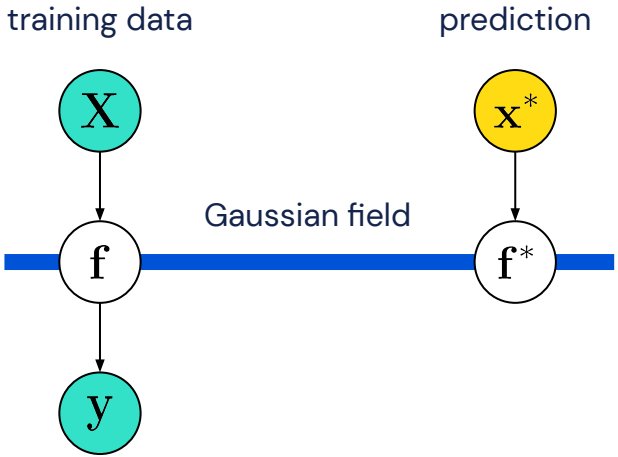
Sparse Orthogonal Variational Inference for Gaussian Processes

Jiaxin Shi (Tsinghua University)

Michalis Titsias, Andriy Mnih (DeepMind)

Gaussian Processes



$$\mathbf{f} = f(\mathbf{X}) := [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$$

Definition

mean function

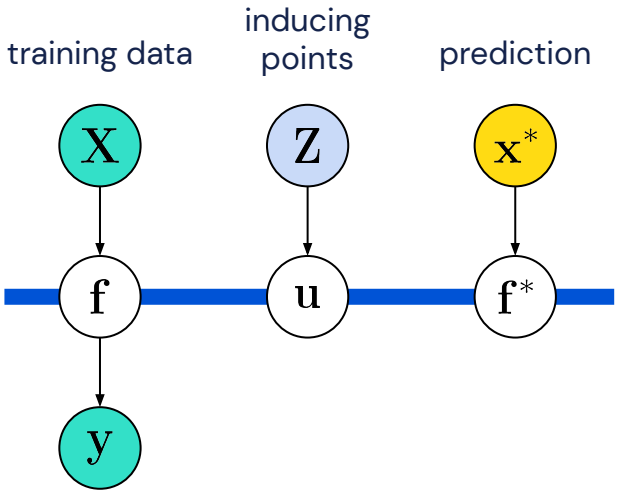
$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$$

covariance function / kernel

$$p(\mathbf{f}, \mathbf{f}^*) := \mathcal{N}\left(\begin{bmatrix}\mathbf{f} \\ \mathbf{f}^*\end{bmatrix} \middle| \begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{x}^*)\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{\mathbf{ff}} & \mathbf{K}_{\mathbf{f}*} \\ \mathbf{K}_{*\mathbf{f}} & \mathbf{K}_{**}\end{bmatrix}\right)$$

$\mathcal{O}(N^3)$ complexity

Inducing Points



$$\mathbf{u} = f(\mathbf{Z}) := [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_M)]^\top$$

Sparse variational GP methods (SVGP)

Titsias (2009), Hensman et al. (2013 & 2015)

- Augmented joint distribution:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}, \mathbf{u})$$

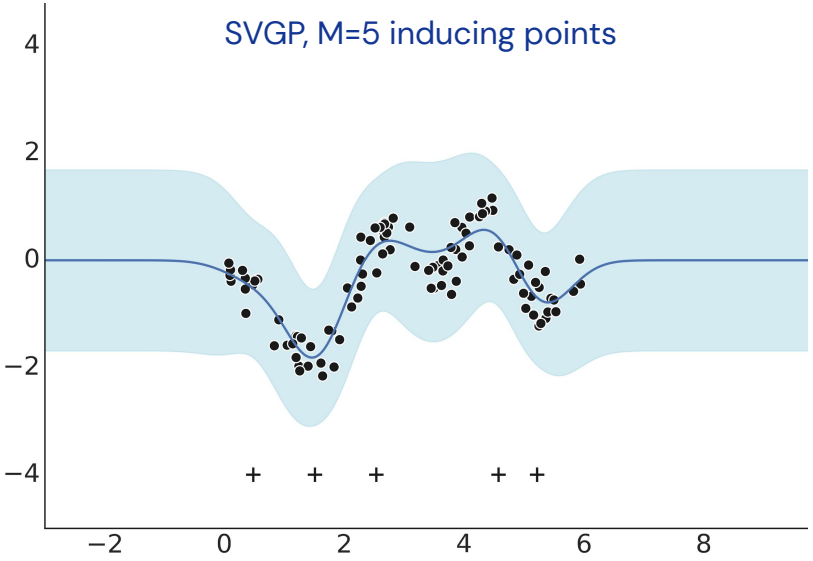
- Variational distribution:

$$q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$$

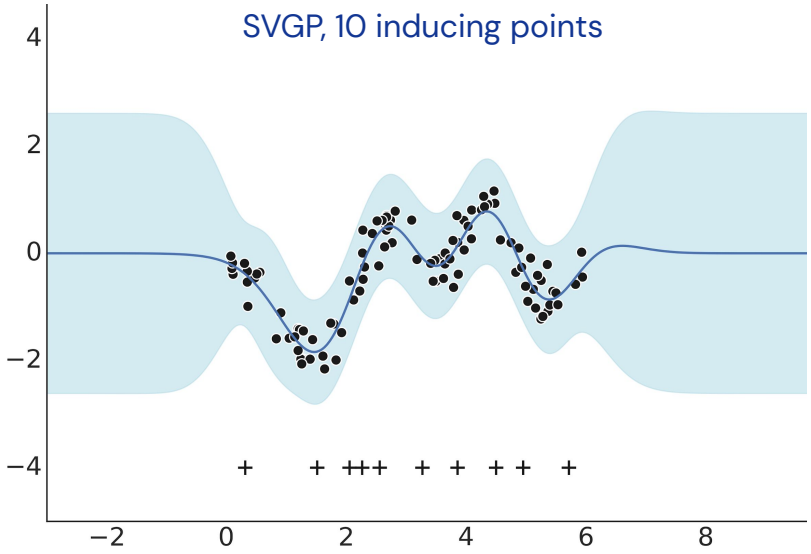
$$\mathrm{KL}\left[q(\mathbf{f}, \mathbf{u}) || p(\mathbf{f}, \mathbf{u}|\mathbf{y})\right] = \mathbb{E}_q \log \frac{p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \cdot p(\mathbf{y})}{p(\mathbf{y}|\mathbf{f}) \cdot p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}$$

