seen in the real world. We further illustrate how our approach may be used to compose pre-trained text-guided diffusion models and generate photorealistic images containing all the details described in the input descriptions, including the binding of certain object attributes that have been shown difficult for DALL-E 2. These results point to the effectiveness of the proposed method in promoting structured generalization for visual generation.

Keywords: Compositionality, Diffusion Models, Energy-based Models, Visual Generation

1 Introduction

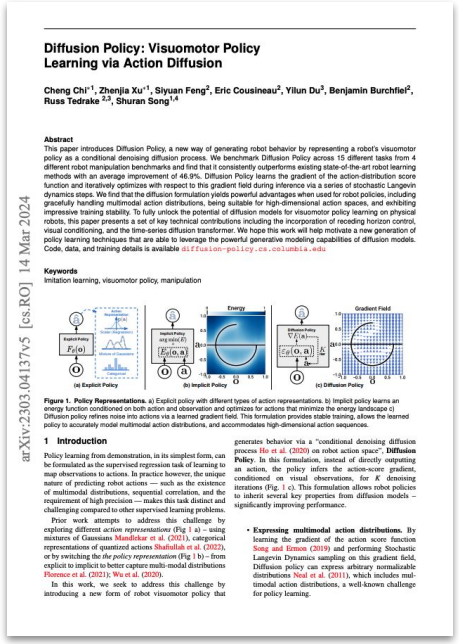
Our understanding of the world is highly compositional in nature. We are able to rapidly understand new objects from their components, or compose words into complex sentences to describe the world states we encounter [26]. We are able

* indicates equal contribution.
Correspondence to: Shuang Li <lishuang@mit.edu>, Yilun Du <yilundu@mit.edu>
Webpage: https://energy-based-model.github.io/Compositional-Visual-Generation-with-Composable-Diffusion-Models/

LESS — gradient-based instruction data selection

Today

After: Next batch of **paper presentations**.



Diffusion Policy: Visuomotor Policy Learning via Action Diffusion

Cheng Chi[*1], Zhenjia Xu[*1], Siyuan Feng[2], Eric Cousineau[2], Yilun Du[3], Benjamin Burchfiel[2], Russ Tedrake[2,3], Shuran Song[1,4]

Abstract

This paper introduces Diffusion Policy, a new way of generating robot behavior by representing a robot's visuomotor policy as a conditional denoising diffusion process. We benchmark Diffusion Policy across 15 different tasks from 4 different robot manipulation benchmarks and find that it consistently outperforms existing state-of-the-art robot learning methods with an average improvement of 46.9%. Diffusion Policy learns the gradient of the action-distribution score function and iteratively optimizes with respect to this gradient field during inference via a series of stochastic Langevin dynamics steps. We find that the diffusion formulation yields powerful advantages when used for robot policies, including gracefully handling multimodal action distributions, being suitable for high-dimensional action spaces, and exhibiting impressive training stability. To fully unlock the potential of diffusion models for visuomotor policy learning on physical robots, this paper presents a set of key technical contributions including the incorporation of receding horizon control, visual conditioning, and the time-series diffusion transformer. We hope this work will help motivate a new generation of policy learning techniques that are able to leverage the powerful generative modeling capabilities of diffusion models. Code, data, and training details is available diffusion-policy.cs.columbia.edu

Keywords

Imitation learning, visuomotor policy, manipulation

Figure 1. Policy Representations. a) Explicit policy with different types of action representations. b) Implicit policy learns an energy function conditioned on both action and observation and optimizes for actions that minimize the energy landscape c) Diffusion policy refines noise into actions via a learned gradient field. This formulation provides stable training, allows the learned policy to accurately model multimodal action distributions, and accommodates high-dimensional action sequences.

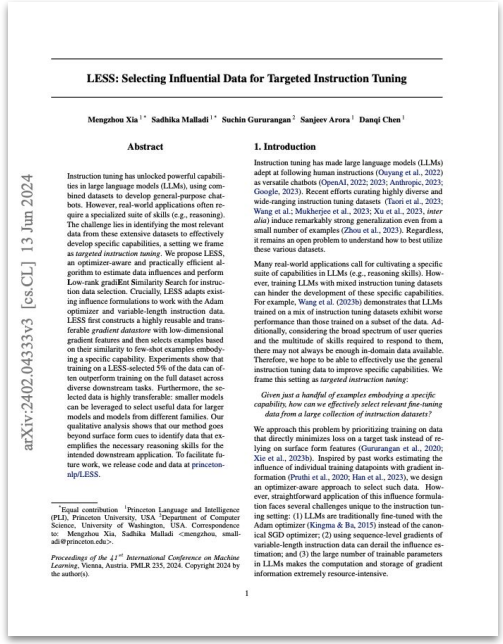
1 Introduction

Policy learning from demonstration, in its simplest form, can be formulated as the supervised regression task of learning to map observations to actions. In practice however, the unique nature of predicting robot actions — such as the existence of multimodal distributions, sequential correlation, and the requirement of high precision — makes this task distinct and challenging compared to other supervised learning problems.

Prior work attempts to address this challenge by exploring different *action representations* (Fig 1 a) – using mixtures of Gaussians Mandlekar et al. (2021), categorical representations of quantized actions Shafiullah et al. (2022), or by switching the *the policy representation* (Fig 1 b) – from explicit to implicit to better capture multi-modal distributions Florence et al. (2021); Wu et al. (2020).

In this work, we seek to address this challenge by introducing a new form of robot visuomotor policy that generates behavior via a "conditional denoising diffusion process Ho et al. (2020) on robot action space", **Diffusion Policy**. In this formulation, instead of directly outputting an action, the policy infers the action-score gradient, conditioned on visual observations, for K denoising iterations (Fig. 1 c). This formulation allows robot policies to inherit several key properties from diffusion models – significantly improving performance.

- **Expressing multimodal action distributions.** By learning the gradient of the action score function Song and Ermon (2019) and performing Stochastic Langevin Dynamics sampling on this gradient field, Diffusion policy can express arbitrary normalizable distributions Neal et al. (2011), which includes multimodal action distributions, a well-known challenge for policy learning.



LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia[1*] Sadhika Malladi[1*] Suchin Gururangan[2] Sanjeev Arora[1] Danqi Chen[1]

Abstract

Instruction tuning has unlocked powerful capabilities in large language models (LLMs), using combined datasets to develop general-purpose chatbots. However, real-world applications often require a specialized suite of skills (e.g., reasoning). The challenge lies in identifying the most relevant data from these extensive datasets to effectively develop specific capabilities, a setting we frame as *targeted instruction tuning*. We propose LESS, an optimizer-aware and practically efficient algorithm to estimate data influences and perform Low-rank gradIEnt Similarity Search for instruction data selection. Crucially, LESS adapts existing influence formulations to work with the Adam optimizer and variable-length instruction data. LESS first constructs a highly reusable and transferable *gradient datastore* with low-dimensional gradient features and then selects examples based on their similarity to few-shot examples embodying a specific capability. Experiments show that training on a LESS-selected 5% of the data can often outperform training on the full dataset across diverse downstream tasks. Furthermore, the selected data is highly transferable: smaller models can be leveraged to select useful data for larger models and models from different families. Our qualitative analysis shows that our method goes beyond surface form cues to identify data that exemplifies the necessary reasoning skills for the intended downstream application. To facilitate future work, we release code and data at princeton-nlp/LESS.

1. Introduction

Instruction tuning has made large language models (LLMs) adept at following human instructions (Ouyang et al., 2022) as versatile chatbots (OpenAI, 2022; 2023; Anthropic, 2023; Google, 2023). Recent efforts curating highly diverse and wide-ranging instruction tuning datasets (Taori et al., 2023; Wang et al.; Mukherjee et al., 2023; Xu et al., 2023, *inter alia*) induce remarkably strong generalization even from a small number of examples (Zhou et al., 2023). Regardless, it remains an open problem to understand how to best utilize these various datasets.

Many real-world applications call for cultivating a specific suite of capabilities in LLMs (e.g., reasoning skills). However, training LLMs with mixed instruction tuning datasets can hinder the development of these specific capabilities. For example, Wang et al. (2023b) demonstrates that LLMs trained on a mix of instruction tuning datasets exhibit worse performance than those trained on a subset of the data. Additionally, considering the broad spectrum of user queries and the multitude of skills required to respond to them, there may not always be enough in-domain data available. Therefore, we hope to be able to effectively use the general instruction tuning data to improve specific capabilities. We frame this setting as *targeted instruction tuning*:

Given just a handful of examples embodying a specific capability, how can we effectively select relevant fine-tuning data from a large collection of instruction datasets?

We approach this problem by prioritizing training on data that directly minimizes loss on a target task instead of relying on surface form features (Gururangan et al., 2020; Xie et al., 2023b). Inspired by past works estimating the influence of individual training datapoints with gradient information (Pruthi et al., 2020; Han et al., 2023), we design an optimizer-aware approach to select such data. However, straightforward application of this influence formulation faces several challenges unique to the instruction tuning setting: (1) LLMs are traditionally fine-tuned with the Adam optimizer (Kingma & Ba, 2015) instead of the canonical SGD optimizer; (2) using sequence-level gradients of variable-length instruction data can derail the influence estimation; and (3) the large number of trainable parameters in LLMs makes the computation and storage of gradient information extremely resource-intensive.

[*]Equal contribution [1]Princeton Language and Intelligence (PLI), Princeton University, USA. [2]Department of Computer Science, University of Washington, USA. Correspondence to: Mengzhou Xia, Sadhika Malladi <mengzhou, smalladi@princeton.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Published as a conference paper at ICLR 2021

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation without adversarial training, yet they require simulating a Markov chain for many steps in order to produce a sample. To accelerate sampling, we present denoising diffusion implicit models (DDIMs), a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPMs. In DDPMs, the generative process is defined as the reverse of a particular Markovian diffusion process. We generalize DDPMs via a class of non-Markovian diffusion processes that lead to the same training objective. These non-Markovian processes can correspond to generative processes that are deterministic, giving rise to implicit models that produce high quality samples much faster. We empirically demonstrate that DDIMs can produce high quality samples $10\times$ to $50\times$ faster in terms of wall-clock time compared to DDPMs, allow us to trade off computation for sample quality, perform semantically meaningful image interpolation directly in the latent space, and reconstruct observations with very low error.

1 INTRODUCTION

Deep generative models have demonstrated the ability to produce high quality samples in many domains (Karras et al., 2020; van den Oord et al., 2016a). In terms of image generation, generative adversarial networks (GANs, Goodfellow et al. (2014)) currently exhibits higher sample quality than likelihood-based methods such as variational autoencoders (Kingma & Welling, 2013), autoregressive models (van den Oord et al., 2016b) and normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2016). However, GANs require very specific choices in optimization and architectures in order to stabilize training (Arjovsky et al., 2017; Gulrajani et al., 2017; Karras et al., 2018; Brock et al., 2018), and could fail to cover modes of the data distribution (Zhao et al., 2018).

Recent works on iterative generative models (Bengio et al., 2014), such as denoising diffusion probabilistic models (DDPM, Ho et al. (2020)) and noise conditional score networks (NCSN, Song & Ermon (2019)) have demonstrated the ability to produce samples comparable to that of GANs, without having to perform adversarial training. To achieve this, many denoising autoencoding models are trained to denoise samples corrupted by various levels of Gaussian noise. Samples are then produced by a Markov chain which, starting from white noise, progressively denoises it into an image. This generative Markov Chain process is either based on Langevin dynamics (Song & Ermon, 2019) or obtained by reversing a forward *diffusion process* that progressively turns an image into noise (Sohl-Dickstein et al., 2015).

A critical drawback of these models is that they require many iterations to produce a high quality sample. For DDPMs, this is because that the generative process (from noise to data) approximates the reverse of the forward *diffusion process* (from data to noise), which could have thousands of steps; iterating over all the steps is required to produce a single sample, which is much slower compared to GANs, which only needs one pass through a network. For example, it takes around 20 hours to sample 50k images of size 32×32 from a DDPM, but less than a minute to do so from a GAN on a Nvidia 2080 Ti GPU. This becomes more problematic for larger images as sampling 50k images of size 256×256 could take nearly 1000 hours on the same GPU.

To close this efficiency gap between DDPMs and GANs, we present denoising diffusion implicit models (DDIMs). DDIMs are implicit probabilistic models (Mohamed & Lakshminarayanan, 2016) and are closely related to DDPMs, in the sense that they are trained with the same objective function.

Published as a conference paper at ICLR 2023

DIFFDOCK: DIFFUSION STEPS, TWISTS, AND TURNS FOR MOLECULAR DOCKING

Gabriele Corso*, Hannes Stärk*, Bowen Jing*, Regina Barzilay & Tommi Jaakkola
CSAIL, Massachusetts Institute of Technology

ABSTRACT

Predicting the binding structure of a small molecule ligand to a protein—a task known as *molecular docking*—is critical to drug design. Recent deep learning methods that treat docking as a regression problem have decreased runtime compared to traditional search-based methods but have yet to offer substantial improvements in accuracy. We instead frame molecular docking as a *generative* modeling problem and develop DIFFDOCK, a diffusion generative model over the non-Euclidean manifold of ligand poses. To do so, we map this manifold to the product space of the degrees of freedom (translational, rotational, and torsional) involved in docking and develop an efficient diffusion process on this space. Empirically, DIFFDOCK obtains a 38% top-1 success rate (RMSD<2Å) on PDBBind, significantly outperforming the previous state-of-the-art of traditional docking (23%) and deep learning (20%) methods. Moreover, while previous methods are not able to dock on computationally folded structures (maximum accuracy 10.4%), DIFFDOCK maintains significantly higher precision (21.7%). Finally, DIFFDOCK has fast inference times and provides confidence estimates with high selective accuracy.

1 INTRODUCTION

The biological functions of proteins can be modulated by small molecule ligands (such as drugs) binding to them. Thus, a crucial task in computational drug design is *molecular docking*—predicting the position, orientation, and conformation of a ligand when bound to a target protein—from which the effect of the ligand (if any) might be inferred. Traditional approaches for docking [Trott & Olson, 2010; Halgren et al., 2004] rely on scoring-functions that estimate the correctness of a proposed structure or pose, and an optimization algorithm that searches for the global maximum of the scoring function. However, since the search space is vast and the landscape of the scoring functions rugged, these methods tend to be too slow and inaccurate, especially for high-throughput workflows.

Recent works [Stärk et al., 2022; Lu et al., 2022] have developed deep learning models to predict the binding pose in one shot, treating docking as a regression problem. While these methods are much faster than traditional search-based methods, they have yet to demonstrate significant improvements in accuracy. We argue that this may be because the regression-based paradigm corresponds imperfectly with the objectives of molecular docking, which is reflected in the fact that standard accuracy metrics resemble the *likelihood* of the data under the predictive model rather than a regression loss. We thus frame molecular docking as a *generative modeling problem*—given a ligand and target protein structure, we learn a distribution over ligand poses.

To this end, we develop DIFFDOCK, a diffusion generative model (DGM) over the space of ligand poses for molecular docking. We define a diffusion process over the degrees of freedom involved in docking: the position of the ligand relative to the protein (locating the binding pocket), its orientation in the pocket, and the torsion angles describing its conformation. DIFFDOCK samples poses by running the learned (reverse) diffusion process, which iteratively transforms an uninformed, noisy prior distribution over ligand poses into the learned model distribution (Figure 1). Intuitively, this process can be viewed as the progressive refinement of random poses via updates of their translations, rotations, and torsion angles.

*Equal contribution. Correspondence to {gcorso, hstark, bjing}@mit.edu.

Compositional Visual Generation with Composable Diffusion Models

Nan Liu[1*], Shuang Li[2*], Yilun Du[2*]
Antonio Torralba[2], and Joshua B. Tenenbaum[2]

[1] University of Illinois Urbana-Champaign
[2] Massachusetts Institute of Technology
nanliu6@illinois.edu, {lishuang,yilundu,torralba,jbt}@mit.edu

Abstract. Large text-guided diffusion models, such as DALL-E 2, are able to generate stunning photorealistic images given natural language descriptions. While such models are highly flexible, they struggle to understand the composition of certain concepts, such as confusing the attributes of different objects or relations between objects. In this paper, we propose an alternative structured approach for compositional generation using diffusion models. An image is generated by composing a set of diffusion models, with each of them modeling a certain component of the image. To do this, we interpret diffusion models as energy-based models in which the data distributions defined by the energy functions may be explicitly combined. The proposed method can generate scenes at test time that are substantially more complex than those seen in training, composing sentence descriptions, object relations, human facial attributes, and even generalizing to new combinations that are rarely seen in the real world. We further illustrate how our approach may be used to compose pre-trained text-guided diffusion models and generate photorealistic images containing all the details described in the input descriptions, including the binding of certain object attributes that have been shown difficult for DALL-E 2. These results point to the effectiveness of the proposed method in promoting structured generalization for visual generation.

Keywords: Compositionality, Diffusion Models, Energy-based Models, Visual Generation

1 Introduction

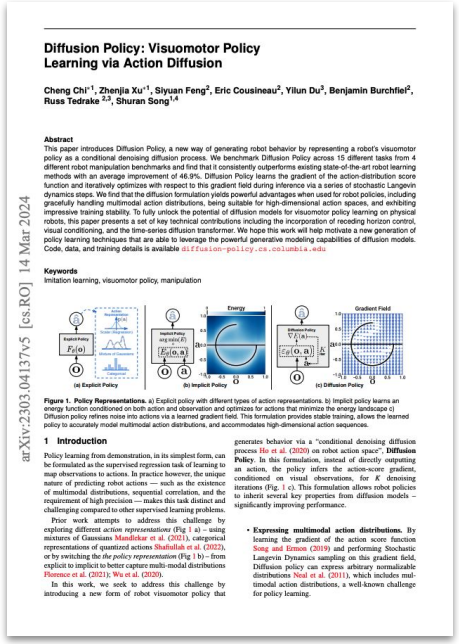
Our understanding of the world is highly compositional in nature. We are able to rapidly understand new objects from their components, or compose words into complex sentences to describe the world states we encounter [26]. We are able

* indicates equal contribution.
Correspondence to: Shuang Li <lishuang@mit.edu>, Yilun Du <yilundu@mit.edu>
Webpage: https://energy-based-model.github.io/Compositional-Visual-Generation-with-Composable-Diffusion-Models/

DDIM — classic method for efficient generation

Today

After: Next batch of **paper presentations**.



Diffusion Policy: Visuomotor Policy Learning via Action Diffusion

Cheng Chi[*1], Zhenjia Xu[*1], Siyuan Feng[2], Eric Cousineau[2], Yilun Du[3], Benjamin Burchfiel[2], Russ Tedrake[2,3], Shuran Song[1,4]

Abstract

This paper introduces Diffusion Policy, a new way of generating robot behavior by representing a robot's visuomotor policy as a conditional denoising diffusion process. We benchmark Diffusion Policy across 15 different tasks from 4 different robot manipulation benchmarks and find that it consistently outperforms existing state-of-the-art robot learning methods with an average improvement of 46.9%. Diffusion Policy learns the gradient of the action-distribution score function and iteratively optimizes with respect to this gradient field during inference via a series of stochastic Langevin dynamics steps. We find that the diffusion formulation yields powerful advantages when used for robot policies, including gracefully handling multimodal action distributions, being suitable for high-dimensional action spaces, and exhibiting impressive training stability. To fully unlock the potential of diffusion models for visuomotor policy learning on physical robots, this paper presents a set of key technical contributions including the incorporation of receding horizon control, visual conditioning, and the time-series diffusion transformer. We hope this work will help motivate a new generation of policy learning techniques that are able to leverage the powerful generative modeling capabilities of diffusion models. Code, data, and training details is available diffusion-policy.cs.columbia.edu

Keywords

Imitation learning, visuomotor policy, manipulation

Figure 1. Policy Representations. a) Explicit policy with different types of action representations. b) Implicit policy learns an energy function conditioned on both action and observation and optimizes for actions that minimize the energy landscape c) Diffusion policy refines noise into actions via a learned gradient field. This formulation provides stable training, allows the learned policy to accurately model multimodal action distributions, and accommodates high-dimensional action sequences.

1 Introduction

Policy learning from demonstration, in its simplest form, can be formulated as the supervised regression task of learning to map observations to actions. In practice however, the unique nature of predicting robot actions — such as the existence of multimodal distributions, sequential correlation, and the requirement of high precision — makes this task distinct and challenging compared to other supervised learning problems.

Prior work attempts to address this challenge by exploring different *action representations* (Fig 1 a) – using mixtures of Gaussians Mandlekar et al. (2021), categorical representations of quantized actions Shafiullah et al. (2022), or by switching the *the policy representation* (Fig 1 b) – from explicit to implicit to better capture multi-modal distributions Florence et al. (2021); Wu et al. (2020).

In this work, we seek to address this challenge by introducing a new form of robot visuomotor policy that generates behavior via a "conditional denoising diffusion process Ho et al. (2020) on robot action space", **Diffusion Policy**. In this formulation, instead of directly outputting an action, the policy infers the action-score gradient, conditioned on visual observations, for K denoising iterations (Fig. 1 c). This formulation allows robot policies to inherit several key properties from diffusion models – significantly improving performance.

- **Expressing multimodal action distributions.** By learning the gradient of the action score function Song and Ermon (2019) and performing Stochastic Langevin Dynamics sampling on this gradient field, Diffusion policy can express arbitrary normalizable distributions Neal et al. (2011), which includes multimodal action distributions, a well-known challenge for policy learning.

LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia[1*] Sadhika Malladi[1*] Suchin Gururangan[2] Sanjeev Arora[1] Danqi Chen[1]

Abstract

Instruction tuning has unlocked powerful capabilities in large language models (LLMs), using combined datasets to develop general-purpose chatbots. However, real-world applications often require a specialized suite of skills (e.g., reasoning). The challenge lies in identifying the most relevant data from these extensive datasets to effectively develop specific capabilities, a setting we frame as *targeted instruction tuning*. We propose LESS, an optimizer-aware and practically efficient algorithm to estimate data influences and perform Low-rank gradIEnt Similarity Search for instruction data selection. Crucially, LESS adapts existing influence formulations to work with the Adam optimizer and variable-length instruction data. LESS first constructs a highly reusable and transferable *gradient datastore* with low-dimensional gradient features and then selects examples based on their similarity to few-shot examples embodying a specific capability. Experiments show that training on a LESS-selected 5% of the data can often outperform training on the full dataset across diverse downstream tasks. Furthermore, the selected data is highly transferable: smaller models can be leveraged to select useful data for larger models and models from different families. Our qualitative analysis shows that our method goes beyond surface form cues to identify data that exemplifies the necessary reasoning skills for the intended downstream application. To facilitate future work, we release code and data at princeton-nlp/LESS.

1. Introduction

Instruction tuning has made large language models (LLMs) adept at following human instructions (Ouyang et al., 2022) as versatile chatbots (OpenAI, 2022; 2023; Anthropic, 2023; Google, 2023). Recent efforts curating highly diverse and wide-ranging instruction tuning datasets (Taori et al., 2023; Wang et al.; Mukherjee et al., 2023; Xu et al., 2023, *inter alia*) induce remarkably strong generalization even from a small number of examples (Zhou et al., 2023). Regardless, it remains an open problem to understand how to best utilize these various datasets.

Many real-world applications call for cultivating a specific suite of capabilities in LLMs (e.g., reasoning skills). However, training LLMs with mixed instruction tuning datasets can hinder the development of these specific capabilities. For example, Wang et al. (2023b) demonstrates that LLMs trained on a mix of instruction tuning datasets exhibit worse performance than those trained on a subset of the data. Additionally, considering the broad spectrum of user queries and the multitude of skills required to respond to them, there may not always be enough in-domain data available. Therefore, we hope to be able to effectively use the general instruction tuning data to improve specific capabilities. We frame this setting as *targeted instruction tuning*:

Given just a handful of examples embodying a specific capability, how can we effectively select relevant fine-tuning data from a large collection of instruction datasets?

We approach this problem by prioritizing training on data that directly minimizes loss on a target task instead of relying on surface form features (Gururangan et al., 2020; Xie et al., 2023b). Inspired by past works estimating the influence of individual training datapoints with gradient information (Pruthi et al., 2020; Han et al., 2023), we design an optimizer-aware approach to select such data. However, straightforward application of this influence formulation faces several challenges unique to the instruction tuning setting: (1) LLMs are traditionally fine-tuned with the Adam optimizer (Kingma & Ba, 2015) instead of the canonical SGD optimizer; (2) using sequence-level gradients of variable-length instruction data can derail the influence estimation; and (3) the large number of trainable parameters in LLMs makes the computation and storage of gradient information extremely resource-intensive.

[*]Equal contribution [1]Princeton Language and Intelligence (PLI), Princeton University, USA. [2]Department of Computer Science, University of Washington, USA. Correspondence to: Mengzhou Xia, Sadhika Malladi <{mengzhou, smalladi}@princeton.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Published as a conference paper at ICLR 2021

Denoising Diffusion Implicit Models

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

Abstract

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation without adversarial training, yet they require simulating a Markov chain for many steps in order to produce a sample. To accelerate sampling, we present denoising diffusion implicit models (DDIMs), a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPMs. In DDPMs, the generative process is defined as the reverse of a particular Markovian diffusion process. We generalize DDPMs via a class of non-Markovian diffusion processes that lead to the same training objective. These non-Markovian processes can correspond to generative processes that are deterministic, giving rise to implicit models that produce high quality samples much faster. We empirically demonstrate that DDIMs can produce high quality samples $10\times$ to $50\times$ faster in terms of wall-clock time compared to DDPMs, allow us to trade off computation for sample quality, perform semantically meaningful image interpolation directly in the latent space, and reconstruct observations with very low error.

1 Introduction

Deep generative models have demonstrated the ability to produce high quality samples in many domains (Karras et al., 2020; van den Oord et al., 2016a). In terms of image generation, generative adversarial networks (GANs, Goodfellow et al. (2014)) currently exhibits higher sample quality than likelihood-based methods such as variational autoencoders (Kingma & Welling, 2013), autoregressive models (van den Oord et al., 2016b) and normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2016). However, GANs require very specific choices in optimization and architectures in order to stabilize training (Arjovsky et al., 2017; Gulrajani et al., 2017; Karras et al., 2018; Brock et al., 2018), and could fail to cover modes of the data distribution (Zhao et al., 2018).

Recent works on iterative generative models (Bengio et al., 2014), such as denoising diffusion probabilistic models (DDPM, Ho et al. (2020)) and noise conditional score networks (NCSN, Song & Ermon (2019)) have demonstrated the ability to produce samples comparable to that of GANs, without having to perform adversarial training. To achieve this, many denoising autoencoding models are trained to denoise samples corrupted by various levels of Gaussian noise. Samples are then produced by a Markov chain which, starting from white noise, progressively denoises it into an image. This generative Markov Chain process is either based on Langevin dynamics (Song & Ermon, 2019) or obtained by reversing a forward *diffusion process* that progressively turns an image into noise (Sohl-Dickstein et al., 2015).

A critical drawback of these models is that they require many iterations to produce a high quality sample. For DDPMs, this is because that the generative process (from noise to data) approximates the reverse of the forward *diffusion process* (from data to noise), which could have thousands of steps; iterating over all the steps is required to produce a single sample, which is much slower compared to GANs, which only needs one pass through a network. For example, it takes around 20 hours to sample 50k images of size 32×32 from a DDPM, but less than a minute to do so from a GAN on a Nvidia 2080 Ti GPU. This becomes more problematic for larger images as sampling 50k images of size 256×256 could take nearly 1000 hours on the same GPU.

To close this efficiency gap between DDPMs and GANs, we present denoising diffusion implicit models (DDIMs). DDIMs are implicit probabilistic models (Mohamed & Lakshminarayanan, 2016) and are closely related to DDPMs, in the sense that they are trained with the same objective function.

Published as a conference paper at ICLR 2023

DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking

Gabriele Corso*, Hannes Stärk*, Bowen Jing*, Regina Barzilay & Tommi Jaakkola
CSAIL, Massachusetts Institute of Technology

Abstract

Predicting the binding structure of a small molecule ligand to a protein—a task known as *molecular docking*—is critical to drug design. Recent deep learning methods that treat docking as a regression problem have decreased runtime compared to traditional search-based methods but have yet to offer substantial improvements in accuracy. We instead frame molecular docking as a *generative* modeling problem and develop DiffDock, a diffusion generative model over the non-Euclidean manifold of ligand poses. To do so, we map this manifold to the product space of the degrees of freedom (translational, rotational, and torsional) involved in docking and develop an efficient diffusion process on this space. Empirically, DiffDock obtains a 38% top-1 success rate (RMSD<2Å) on PDBBind, significantly outperforming the previous state-of-the-art of traditional docking (23%) and deep learning (20%) methods. Moreover, while previous methods are not able to dock on computationally folded structures (maximum accuracy 10.4%), DiffDock maintains significantly higher precision (21.7%). Finally, DiffDock has fast inference times and provides confidence estimates with high selective accuracy.

1 Introduction

The biological functions of proteins can be modulated by small molecule ligands (such as drugs) binding to them. Thus, a crucial task in computational drug design is *molecular docking*—predicting the position, orientation, and conformation of a ligand when bound to a target protein—from which the effect of the ligand (if any) might be inferred. Traditional approaches for docking [Trott & Olson, 2010; Halgren et al., 2004] rely on scoring-functions that estimate the correctness of a proposed structure or pose, and an optimization algorithm that searches for the global maximum of the scoring function. However, since the search space is vast and the landscape of the scoring functions rugged, these methods tend to be too slow and inaccurate, especially for high-throughput workflows.

Recent works [Stärk et al., 2022; Lu et al., 2022] have developed deep learning models to predict the binding pose in one shot, treating docking as a regression problem. While these methods are much faster than traditional search-based methods, they have yet to demonstrate significant improvements in accuracy. We argue that this may be because the regression-based paradigm corresponds imperfectly with the objectives of molecular docking, which is reflected in the fact that standard accuracy metrics resemble the *likelihood* of the data under the predictive model rather than a regression loss. We thus frame molecular docking as a *generative modeling problem*—given a ligand and target protein structure, we learn a distribution over ligand poses.

To this end, we develop DiffDock, a diffusion generative model (DGM) over the space of ligand poses for molecular docking. We define a diffusion process over the degrees of freedom involved in docking: the position of the ligand relative to the protein (locating the binding pocket), its orientation in the pocket, and the torsion angles describing its conformation. DiffDock samples poses by running the learned (reverse) diffusion process, which iteratively transforms an uninformed, noisy prior distribution over ligand poses into the learned model distribution (Figure 1). Intuitively, this process can be viewed as the progressive refinement of random poses via updates of their translations, rotations, and torsion angles.

*Equal contribution. Correspondence to {gcorso, hstark, bjing}@mit.edu.

Compositional Visual Generation with Composable Diffusion Models

Nan Liu[1*], Shuang Li[2*], Yilun Du[2*]
Antonio Torralba[2], and Joshua B. Tenenbaum[2]

[1] University of Illinois Urbana-Champaign
[2] Massachusetts Institute of Technology
nanliu6@illinois.edu, {lishuang,yilundu,torralba,jbt}@mit.edu

Abstract. Large text-guided diffusion models, such as DALL-E 2, are able to generate stunning photorealistic images given natural language descriptions. While such models are highly flexible, they struggle to understand the composition of certain concepts, such as confusing the attributes of different objects or relations between objects. In this paper, we propose an alternative structured approach for compositional generation using diffusion models. An image is generated by composing a set of diffusion models, with each of them modeling a certain component of the image. To do this, we interpret diffusion models as energy-based models in which the data distributions defined by the energy functions may be explicitly combined. The proposed method can generate scenes at test time that are substantially more complex than those seen in training, composing sentence descriptions, object relations, human facial attributes, and even generalizing to new combinations that are rarely seen in the real world. We further illustrate how our approach may be used to compose pre-trained text-guided diffusion models and generate photorealistic images containing all the details described in the input descriptions, including the binding of certain object attributes that have been shown difficult for DALL-E 2. These results point to the effectiveness of the proposed method in promoting structured generalization for visual generation.