$\mathcal{O}(M^3)$ complexity per update

SVGP: Inducing Points Are Expensive

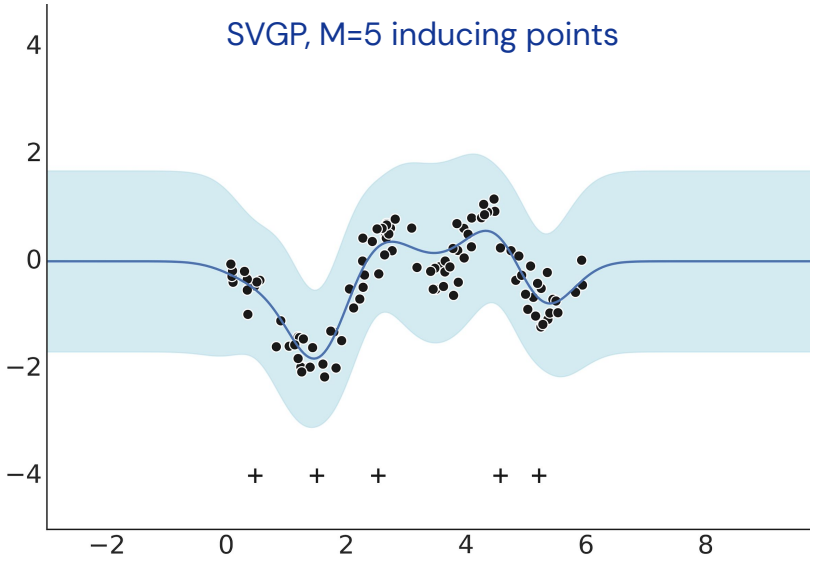


Cholesky: cM^3

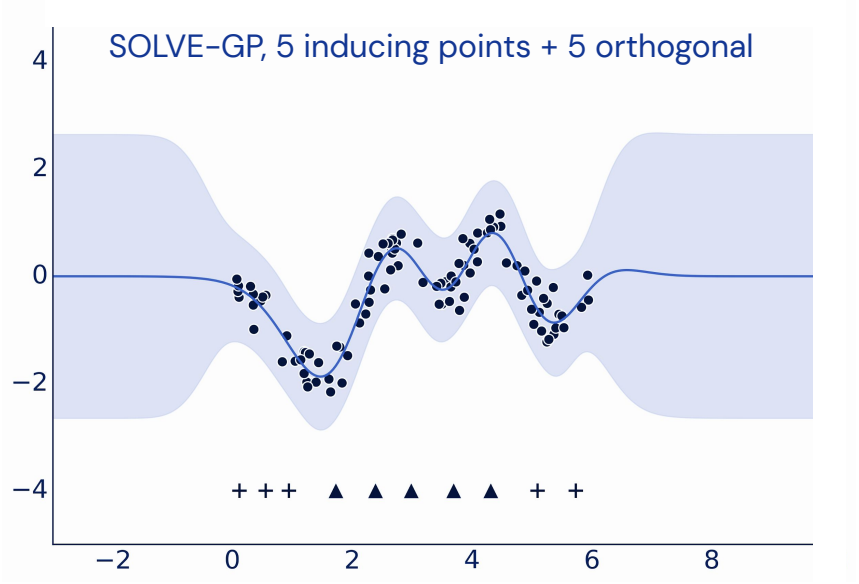


Cholesky: $8cM^3$

SOLVE-GP: Orthogonal Inducing Points Are Cheaper



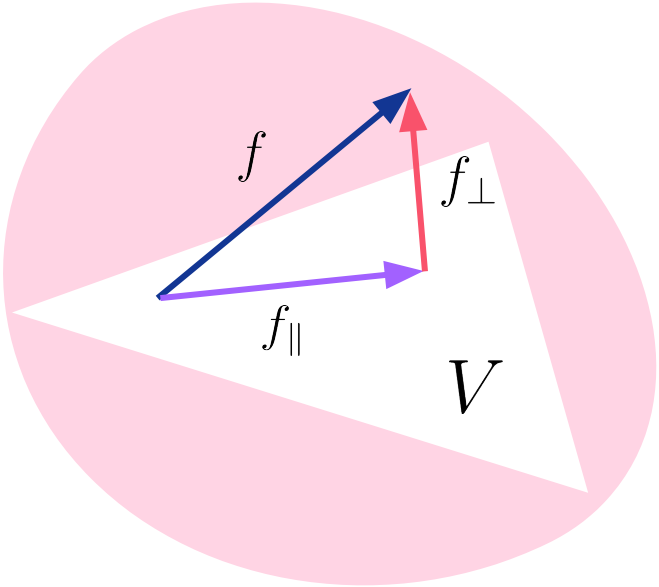
Cholesky: cM^3

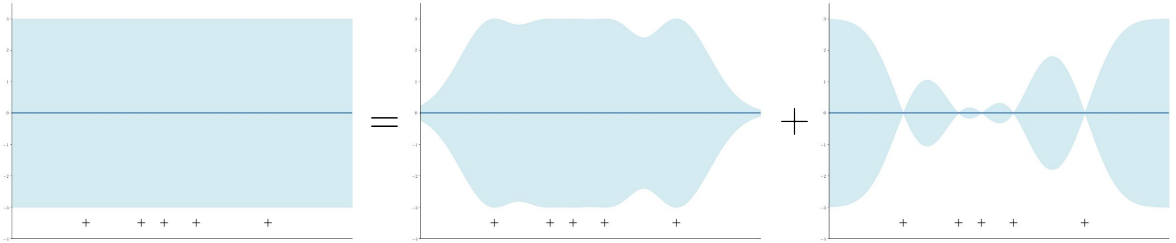


Cholesky: $2cM^3$

+ Inducing points ▲ Orthogonal inducing points

Orthogonal Decomposition





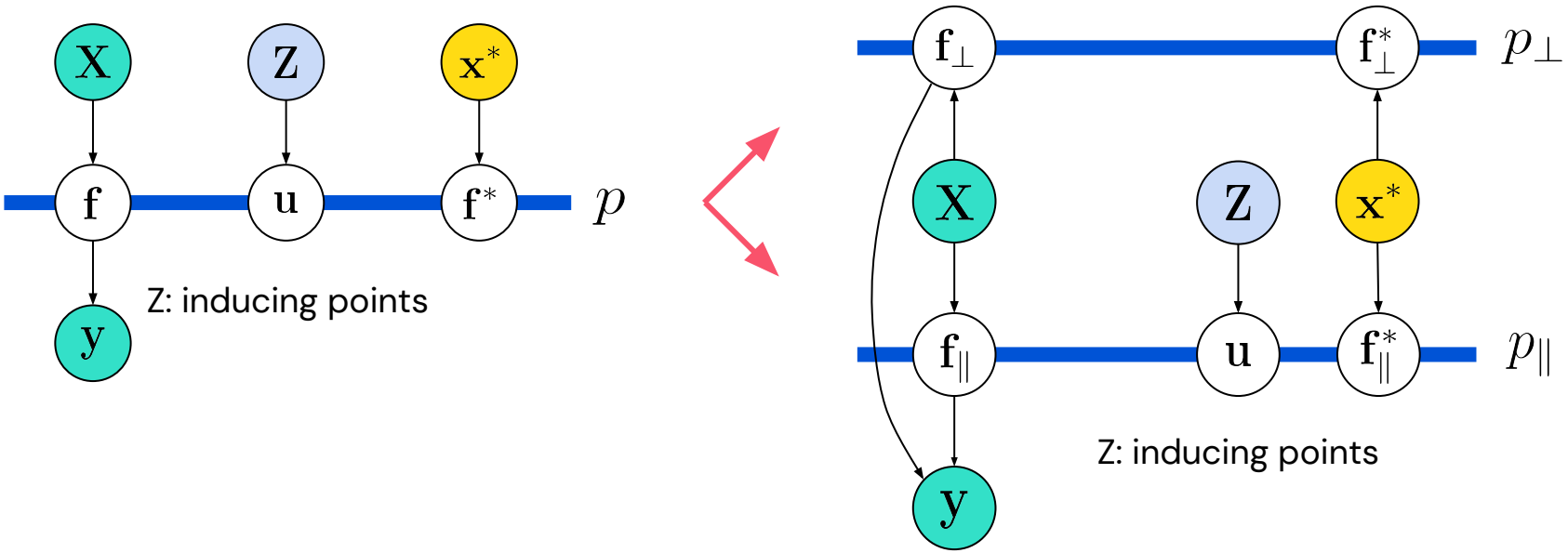
$$V = \left\{ \sum_{i=1}^{M} a_i k(\mathbf{z}_i, \cdot) \;\middle|\; \mathbf{a} \in \mathbb{R}^M \right\}$$

$$p_\parallel : f_\parallel = \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{u} \sim \mathcal{GP}(0, \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{k_u}(\mathbf{x}'))$$

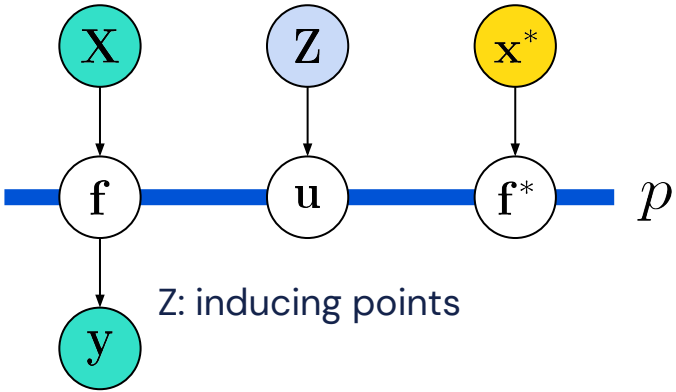
$$p_\perp : f_\perp \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}') - \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{k_u}(\mathbf{x}'))$$

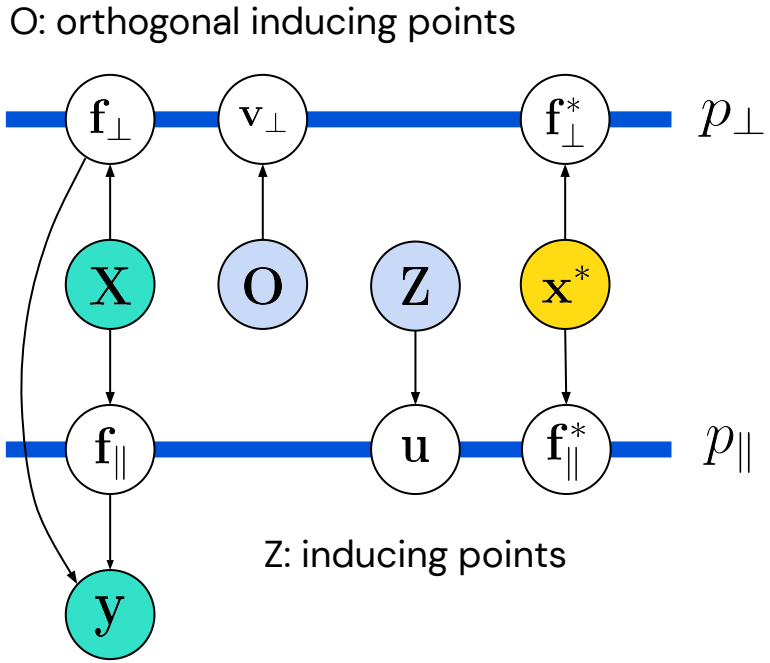
$$p : f = f_\perp + f_\parallel \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$

Orthogonal Inducing Points

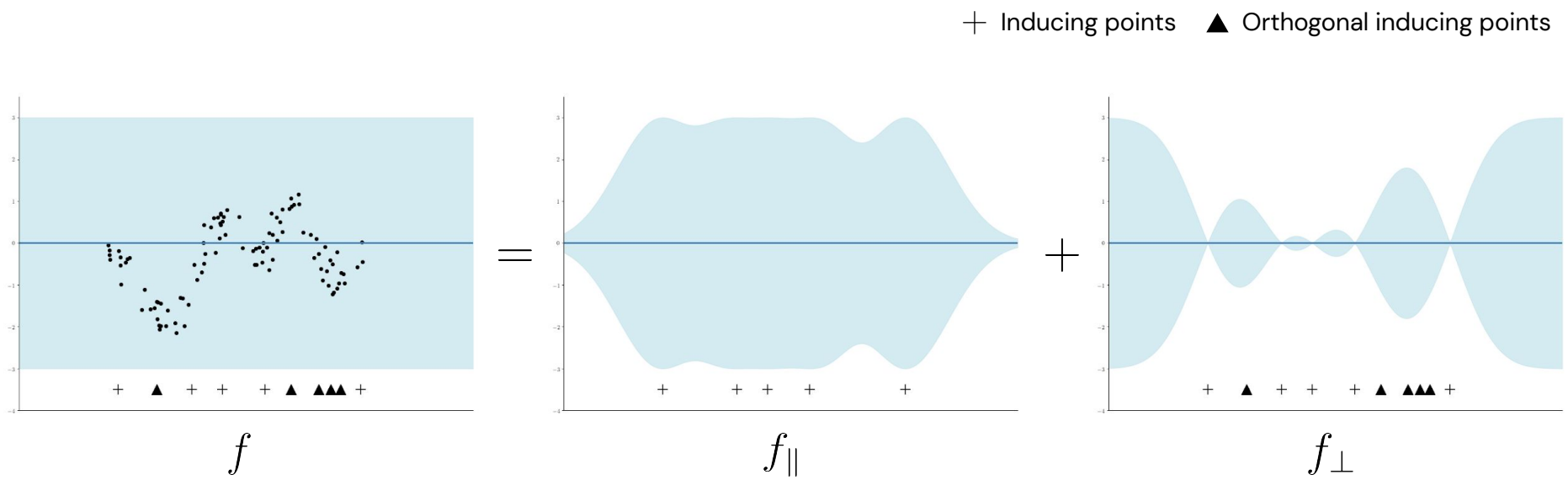


Orthogonal Inducing Points





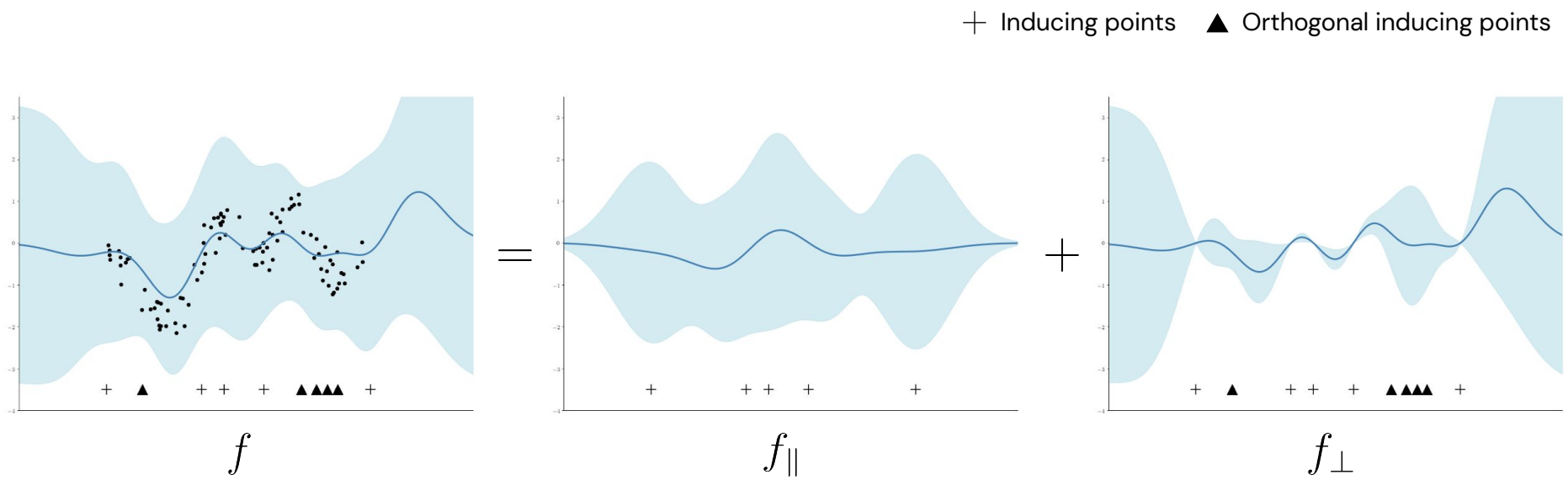
Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

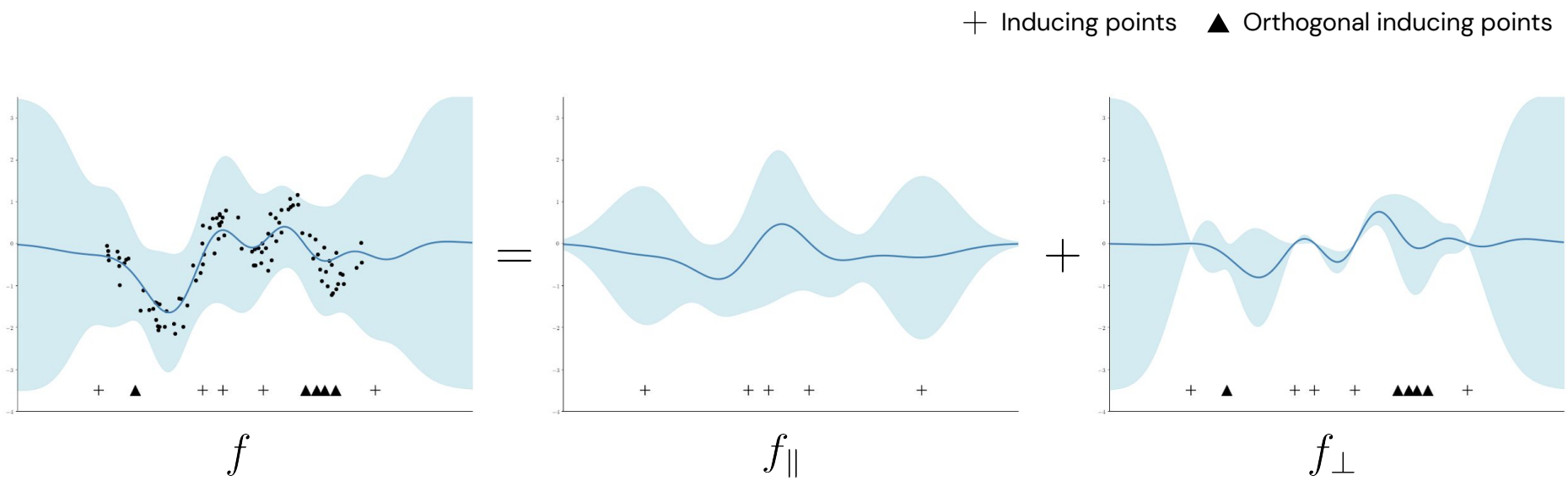
Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

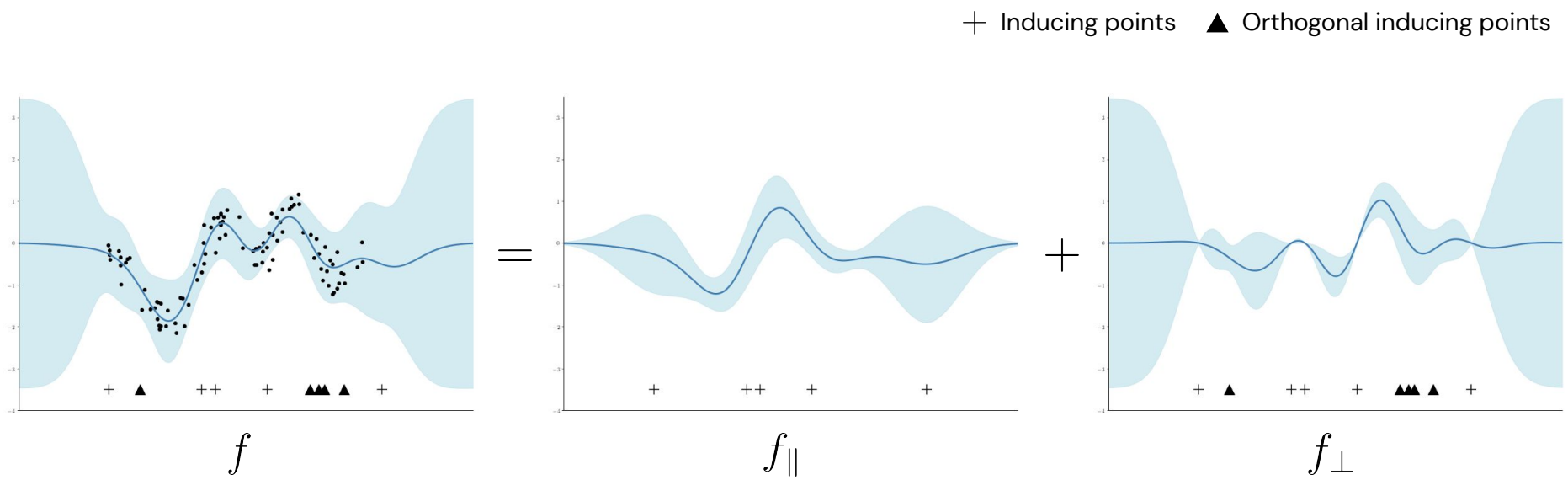
Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

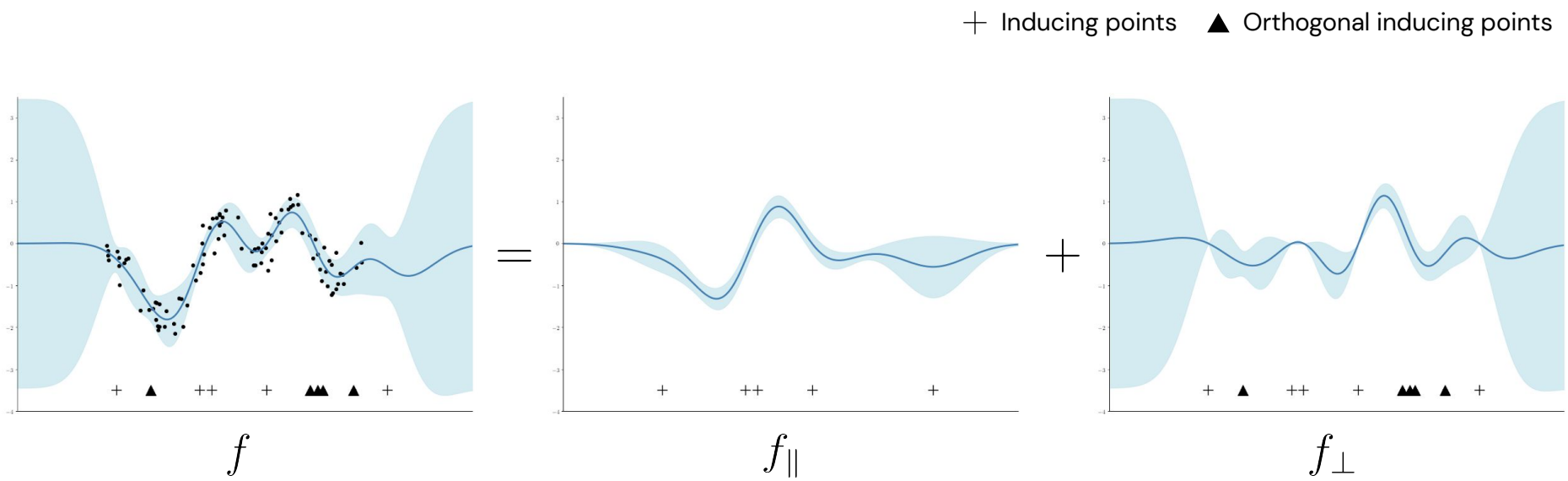
Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