Keywords: Compositionality, Diffusion Models, Energy-based Models, Visual Generation

1 Introduction

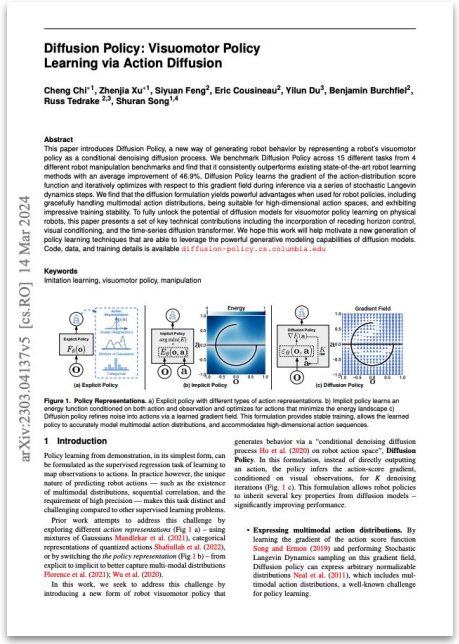
Our understanding of the world is highly compositional in nature. We are able to rapidly understand new objects from their components, or compose words into complex sentences to describe the world states we encounter [26]. We are able

* indicates equal contribution.
Correspondence to: Shuang Li <lishuang@mit.edu>, Yilun Du <yilundu@mit.edu>
Webpage: https://energy-based-model.github.io/Compositional-Visual-Generation-with-Composable-Diffusion-Models/

DiffDock – diffusion for molecular docking

Today

After: Next batch of **paper presentations**.



Diffusion Policy: Visuomotor Policy Learning via Action Diffusion

Cheng Chi[*1], Zhenjia Xu[*1], Siyuan Feng[2], Eric Cousineau[2], Yilun Du[3], Benjamin Burchfiel[2], Russ Tedrake[2,3], Shuran Song[1,4]

Abstract

This paper introduces Diffusion Policy, a new way of generating robot behavior by representing a robot's visuomotor policy as a conditional denoising diffusion process. We benchmark Diffusion Policy across 15 different tasks from 4 different robot manipulation benchmarks and find that it consistently outperforms existing state-of-the-art robot learning methods with an average improvement of 46.9%. Diffusion Policy learns the gradient of the action-distribution score function and iteratively optimizes with respect to this gradient field during inference via a series of stochastic Langevin dynamics steps. We find that the diffusion formulation yields powerful advantages when used for robot policies, including gracefully handling multimodal action distributions, being suitable for high-dimensional action spaces, and exhibiting impressive training stability. To fully unlock the potential of diffusion models for visuomotor policy learning on physical robots, this paper presents a set of key technical contributions including the incorporation of receding horizon control, visual conditioning, and the time-series diffusion transformer. We hope this work will help motivate a new generation of policy learning techniques that are able to leverage the powerful generative modeling capabilities of diffusion models. Code, data, and training details is available diffusion-policy.cs.columbia.edu

Keywords

Imitation learning, visuomotor policy, manipulation

Figure 1. Policy Representations. a) Explicit policy with different types of action representations. b) Implicit policy learns an energy function conditioned on both action and observation and optimizes for actions that minimize the energy landscape c) Diffusion policy refines noise into actions via a learned gradient field. This formulation provides stable training, allows the learned policy to accurately model multimodal action distributions, and accommodates high-dimensional action sequences.

1 Introduction

Policy learning from demonstration, in its simplest form, can be formulated as the supervised regression task of learning to map observations to actions. In practice however, the unique nature of predicting robot actions — such as the existence of multimodal distributions, sequential correlation, and the requirement of high precision — makes this task distinct and challenging compared to other supervised learning problems.

Prior work attempts to address this challenge by exploring different *action representations* (Fig 1 a) – using mixtures of Gaussians Mandlekar et al. (2021), categorical representations of quantized actions Shafiullah et al. (2022), or by switching the *the policy representation* (Fig 1 b) – from explicit to implicit to better capture multi-modal distributions Florence et al. (2021); Wu et al. (2020).

In this work, we seek to address this challenge by introducing a new form of robot visuomotor policy that generates behavior via a "conditional denoising diffusion process Ho et al. (2020) on robot action space", **Diffusion Policy**. In this formulation, instead of directly outputting an action, the policy infers the action-score gradient, conditioned on visual observations, for K denoising iterations (Fig. 1 c). This formulation allows robot policies to inherit several key properties from diffusion models – significantly improving performance.

- **Expressing multimodal action distributions.** By learning the gradient of the action score function Song and Ermon (2019) and performing Stochastic Langevin Dynamics sampling on this gradient field, Diffusion policy can express arbitrary normalizable distributions Neal et al. (2011), which includes multimodal action distributions, a well-known challenge for policy learning.

LESS: Selecting Influential Data for Targeted Instruction Tuning

Mengzhou Xia[1*] Sadhika Malladi[1*] Suchin Gururangan[2] Sanjeev Arora[1] Danqi Chen[1]

Abstract

Instruction tuning has unlocked powerful capabilities in large language models (LLMs), using combined datasets to develop general-purpose chatbots. However, real-world applications often require a specialized suite of skills (e.g., reasoning). The challenge lies in identifying the most relevant data from these extensive datasets to effectively develop specific capabilities, a setting we frame as *targeted instruction tuning*. We propose LESS, an optimizer-aware and practically efficient algorithm to estimate data influences and perform Low-rank gradIEnt Similarity Search for instruction data selection. Crucially, LESS adapts existing influence formulations to work with the Adam optimizer and variable-length instruction data. LESS first constructs a highly reusable and transferable *gradient datastore* with low-dimensional gradient features and then selects examples based on their similarity to few-shot examples embodying a specific capability. Experiments show that training on a LESS-selected 5% of the data can often outperform training on the full dataset across diverse downstream tasks. Furthermore, the selected data is highly transferable: smaller models can be leveraged to select useful data for larger models and models from different families. Our qualitative analysis shows that our method goes beyond surface form cues to identify data that exemplifies the necessary reasoning skills for the intended downstream application. To facilitate future work, we release code and data at princeton-nlp/LESS.

1. Introduction

Instruction tuning has made large language models (LLMs) adept at following human instructions (Ouyang et al., 2022) as versatile chatbots (OpenAI, 2022; 2023; Anthropic, 2023; Google, 2023). Recent efforts curating highly diverse and wide-ranging instruction tuning datasets (Taori et al., 2023; Wang et al.; Mukherjee et al., 2023; Xu et al., 2023, *inter alia*) induce remarkably strong generalization even from a small number of examples (Zhou et al., 2023). Regardless, it remains an open problem to understand how to best utilize these various datasets.

Many real-world applications call for cultivating a specific suite of capabilities in LLMs (e.g., reasoning skills). However, training LLMs with mixed instruction tuning datasets can hinder the development of these specific capabilities. For example, Wang et al. (2023b) demonstrates that LLMs trained on a mix of instruction tuning datasets exhibit worse performance than those trained on a subset of the data. Additionally, considering the broad spectrum of user queries and the multitude of skills required to respond to them, there may not always be enough in-domain data available. Therefore, we hope to be able to effectively use the general instruction tuning data to improve specific capabilities. We frame this setting as *targeted instruction tuning*:

Given just a handful of examples embodying a specific capability, how can we effectively select relevant fine-tuning data from a large collection of instruction datasets?

We approach this problem by prioritizing training on data that directly minimizes loss on a target task instead of relying on surface form features (Gururangan et al., 2020; Xie et al., 2023b). Inspired by past works estimating the influence of individual training datapoints with gradient information (Pruthi et al., 2020; Han et al., 2023), we design an optimizer-aware approach to select such data. However, straightforward application of this influence formulation faces several challenges unique to the instruction tuning setting: (1) LLMs are traditionally fine-tuned with the Adam optimizer (Kingma & Ba, 2015) instead of the canonical SGD optimizer; (2) using sequence-level gradients of variable-length instruction data can derail the influence estimation; and (3) the large number of trainable parameters in LLMs makes the computation and storage of gradient information extremely resource-intensive.

[*]Equal contribution [1]Princeton Language and Intelligence (PLI), Princeton University, USA. [2]Department of Computer Science, University of Washington, USA. Correspondence to: Mengzhou Xia, Sadhika Malladi <mengzhou, smalladi@princeton.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Published as a conference paper at ICLR 2021

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation without adversarial training, yet they require simulating a Markov chain for many steps in order to produce a sample. To accelerate sampling, we present denoising diffusion implicit models (DDIMs), a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPMs. In DDPMs, the generative process is defined as the reverse of a particular Markovian diffusion process. We generalize DDPMs via a class of non-Markovian diffusion processes that lead to the same training objective. These non-Markovian processes can correspond to generative processes that are deterministic, giving rise to implicit models that produce high quality samples much faster. We empirically demonstrate that DDIMs can produce high quality samples $10\times$ to $50\times$ faster in terms of wall-clock time compared to DDPMs, allow us to trade off computation for sample quality, perform semantically meaningful image interpolation directly in the latent space, and reconstruct observations with very low error.

1 INTRODUCTION

Deep generative models have demonstrated the ability to produce high quality samples in many domains (Karras et al., 2020; van den Oord et al., 2016a). In terms of image generation, generative adversarial networks (GANs, Goodfellow et al. (2014)) currently exhibits higher sample quality than likelihood-based methods such as variational autoencoders (Kingma & Welling, 2013), autoregressive models (van den Oord et al., 2016b) and normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2016). However, GANs require very specific choices in optimization and architectures in order to stabilize training (Arjovsky et al., 2017; Gulrajani et al., 2017; Karras et al., 2018; Brock et al., 2018), and could fail to cover modes of the data distribution (Zhao et al., 2018).

Recent works on iterative generative models (Bengio et al., 2014), such as denoising diffusion probabilistic models (DDPM, Ho et al. (2020)) and noise conditional score networks (NCSN, Song & Ermon (2019)) have demonstrated the ability to produce samples comparable to that of GANs, without having to perform adversarial training. To achieve this, many denoising autoencoding models are trained to denoise samples corrupted by various levels of Gaussian noise. Samples are then produced by a Markov chain which, starting from white noise, progressively denoises it into an image. This generative Markov Chain process is either based on Langevin dynamics (Song & Ermon, 2019) or obtained by reversing a forward *diffusion process* that progressively turns an image into noise (Sohl-Dickstein et al., 2015).

A critical drawback of these models is that they require many iterations to produce a high quality sample. For DDPMs, this is because that the generative process (from noise to data) approximates the reverse of the forward *diffusion process* (from data to noise), which could have thousands of steps; iterating over all the steps is required to produce a single sample, which is much slower compared to GANs, which only needs one pass through a network. For example, it takes around 20 hours to sample 50k images of size 32×32 from a DDPM, but less than a minute to do so from a GAN on a Nvidia 2080 Ti GPU. This becomes more problematic for larger images as sampling 50k images of size 256×256 could take nearly 1000 hours on the same GPU.

To close this efficiency gap between DDPMs and GANs, we present denoising diffusion implicit models (DDIMs). DDIMs are implicit probabilistic models (Mohamed & Lakshminarayanan, 2016) and are closely related to DDPMs, in the sense that they are trained with the same objective function.

Published as a conference paper at ICLR 2023

DIFFDOCK: DIFFUSION STEPS, TWISTS, AND TURNS FOR MOLECULAR DOCKING

Gabriele Corso*, Hannes Stärk*, Bowen Jing*, Regina Barzilay & Tommi Jaakkola
CSAIL, Massachusetts Institute of Technology

ABSTRACT

Predicting the binding structure of a small molecule ligand to a protein—a task known as *molecular docking*—is critical to drug design. Recent deep learning methods that treat docking as a regression problem have decreased runtime compared to traditional search-based methods but have yet to offer substantial improvements in accuracy. We instead frame molecular docking as a *generative* modeling problem and develop DIFFDOCK, a diffusion generative model over the non-Euclidean manifold of ligand poses. To do so, we map this manifold to the product space of the degrees of freedom (translational, rotational, and torsional) involved in docking and develop an efficient diffusion process on this space. Empirically, DIFFDOCK obtains a 38% top-1 success rate (RMSD<2Å) on PDBBind, significantly outperforming the previous state-of-the-art of traditional docking (23%) and deep learning (20%) methods. Moreover, while previous methods are not able to dock on computationally folded structures (maximum accuracy 10.4%), DIFFDOCK maintains significantly higher precision (21.7%). Finally, DIFFDOCK has fast inference times and provides confidence estimates with high selective accuracy.

1 INTRODUCTION

The biological functions of proteins can be modulated by small molecule ligands (such as drugs) binding to them. Thus, a crucial task in computational drug design is *molecular docking*—predicting the position, orientation, and conformation of a ligand when bound to a target protein—from which the effect of the ligand (if any) might be inferred. Traditional approaches for docking [Trott & Olson, 2010; Halgren et al., 2004] rely on scoring-functions that estimate the correctness of a proposed structure or pose, and an optimization algorithm that searches for the global maximum of the scoring function. However, since the search space is vast and the landscape of the scoring functions rugged, these methods tend to be too slow and inaccurate, especially for high-throughput workflows.

Recent works [Stärk et al., 2022; Lu et al., 2022] have developed deep learning models to predict the binding pose in one shot, treating docking as a regression problem. While these methods are much faster than traditional search-based methods, they have yet to demonstrate significant improvements in accuracy. We argue that this may be because the regression-based paradigm corresponds imperfectly with the objectives of molecular docking, which is reflected in the fact that standard accuracy metrics resemble the *likelihood* of the data under the predictive model rather than a regression loss. We thus frame molecular docking as a *generative modeling problem*—given a ligand and target protein structure, we learn a distribution over ligand poses.

To this end, we develop DIFFDOCK, a diffusion generative model (DGM) over the space of ligand poses for molecular docking. We define a diffusion process over the degrees of freedom involved in docking: the position of the ligand relative to the protein (locating the binding pocket), its orientation in the pocket, and the torsion angles describing its conformation. DIFFDOCK samples poses by running the learned (reverse) diffusion process, which iteratively transforms an uninformed, noisy prior distribution over ligand poses into the learned model distribution (Figure 1). Intuitively, this process can be viewed as the progressive refinement of random poses via updates of their translations, rotations, and torsion angles.

*Equal contribution. Correspondence to {gcorso, hstark, bjing}@mit.edu.

Compositional Visual Generation with Composable Diffusion Models

Nan Liu[1*], Shuang Li[2*], Yilun Du[2*]
Antonio Torralba[2], and Joshua B. Tenenbaum[2]

[1] University of Illinois Urbana-Champaign
[2] Massachusetts Institute of Technology
nanliu6@illinois.edu, {lishuang,yilundu,torralba,jbt}@mit.edu

Abstract. Large text-guided diffusion models, such as DALL-E 2, are able to generate stunning photorealistic images given natural language descriptions. While such models are highly flexible, they struggle to understand the composition of certain concepts, such as confusing the attributes of different objects or relations between objects. In this paper, we propose an alternative structured approach for compositional generation using diffusion models. An image is generated by composing a set of diffusion models, with each of them modeling a certain component of the image. To do this, we interpret diffusion models as energy-based models in which the data distributions defined by the energy functions may be explicitly combined. The proposed method can generate scenes at test time that are substantially more complex than those seen in training, composing sentence descriptions, object relations, human facial attributes, and even generalizing to new combinations that are rarely seen in the real world. We further illustrate how our approach may be used to compose pre-trained text-guided diffusion models and generate photorealistic images containing all the details described in the input descriptions, including the binding of certain object attributes that have been shown difficult for DALL-E 2. These results point to the effectiveness of the proposed method in promoting structured generalization for visual generation.

Keywords: Compositionality, Diffusion Models, Energy-based Models, Visual Generation

1 Introduction

Our understanding of the world is highly compositional in nature. We are able to rapidly understand new objects from their components, or compose words into complex sentences to describe the world states we encounter [26]. We are able

* indicates equal contribution.
Correspondence to: Shuang Li <lishuang@mit.edu>, Yilun Du <yilundu@mit.edu>
Webpage: https://energy-based-model.github.io/Compositional-Visual-Generation-with-Composable-Diffusion-Models/

Combining diffusion models for compositional generation

Course Projects

Course Projects

Final Presentations — 25 to 30 minute presentation! Cover the full project scope, including:

Course Projects

Final Presentations — 25 to 30 minute presentation! Cover the full project scope, including:

- **Introduction to the topic and motivation for your project:** the problem you are trying to solve, why it's useful in the world/downstream applications, why it's difficult to solve.

Course Projects

Final Presentations — 25 to 30 minute presentation! Cover the full project scope, including:

- **Introduction to the topic and motivation for your project:** the problem you are trying to solve, why it's useful in the world/downstream applications, why it's difficult to solve.
- **Prior work:** previous methods that are related to your work, high-level overview on any prior material needed to understand your work.

Course Projects

Final Presentations — 25 to 30 minute presentation! Cover the full project scope, including:

- **Introduction to the topic and motivation for your project:** the problem you are trying to solve, why it's useful in the world/downstream applications, why it's difficult to solve.
- **Prior work:** previous methods that are related to your work, high-level overview on any prior material needed to understand your work.
- **Technical contribution:** your main *contributions*, *methods*, or *technical work* in this project.

Course Projects

Final Presentations — 25 to 30 minute presentation! Cover the full project scope, including:

- **Introduction to the topic and motivation for your project:** the problem you are trying to solve, why it's useful in the world/downstream applications, why it's difficult to solve.
- **Prior work:** previous methods that are related to your work, high-level overview on any prior material needed to understand your work.
- **Technical contribution:** your main *contributions*, *methods*, or *technical work* in this project.
- **Experimental results:** the experiments you carried out and results you achieved, tying it back to your original project goals and motivation.

Course Projects

Final Presentations — 25 to 30 minute presentation! Cover the full project scope, including:

- **Introduction to the topic and motivation for your project:** the problem you are trying to solve, why it's useful in the world/downstream applications, why it's difficult to solve.
- **Prior work:** previous methods that are related to your work, high-level overview on any prior material needed to understand your work.
- **Technical contribution:** your main *contributions*, *methods*, or *technical work* in this project.
- **Experimental results:** the experiments you carried out and results you achieved, tying it back to your original project goals and motivation.
- **Discussion:** Thoughts/debrief on your project — how did the idea work out, and did you accomplish your goal? Any next steps for the project? Any new perspectives on promising directions in this subfield?

Course Projects

Final Presentations — 25 to 30 minute presentation! Cover the full project scope, including:

- First presentation day: **April 25**
  - Groups: 1, 3, 9 → Thank you for presenting first!

Course Projects

Final Presentations — 25 to 30 minute presentation! Cover the full project scope, including:

- First presentation day: **April 25**
  - Groups: 1, 3, 9 → Thank you for presenting first!
- First presentation day: **May 2**
  - Groups: 2, 4, 5, 6, 7, 8 → Thank you also :-).

We will stay on schedule to get through everyone!

Course Projects

Final Reports

Course Projects

Final Reports — 8-10 pages long. Similar scope as final presentation, including:

Course Projects

Final Reports — 8-10 pages long. Similar scope as final presentation, including:

- **Introduction**
- **Related Work**
- **Methods and Technical Contribution**
- **Experiments and Results**
- **Discussion**

Course Projects

Final Reports — 8-10 pages long. Similar scope as final presentation, including:

- **Introduction**
- **Related Work**
- **Methods and Technical Contribution**
- **Experiments and Results**
- **Discussion**

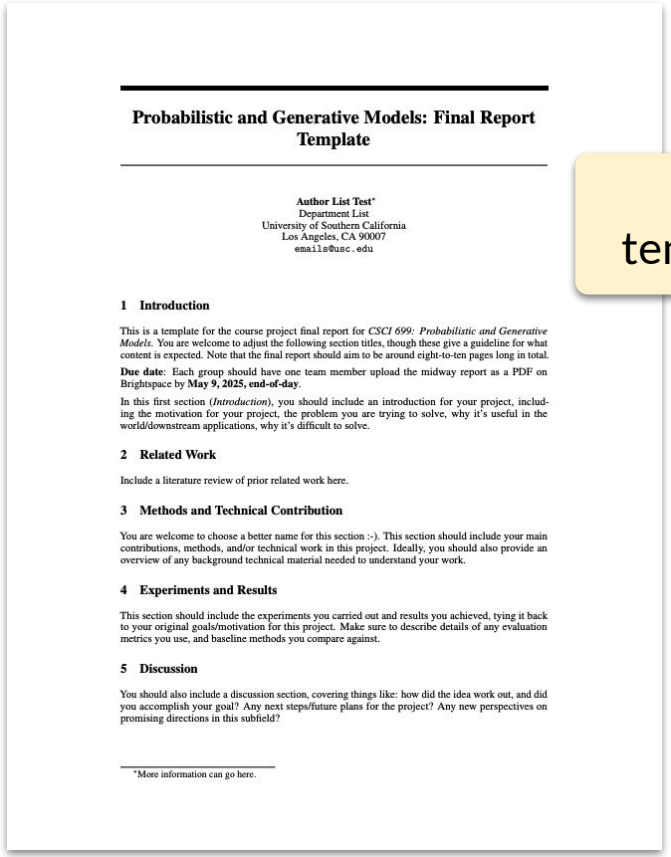
Due May 9th!

Course Projects

Final Reports — 8-10 pages long. Similar scope as final presentation, including:

- **Introduction**
- **Related Work**
- **Methods and Technical Contribution**
- **Experiments and Results**
- **Discussion**

Due May 9th!



Probabilistic and Generative Models: Final Report Template

Author List Test*
Department List
University of Southern California
Los Angeles, CA 90007
emails@usc.edu

1 Introduction

This is a template for the course project final report for *CSCI 699: Probabilistic and Generative Models*. You are welcome to adjust the following section titles, though these give a guideline for what content is expected. Note that the final report should aim to be around eight-to-ten pages long in total.

Due date: Each group should have one team member upload the midway report as a PDF on Brightspace by **May 9, 2025, end-of-day**.

In this first section (*Introduction*), you should include an introduction for your project, including the motivation for your project, the problem you are trying to solve, why it's useful in the world/downstream applications, why it's difficult to solve.

2 Related Work

Include a literature review of prior related work here.

3 Methods and Technical Contribution

You are welcome to choose a better name for this section :-). This section should include your main contributions, methods, and/or technical work in this project. Ideally, you should also provide an overview of any background technical material needed to understand your work.

4 Experiments and Results

This section should include the experiments you carried out and results you achieved, tying it back to your original goals/motivation for this project. Make sure to describe details of any evaluation metrics you use, and baseline methods you compare against.

5 Discussion

You should also include a discussion section, covering things like: how did the idea work out, and did you accomplish your goal? Any next steps/future plans for the project? Any new perspectives on promising directions in this subfield?

*More information can go here.

Have LaTeX template to share

Active Learning and Bayesian Optimization

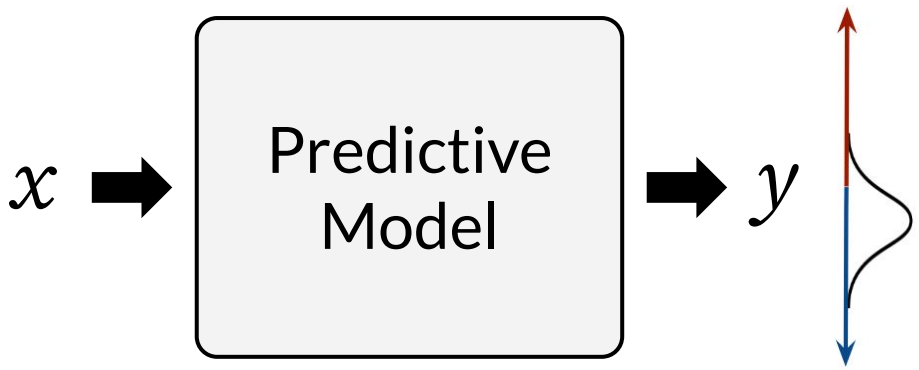
Bayesian Optimization — Review

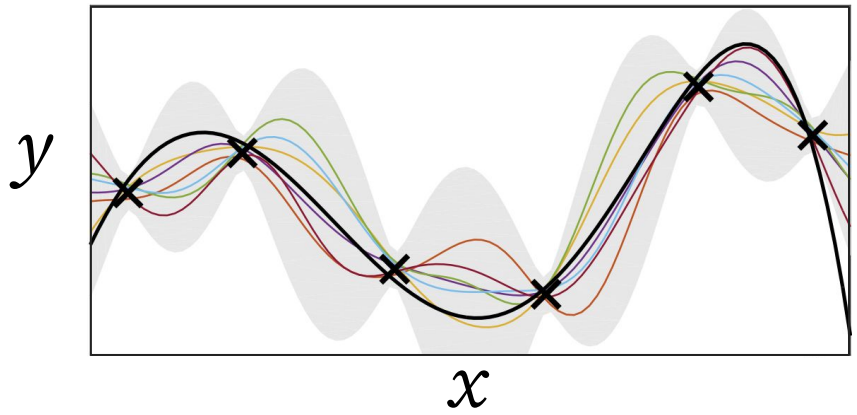
Bayesian Optimization — Review

What is Bayesian Opt? Uses a (probabilistic) predictive model to aid in optimization.

Bayesian Optimization — Review

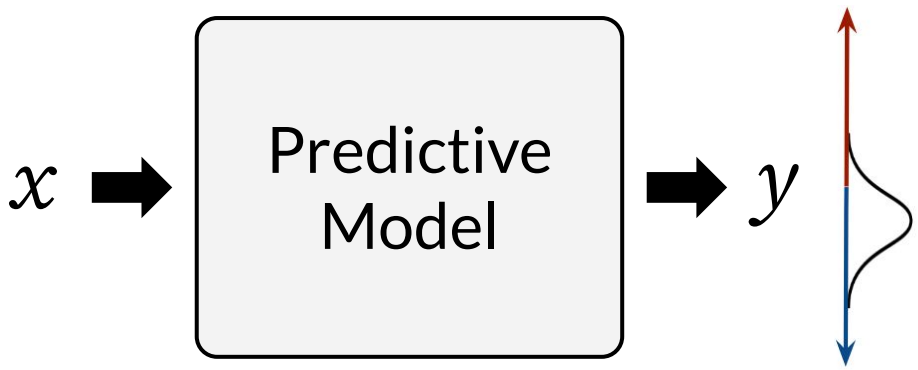
What is Bayesian Opt? Uses a (probabilistic) predictive model to aid in optimization.

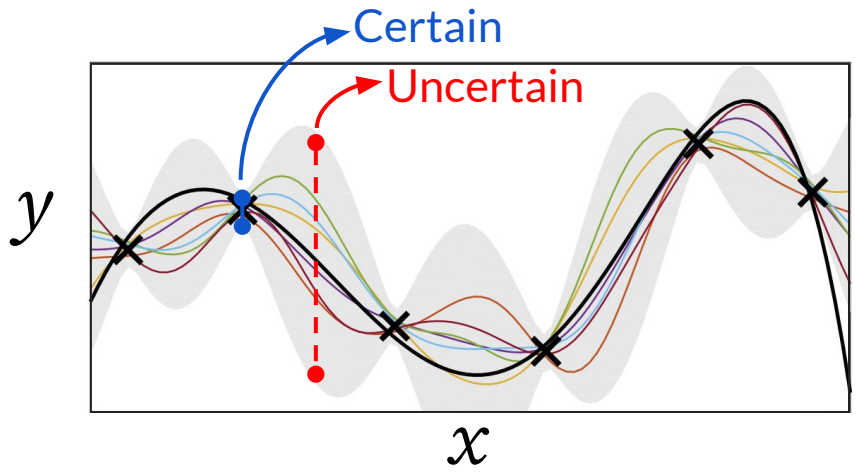




Bayesian Optimization — Review

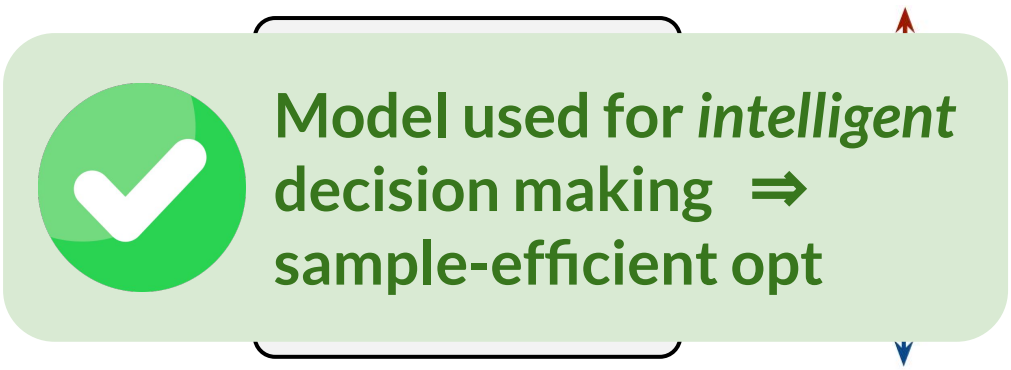
What is Bayesian Opt? Uses a (probabilistic) predictive model to aid in optimization.

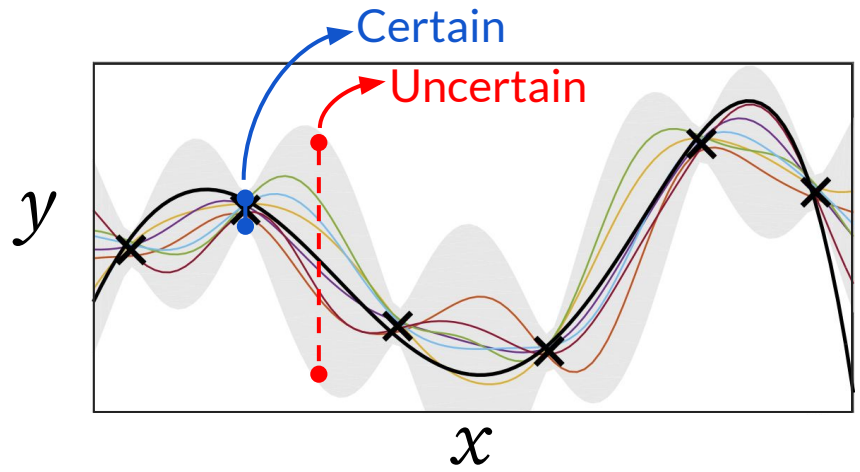




Bayesian Optimization — Review

What is Bayesian Opt? Uses a (probabilistic) predictive model to aid in optimization.