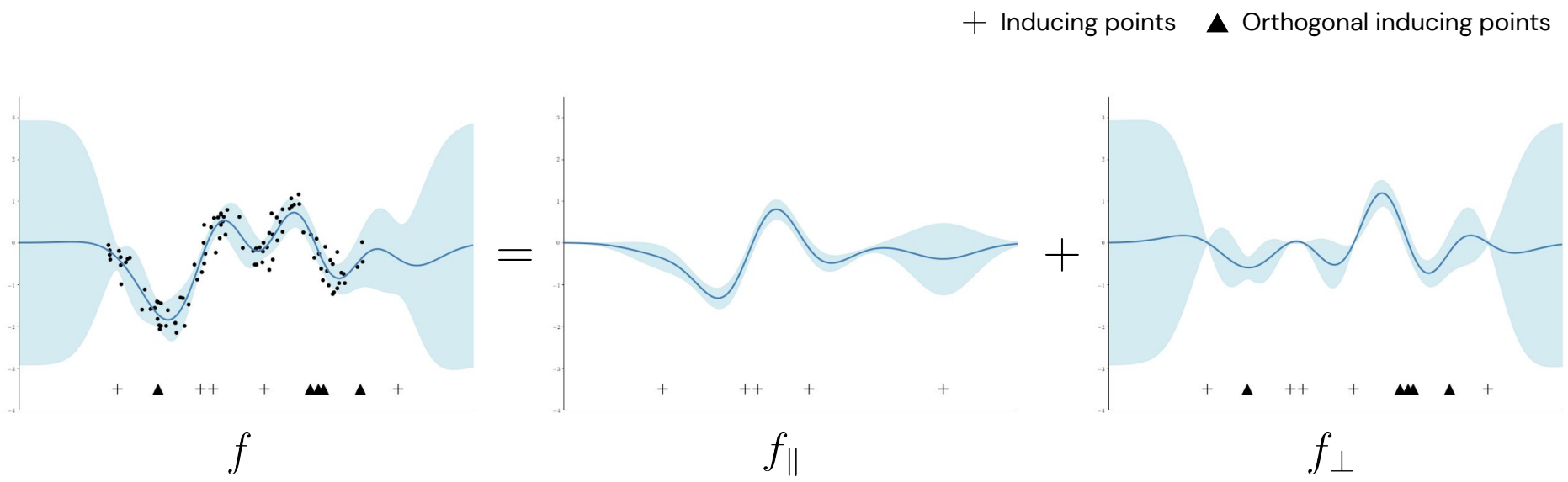
Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

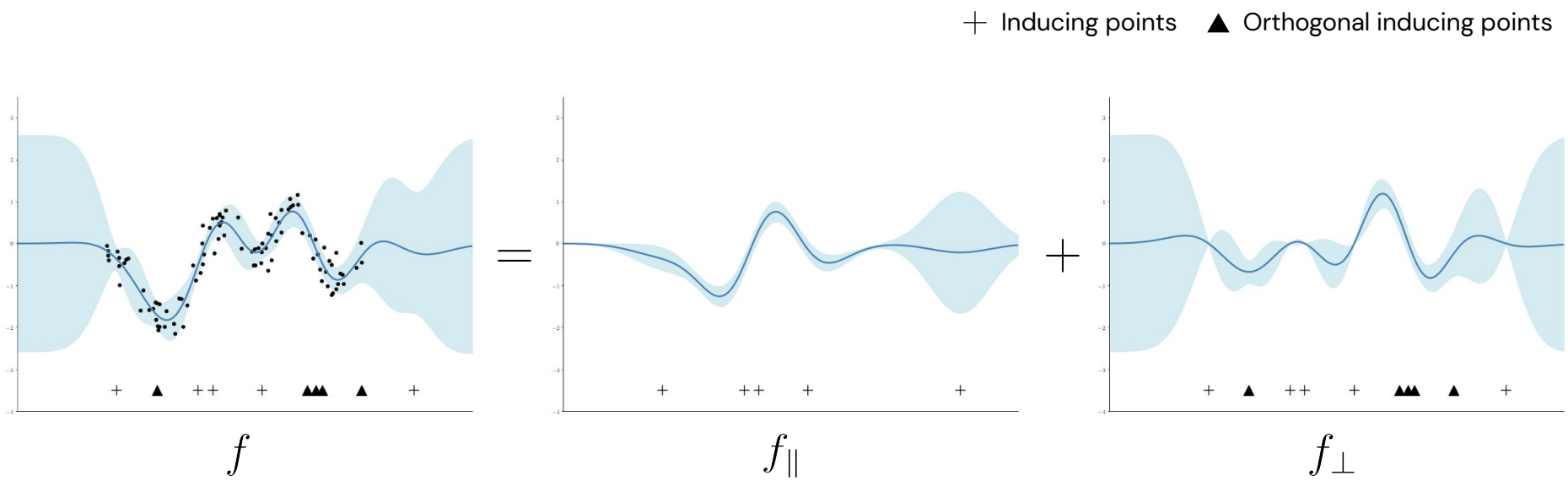
Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

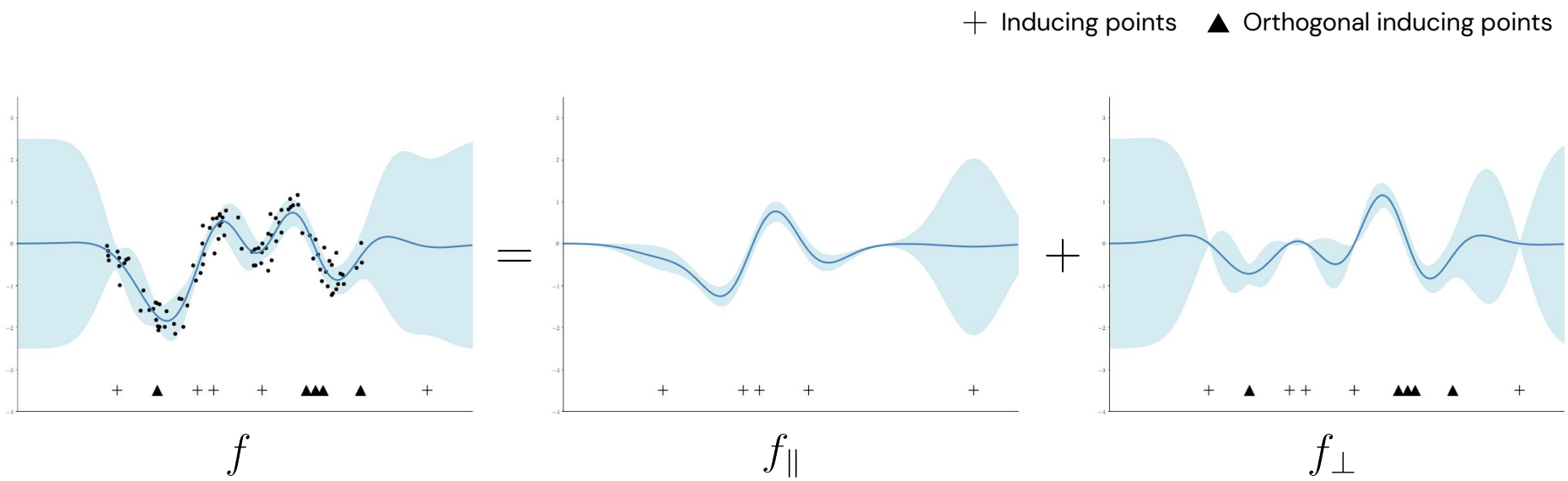
Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

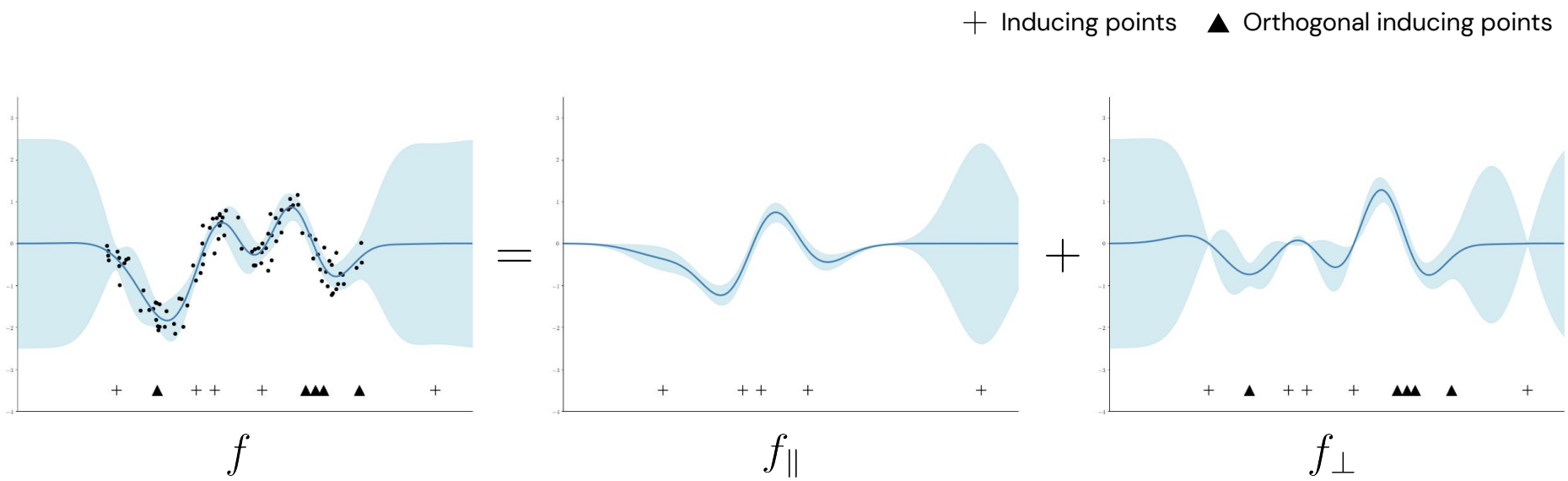
Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

Posterior Approximation during Training



SOLVE-GP lower bound

$$\mathbb{E}_{q(\mathbf{u})q(\mathbf{v}_\perp)p_\perp(\mathbf{f}_\perp|\mathbf{v}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{v}_\perp)\|p_\perp(\mathbf{v}_\perp)\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

CIFAR 10 Classification

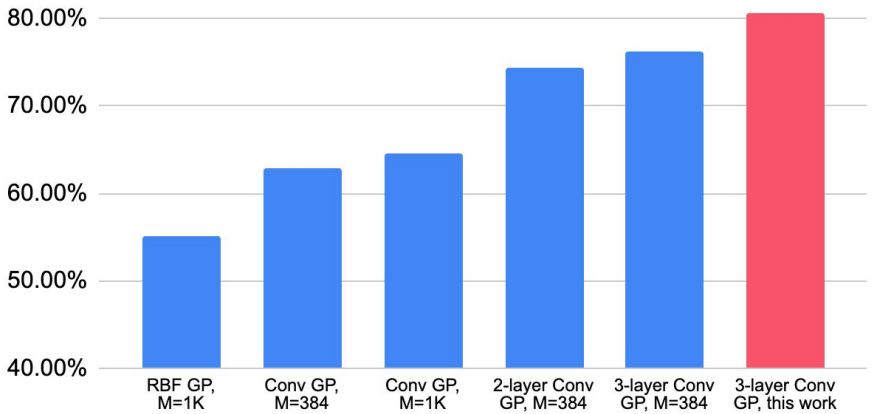
Convolutional Gaussian Process

new SOTA: 64.6% -> 68.2%

Methods	SVGP		SOLVE-GP	SVGP
M	1K	1.6K	1K+1K	2K*
Test LL	-1.59	-1.54	**-1.51**	**-1.48**
Test Acc	66.07%	67.18%	**68.19%**	68.06%
Time /iter	0.241	0.380	0.370	0.474

Deep Convolutional Gaussian Process

new SOTA: 76.2% -> 80.3%



- No neural networks; no data augmentation.
- Better results compared to exact GPs derived from infinite-width neural networks: CNN-GP 67.1% (Novak et al., 2019); CNTK 77.4% (Arora et al., 2019).

Skoltech

Skolkovo Institute of Science and Technology

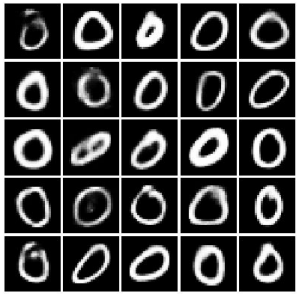
BooVAE: A scalable framework for continual VAE learning under boosting approach

Anna Kuzina* and Evgenii Egorov*,
Evgeny Burnaev

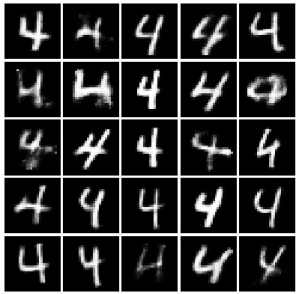
*Equally contributed

Problem: how to train VAEs incrementally?

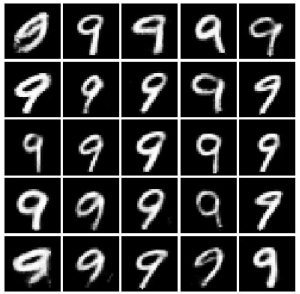
1 task



5 tasks

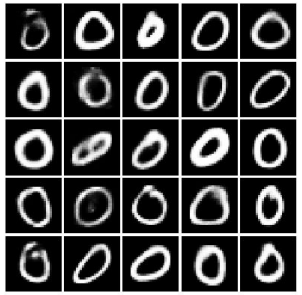


10 tasks

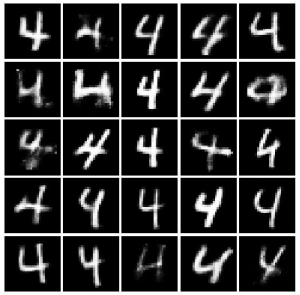


Problem: how to train VAEs incrementally?

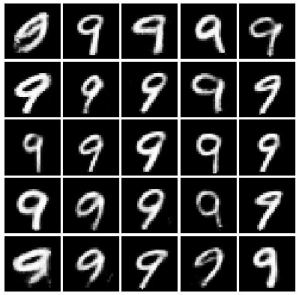
1 task



5 tasks



10 tasks

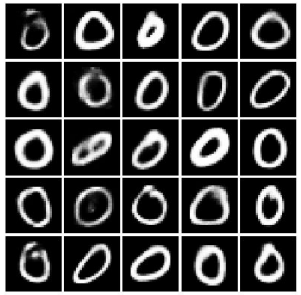


Current Methods

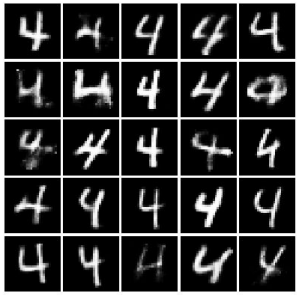
Regularization	(EWC)
Model Expansion	(Multihead architectures)
Rehearsal	(Generative replay)

Problem: how to train VAEs incrementally?

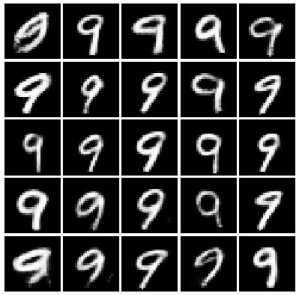
1 task



5 tasks



10 tasks



Current Methods

Regularization	(EWC)
Model Expansion	(Multihead architectures)
Rehearsal	(Generative replay)

BooVAE

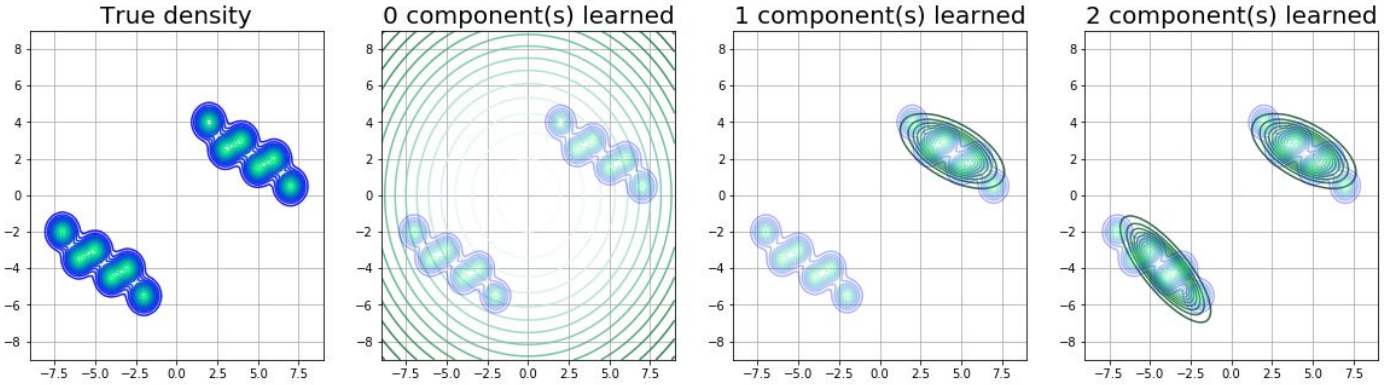
- ✓ Expand the prior, keeping the architecture *fixed*
- ✓ Knowledge of task boundaries is *not required*
- ✓ Can be *combined* with your favourite regularization method