Bayesian Optimization — Algorithm Overview

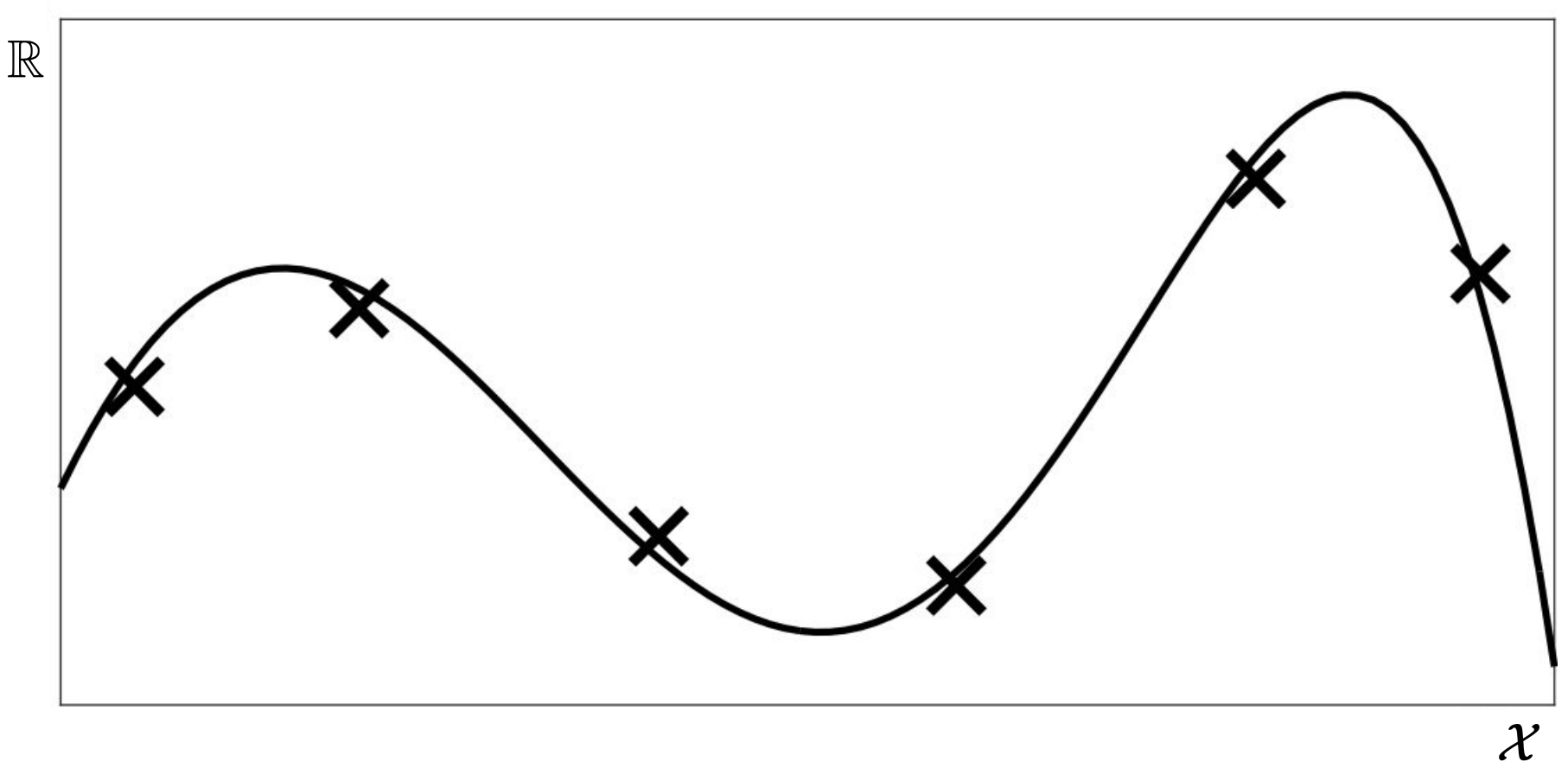
At each iteration of BO, three steps:

Bayesian Optimization — Algorithm Overview

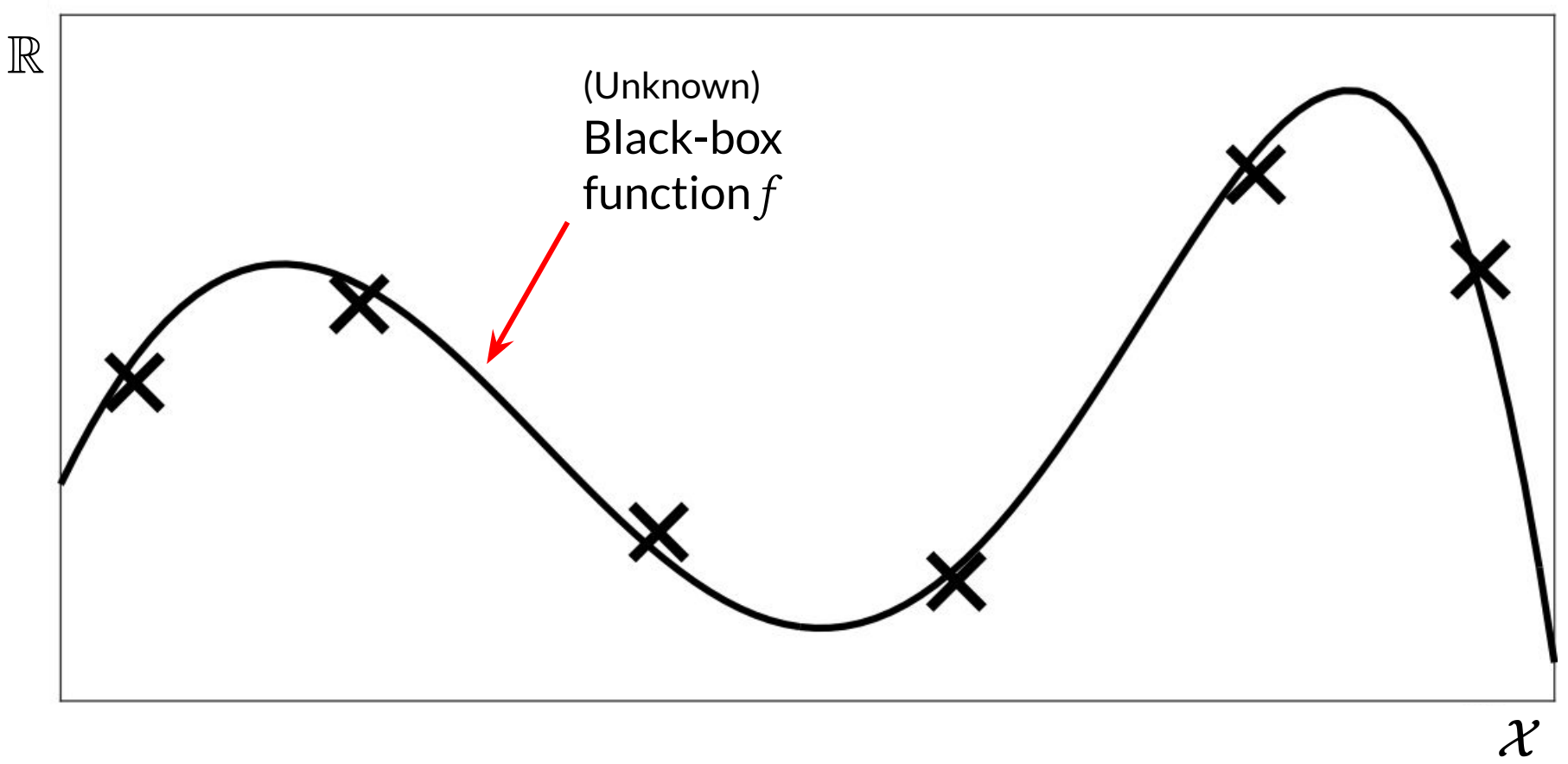
At each iteration of BO, three steps:

1. Fit model to dataset.
2. Optimize an *acquisition function* to select next point to query.
3. Query black-box function.

Bayesian Optimization — Visualization

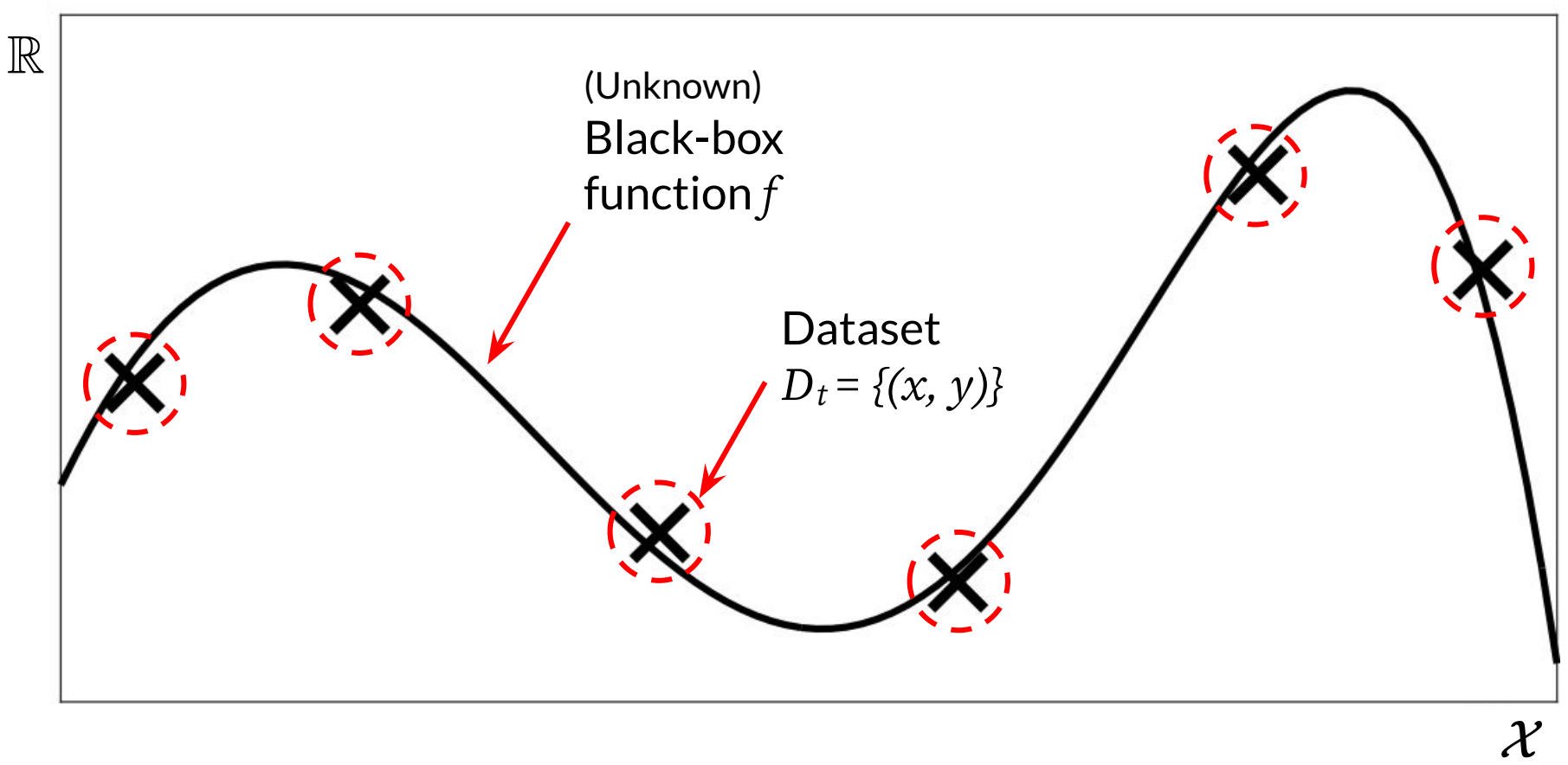


Bayesian Optimization — Visualization



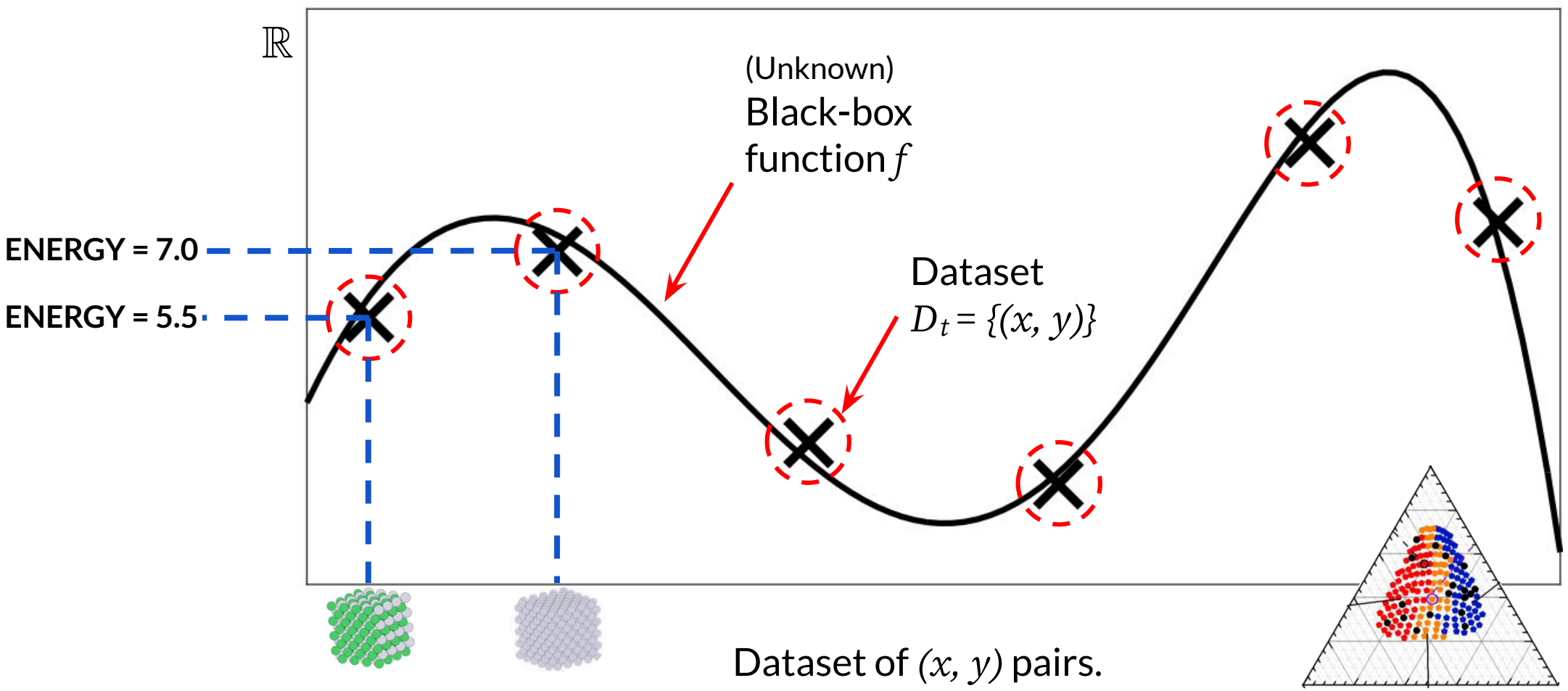
Unknown black-box f

Bayesian Optimization — Visualization

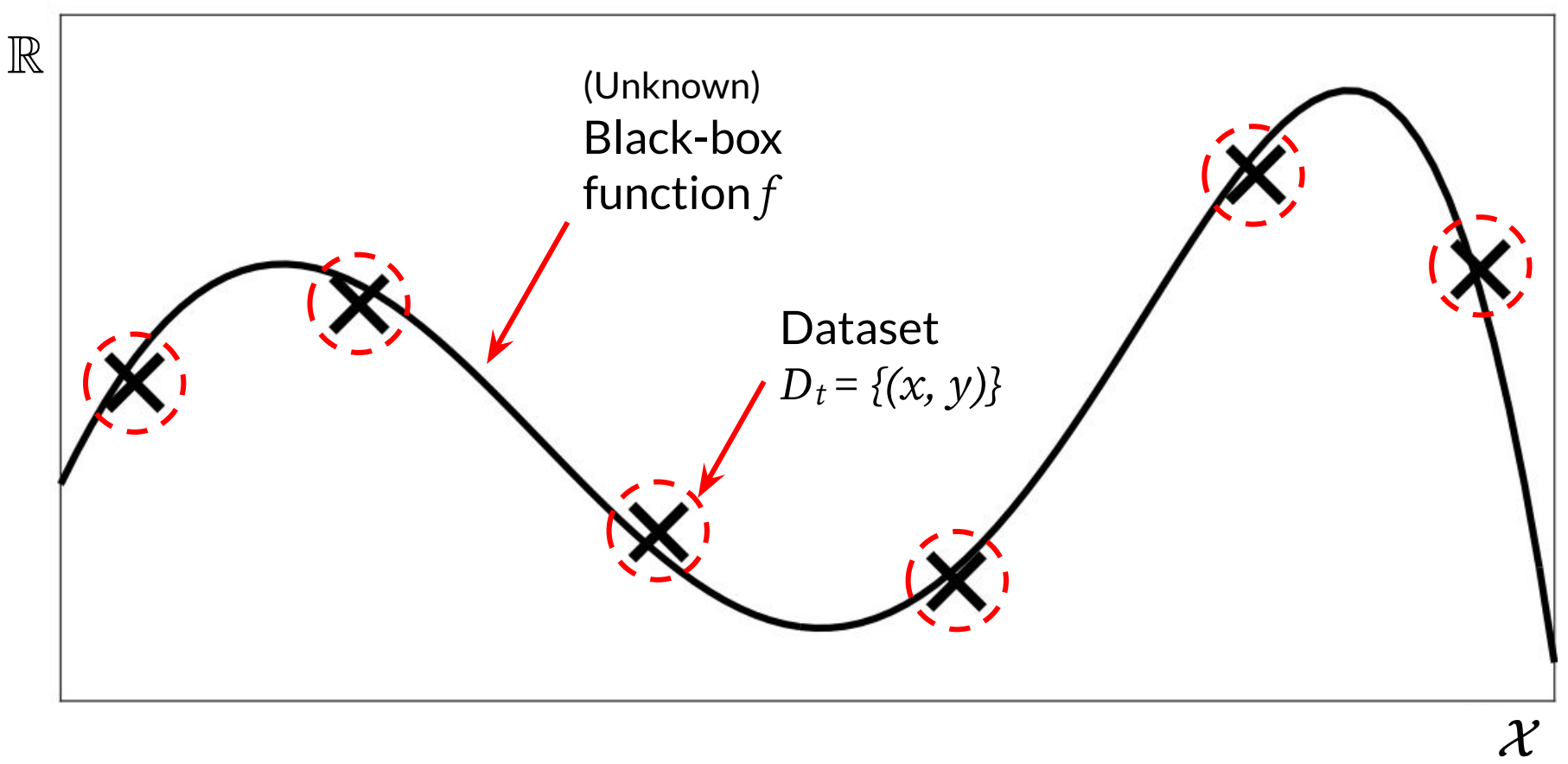


Unknown black-box f, and dataset of (x, y) pairs.

Bayesian Optimization — Visualization

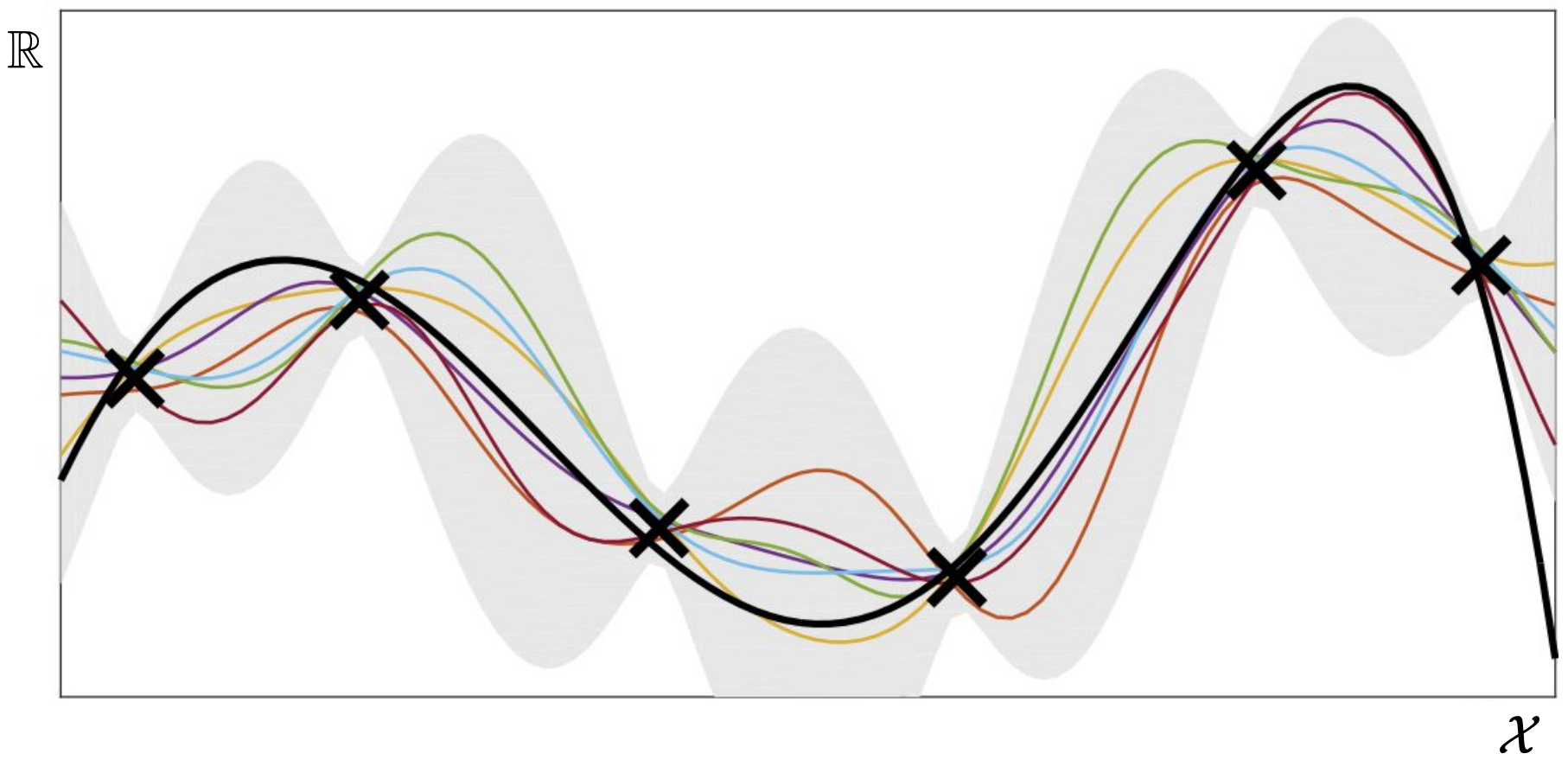


Bayesian Optimization — Visualization



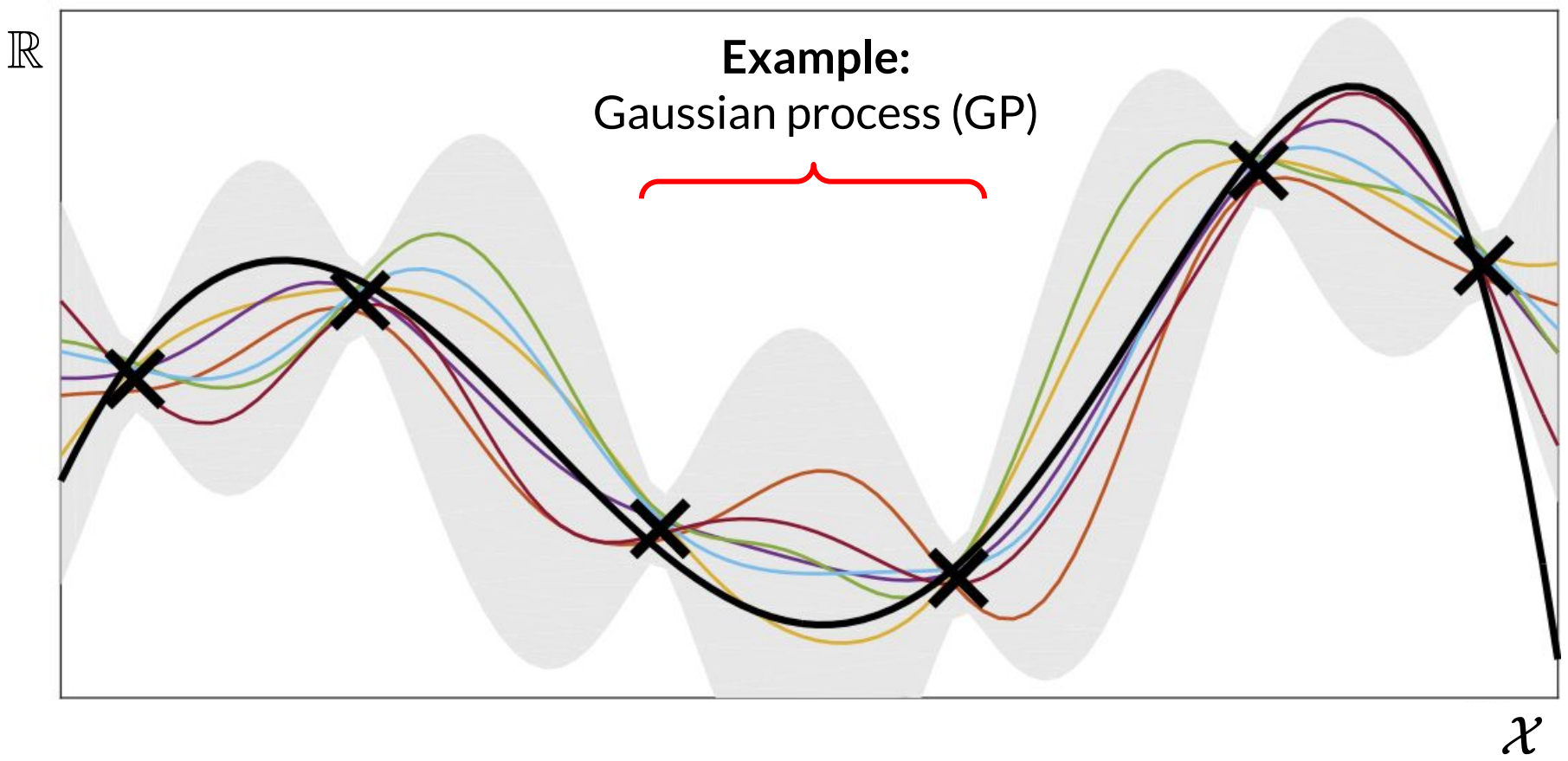
Dataset of (x, y) pairs.

Bayesian Optimization — Visualization



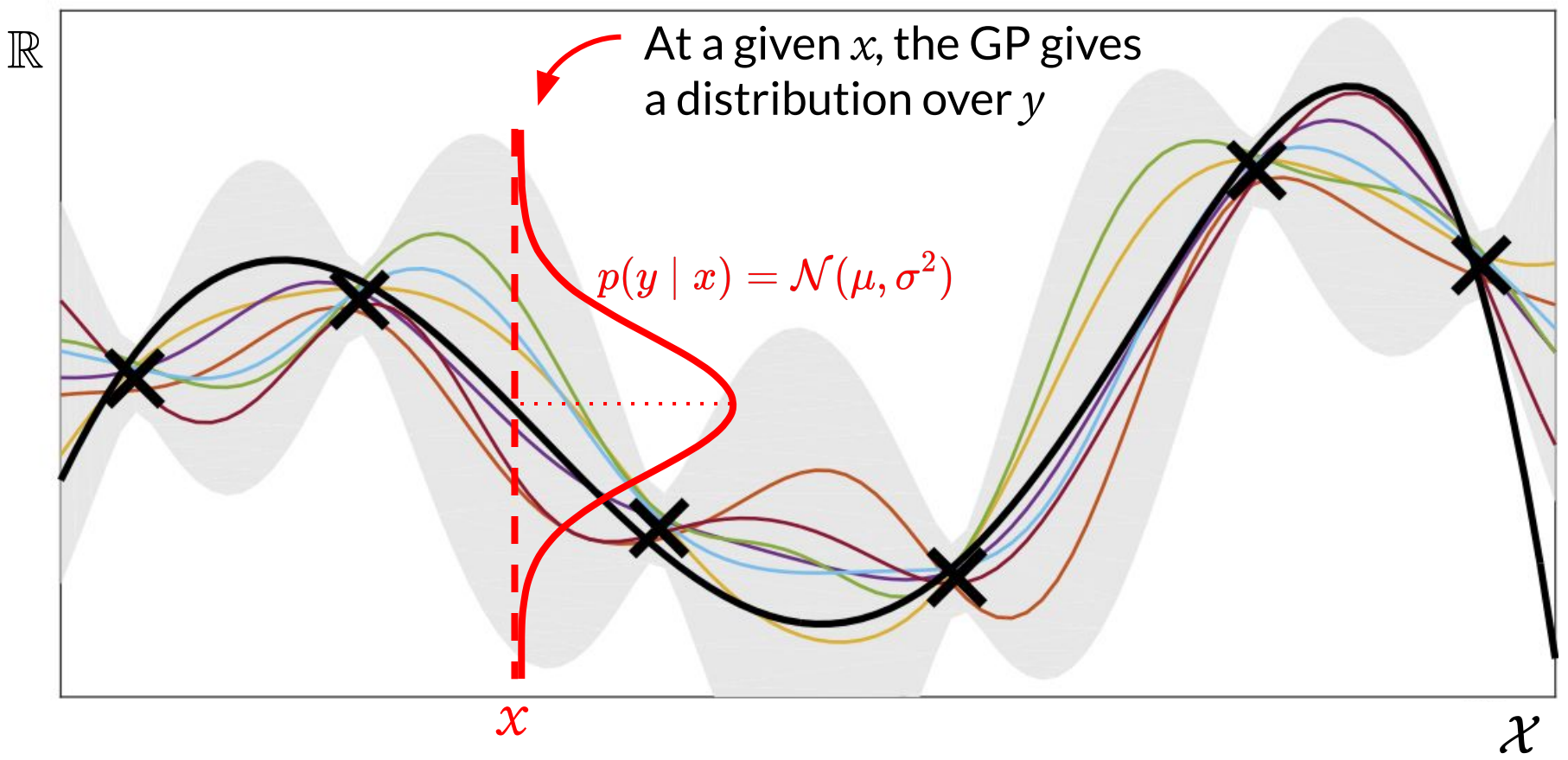
Given this dataset, can fit probabilistic model.

Bayesian Optimization — Visualization



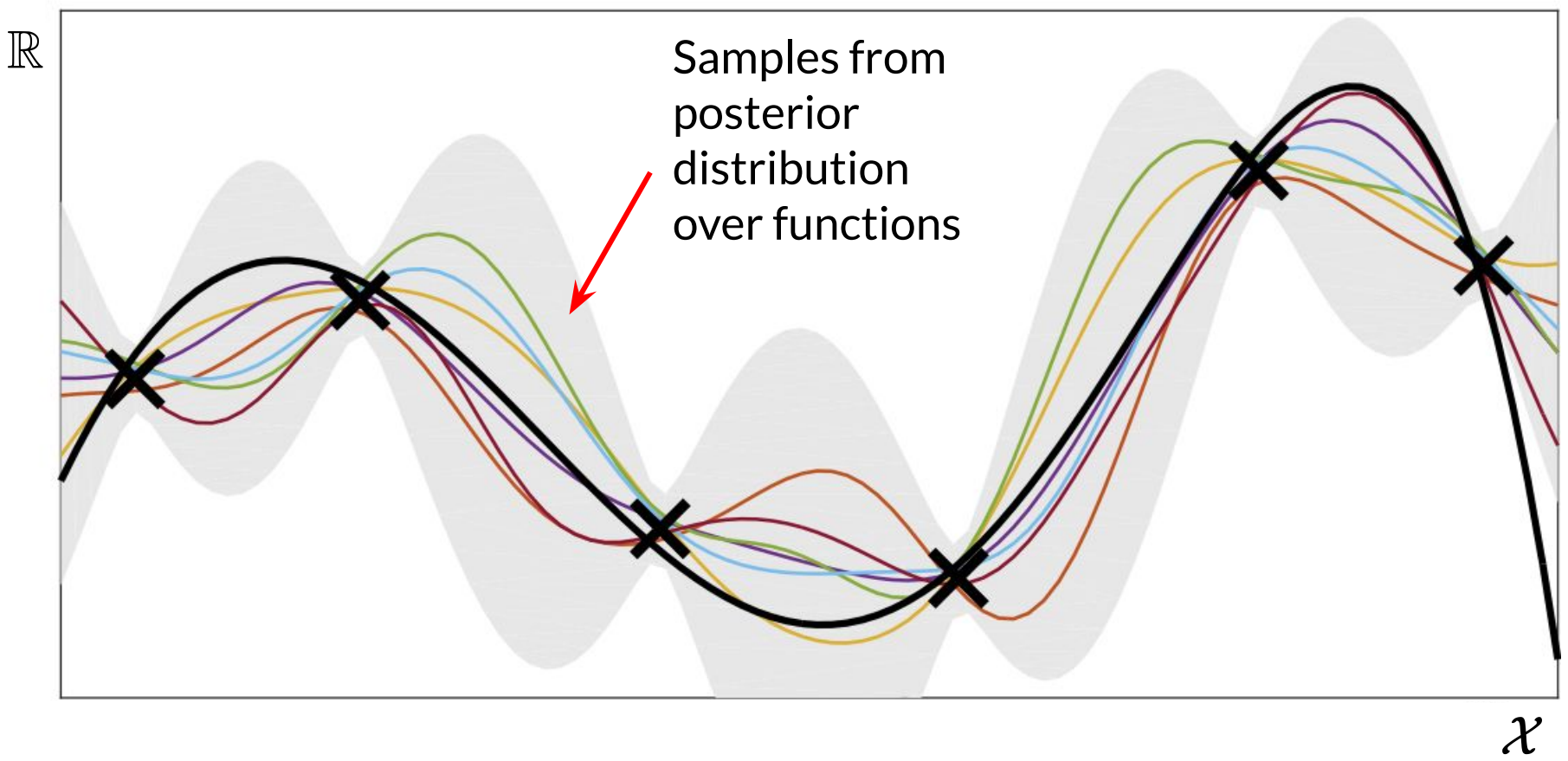
A popular choice for this model is a **Gaussian Process**.