Boosting: greedy density approximation

Approximate density by a mixture

$$p^*(x) \approx p_K(x)$$

$$p_K(x) = \sum_{i=1}^{K} \alpha_i q_i(x)$$

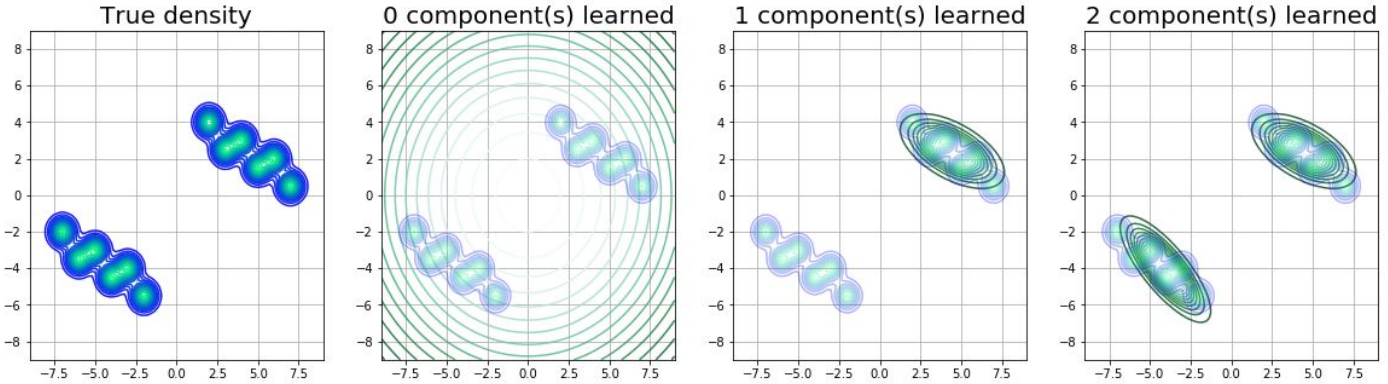


Boosting: greedy density approximation

Approximate density by a mixture

$$p^*(x) \approx p_K(x)$$

$$p_K(x) = \sum_{i=1}^{K} \alpha_i q_i(x)$$



Mixture update

$$p_i = \alpha_i h + (1 - \alpha_i) p_{i-1}$$

is performed in two steps

1. Find a new component

$$\max_{h \in Q} \mathcal{H}(h)$$

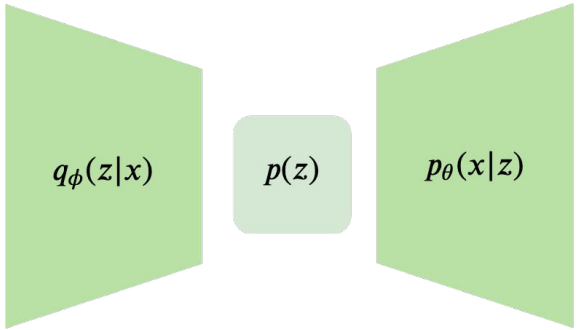
$$D_{\mathrm{KL}}(p_{t-1}|p^*) - D_{\mathrm{KL}}(p_t|p^*) > 0$$

+ linearization

2. Find its weight in the mixture

$$\min_{\alpha \in (0,1)} D_{\mathrm{KL}}(\alpha_i h + (1 - \alpha_i) p_{i-1} || p^*).$$

BooVAE: learns prior greedily



$$\log p(x) \geq \mathcal{L}(x;\theta;q) = \mathbb{E}_{z\sim q(z)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}[q(z)\|p(z)]$$

Target: optimal prior density (Empirical Bayes)

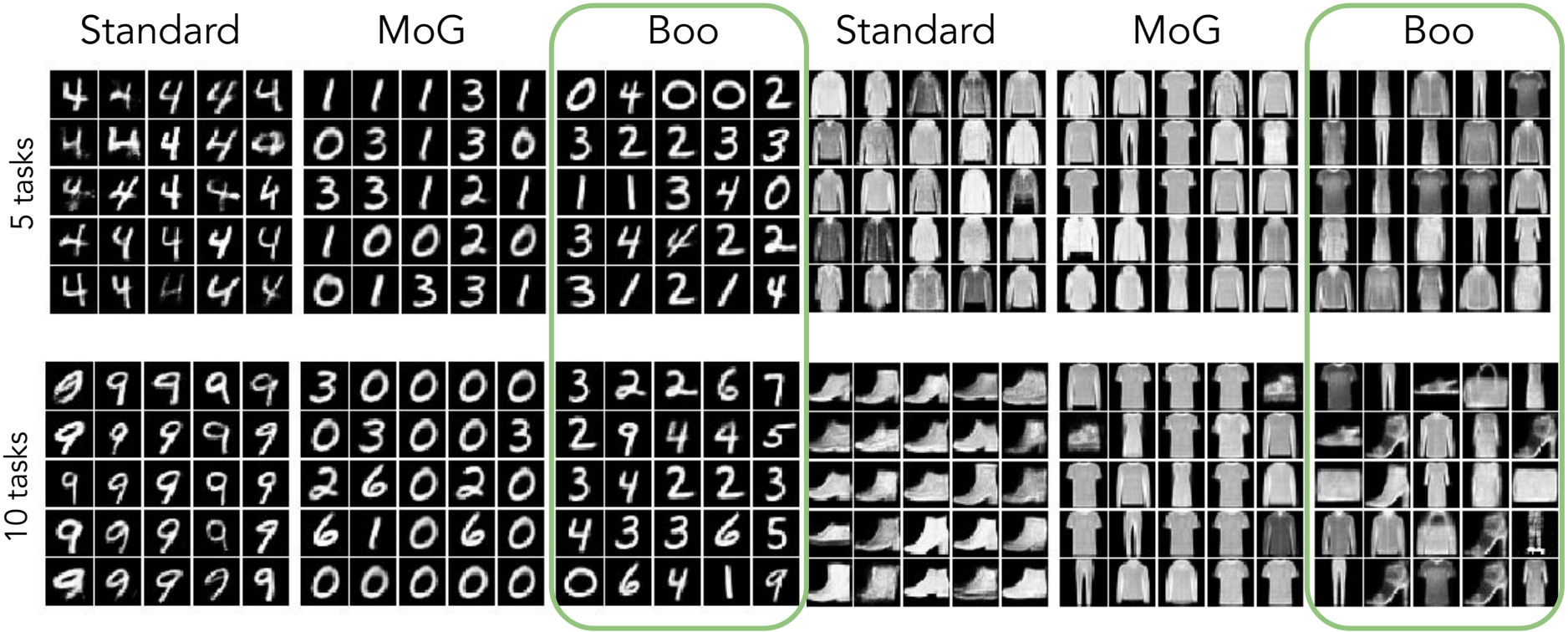
$$p^*(z) = \arg\max_{p(z)} \mathbf{ELBO} = \frac{1}{N}\sum_{n=1}^{N} q_\phi(z|x_n).$$

We Approximate it by boosting

- MaxEntropy Approach reduces number of components
- Use pseudoinputs, following VampPrior idea

Alternate between Encoder/Decoder and Prior updates during training

Results: 10 tasks with known bounds



Come and see our poster (id 41)

arXiv:1908.11853

Anna Kuzina and Evgenii Egorov, Evgeny Burnaev

PhD-wanna-be
akuzina.github.io

a.kuzina@skoltech.ru
scholar

PhD student
evgenii-egorov.github.io

evgenii.egorov@skoltech.ru
scholar

Associate Professor
adase.group

e.burnaev@skoltech.ru
scholar

Variational Gaussian Process Models without Matrix Inverses

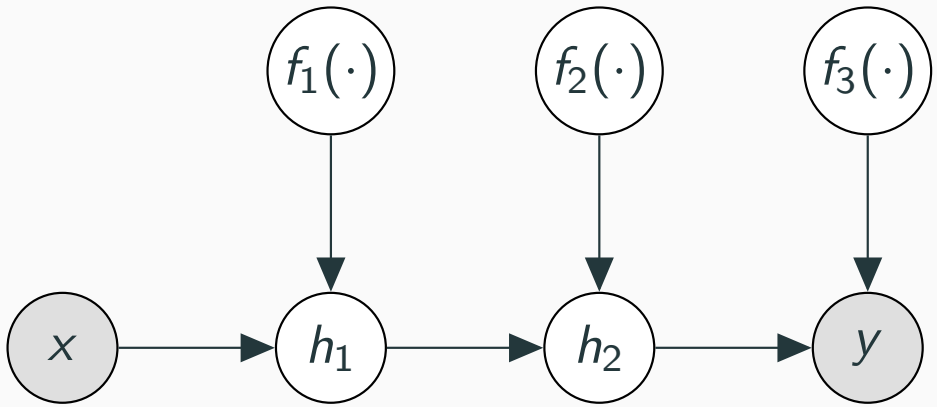
Mark van der Wilk, ST John, Artem Artemev, James Hensman

December 8, 2019

PROWLER.io

Gaussian processes as building blocks

Q: Can Gaussian processes be building blocks
for Bayesian deep learning?



Variational inference is **general** and **actually works** in GP models:

- the ELBO is tight enough for **hyperparameter selection**.
- the variational posterior predicts with **non-parametric error bars**.

Scalability

Cost of minibatch iteration: $\mathcal{O}(BM^2 + M^3)$
for batch size B and model capacity M.

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n \mid f(\mathbf{x}_n)] - \mathrm{KL}[q(f(\cdot))||p(f(\cdot))]$$

$$q(f(\mathbf{x}_n)) = \mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m},\, k_{f_n f_n} - \underbrace{\mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}}_{\text{difficult}} + \mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}\right)$$

Scalability

Cost of minibatch iteration: $\mathcal{O}(BM^2 + M^3)$
for batch size B and model capacity M.

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n \mid f(\mathbf{x}_n)] - \mathrm{KL}[q(f(\cdot))||p(f(\cdot))]$$

$$q(f(\mathbf{x}_n)) = \mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m},\, k_{f_n f_n} - \underbrace{\mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}}_{\text{difficult}} + \mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}\right)$$

Q: Can Gaussian processes be building blocks
for Bayesian deep learning?

Scalability

Cost of minibatch iteration: $\mathcal{O}(BM^2 + M^3)$
for batch size B and model capacity M.

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n \mid f(\mathbf{x}_n)] - \mathrm{KL}[q(f(\cdot))||p(f(\cdot))]$$

$$q(f(\mathbf{x}_n)) = \mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m},\, k_{f_n f_n} - \underbrace{\mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}}_{\text{difficult}} + \mathbf{k}_{\mathbf{u}f_n}^\top \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}\right)$$

Q: Can Gaussian processes be building blocks for Bayesian deep learning?

A: As long as they need a matrix inverse... probably not.

Inverse-free Gaussian Processes

1. New variational posterior:

   Hensman et al [2013] posterior:

   $$\mathcal{N}\big(\mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{m},\ k_{f_nf_n} - \mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n} + \mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}\big)$$

   Our posterior:

   $$\mathcal{N}\big(\mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{m},\ k_{f_nf_n} + \mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{T}\mathbf{K}_{\mathbf{uu}}\mathbf{T}\mathbf{k}_{\mathbf{u}f_n} - 2\mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{T}\mathbf{k}_{\mathbf{u}f_n} + \mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{S}'\mathbf{k}_{\mathbf{u}f_n}\big)$$

Inverse-free Gaussian Processes

1. New variational posterior:

   Hensman et al [2013] posterior:

   $$\mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{m},\ k_{f_nf_n} - \mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n} + \mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}\right)$$

   Our posterior:

   $$\mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{m},\ k_{f_nf_n} + \mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{T}\mathbf{K}_{\mathbf{uu}}\mathbf{T}\mathbf{k}_{\mathbf{u}f_n} - 2\mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{T}\mathbf{k}_{\mathbf{u}f_n} + \mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{S}'\mathbf{k}_{\mathbf{u}f_n}\right)$$

2. New variational bound also a function of $\mathbf{T}$, with property:

   $$\mathbf{T}^* = \operatorname*{argmax}_{\mathbf{T}} \mathcal{L}(..., \mathbf{T}) = \mathbf{K}_{\mathbf{uu}}^{-1}$$

Inverse-free Gaussian Processes

1. New variational posterior:

   Hensman et al [2013] posterior:

   $$\mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{m},\ k_{f_nf_n} - \mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n} + \mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}\right)$$

   Our posterior:

   $$\mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{m},\ k_{f_nf_n} + \mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{T}\mathbf{K}_{\mathbf{uu}}\mathbf{T}\mathbf{k}_{\mathbf{u}f_n} - 2\mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{T}\mathbf{k}_{\mathbf{u}f_n} + \mathbf{k}_{\mathbf{u}f_n}^\mathsf{T}\mathbf{S}'\mathbf{k}_{\mathbf{u}f_n}\right)$$

2. New variational bound also a function of $\mathbf{T}$, with property:

   $$\mathbf{T}^* = \operatorname*{argmax}_{\mathbf{T}} \mathcal{L}(..., \mathbf{T}) = \mathbf{K}_{\mathbf{uu}}^{-1}$$

3. Stochastic estimate of bound costs $\mathcal{O}(BM^2)$

Inverse-free Gaussian Processes

1. New variational posterior:

   Hensman et al [2013] posterior:

   $$\mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{m},\ k_{f_nf_n} - \mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n} + \mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}f_n}\right)$$

   Our posterior:

   $$\mathcal{N}\left(\mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{m},\ k_{f_nf_n} + \mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{T}\mathbf{K}_{\mathbf{uu}}\mathbf{T}\mathbf{k}_{\mathbf{u}f_n} - 2\mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{T}\mathbf{k}_{\mathbf{u}f_n} + \mathbf{k}_{\mathbf{u}f_n}^{\mathsf{T}}\mathbf{S}'\mathbf{k}_{\mathbf{u}f_n}\right)$$

2. New variational bound also a function of $\mathbf{T}$, with property:

   $$\mathbf{T}^* = \operatorname*{argmax}_{\mathbf{T}} \mathcal{L}(..., \mathbf{T}) = \mathbf{K}_{\mathbf{uu}}^{-1}$$

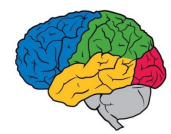
3. Stochastic estimate of bound costs $\mathcal{O}(BM^2)$

See more at our poster #49!

Scalable Gradients for Stochastic Differential Equations

Xuechen Li [3] Ting-Kam Leonard Wong [1]
Ricky T. Q. Chen [1,2] David Duvenaud [1,2]

[1]University of Toronto [2]Vector Institute [3]Google Brain



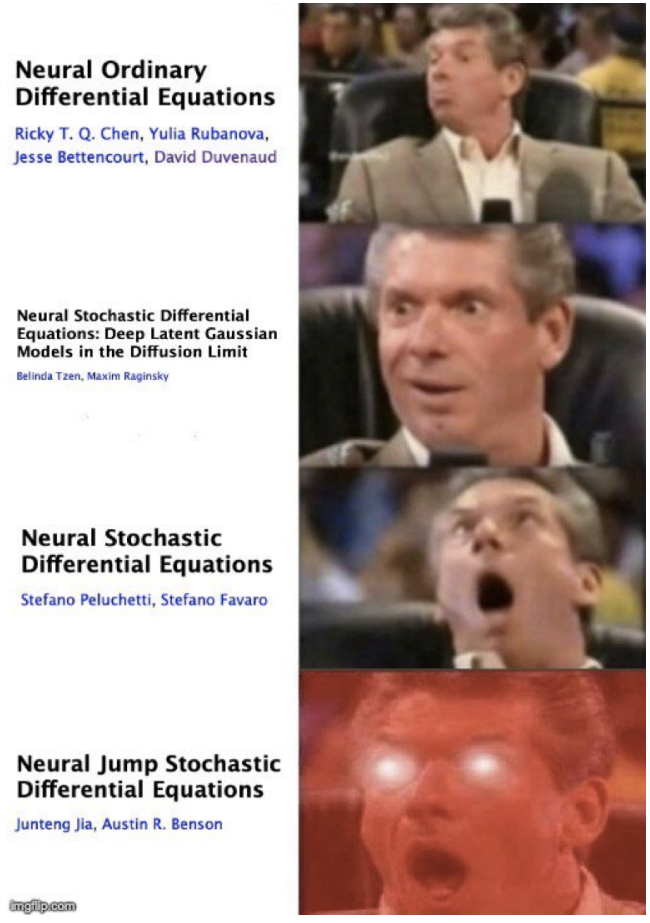
Background: The Adjoint Method for ODEs

- The *adjoint sensitivity method* can be combined with reverse-mode AD [1] to train neural ODEs.
- Define adjoint $a(t) = \partial L / \partial z_t$

$$a(0) = a(T) - \int_T^0 a(t) \frac{\partial f(z(t), t)}{\partial z(t)}^\top \mathrm{d}t.$$

- Can be implemented using calls to *vector-Jacobian product*.
- Doesn't store anything except current state.
- An extension to **stochastic differential equations**?

Related Work



We still don't know how to extend the adjoint method to SDEs for time-efficient and $O(1)$ memory backprop!

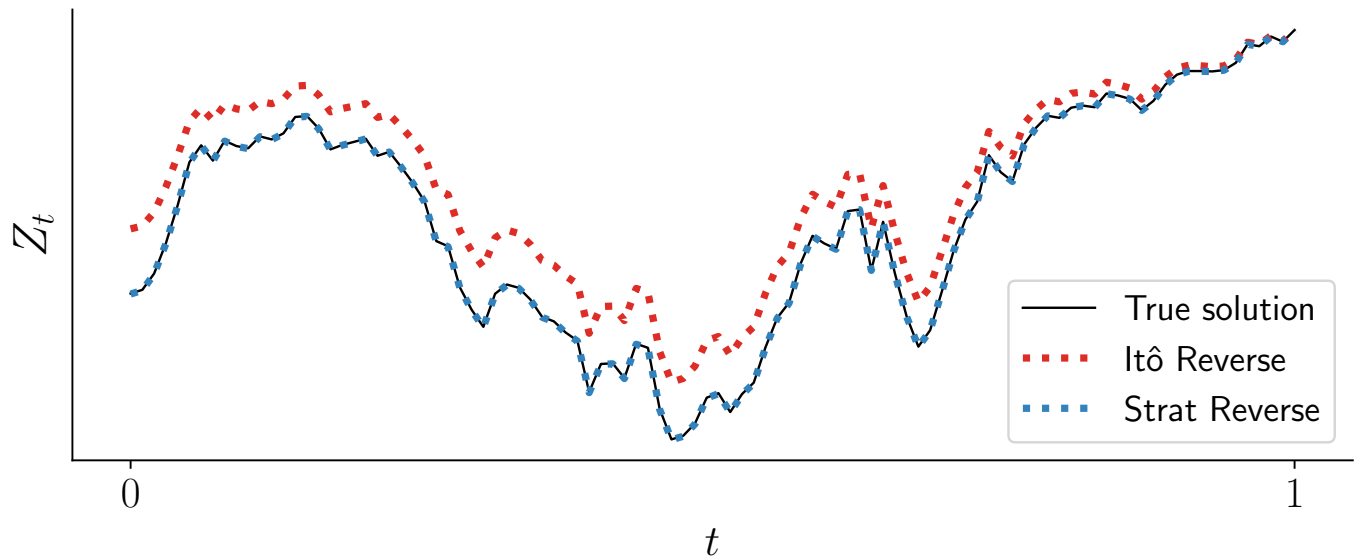
Related Work

Unfortunately, there is no straightforward way to port this construction to SDEs. While the analogue of the method of adjoints is available for SDEs (see, e.g., Yong and Zhou (1999, Chap. 3)), the adjoint equation is an SDE that has to be solved backward in time with a terminal condition that depends on the entire Wiener path $W = \{W_t\}_{t\in[0,1]}$, but the solution at each time $t \in [0, 1]$ must still be measurable w.r.t. the "past" $\{W_s\}_{s\in[0,t]}$.

–Tzen & Raginsky (2019).