Bayesian Optimization — Visualization



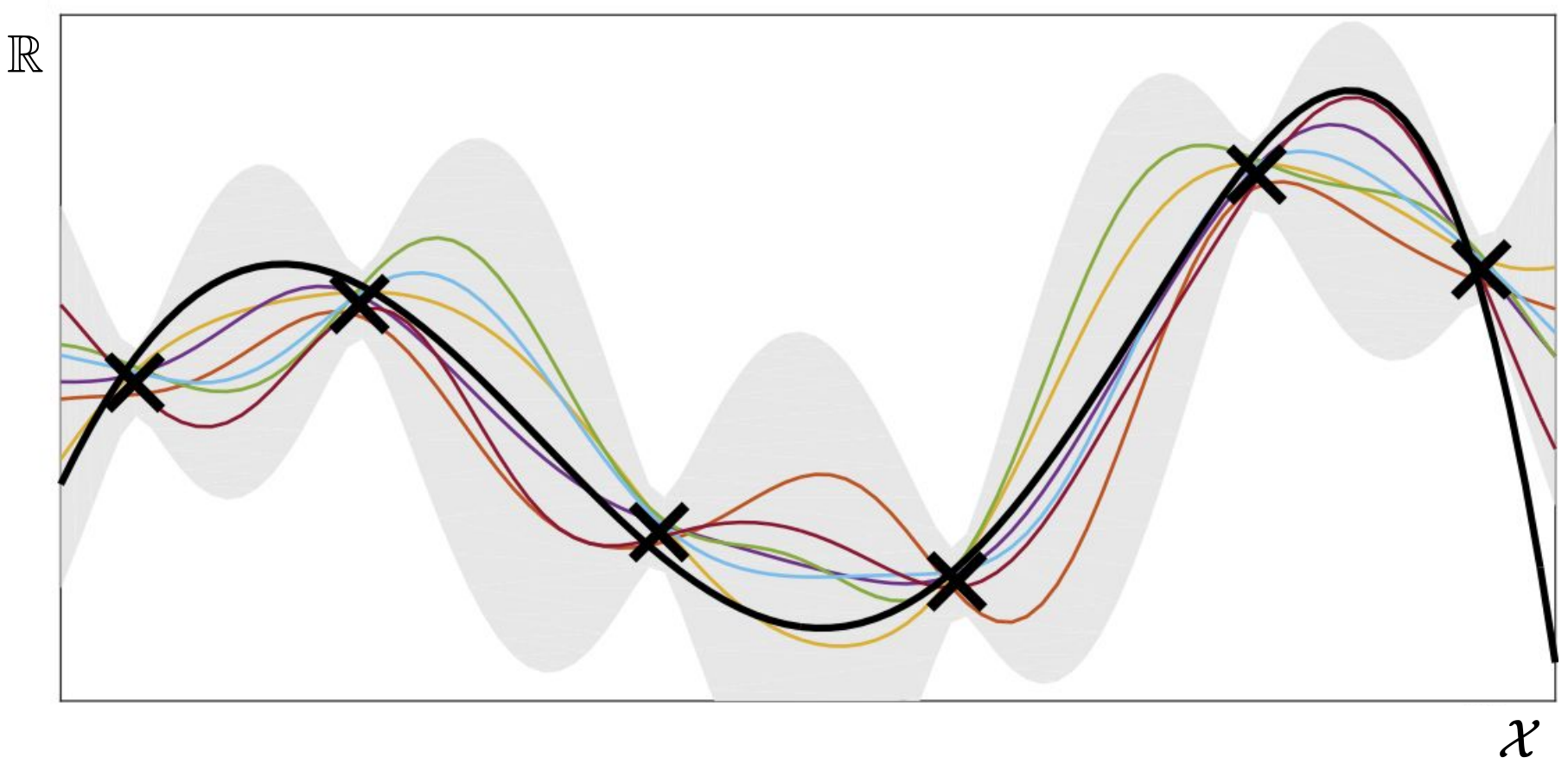
A popular choice for this model is a **Gaussian Process**.

Bayesian Optimization — Visualization



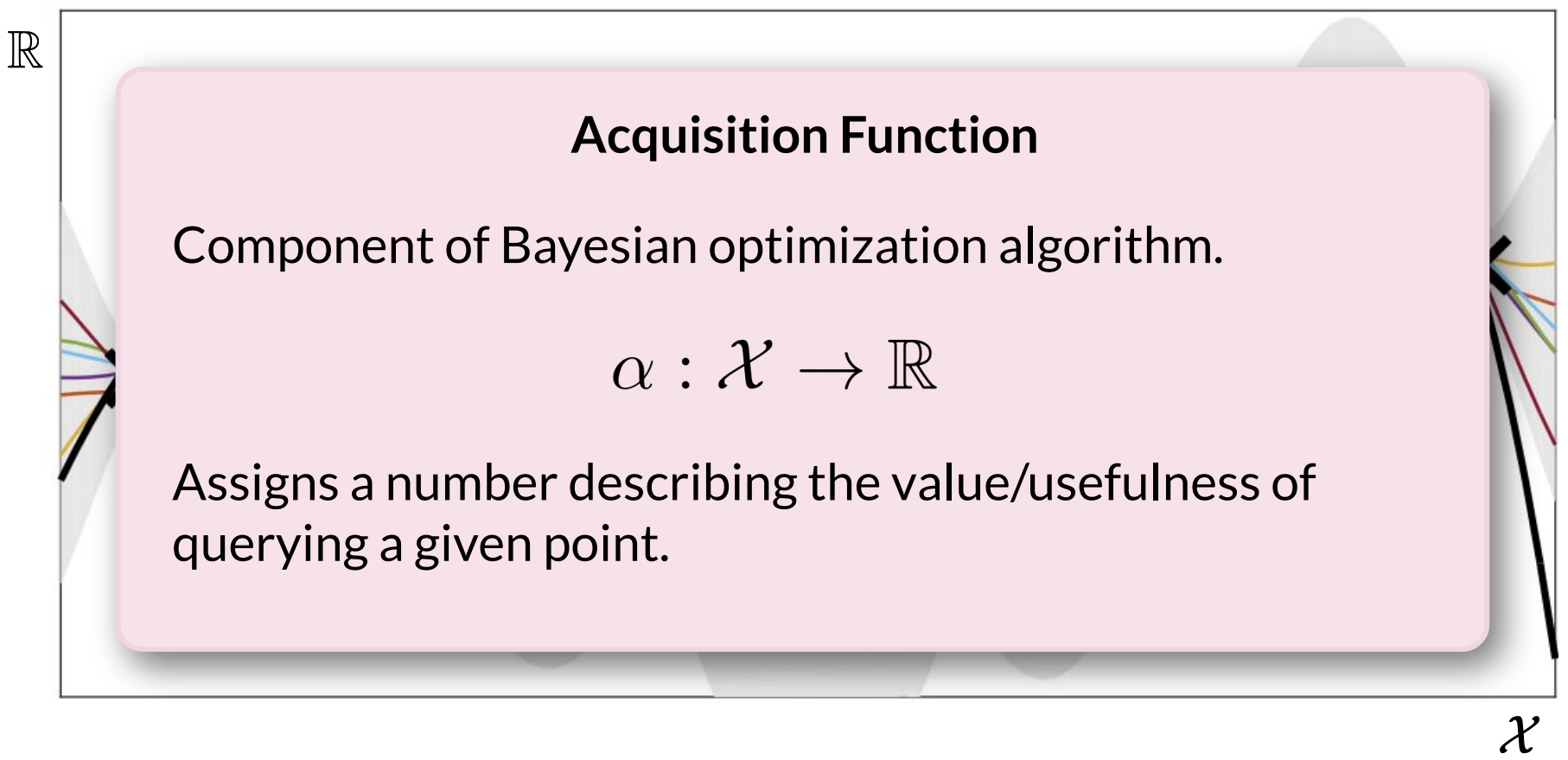
A popular choice for this model is a **Gaussian Process**.

Bayesian Optimization — Visualization



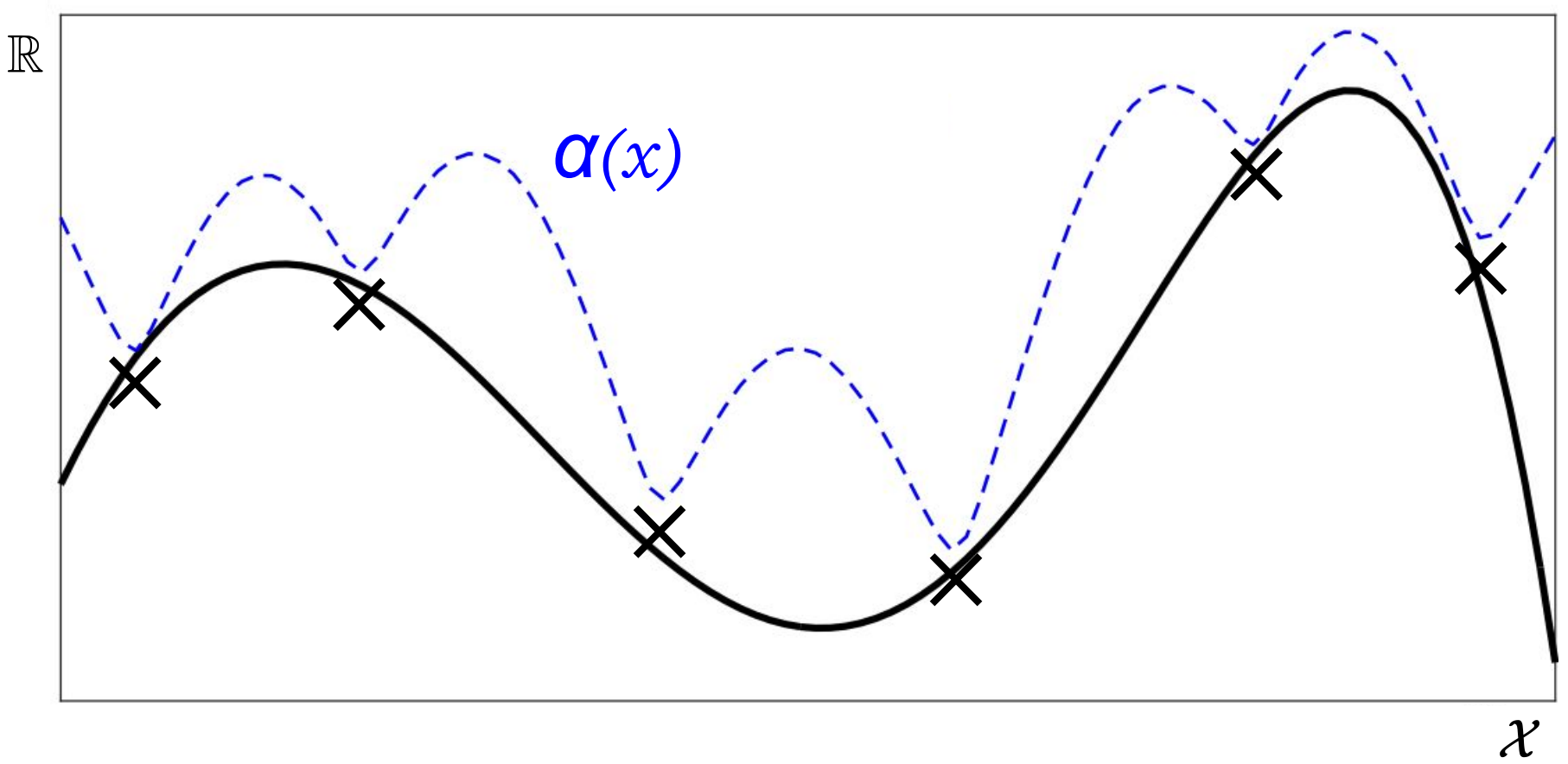
Define **acquisition function** using model.

Bayesian Optimization — Visualization



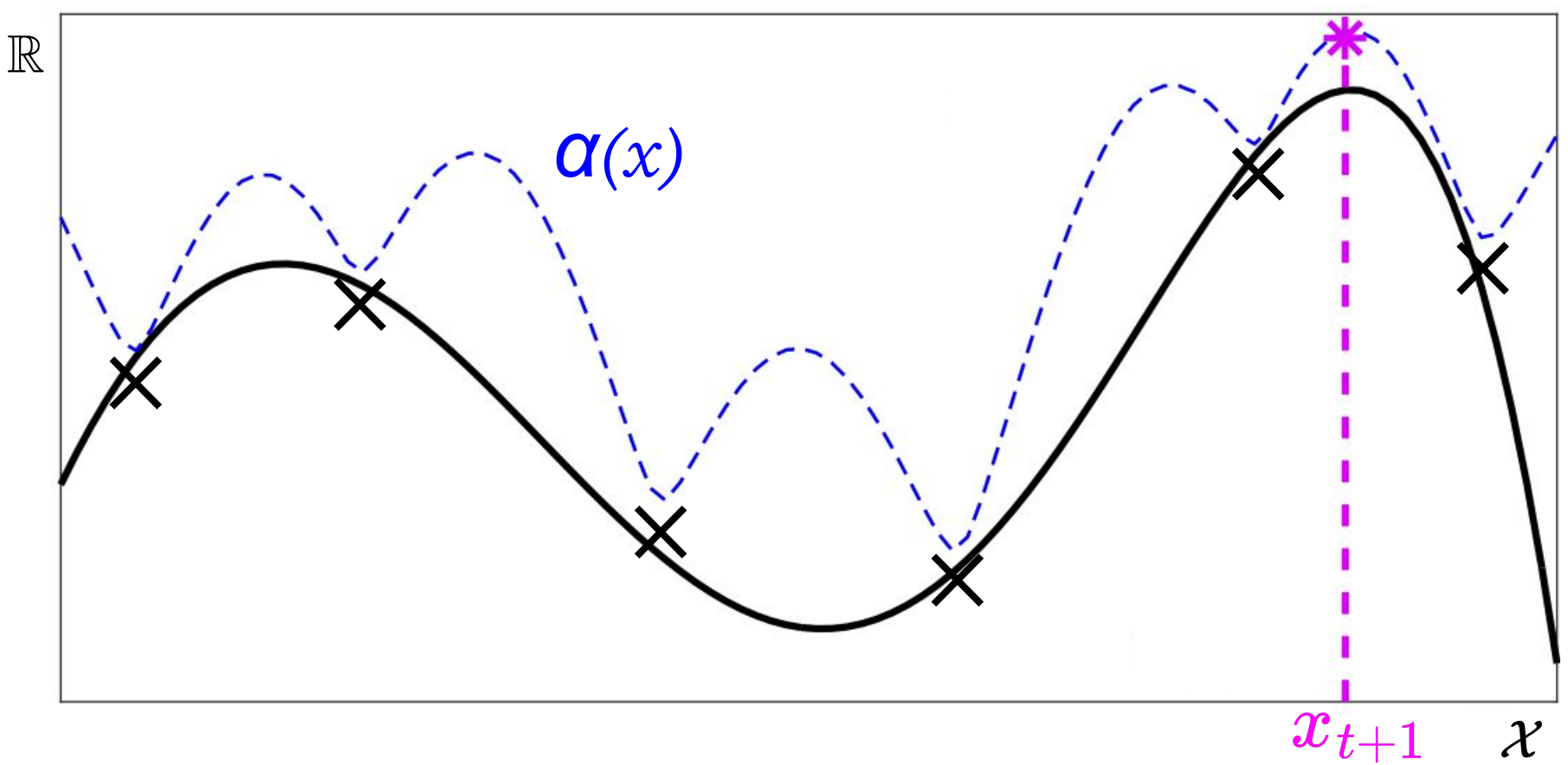
Define **acquisition function** using model.

Bayesian Optimization — Visualization



Define **acquisition function** using model.

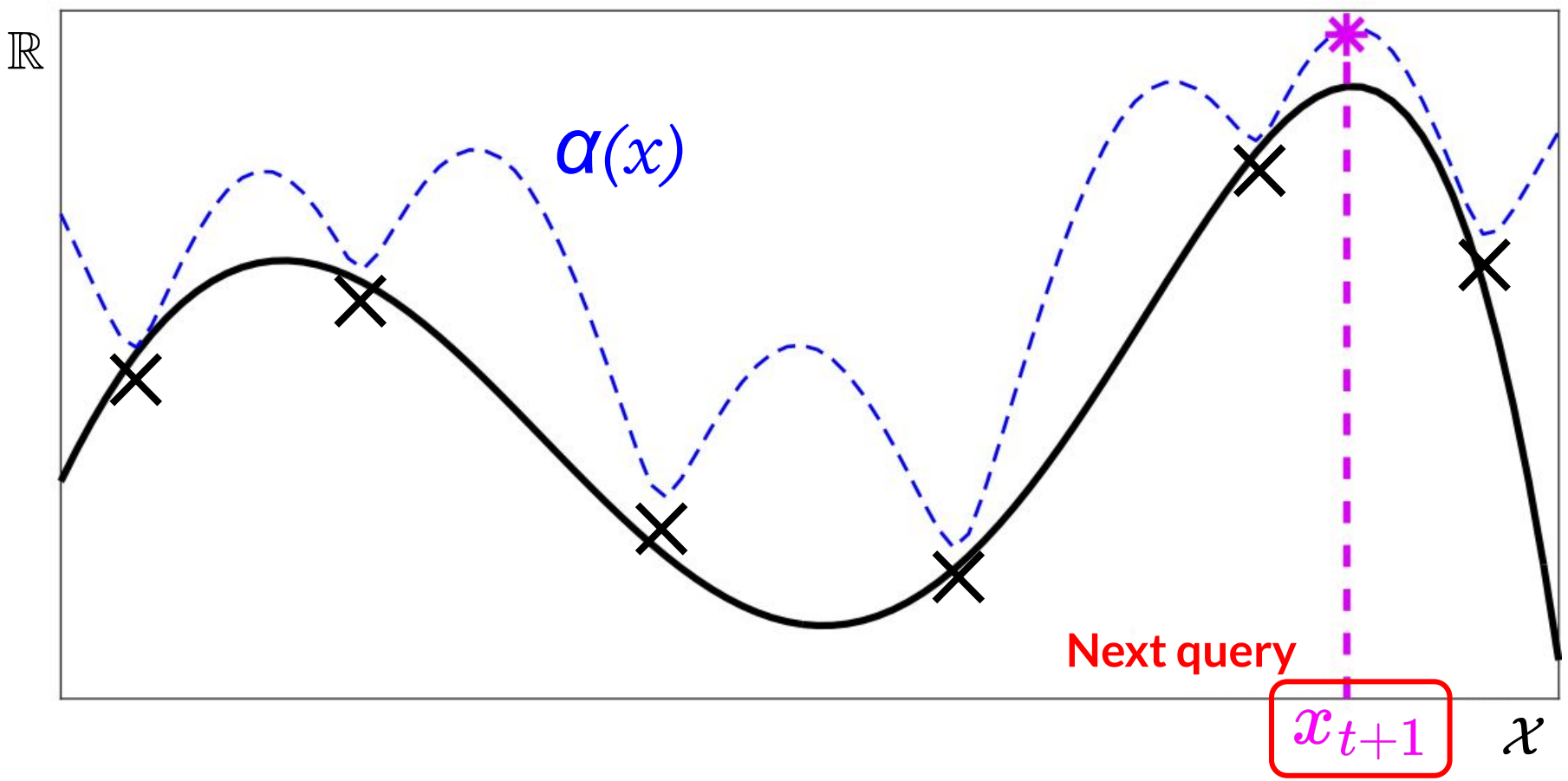
Bayesian Optimization — Visualization



Optimize acquisition function ⇒ yields next point to query.

$$x_{t+1} = \arg\max_{x \in \mathcal{X}} \alpha(x)$$

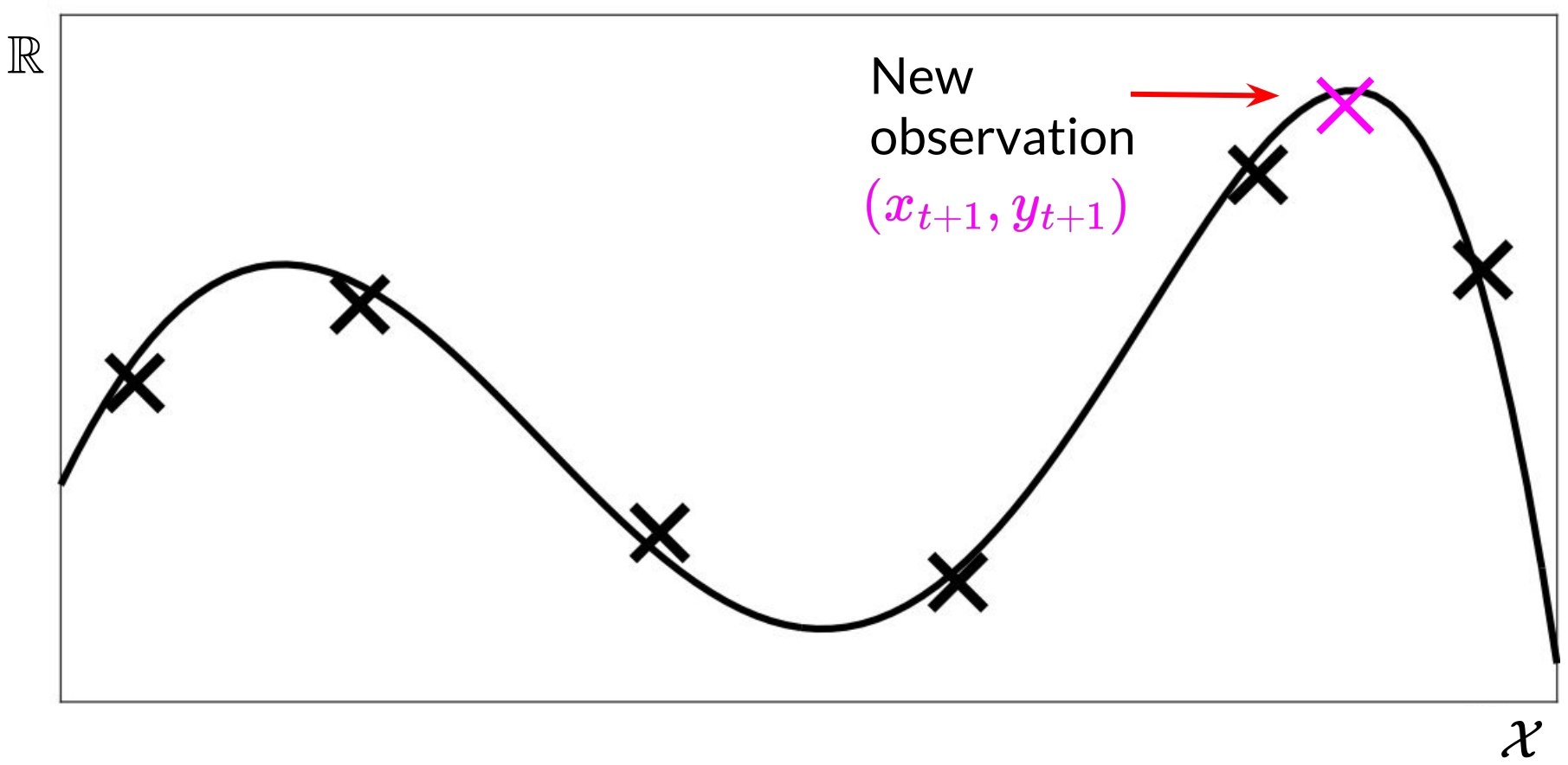
Bayesian Optimization — Visualization



Optimize acquisition function ⇒ yields next point to query.

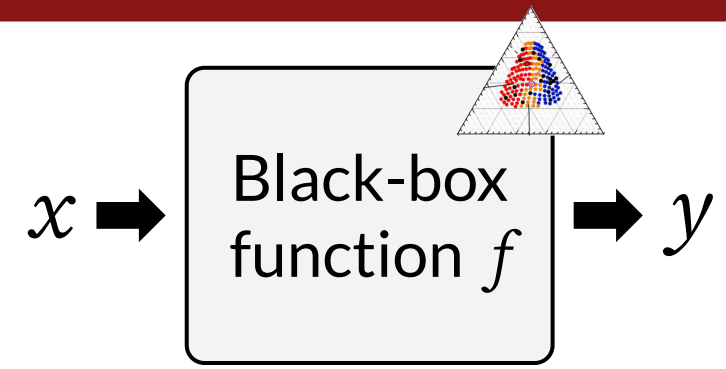
$$x_{t+1} = \arg\max_{x \in \mathcal{X}} \alpha(x)$$

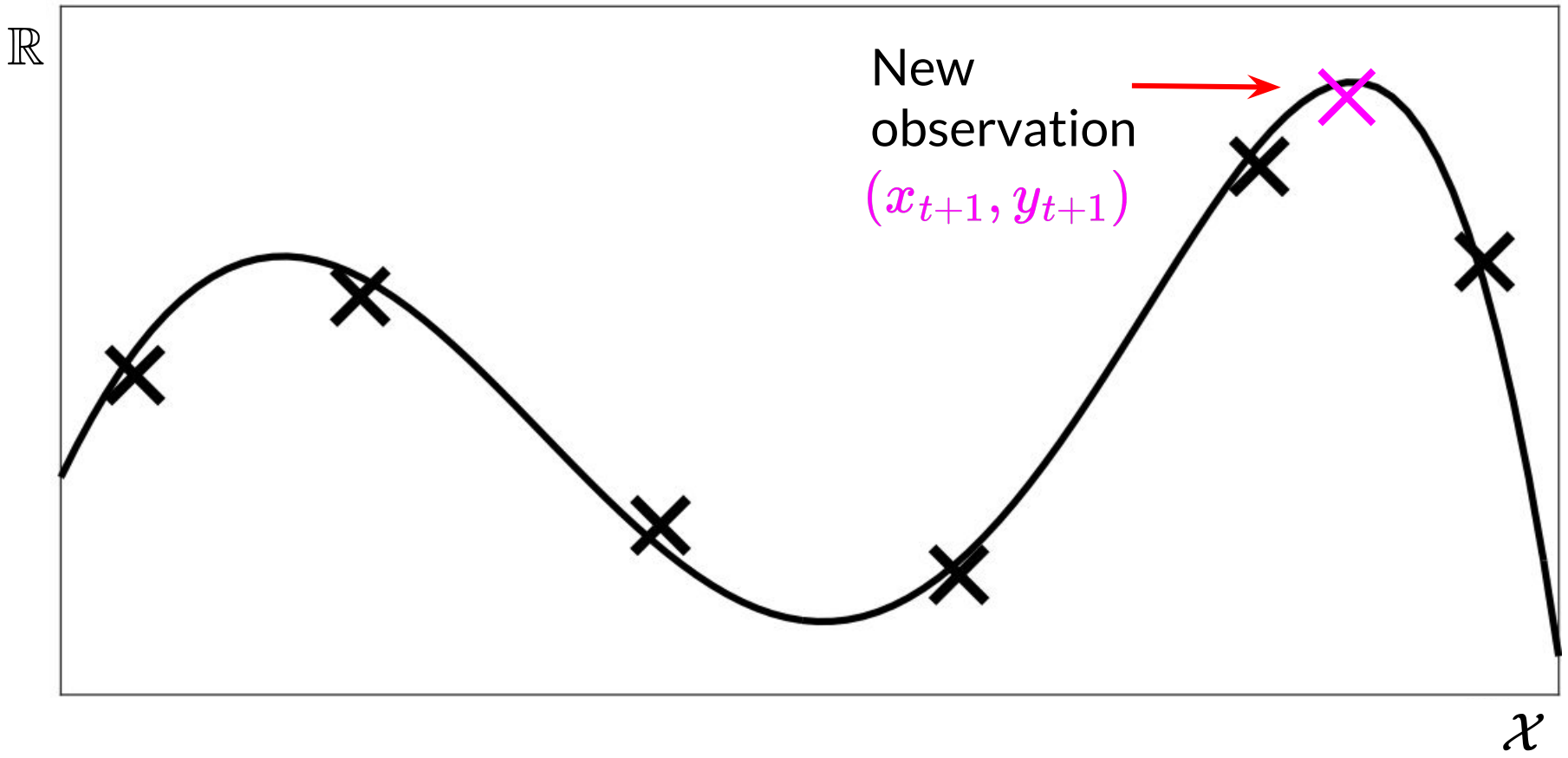
Bayesian Optimization — Visualization



Query black-box f at x, observe y, and update dataset.

Bayesian Optimization — Visualization





Query black-box f at x, observe y, and update dataset.

Acquisition Function — Key Component of BO

Acquisition Function — Key Component of BO

Acquisition functions are, effectively, (one of) the main things that change between different BO procedures.

⇒ they dictate the BO algorithm, and choose the point queried at each iteration.

Acquisition Function — Key Component of BO

Acquisition functions are, effectively, (one of) the main things that change between different BO procedures.

⇒ they dictate the BO algorithm, and choose the point queried at each iteration.

Goal when designing acquisition functions:

- Want to be “optimal” in some sense; to choose the point that is *most informative*.
- And/or: most-likely to accomplish goals of algorithm, *e.g.*, carry out optimization.

Acquisition Function — Key Component of BO

Next we will go through a few popular acquisition functions (for Bayes opt) in more detail:

Acquisition Function — Key Component of BO

Next we will go through a few popular acquisition functions (for Bayes opt) in more detail:

UCB (*upper confidence bound*) — which we saw visualized before.

Acquisition Function — Key Component of BO

Next we will go through a few popular acquisition functions (for Bayes opt) in more detail:

UCB (*upper confidence bound*) — which we saw visualized before.

PI (*probability of improvement*) — one of the simplest acquisition functions.

Acquisition Function — Key Component of BO

Next we will go through a few popular acquisition functions (for Bayes opt) in more detail:

UCB (*upper confidence bound*) — which we saw visualized before.

PI (*probability of improvement*) — one of the simplest acquisition functions.

EI (*expected improvement*) — perhaps the most popular of the acquisition functions.

Acquisition Function — Key Component of BO

Next we will go through a few popular acquisition functions (for Bayes opt) in more detail:

UCB (*upper confidence bound*) — which we saw visualized before.

PI (*probability of improvement*) — one of the simplest acquisition functions.

EI (*expected improvement*) — perhaps the most popular of the acquisition functions.

KG (*knowledge gradient*) — bit more complex, but performs well and is well-justified.

Acquisition Function — Key Component of BO

Next we will go through a few popular acquisition functions (for Bayes opt) in more detail:

UCB (*upper confidence bound*) — which we saw visualized before.

PI (*probability of improvement*) — one of the simplest acquisition functions.

EI (*expected improvement*) — perhaps the most popular of the acquisition functions.

KG (*knowledge gradient*) — bit more complex, but performs well and is well-justified.

ES (*entropy search*) — rooted in information theory and *Bayesian optimal experimental design (BOED)*.

Acquisition Function — Key Component of BO

Next we will go through a few popular acquisition functions (for Bayes opt) in more detail:

UCB (*upper confidence bound*) — which we saw visualized before.

PI (*probability of improvement*) — one of the simplest acquisition functions.

EI (*expected improvement*) — perhaps the most popular of the acquisition functions.

KG (*knowledge gradient*) — bit more complex, but performs well and is well-justified.

ES (*entropy search*) — rooted in information theory and *Bayesian optimal experimental design (BOED)*.

***But first*, a quick discussion on terminology!**

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

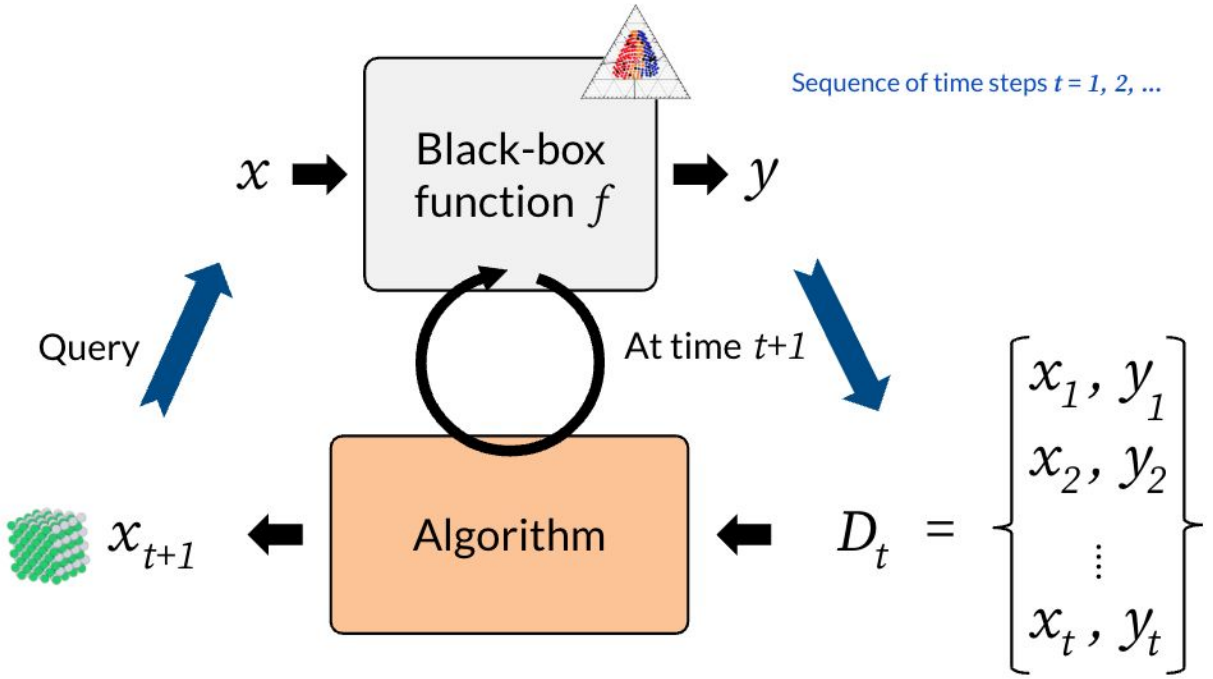
These terms all refer to related procedures, but there are slight differences between them…

Similarities: these all refer to sequential decision making (data acquisition) procedures.

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

These terms all refer to related procedures, but there are slight differences between them…

Similarities: these all refer to sequential decision making (data acquisition) procedures.



Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

These terms all refer to related procedures, but there are slight differences between them…

Differences:

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

These terms all refer to related procedures, but there are slight differences between them…

Differences:

Active Learning

Originally from CS/ML.

Classically: iteratively choosing points to label from a set.

More recently: any learning method where you choose a data point to observe at each iteration .

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

These terms all refer to related procedures, but there are slight differences between them…

Differences:

Active Learning

Originally from CS/ML.

Classically: iteratively choosing points to label from a set.

More recently: any learning method where you choose a data point to observe at each iteration .

Bayesian Optimization

Originally from operations research (among other fields).

Can view as: a special case of active learning, aiming to carry out the task of optimization.

(Which could potentially use an experimental design strategy, *e.g.*, in *entropy search*).

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

These terms all refer to related procedures, but there are slight differences between them…

Differences:

Active Learning

Originally from CS/ML.

Classically: iteratively choosing points to label from a set.

More recently: any learning method where you choose a data point to observe at each iteration .

Bayesian Optimization

Originally from operations research (among other fields).

Can view as: a special case of active learning, aiming to carry out the task of optimization.

(Which could potentially use an experimental design strategy, *e.g.*, in *entropy search*).

Experimental Design

Originally from statistics.

Classically: goal is to efficiently estimate a model parameter (often by reducing posterior entropy).

Can view parameter estimation as one strategy for active learning/Bayesian optimization.

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

A bit more on experimental design:

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

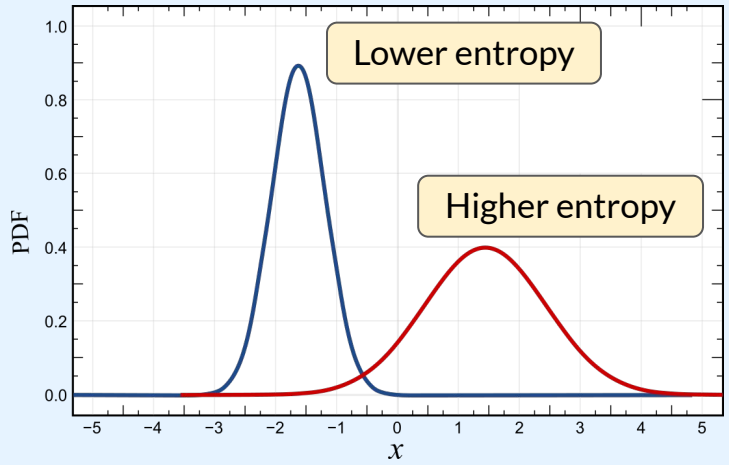
A bit more on experimental design: often referred to as *information-based experimental design* (if it involves reducing posterior entropy), or *Bayesian optimal experimental design (BOED)*.

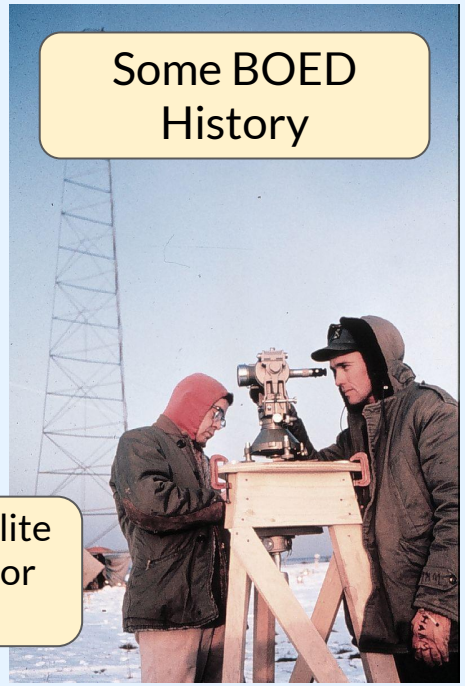
Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

A bit more on experimental design: often referred to as *information-based experimental design* (if it involves reducing posterior entropy), or *Bayesian optimal experimental design (BOED)*.

BOED: have model with an (unknown) parameter of interest.

- Choose experiments that most reduce uncertainty about parameter.
- *Uncertainty*: **entropy** of posterior distribution over parameter.





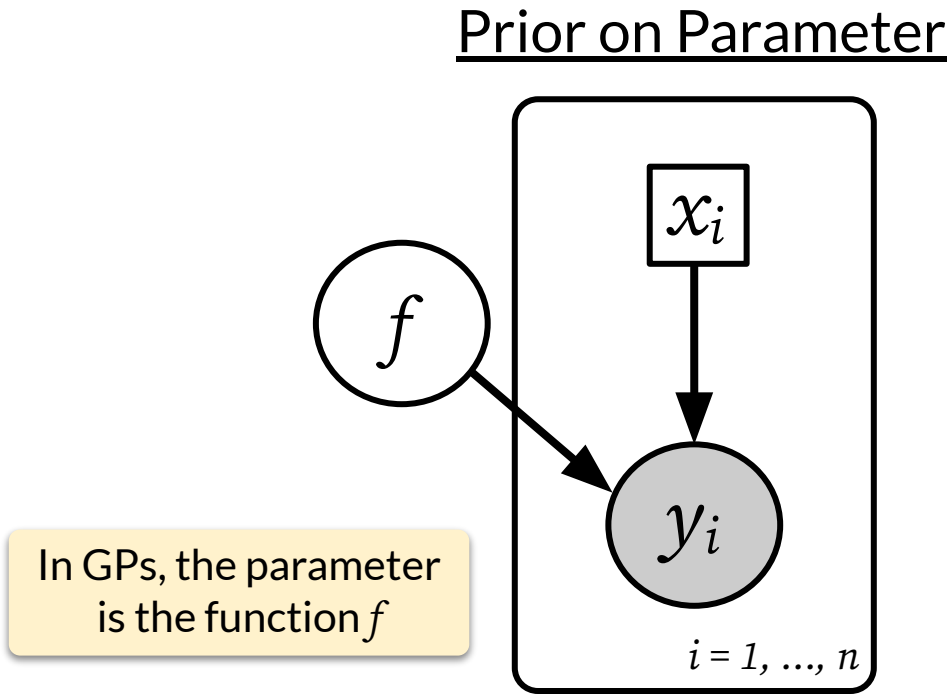
Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

How would experimental design look for GPs?

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

How would experimental design look for GPs?

Recall the PGM for GPs:

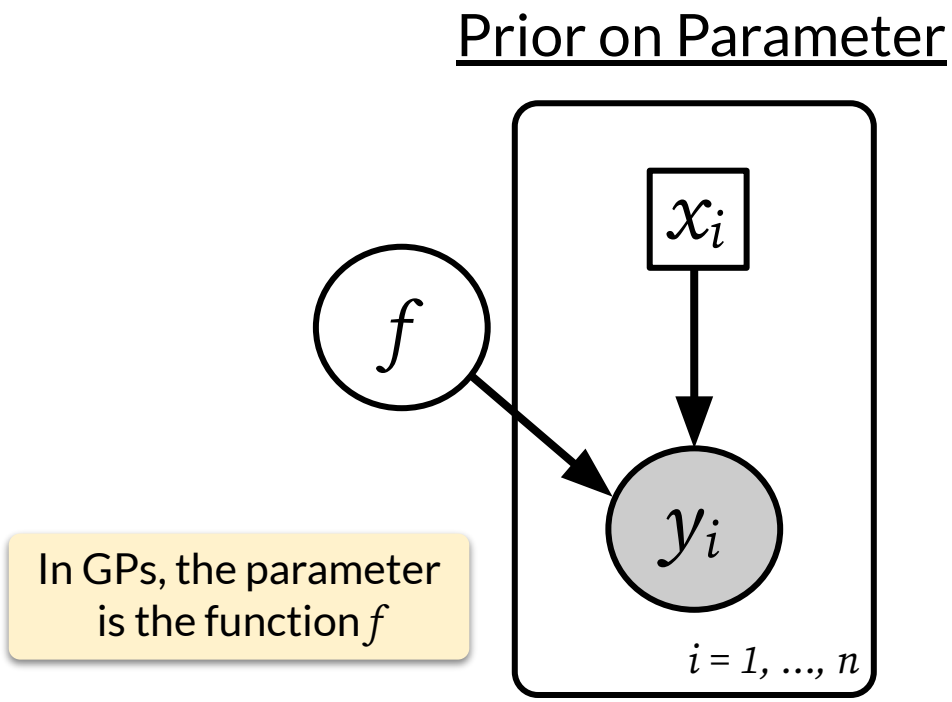


Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

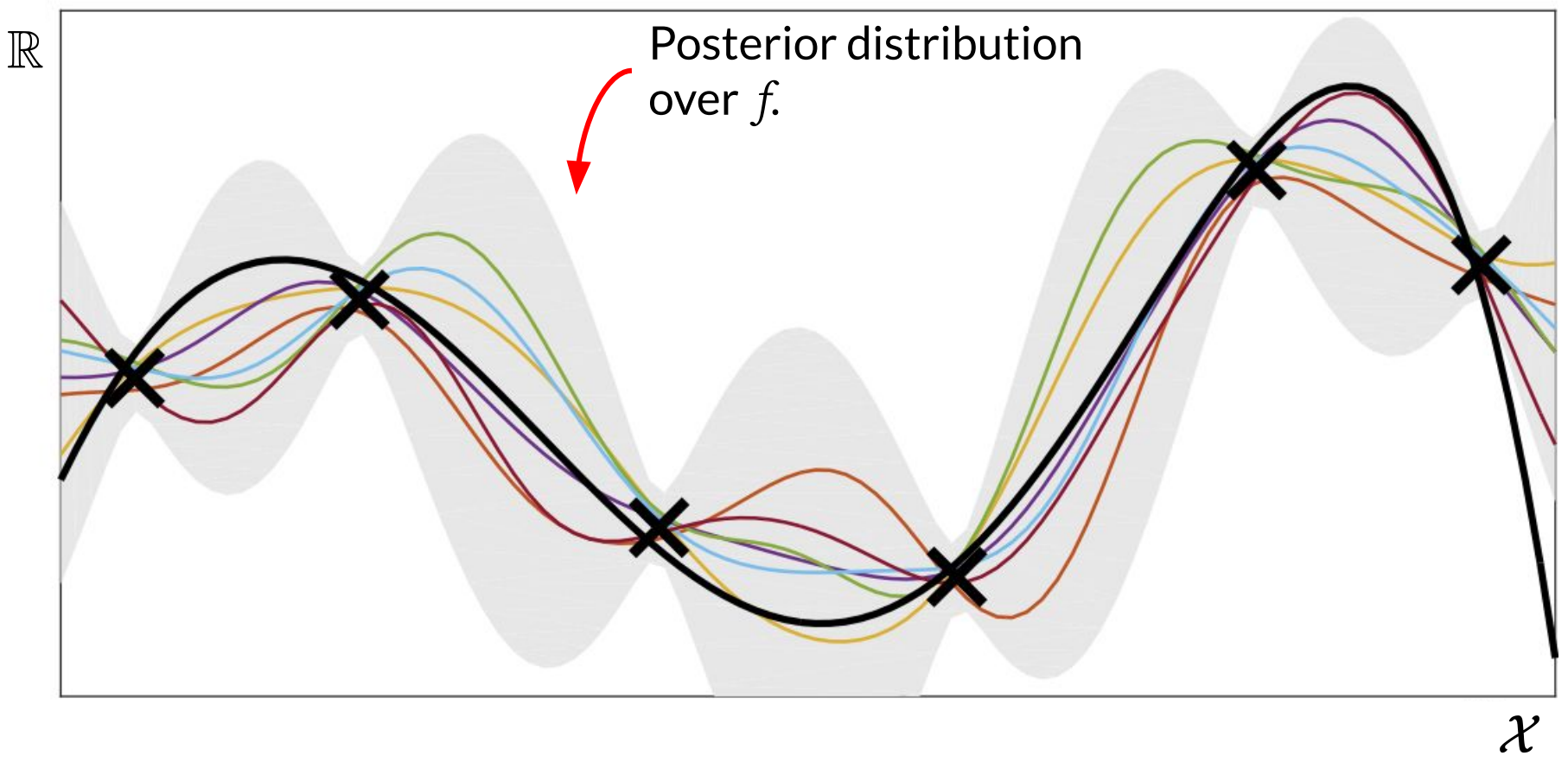
How would experimental design look for GPs?

Recall the PGM for GPs:

- We have a prior on the model parameter.
- Where the model parameter can be viewed as the function f.
- And via Bayesian inference we compute a posterior distribution over f.

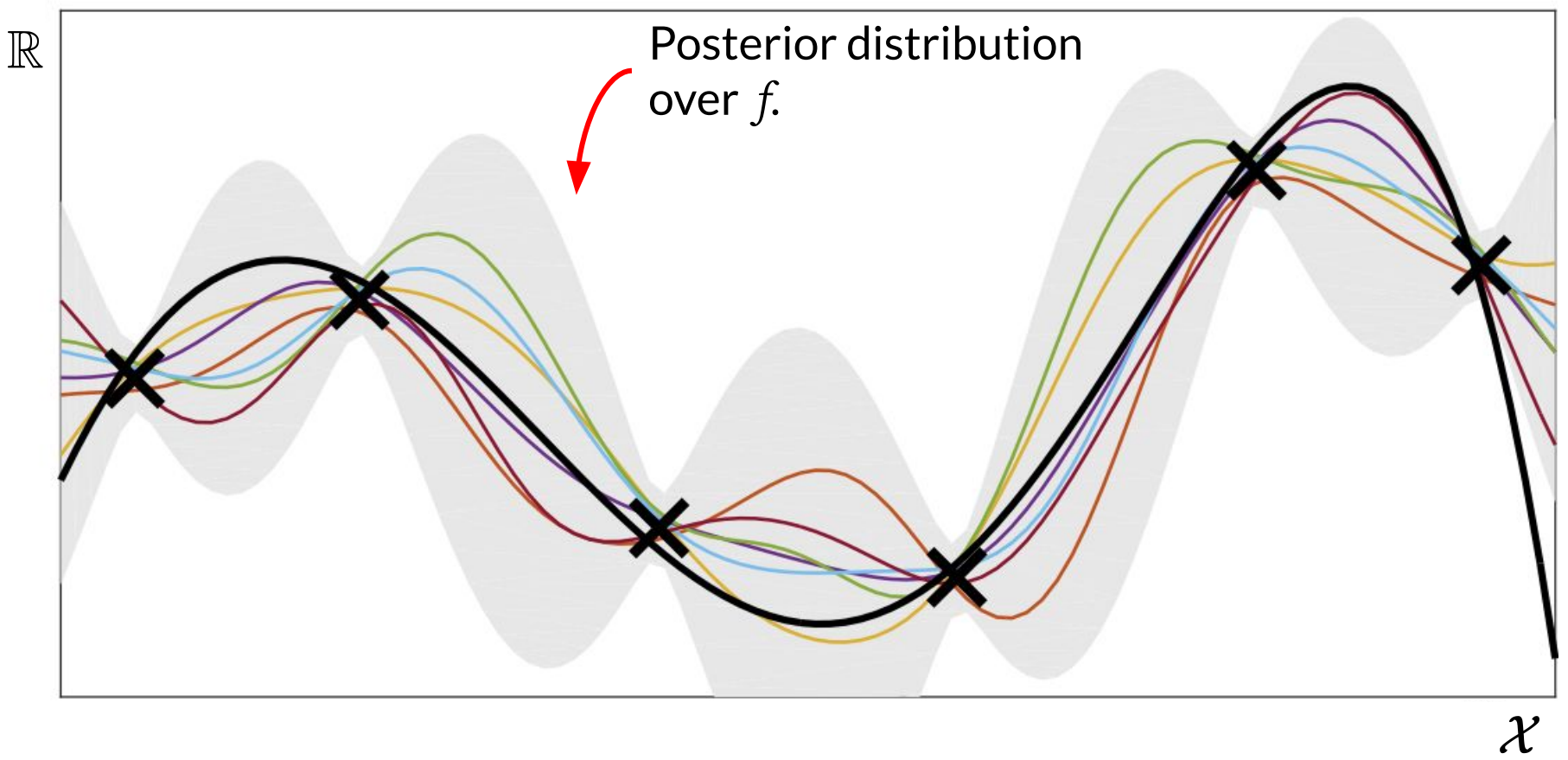


Terminology — Active Learning, Bayes Opt, Experimental Design, etc.



Every time we get an observation, we reduce the posterior uncertainty over f.

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.



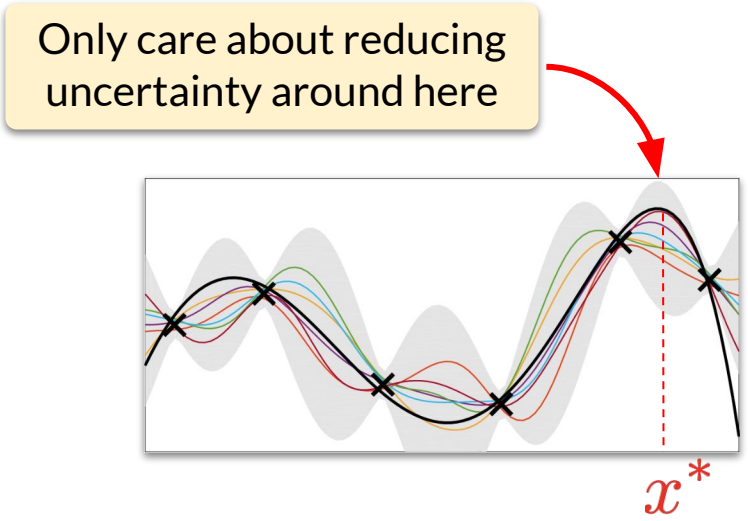
Experimental design ⇒ choose a point to maximally shrink the uncertainty (entropy).

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

As mentioned, this can be used for BO — but we don’t care about reducing uncertainty over full function, just over the region near the optima.

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

As mentioned, this can be used for BO — but we don't care about reducing uncertainty over full function, just over the region near the optima.



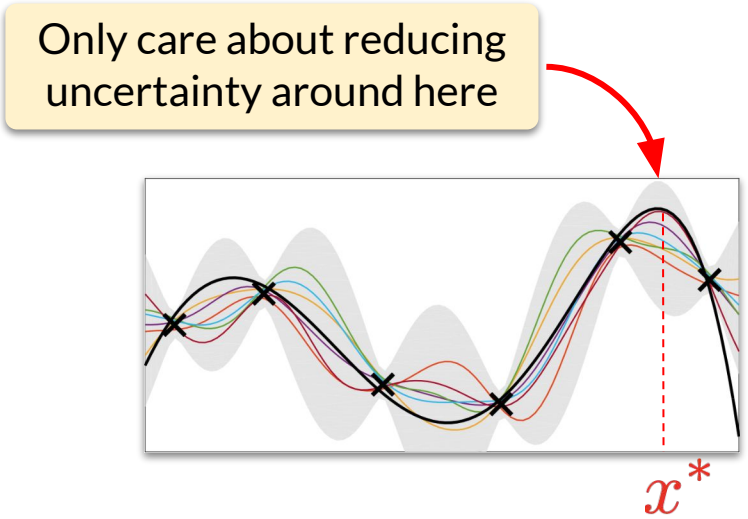
Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

As mentioned, this can be used for BO — but we don't care about reducing uncertainty over full function, just over the region near the optima.

This leads to *information-based Bayesian optimization*:

Info-based BO: entropy search (ES) , predictive ES , max-value ES .

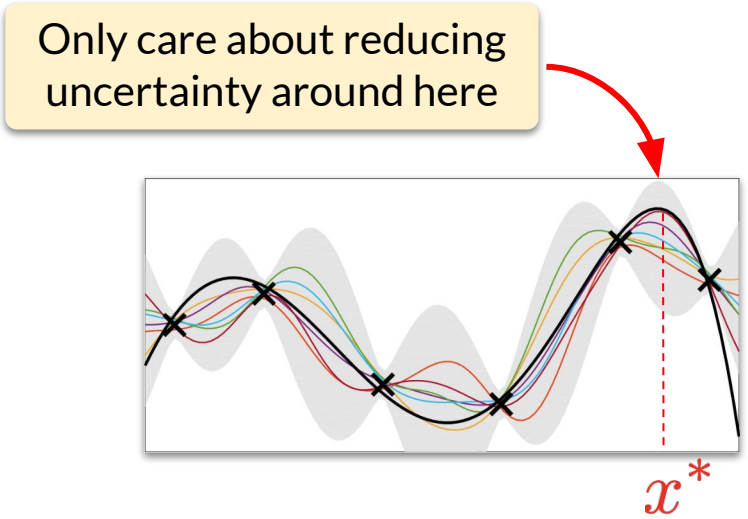
- All rooted in **Bayesian optimal experimental design (BOED)**.



Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

As mentioned, this can be used for BO — but we don't care about reducing uncertainty over full function, just over the region near the optima.

This leads to *information-based Bayesian optimization*:



Info-based BO: entropy search (ES) , predictive ES , max-value ES .

- All rooted in **Bayesian optimal experimental design (BOED)**.

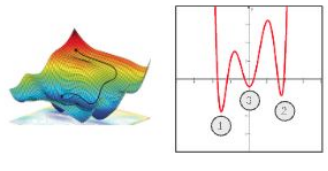
Which we will talk about a little later…

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

Side note: there are many other potential special cases of active learning beyond global optimization, as in BO…

Terminology — Active Learning, Bayes Opt, Experimental Design, etc.

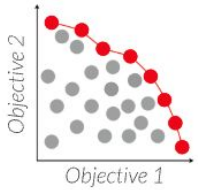
Optimization Variants
(global, local, top-k optima)



Applications

- *ML model training*
- *HPO / NAS*
- *Systems tuning*

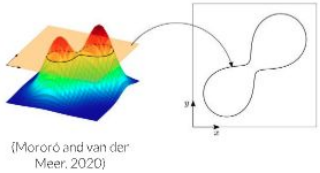
Multi-objective Opt.
(Pareto frontiers)



Applications

- *Process optimization*
- *Portfolio optimization*
- *Laboratory equipment / machines*

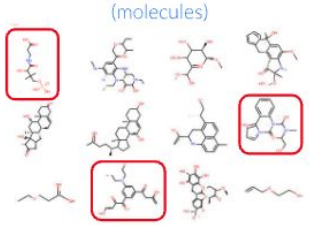
Level Set Estimation
(superlevel set, sublevel set)



Applications

- *Catalyst design*
- *Active learning / weak supervision*
- *Environmental monitoring*

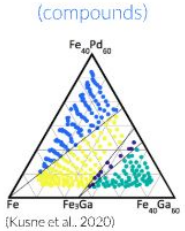
Search
(subset matching criteria)



Applications

- *Drug discovery*
- *Fraud detection*
- *Targeted opinion polling*

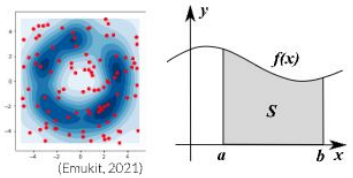
Phase Identification
(boundaries, partitions)



Applications

- *High throughput materials design/discovery*

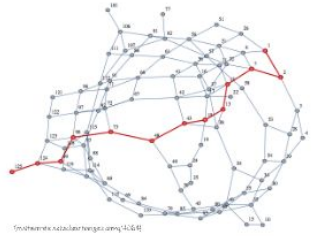
Quadrature, Integration
(integrals, expectations)



Applications

- *Probabilistic modeling (marginal distributions, normalization constants)*
- *Estimating center of mass (and centroids)*

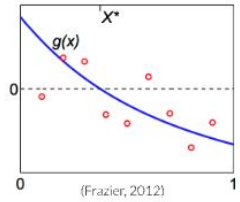
Graph Algorithms
(shortest paths)



Applications

- *Transportation networks*
- *Shipping networks*
- *Social networks*

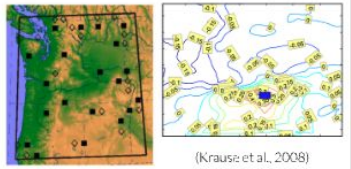
Root Finding, Bisection
(roots)



Applications

- *Target localization (airborne radar)*
- *Edge detection (computer vision)*
- *Biology / microbiology (phase shifts)*

Sensor Placement
(function value at locations)



Applications

- *Water distribution systems*
- *Outbreak detection in networks*
- *Weather monitoring*

Acquisition Functions — Back to Acquisition Functions

Acquisition Functions — Back to Acquisition Functions

Acquisition Function

Component of Bayesian optimization algorithm.

$$\alpha : \mathcal{X} \to \mathbb{R}$$

Assigns a number describing the value/usefulness of querying a given point.

Acquisition Functions — Upper Confidence Bound (UCB)

Acquisition Functions — Upper Confidence Bound (UCB)

The UCB acquisition function is often written:

$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$

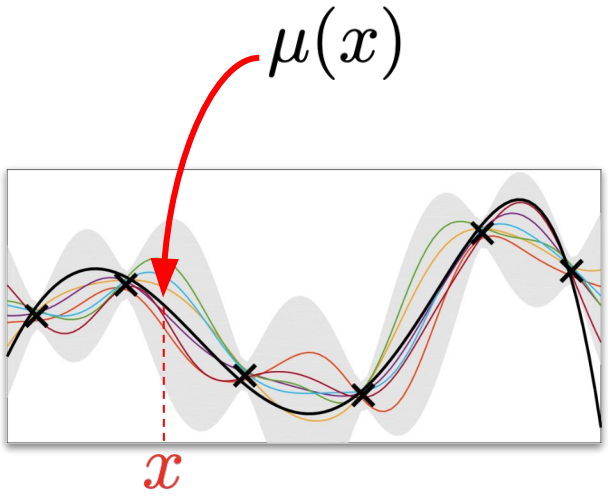
Acquisition Functions — Upper Confidence Bound (UCB)

The UCB acquisition function is often written:

$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$

where:

$\mu(x)$ - Mean of *marginal posterior* at input x.



Acquisition Functions — Upper Confidence Bound (UCB)

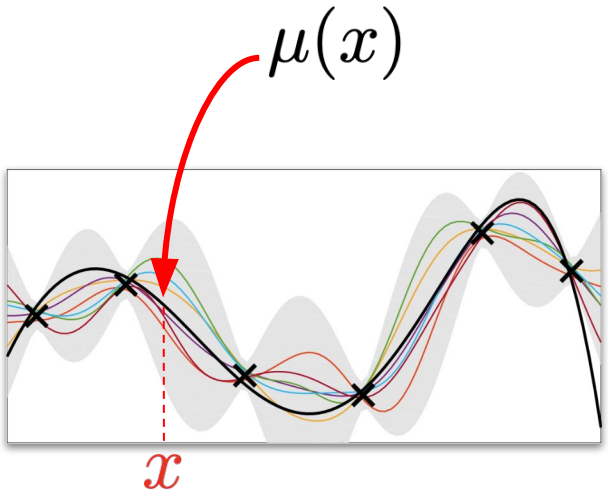
The UCB acquisition function is often written:

$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$

where:

$p(f(x) \mid \mathcal{D}_t)$

$\mu(x)$ - Mean of *marginal posterior* at input x.



Acquisition Functions — Upper Confidence Bound (UCB)

The UCB acquisition function is often written:

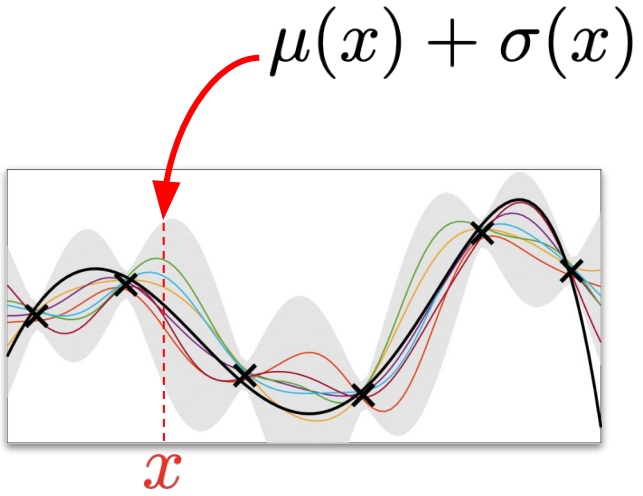
$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$

where:

$p(f(x) \mid \mathcal{D}_t)$

$\mu(x)$ - Mean of *marginal posterior* at input x.

$\sigma(x)$ - Standard deviation of *marginal posterior* at input x.



Acquisition Functions — Upper Confidence Bound (UCB)

The UCB acquisition function is often written:

$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$

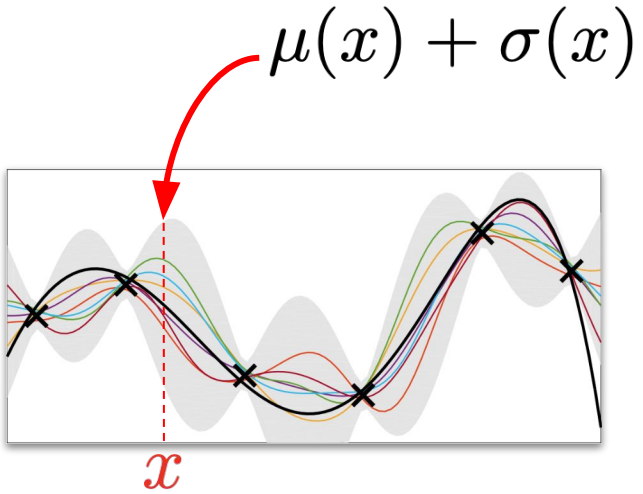
where:

$p(f(x) \mid \mathcal{D}_t)$

$\mu(x)$ - Mean of *marginal posterior* at input x.

$\sigma(x)$ - Standard deviation of *marginal posterior* at input x.

β_t - Hyperparameter controlling tradeoff between exploration and exploitation.

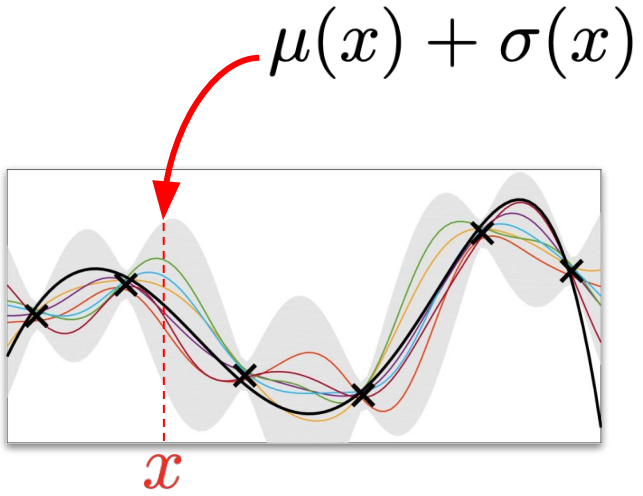


Acquisition Functions — Upper Confidence Bound (UCB)

The UCB acquisition function is often written:

$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$

Intuition:



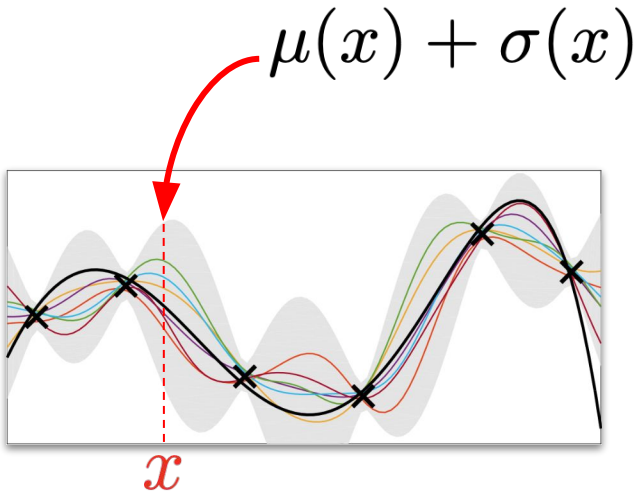
Acquisition Functions — Upper Confidence Bound (UCB)

The UCB acquisition function is often written:

$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$

Intuition:

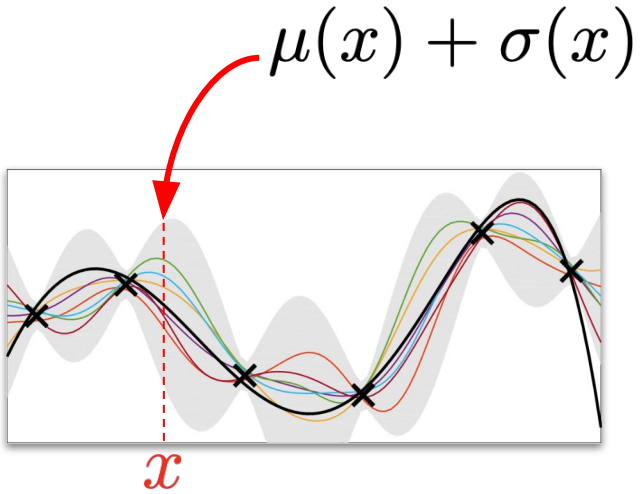
Maximizing std-dev → queries *most unknown* part of function
⇒ most "informative".



Acquisition Functions — Upper Confidence Bound (UCB)

The UCB acquisition function is often written:

$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$



Intuition:

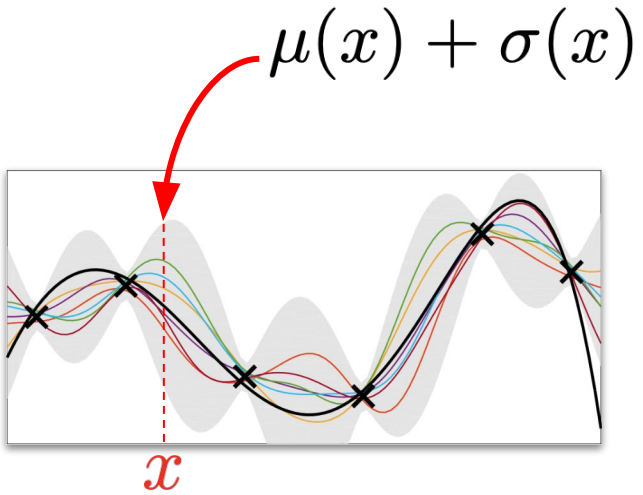
Maximizing std-dev → queries *most unknown* part of function ⇒ most "informative".

Maximizing mean → queries *best guess of highest value*, based on current knowledge of function.

Acquisition Functions — Upper Confidence Bound (UCB)

The UCB acquisition function is often written:

$$\alpha_t(x) = \mu(x) + \beta_t \sigma(x)$$



Intuition:

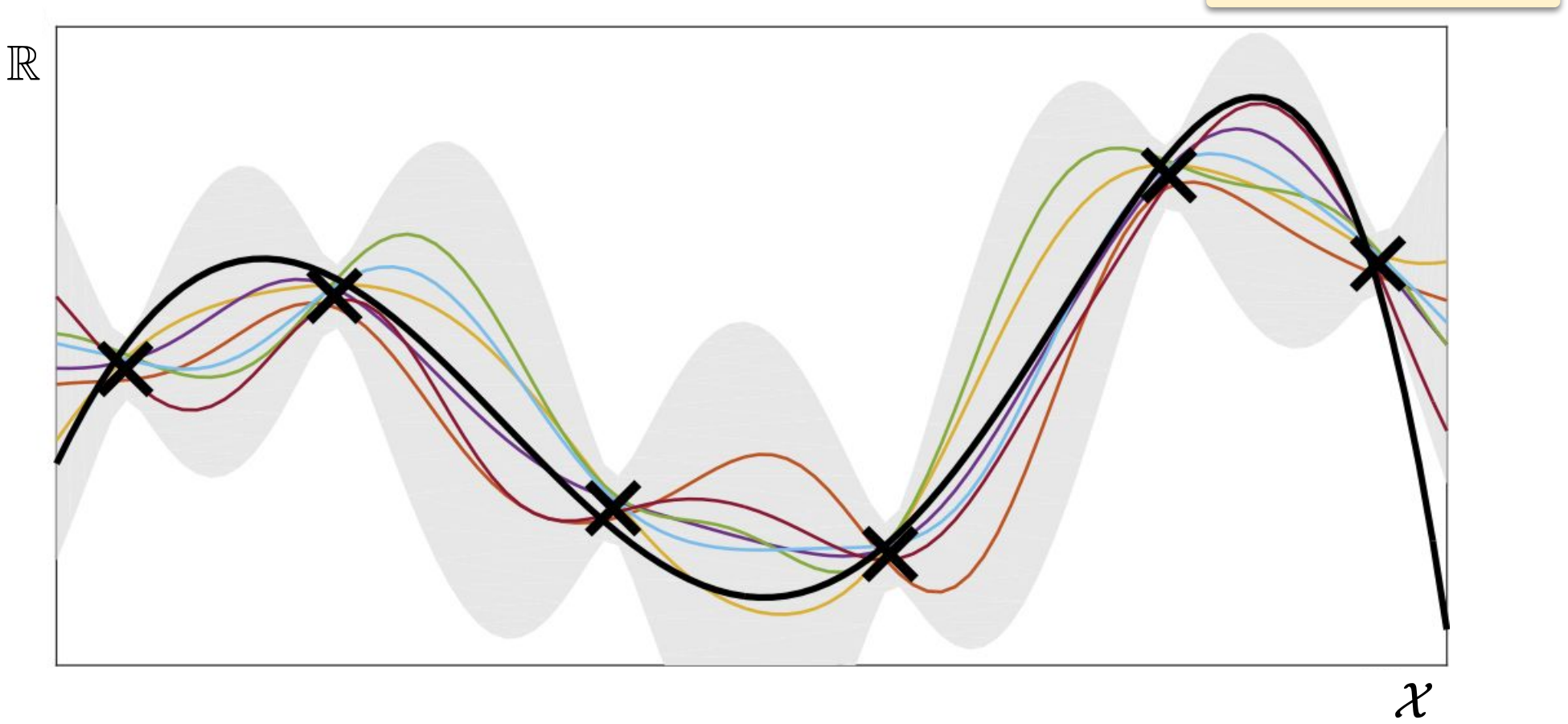
Maximizing std-dev → queries *most unknown* part of function ⇒ most "informative".

Maximizing mean → queries *best guess of highest value*, based on current knowledge of function.

We want to trade off between these two quantities.

Acquisition Functions — Upper Confidence Bound (UCB)

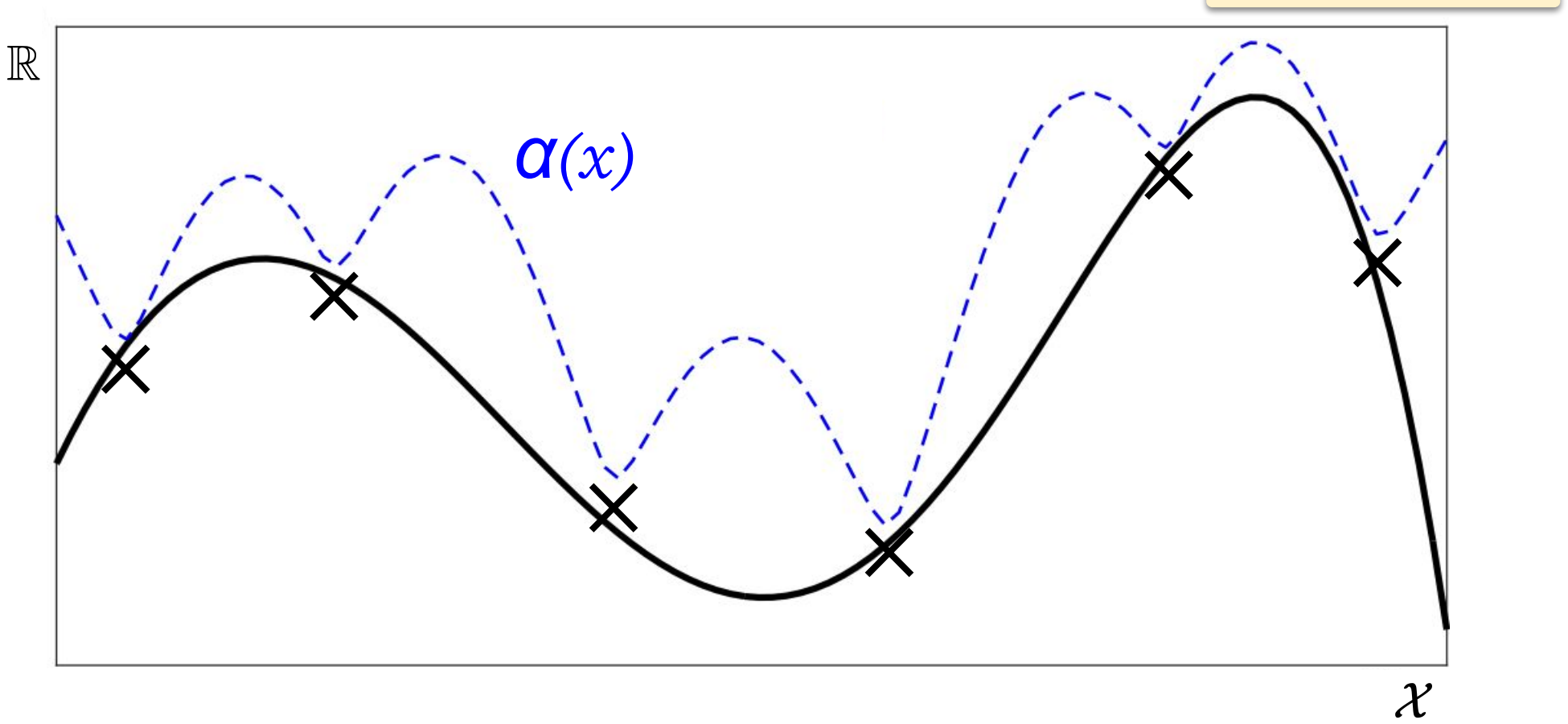
Visualizing UCB



Given our probabilistic model…

Acquisition Functions — Upper Confidence Bound (UCB)

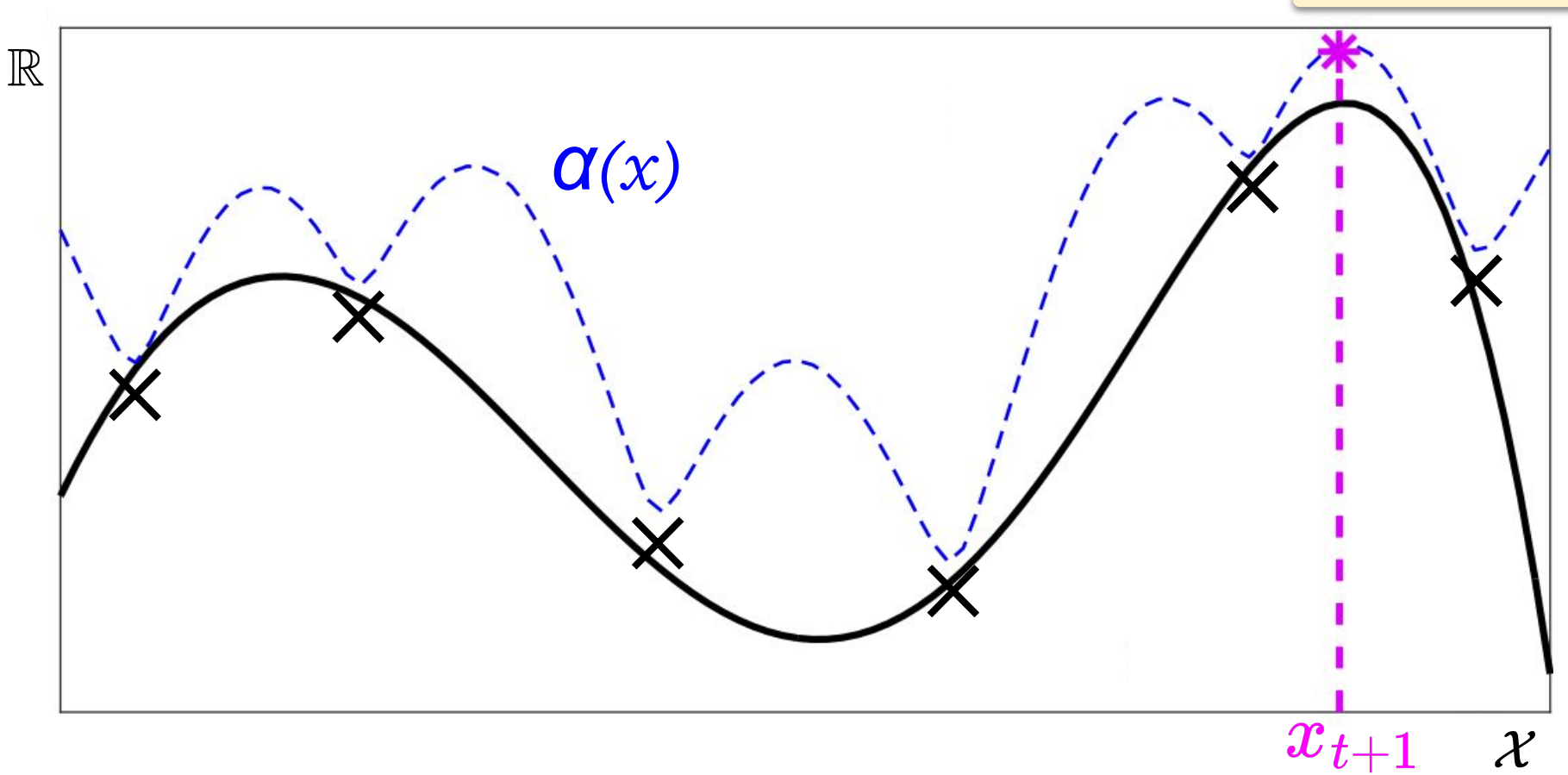
Visualizing UCB



… define the **UCB acquisition function**.

Acquisition Functions — Upper Confidence Bound (UCB)

Visualizing UCB



Optimizing UCB ⇒ yields next point to query.

$$x_{t+1} = \arg\max_{x \in \mathcal{X}} \alpha(x)$$

Acquisition Functions — Experimental Design (EIG about f)

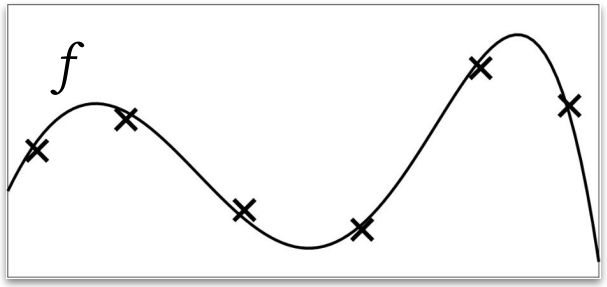
Quick aside on experimental design, to give some perspective on UCB:

Acquisition Functions — Experimental Design (EIG about f)

Quick aside on experimental design, to give some perspective on UCB:

Suppose you don't want to do optimization, but instead want to learn the full function landscape (*i.e.*, learn "all of f").

⇒ technically speaking, this is not *optimization* anymore.

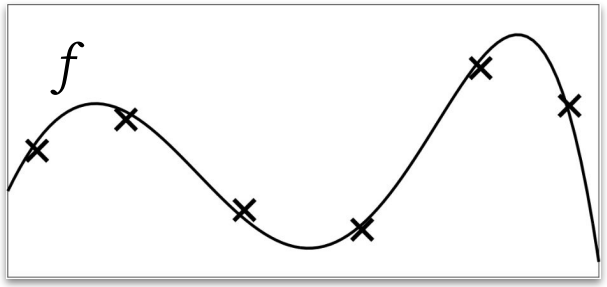


Acquisition Functions — Experimental Design (EIG about f)

Quick aside on experimental design, to give some perspective on UCB:

Suppose you don't want to do optimization, but instead want to learn the full function landscape (*i.e.*, learn "all of f").

⇒ technically speaking, this is not *optimization* anymore.



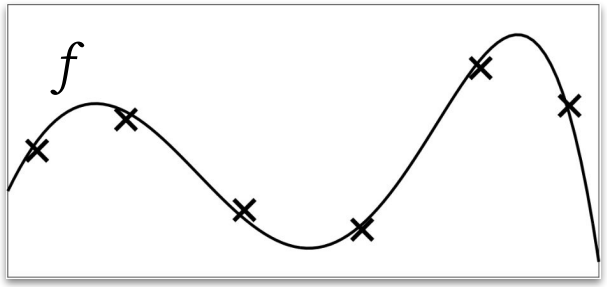
An *experimental design approach* would be to query the point that **maximally reduces the posterior entropy about** f**.**

Acquisition Functions — Experimental Design (EIG about f)

Quick aside on experimental design, to give some perspective on UCB:

Suppose you don't want to do optimization, but instead want to learn the full function landscape (*i.e.*, learn "all of f").

⇒ technically speaking, this is not *optimization* anymore.



An *experimental design approach* would be to query the point that **maximally reduces the posterior entropy about** f**.**

(This is equivalently also commonly called: *expected information gain*, or *EIG, about* f.)

Acquisition Functions — Experimental Design (EIG about f)

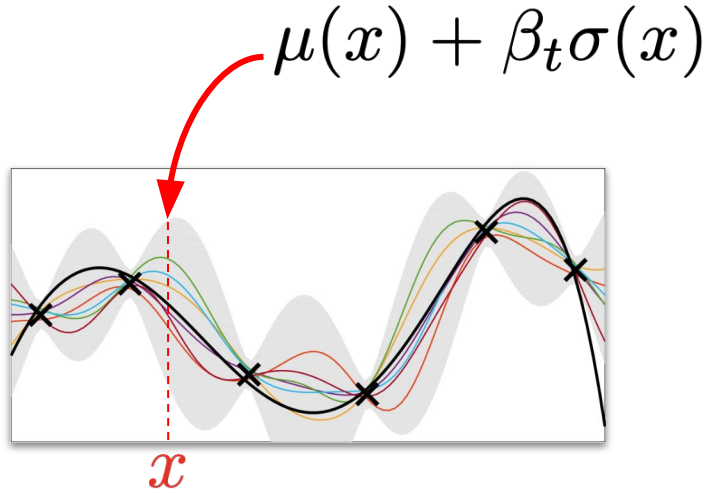
You can prove (at least for GP models) that maximally reducing posterior entropy is equivalent to querying point with maximum standard deviation (of posterior marginal).

Acquisition Functions — Experimental Design (EIG about f)

You can prove (at least for GP models) that maximally reducing posterior entropy is equivalent to querying point with maximum standard deviation (of posterior marginal).

So in **UCB** ⇒

The second term (standard deviation term) is in fact balancing exploration (EIG about f) with exploitation (best expected input x).



Acquisition Functions — Probability of Improvement (PI)

Acquisition Functions — Probability of Improvement (PI)

Next is one of the simplest acquisition functions (though not used a lot in practice).

Acquisition Functions — Probability of Improvement (PI)

Next is one of the simplest acquisition functions (though not used a lot in practice).

The probability of improvement (PI) acquisition function is written:

Acquisition Functions — Probability of Improvement (PI)

Next is one of the simplest acquisition functions (though not used a lot in practice).

The probability of improvement (PI) acquisition function is written:

$$\alpha_t(x) = p(f(x) > f(x^*) \mid \mathcal{D}_t)$$

Acquisition Functions — Probability of Improvement (PI)

Next is one of the simplest acquisition functions (though not used a lot in practice).

The probability of improvement (PI) acquisition function is written:

$$\alpha_t(x) = p(f(x) > f(x^*) \mid \mathcal{D}_t)$$

where $f(x^*)$ is the function value of the best point observed so far in $\mathcal{D}_t$.

I.e., the posterior probability that the f(x) is greater than the best point observed so far.

Acquisition Functions — Probability of Improvement (PI)

Next is one of the simplest acquisition functions (though not used a lot in practice).

The probability of improvement (PI) acquisition function is written:

$$\alpha_t(x) = p(f(x) > f(x^*) \mid \mathcal{D}_t)$$

where $f(x^*)$ is the function value of the best point observed so far in $\mathcal{D}_t$.

I.e., the posterior probability that the f(x) is greater than the best point observed so far.

You can prove this $\alpha_t(x)$ is equal to …

Acquisition Functions — Probability of Improvement (PI)

Next is one of the simplest acquisition functions (though not used a lot in practice).

The probability of improvement (PI) acquisition function is written:

$$\begin{aligned}\alpha_t(x) &= p(f(x) > f(x^*) \mid \mathcal{D}_t) \\ &= \Phi\left(\frac{\mu(x) - f(x^*)}{\sigma(x)}\right)\end{aligned}$$

Acquisition Functions — Probability of Improvement (PI)

Next is one of the simplest acquisition functions (though not used a lot in practice).

The probability of improvement (PI) acquisition function is written:

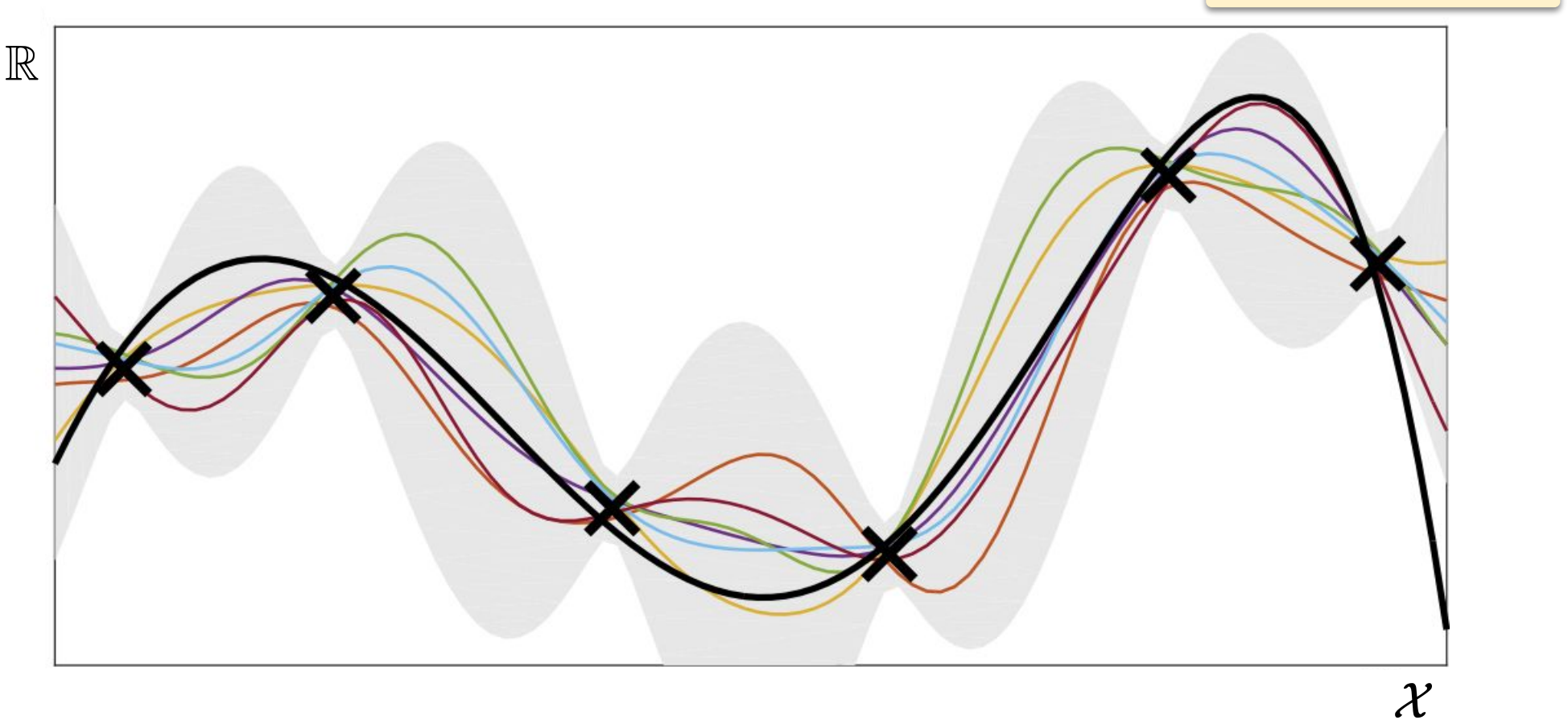
$$\begin{aligned}\alpha_t(x) &= p(f(x) > f(x^*) \mid \mathcal{D}_t) \\ &= \Phi\left(\frac{\mu(x) - f(x^*)}{\sigma(x)}\right)\end{aligned}$$

where $\Phi(\cdot)$ denotes the CDF of the standard normal distribution.

(And, as before, $\mu(x)$ and $\sigma(x)$ are the mean and SD of marginal posterior at x.)

Acquisition Functions — Probability of Improvement (PI)

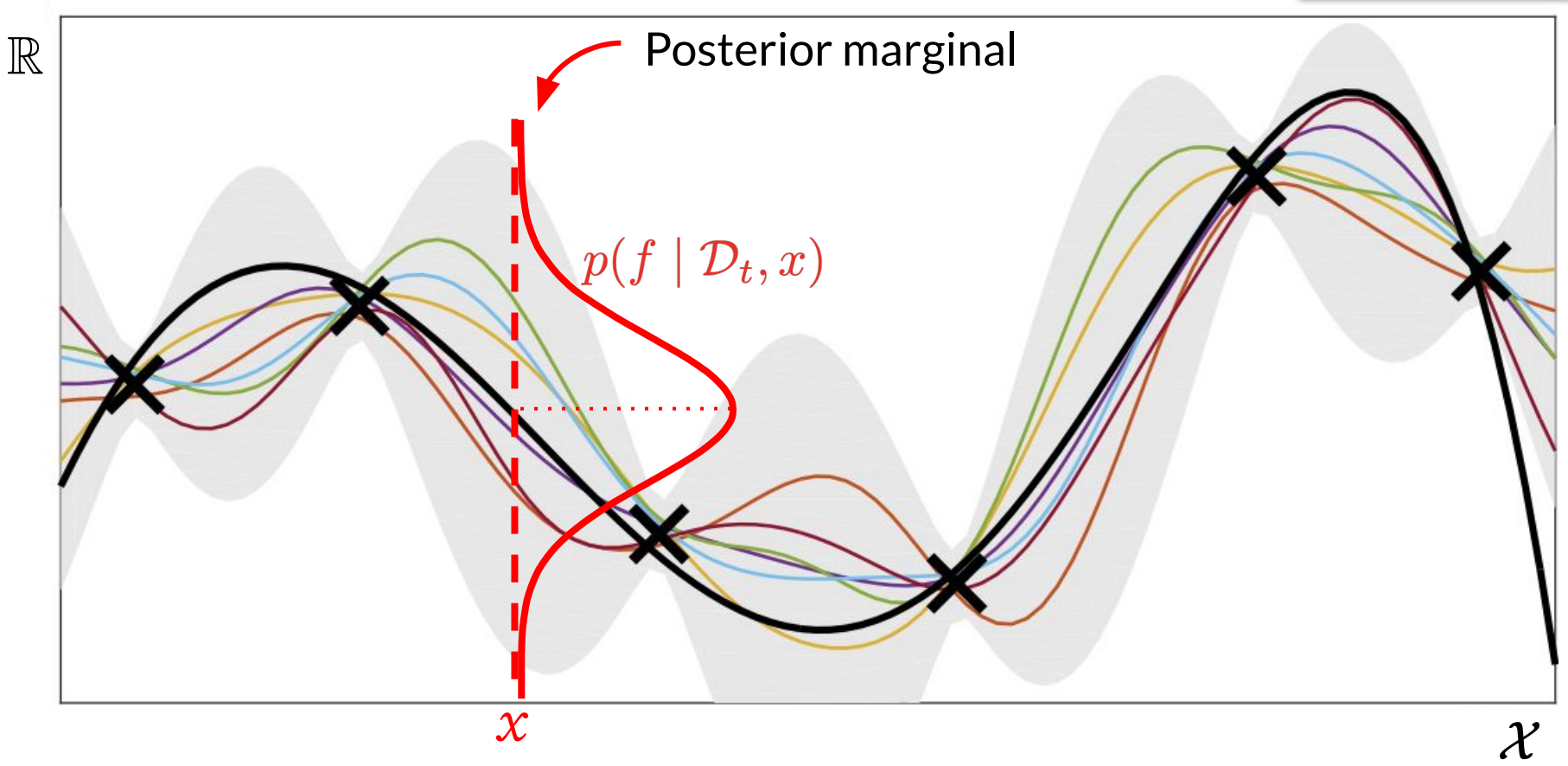
Visualizing PI



Given our probabilistic model…

Acquisition Functions — Probability of Improvement (PI)

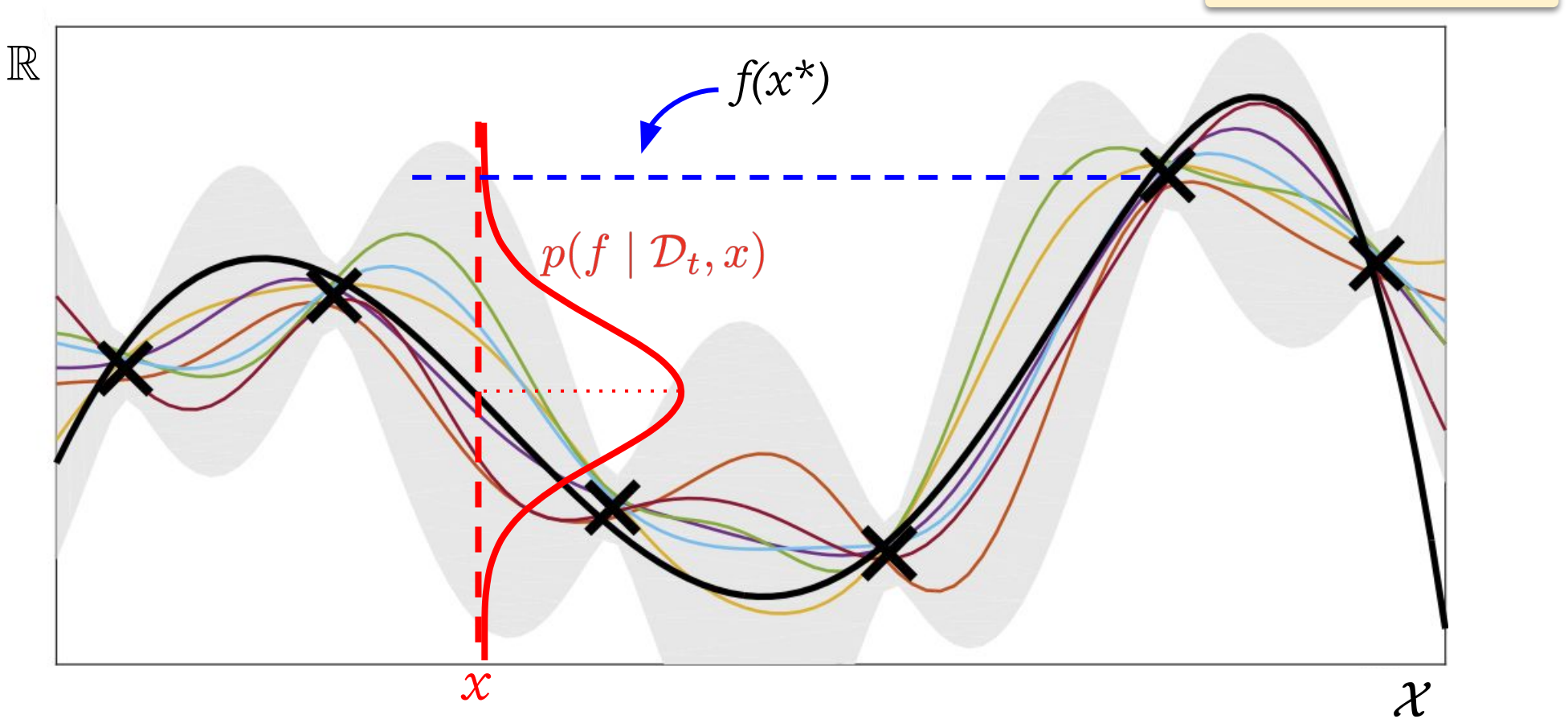
Visualizing PI



Given our probabilistic model…

Acquisition Functions — Probability of Improvement (PI)

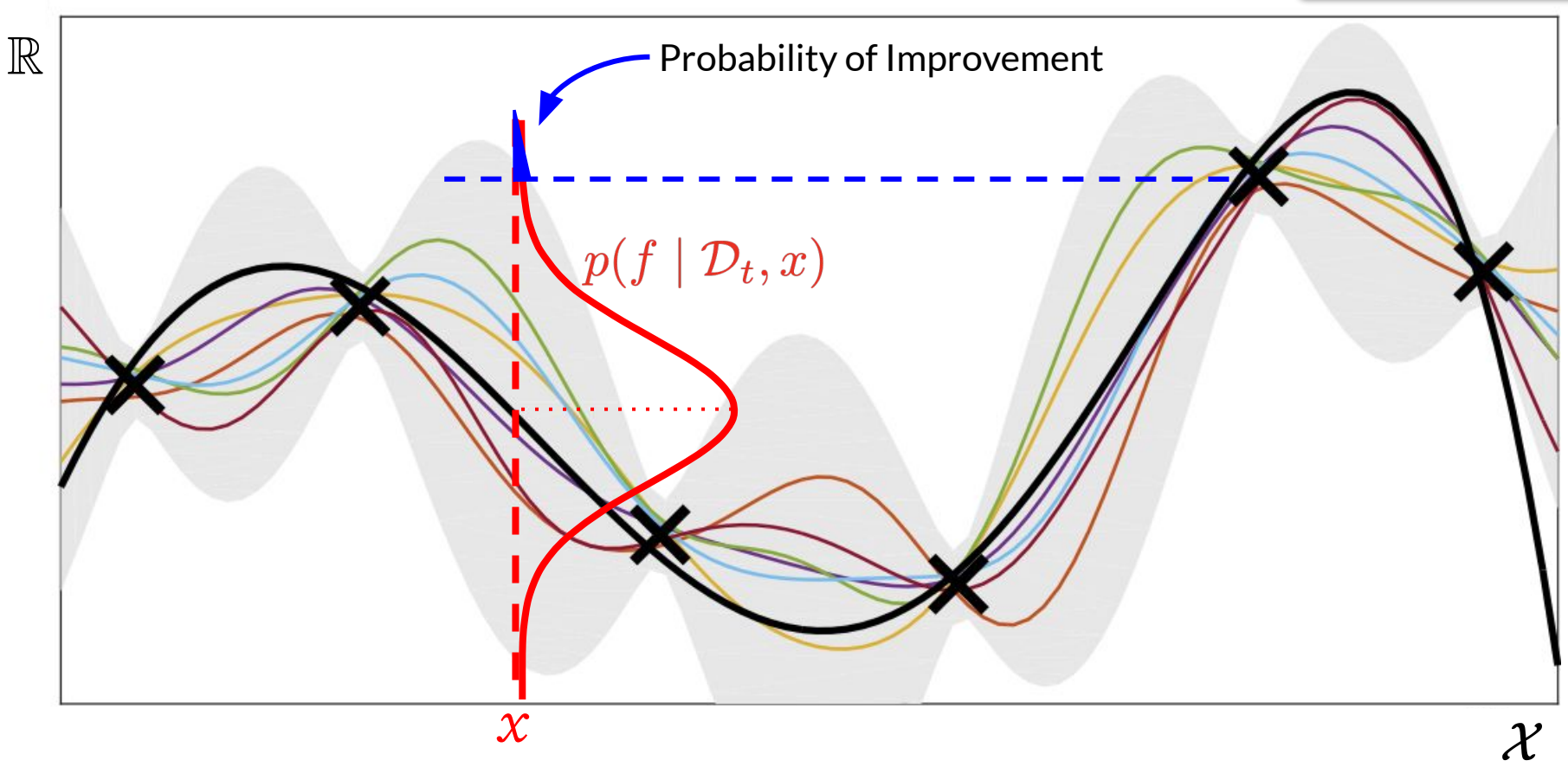
Visualizing PI



Given our probabilistic model…

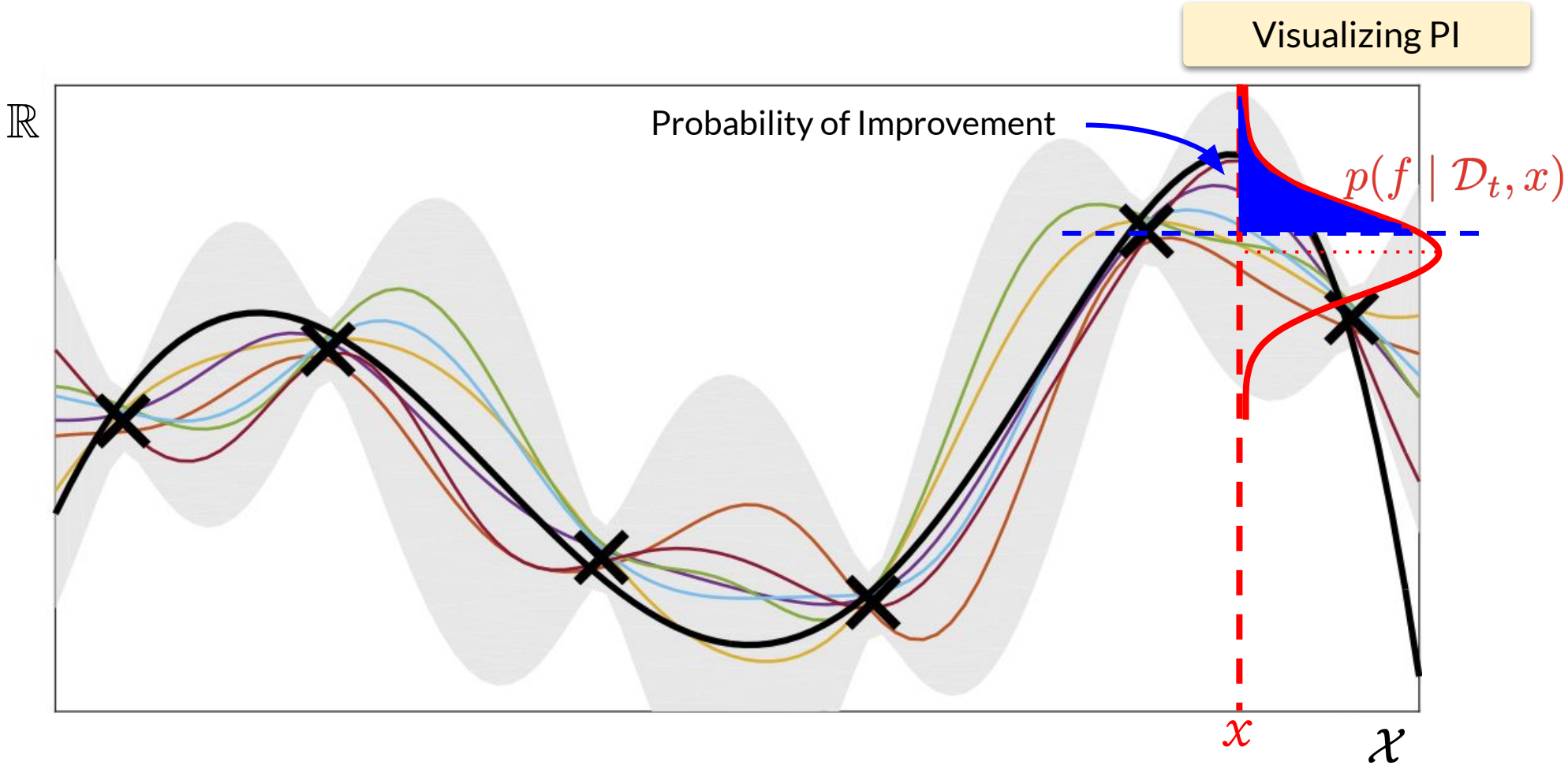
Acquisition Functions — Probability of Improvement (PI)

Visualizing PI



… define the **PI acquisition function**.

Acquisition Functions — Probability of Improvement (PI)



… define the **PI acquisition function**.

Acquisition Functions — Probability of Improvement (PI)

However, note that:

Acquisition Functions — Probability of Improvement (PI)

However, note that:

PI maximizes the **probability** that $f(x)$ is greater than then best point so far.

Does *not* try to maximize the **amount** that $f(x)$ is greater than the best point so far.

Acquisition Functions — Probability of Improvement (PI)

However, note that:

PI maximizes the **probability** that $f(x)$ is greater than then best point so far.

Does *not* try to maximize the **amount** that $f(x)$ is greater than the best point so far.

(*i.e.*, it might choose an x that is only *slightly better* than best point so far, if it is more confident about it — rather than a point that is *significantly greater in expectation*, but it is less confident about it).

Acquisition Functions — Probability of Improvement (PI)

However, note that:

PI maximizes the **probability** that $f(x)$ is greater than then best point so far.

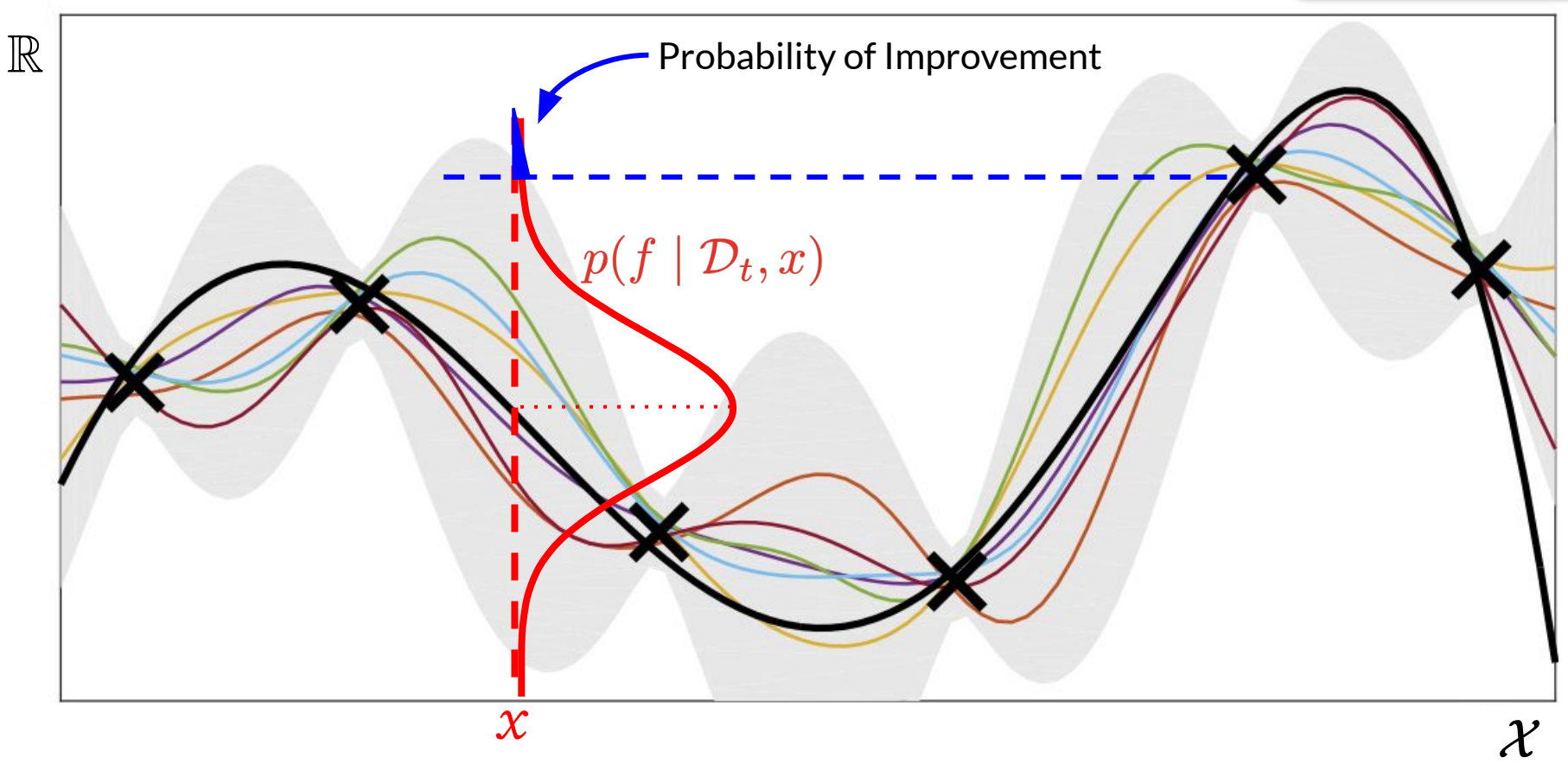
Does *not* try to maximize the **amount** that $f(x)$ is greater than the best point so far.

(*i.e.*, it might choose an x that is only *slightly better* than best point so far, if it is more confident about it — rather than a point that is *significantly greater in expectation*, but it is less confident about it).

Visualizing this…

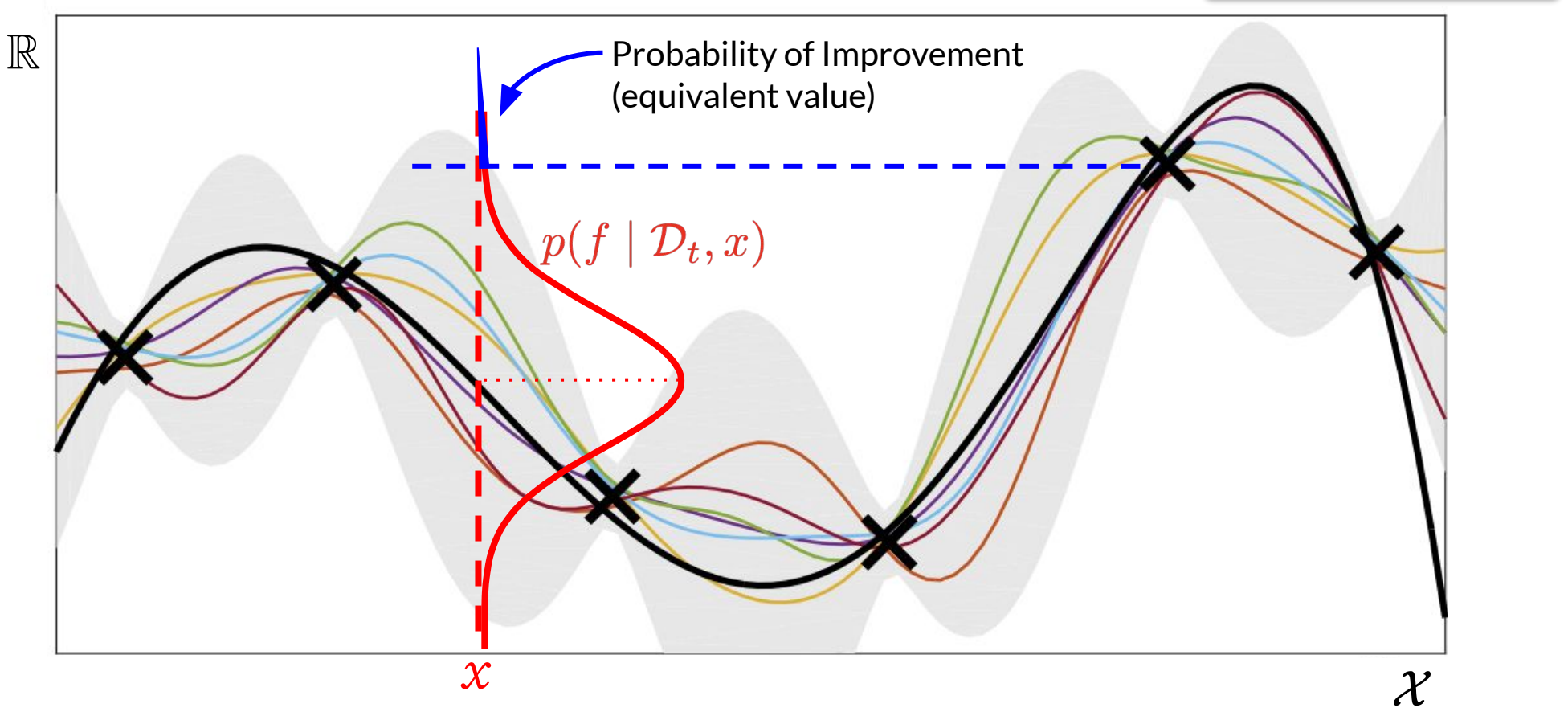
Acquisition Functions — Probability of Improvement (PI)

Visualizing PI



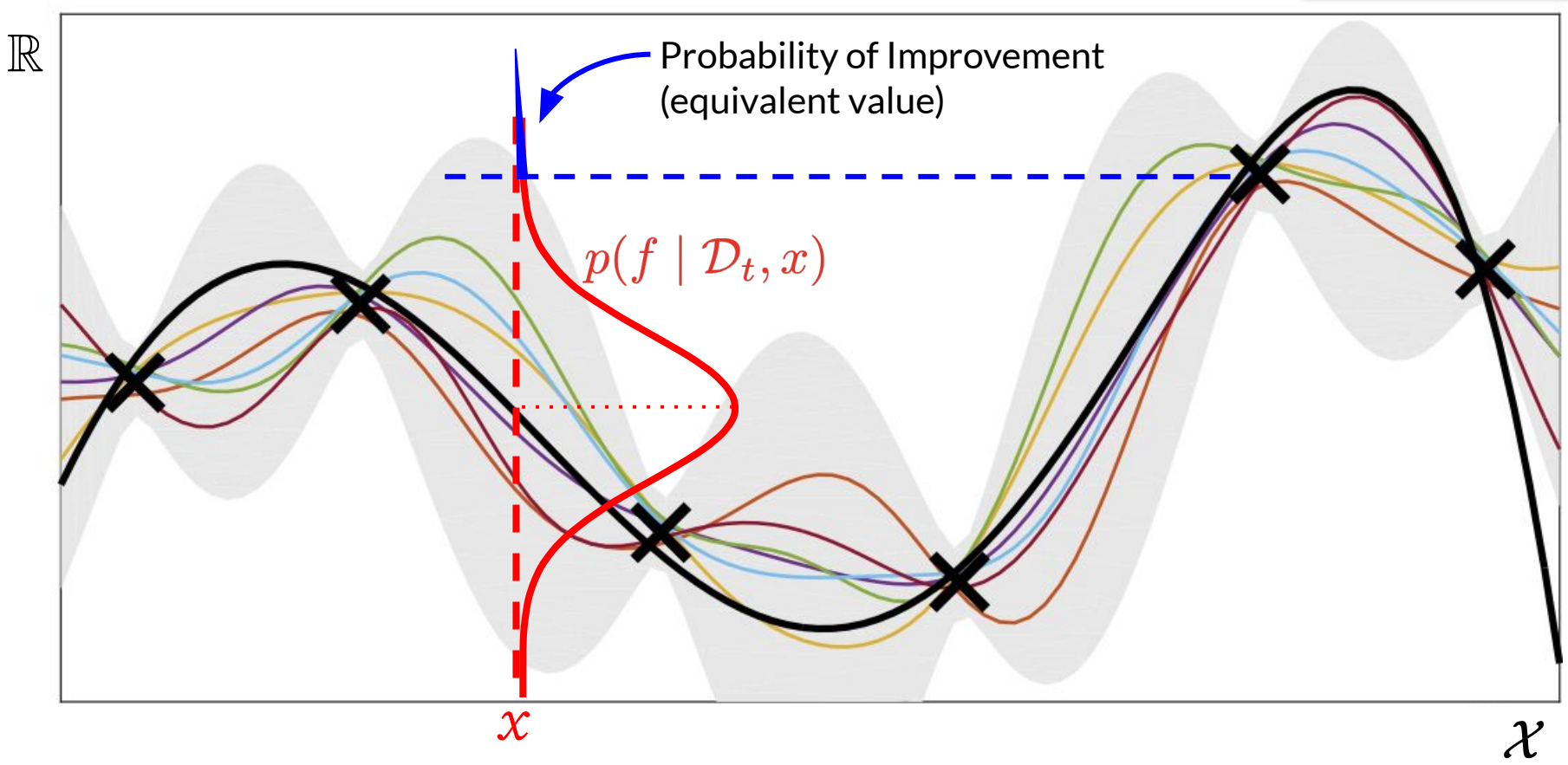
Acquisition Functions — Probability of Improvement (PI)

Visualizing PI



Acquisition Functions — Probability of Improvement (PI)

Visualizing PI



In contrast, we could try to maximize the *expected **amount** of improvement* over the best point so far…

Acquisition Functions — Expected Improvement (EI)

Acquisition Functions — Expected Improvement (EI)

This leads us to one of the most famous and widely used BO acquisition functions (except possibly for UCB).

Acquisition Functions — Expected Improvement (EI)

This leads us to one of the most famous and widely used BO acquisition functions (except possibly for UCB).

The **expected improvement (EI)** acquisition function is written:

Acquisition Functions — Expected Improvement (EI)

This leads us to one of the most famous and widely used BO acquisition functions (except possibly for UCB).

The **expected improvement (EI)** acquisition function is written:

$$\alpha_t(x) = \mathbb{E}_{p(f|\mathcal{D}_t)}\left[\max\{0, f(x) - f(x^*)\}\right]$$

Acquisition Functions — Expected Improvement (EI)

This leads us to one of the most famous and widely used BO acquisition functions (except possibly for UCB).

The **expected improvement (EI)** acquisition function is written:

$$\alpha_t(x) = \mathbb{E}_{p(f|\mathcal{D}_t)}\left[\max\{0, f(x) - f(x^*)\}\right]$$

where $f(x^*)$ is the function value of the best point observed so far in $\mathcal{D}_t$.

Acquisition Functions — Expected Improvement (EI)

This leads us to one of the most famous and widely used BO acquisition functions (except possibly for UCB).

The **expected improvement (EI)** acquisition function is written:

$$\alpha_t(x) = \mathbb{E}_{p(f|\mathcal{D}_t)}\left[\max\{0, f(x) - f(x^*)\}\right]$$

where $f(x^*)$ is the function value of the best point observed so far in $\mathcal{D}_t$.

I.e., the expected amount that $f(x)$ is greater than the best point observed so far (while ignoring the magnitude if $f(x)$ is less than the best point so far).

Acquisition Functions — Expected Improvement (EI)

This leads us to one of the most famous and widely used BO acquisition functions (except possibly for UCB).

The **expected improvement (EI)** acquisition function is written:

$$\alpha_t(x) = \mathbb{E}_{p(f|\mathcal{D}_t)}\left[\max\{0, f(x) - f(x^*)\}\right]$$

where $f(x^*)$ is the function value of the best point observed so far in $\mathcal{D}_t$.

I.e., the expected amount that f(x) is greater than the best point observed so far (while ignoring the magnitude if f(x) is less than the best point so far).

You can prove this is equal to …

Acquisition Functions — Expected Improvement (EI)

This leads us to one of the most famous and widely used BO acquisition functions (except possibly for UCB).

The **expected improvement (EI)** acquisition function is written:

$$\begin{aligned}\alpha_t(x) &= \mathbb{E}_{p(f|\mathcal{D}_t)}\left[\max\{0, f(x) - f(x^*)\}\right] \\ &= (\mu(x) - f(x^*))\Phi\left(\frac{\mu(x) - f(x^*)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\mu(x) - f(x^*)}{\sigma(x)}\right)\end{aligned}$$

Acquisition Functions — Expected Improvement (EI)

This leads us to one of the most famous and widely used BO acquisition functions (except possibly for UCB).

The **expected improvement (EI)** acquisition function is written:

$$\begin{aligned}\alpha_t(x) &= \mathbb{E}_{p(f|\mathcal{D}_t)}\left[\max\{0, f(x) - f(x^*)\}\right] \\ &= (\mu(x) - f(x^*))\Phi\left(\frac{\mu(x) - f(x^*)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\mu(x) - f(x^*)}{\sigma(x)}\end{aligned}$$

where $\Phi(\cdot)$ denotes the CDF of the standard normal distribution and $\phi(\cdot)$ denotes the PDF of the standard normal distribution.

(And, as before, $\mu(x)$ and $\sigma(x)$ are the mean and SD of marginal posterior at x.)

Acquisition Functions — Expected Improvement (EI)

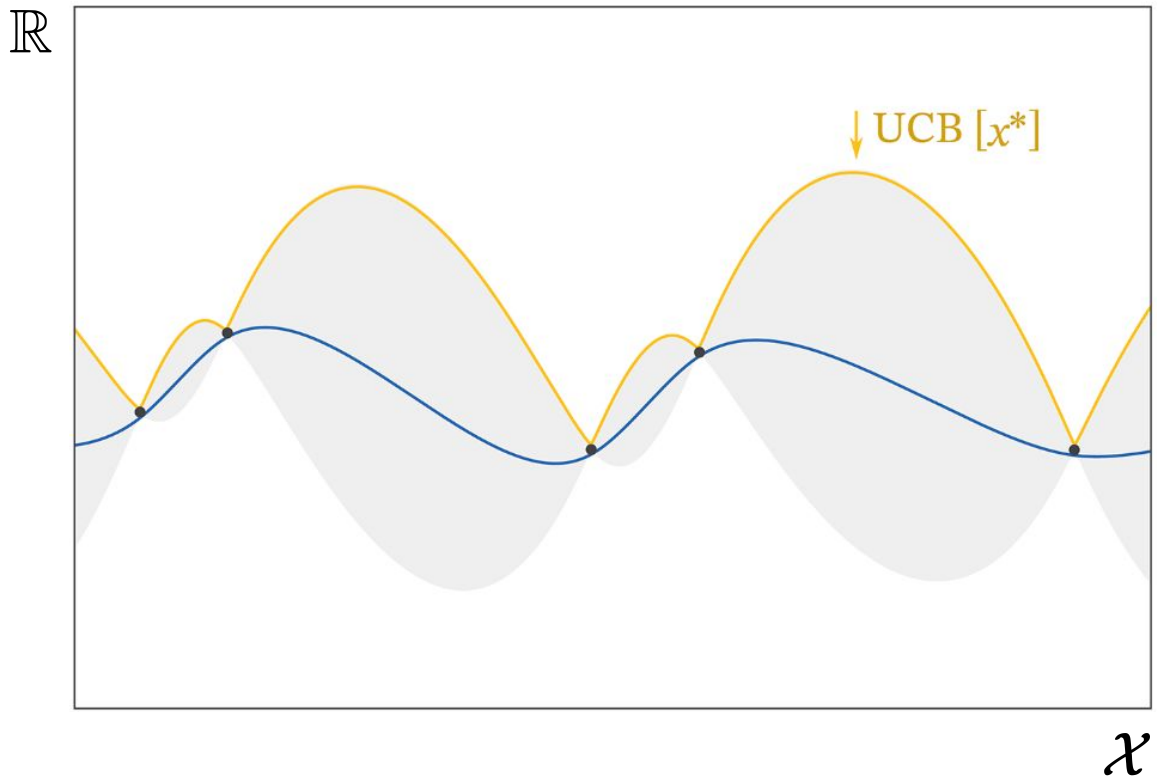
Nice visual comparing UCB *vs.* PI *vs.* EI:

Source: M. O. Ahmed, S. Prince, “Bayesian optimization”

Acquisition Functions — Expected Improvement (EI)

Nice visual comparing UCB *vs.* PI *vs.* EI:

Source: M. O. Ahmed, S. Prince, "Bayesian optimization"

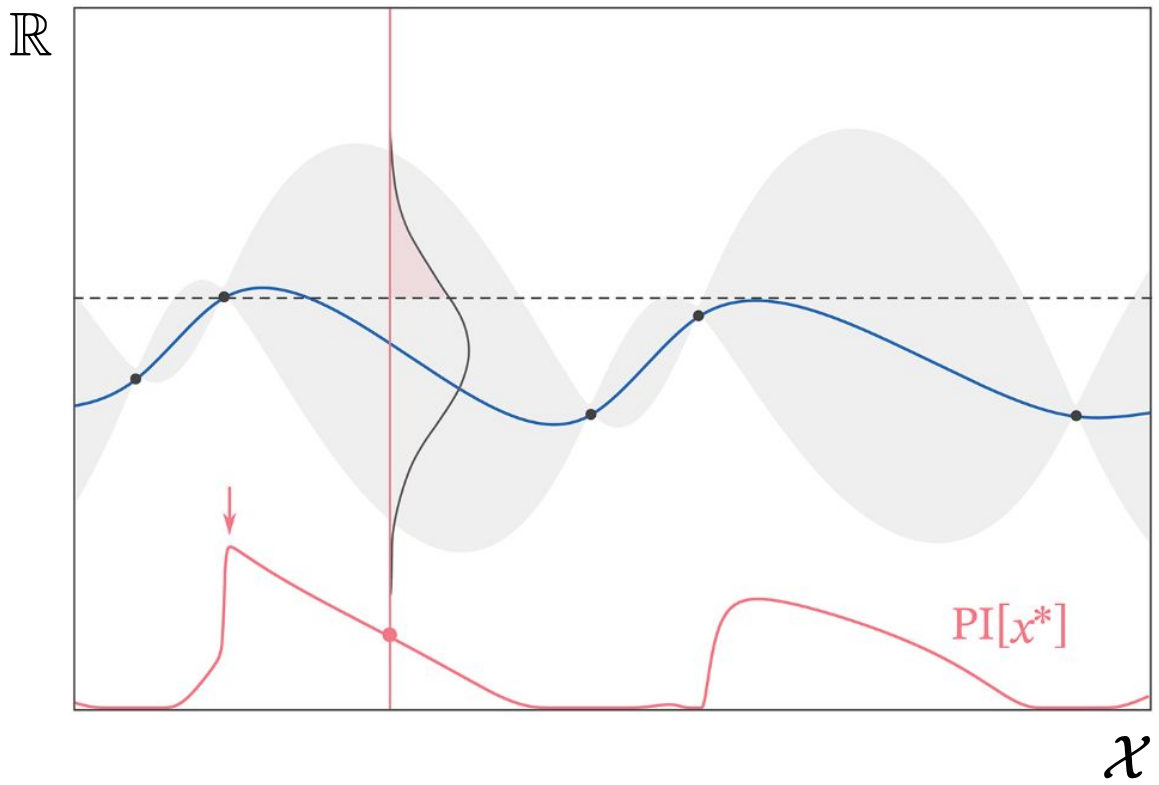


UCB: upper confidence bound of posterior

Acquisition Functions — Expected Improvement (EI)

Nice visual comparing UCB *vs.* PI *vs.* EI:

Source: M. O. Ahmed, S. Prince, "Bayesian optimization"

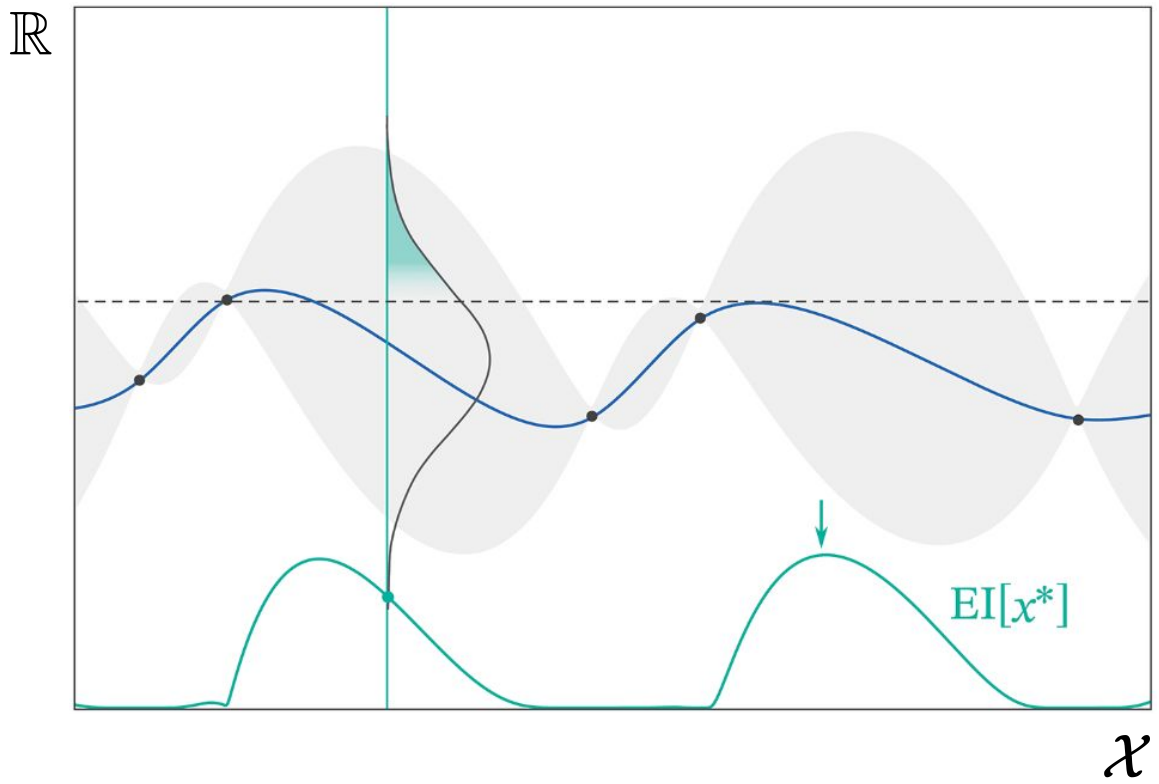


PI: probability that $f(x)$ is greater than best point

Acquisition Functions — Expected Improvement (EI)

Nice visual comparing UCB *vs.* PI *vs.* EI:

Source: M. O. Ahmed, S. Prince, "Bayesian optimization"



EI: *expected amount* $f(x)$ is greater than best point

(and ignoring amount that it's less than best point)

Acquisition Functions — Entropy Search (ES)

Acquisition Functions — Entropy Search (ES)

Another strategy is the “experimental design” strategy, but *applied to BO setting*.

Acquisition Functions — Entropy Search (ES)

Another strategy is the “experimental design” strategy, but *applied to BO setting*.

Specifically, this means:

Reducing posterior uncertainty (entropy) over quantity of interest — in this case, the location x^* of the function maximizer — rather than, e.g., the full landscape of f.

Acquisition Functions — Entropy Search (ES)

Another strategy is the “experimental design” strategy, but *applied to BO setting*.

Specifically, this means:

Reducing posterior uncertainty (entropy) over quantity of interest — in this case, the location x^* of the function maximizer — rather than, e.g., the full landscape of f.

As mentioned, this is an *info-based BO* acquisition function called **entropy search**.

Acquisition Functions — Entropy Search (ES)

Another strategy is the "experimental design" strategy, but *applied to BO setting*.

Specifically, this means:

Reducing posterior uncertainty (entropy) over quantity of interest — in this case, the location x^* of the function maximizer — rather than, e.g., the full landscape of f.

As mentioned, this is an *info-based BO* acquisition function called **entropy search**.

Why would you want to use this acquisition function?

⇒ Intuitively: if you get rewarded for *how good your guess of* x^* *is* — rather than, the value of the maximizer, $f(x^*)$ — ES should be more optimal.

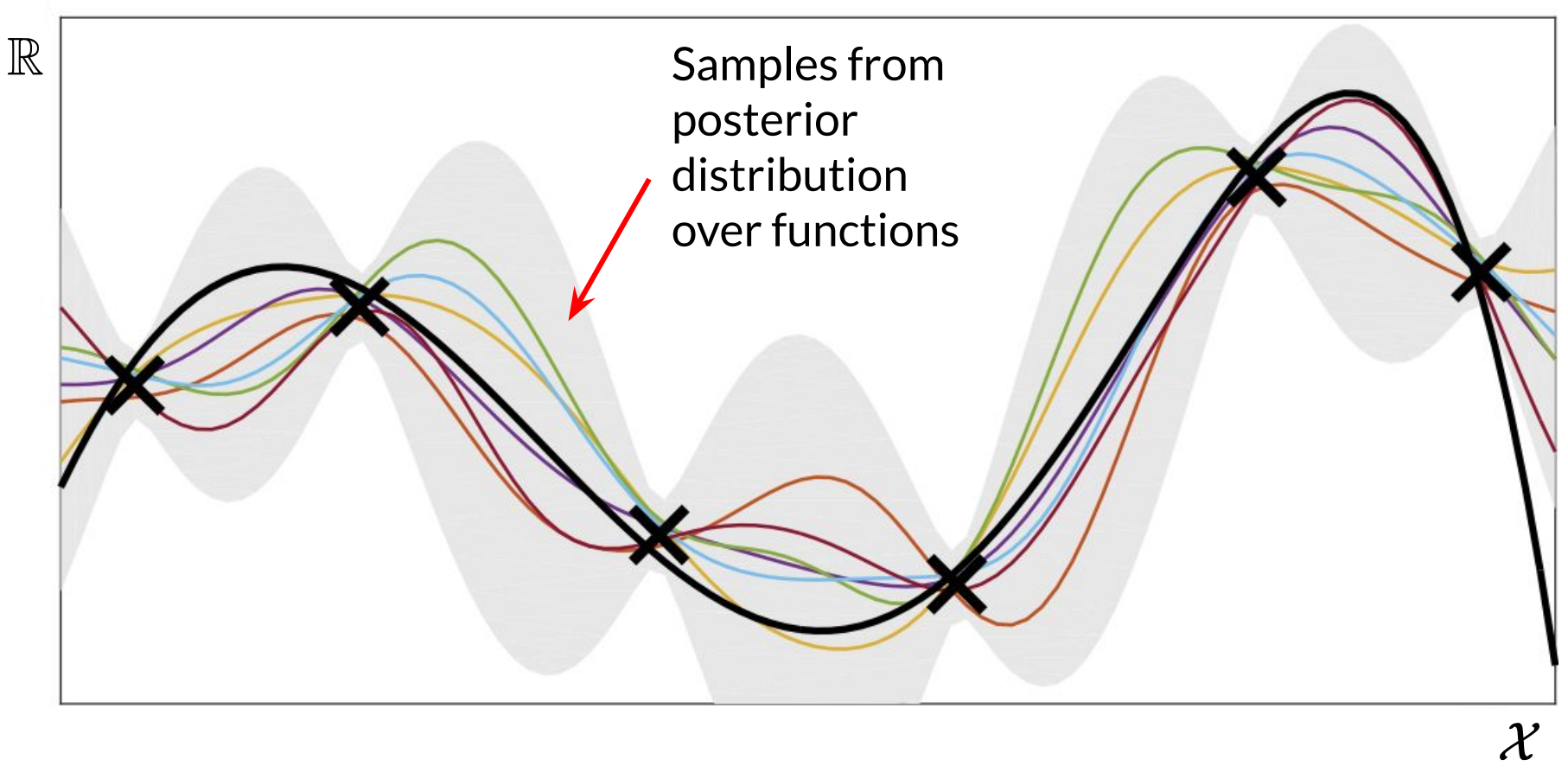
Acquisition Functions — Entropy Search

Visualizing ES

To explain this acquisition function, let's visualize…

Acquisition Functions — Entropy Search

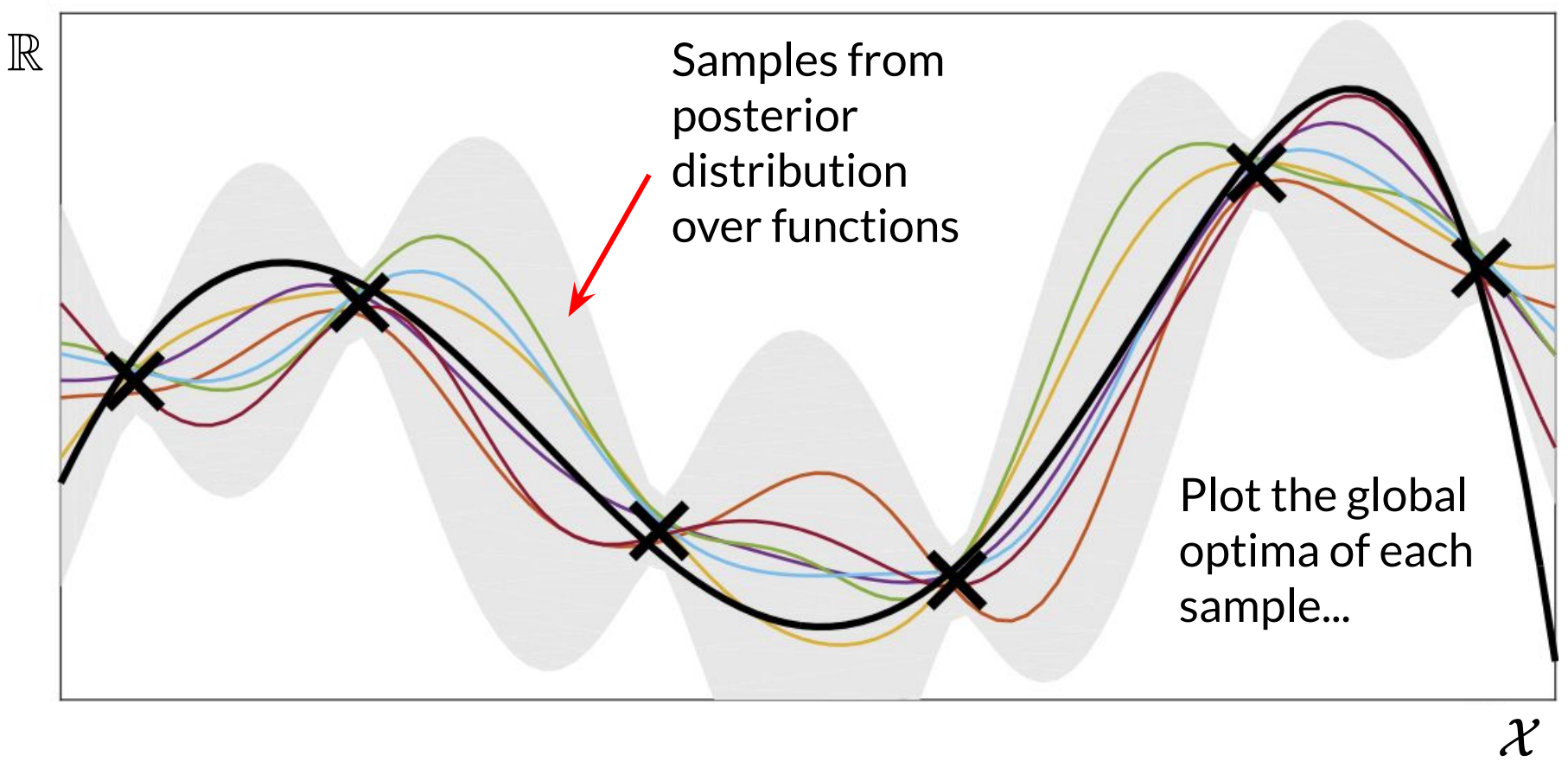
Visualizing ES



Probabilistic model of f, given dataset.

Acquisition Functions — Entropy Search

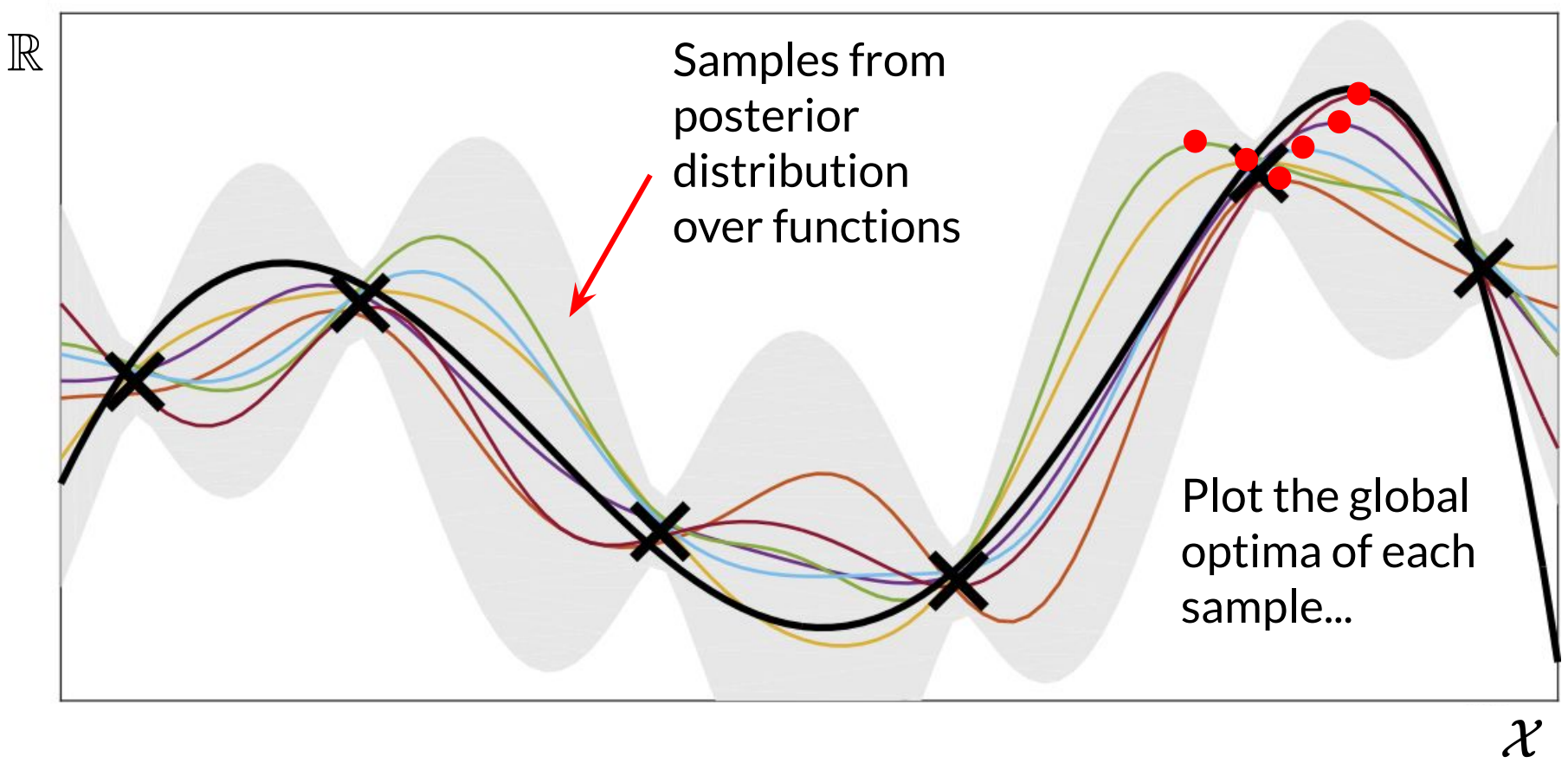
Visualizing ES



Probabilistic model of f, given dataset.

Acquisition Functions — Entropy Search

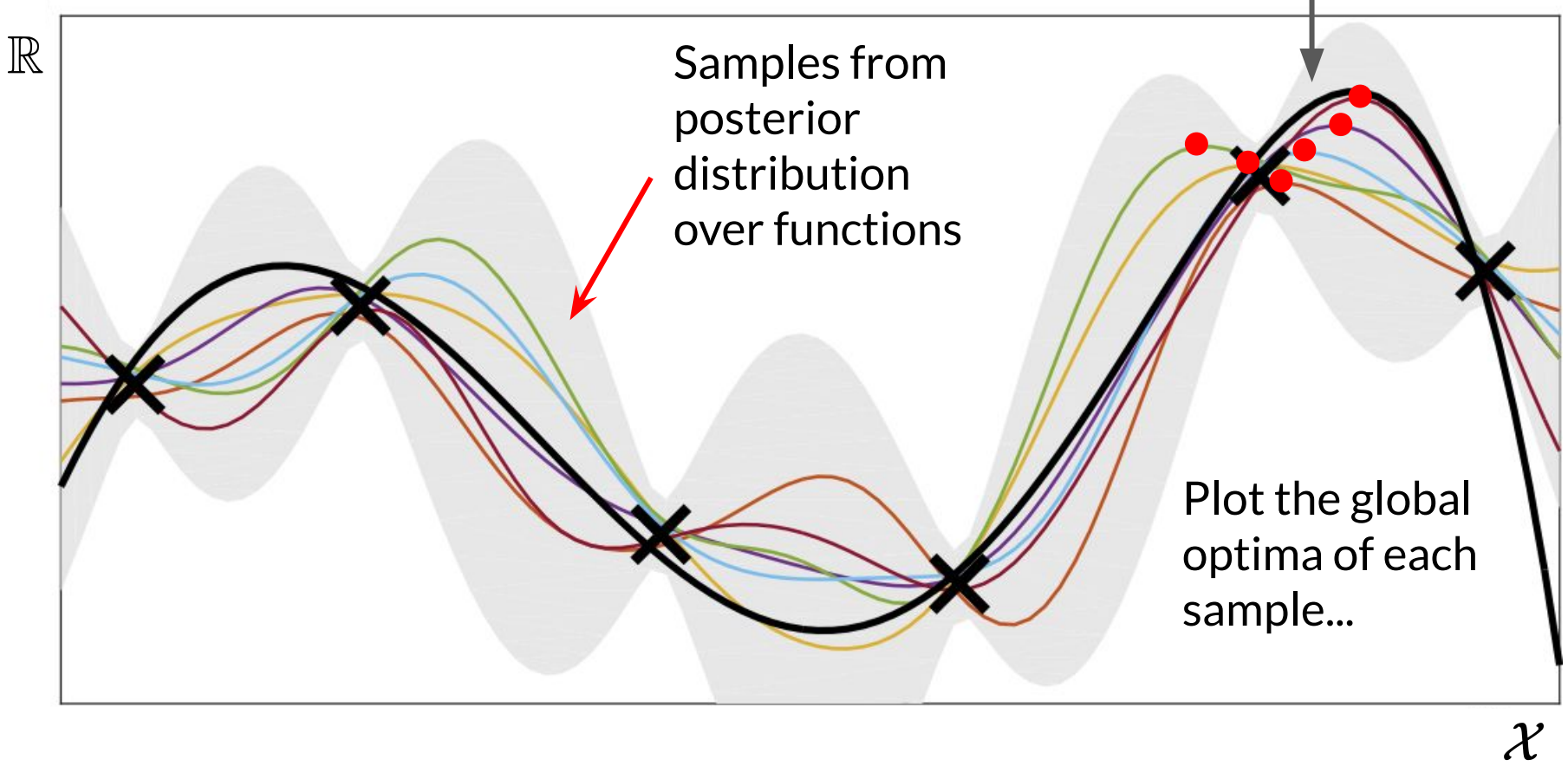
Visualizing ES



Probabilistic model of f, given dataset.

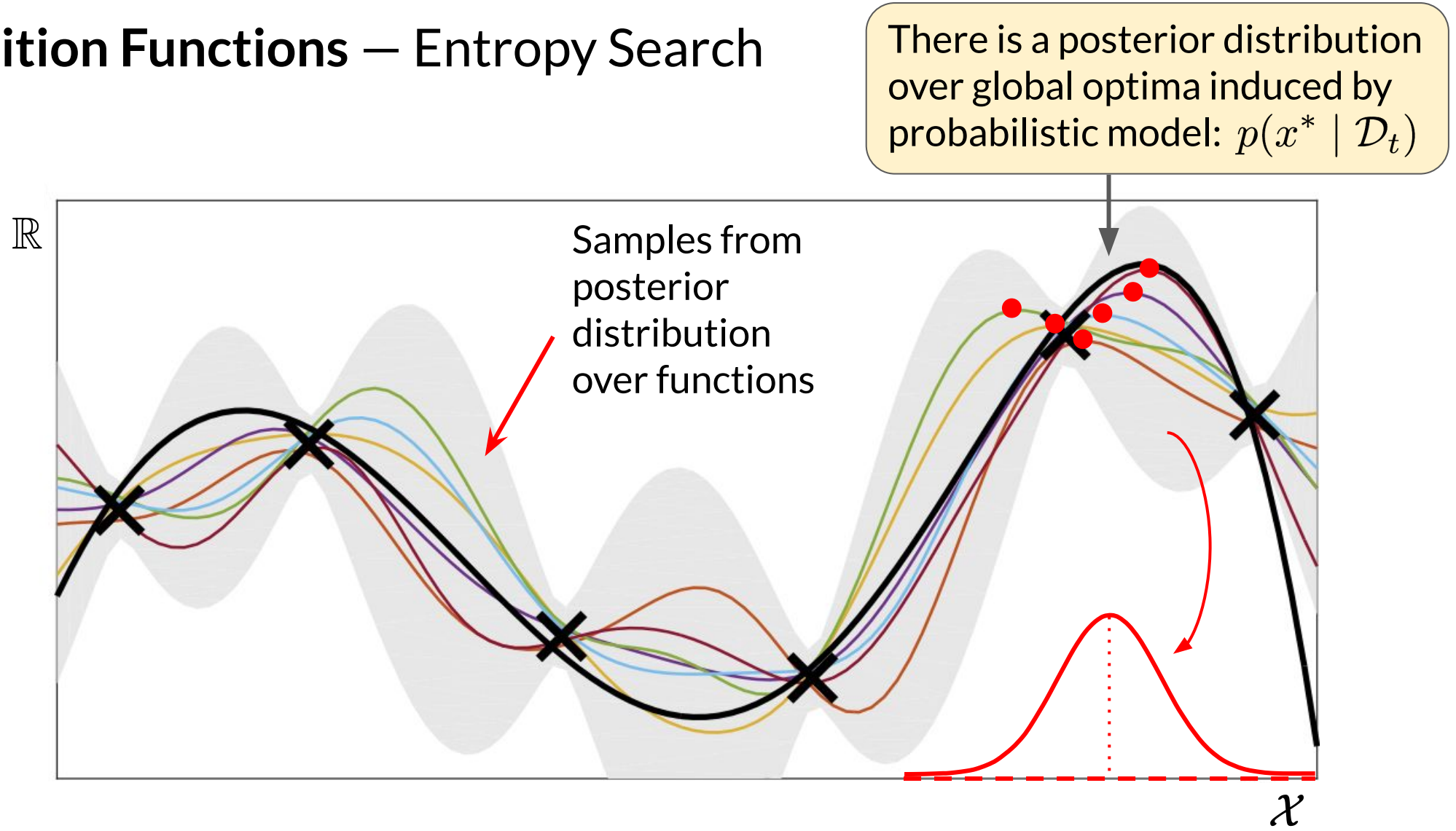
Acquisition Functions — Entropy Search

There is a posterior distribution over global optima induced by probabilistic model: $p(x^* \mid \mathcal{D}_t)$



Probabilistic model of f, given dataset.

Acquisition Functions — Entropy Search



Probabilistic model of f, given dataset.

Acquisition Functions — Entropy Search

This leads us to the acquisition function:

e.g. used in entropy search (ES), predictive entropy search (PES)

"Entropy Search for Information-Efficient Global Optimization", Hennig and Schuler, **JMLR 2012**

"Predictive Entropy Search for Efficient Global Optimization of Black-box Functions", Hernández-Lobato, Hoffman, Ghahramani, **NeurIPS 2014**

Acquisition Functions — Entropy Search

This leads us to the acquisition function:

e.g. used in entropy search (ES), predictive entropy search (PES)

$$\alpha_t(x) = H[p(x^* \mid \mathcal{D}_t)] - \mathbb{E}_{p(y_x \mid \mathcal{D}_t)}[H[p(x^* \mid \mathcal{D}_t \cup \{(x, y_x)\})]]$$

entropy of posterior distribution over global optima x^*

minus

expected entropy of posterior distribution over global optima x^*,
… if we were to make a query at x

"Entropy Search for Information-Efficient Global Optimization", Hennig and Schuler, **JMLR 2012**

"Predictive Entropy Search for Efficient Global Optimization of Black-box Functions", Hernández-Lobato, Hoffman, Ghahramani, **NeurIPS 2014**

Acquisition Functions — Entropy Search

This leads us to the acquisition function:

e.g. used in entropy search (ES), predictive entropy search (PES)

$$\alpha_t(x) = H[p(x^* \mid \mathcal{D}_t)] - \mathbb{E}_{p(y_x|\mathcal{D}_t)}[H[p(x^* \mid \mathcal{D}_t \cup \{(x, y_x)\})]]$$

entropy of posterior distribution over global optima x^*

minus

expected entropy of posterior distribution over global optima x^*,
… if we were to make a query at x

"Entropy Search for Information-Efficient Global Optimization", Hennig and Schuler, **JMLR 2012**
"Predictive Entropy Search for Efficient Global Optimization of Black-box Functions", Hernández-Lobato, Hoffman, Ghahramani, **NeurIPS 2014**

Acquisition Functions — Entropy Search

This leads us to the acquisition function:

e.g. used in entropy search (ES), predictive entropy search (PES)

$$\alpha_t(x) = H[p(x^* \mid \mathcal{D}_t)] - \mathbb{E}_{p(y_x \mid \mathcal{D}_t)}[H[p(x^* \mid \mathcal{D}_t \cup \{(x, y_x)\})]]$$

entropy of posterior distribution over global optima x^*

minus

expected entropy of posterior distribution over global optima x^*,
… if we were to make a query at x

"Entropy Search for Information-Efficient Global Optimization", Hennig and Schuler, **JMLR 2012**

"Predictive Entropy Search for Efficient Global Optimization of Black-box Functions", Hernández-Lobato, Hoffman, Ghahramani, **NeurIPS 2014**

Acquisition Functions — Entropy Search

This leads us to the acquisition function:

e.g. used in entropy search (ES), predictive entropy search (PES)

$$\alpha_t(x) = H[p(x^* \mid \mathcal{D}_t)] - \mathbb{E}_{p(y_x \mid \mathcal{D}_t)}[H[p(x^* \mid \mathcal{D}_t \cup \{(x, y_x)\})]]$$

entropy of posterior distribution over global optima x^*

minus

expected entropy of posterior distribution over global optima x^*,
… if we were to make a query at x

"Entropy Search for Information-Efficient Global Optimization", Hennig and Schuler, **JMLR 2012**

"Predictive Entropy Search for Efficient Global Optimization of Black-box Functions", Hernández-Lobato, Hoffman, Ghahramani, **NeurIPS 2014**

Acquisition Functions — Entropy Search

This leads us to the acquisition function:

e.g. used in entropy search (ES), predictive entropy search (PES)

$$\alpha_t(x) = H[p(x^* \mid \mathcal{D}_t)] - \mathbb{E}_{p(y_x|\mathcal{D}_t)}[H[p(x^* \mid \mathcal{D}_t \cup \{(x, y_x)\})]]$$

entropy of posterior distribution over global optima x^*

minus

expected entropy of posterior distribution over global optima x^*,
… if we were to make a query at x

"Entropy Search for Information-Efficient Global Optimization", Hennig and Schuler, **JMLR 2012**
"Predictive Entropy Search for Efficient Global Optimization of Black-box Functions", Hernández-Lobato, Hoffman, Ghahramani, **NeurIPS 2014**

Acquisition Functions — Entropy Search

This leads us to the acquisition function:

e.g. used in entropy search (ES), predictive entropy search (PES)

$$\alpha_t(x) = H[p(x^* \mid \mathcal{D}_t)] - \mathbb{E}_{p(y_x \mid \mathcal{D}_t)}[H[p(x^* \mid \mathcal{D}_t \cup \{(x, y_x)\})]]$$

entropy of posterior distribution over global optima x^*

minus

expected entropy of posterior distribution over global optima x^*,
… if we were to make a query at x

Expected information gain (EIG) — expected decrease in entropy if we were to query f at x.

Acquisition Functions — Additional

Acquisition Functions — Additional

A few acquisition functions that are useful for further study:

Acquisition Functions — Additional

A few acquisition functions that are useful for further study:

- **Knowledge gradient** — somewhat complex, but one of the *most-optimal* (both theoretically and in practice) BO acquisition functions.

Acquisition Functions — Additional

A few acquisition functions that are useful for further study:

- **Knowledge gradient** — somewhat complex, but one of the *most-optimal* (both theoretically and in practice) BO acquisition functions.
- **Thompson sampling** — a stochastic acquisition function (*i.e.*, draw one posterior, and use it within an acquisition function)

Acquisition Functions — Additional

A few acquisition functions that are useful for further study:

- **Knowledge gradient** — somewhat complex, but one of the *most-optimal* (both theoretically and in practice) BO acquisition functions.
- **Thompson sampling** — a stochastic acquisition function (*i.e.*, draw one posterior, and use it within an acquisition function)
- **Non-BO acq fns.** — acquisition functions for non-optimization tasks, such as level sets, quadrature, phase boundaries, and more-general tasks.

Next Time

Next Time

Final presentations! Groups 1, 3, 9.