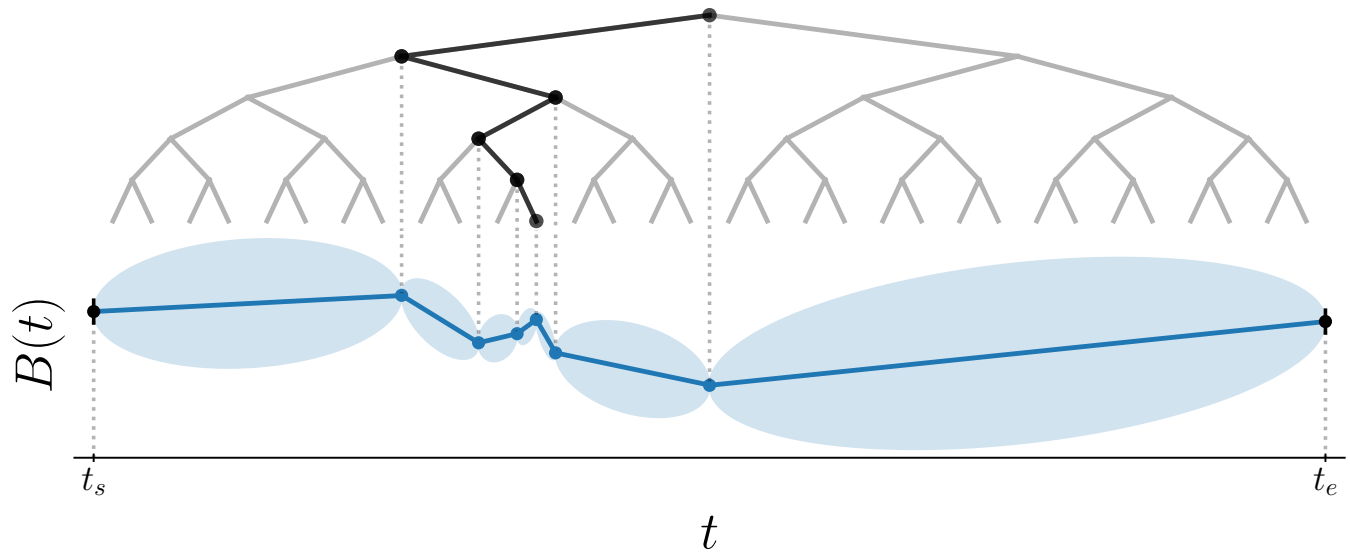
Challenge 1: A Backward SDE Formulation

- $O(1)$ memory backprop implies a need to reconstruct the state.
- Sol'ns to (Stratonovich) SDE are *stochastic diffeomorphism*.
- Stochastic diffeomorphisms solve Kunita backward SDEs [2].
- These backward SDEs can be numerically simulated cheaply.



$$\check{\Psi}_{s,t}(z) = z - \int_s^t b(\check{\Psi}_{r,t}(z), r)\,\mathrm{d}r - \sum_{i=1}^{m} \int_s^t \sigma_i(\check{\Psi}_{r,t}(z), r) \circ \check{\mathrm{d}}B_r^{(i)}.$$

Challenge 2: Reproducing Noise (w/o Storing)



Given a single seed, our *seeded Brownian tree* can generate the whole Brownian motion to a user-specified precision, without storing intermediate computation.

Latent Stochastic Differential Equations

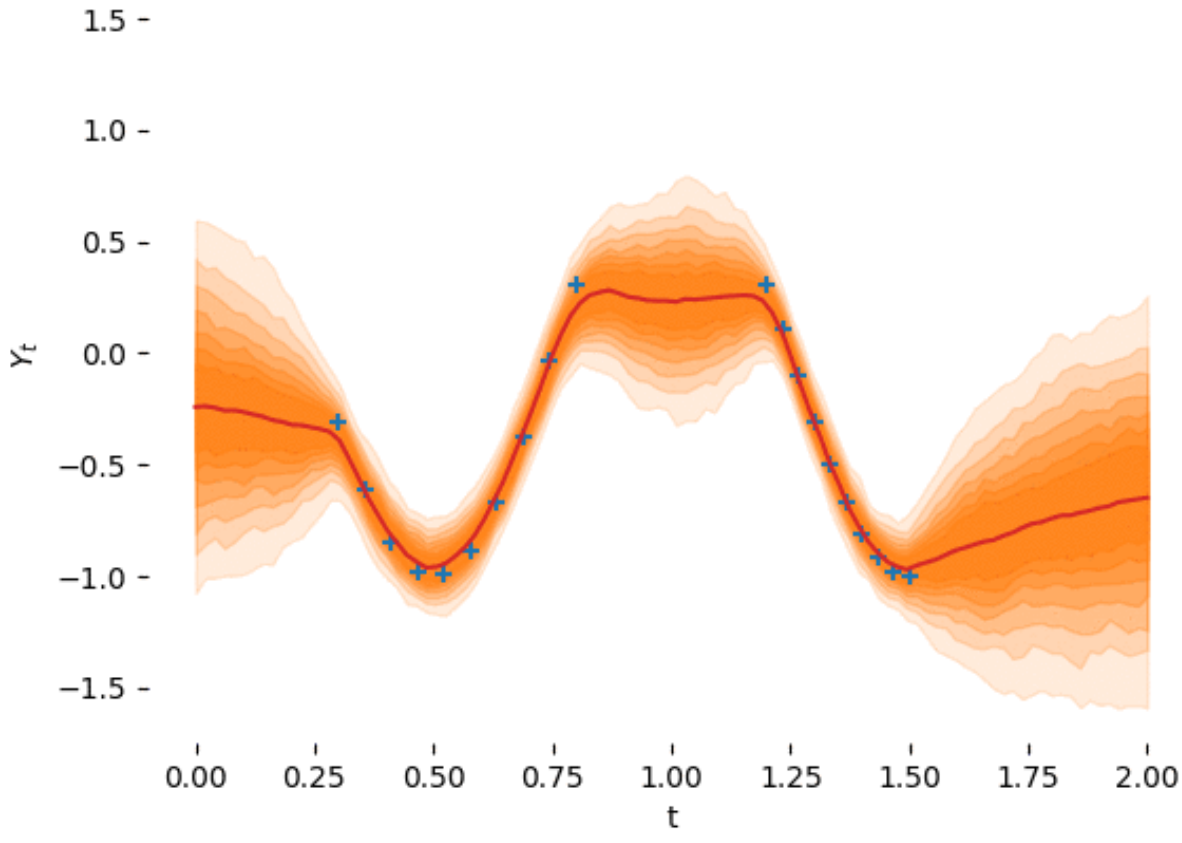
Prior and approximate posterior processes have same diffusion but different drift coefficients.

$$\mathrm{d}p_t = f_p(p_t, t) + \sigma(p_t, t)\,\mathrm{d}W_t, \quad p_0 = x \in \mathbb{R}^d \qquad \text{(Prior)}$$
$$\mathrm{d}q_t = f_q(q_t, t) + \sigma(q_t, t)\,\mathrm{d}W_t, \quad q_0 = x \in \mathbb{R}^d \qquad \text{(Approx. Posterior)}$$

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}\left[-\tfrac{1}{2}\int_0^T |u(q_t, t)|^2 \,\mathrm{d}t + \sum_{i=1}^N \log p(y_{t_i}|q_{t_i})\right],$$

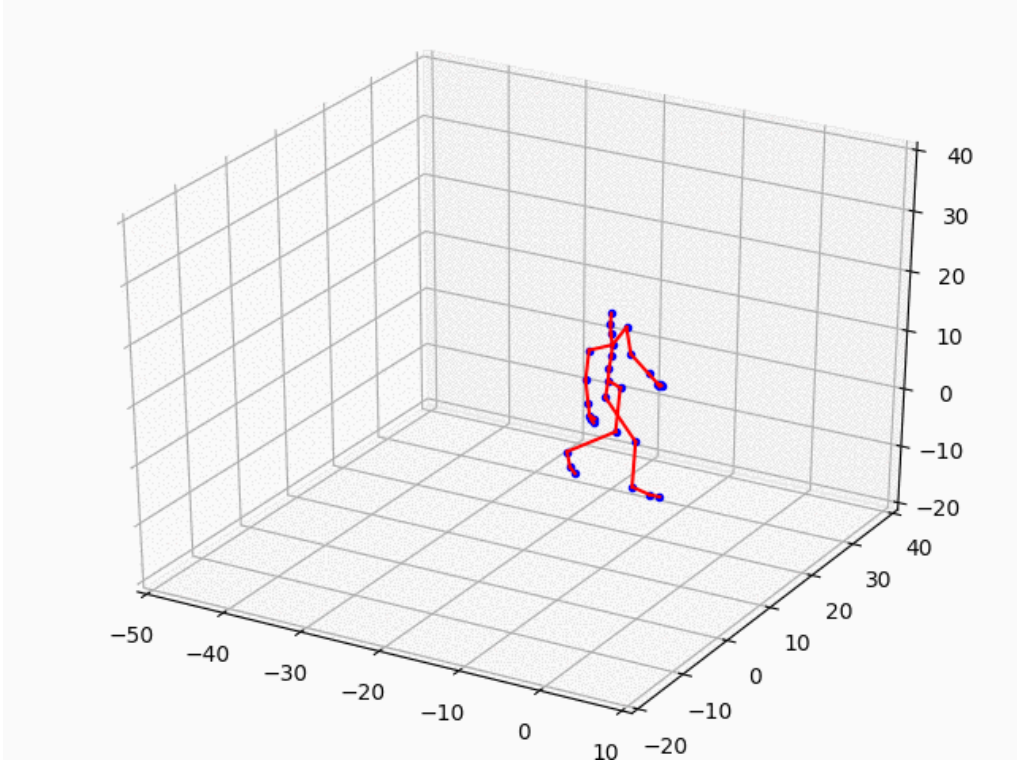
where control $u : \mathbb{R}^d \to \mathbb{R}^m$ solves the linear system $\sigma u = f_p - f_q$.

Applications: Fitting a Single Time-series



Drawback: high variance. Future work: control variates? Common random numbers?

Applications: Fitting Many Time-series



So far, found to be good for datasets with few but non-trivial time series. Strong path space regularization!

Next Steps

- Machine learning:
  - Comprehensive empirical comparison; GPs vs SDEs.
  - Stationary and multi-timescale models.
  - Infinitely deep Bayesian neural nets.
- Numerical analysis:
  - Convert the method to solve high-dimensional PDEs.
  - New theoretical interpretations using rough path theory?
- Other areas of science and engineering:
  - Better models of allele frequency in population genetics.
  - Fast and memory-frugal algorithms for greeks in finance.
- Software: A **PyTorch** package is on the way! Stay tuned!

Thanks to you and my coauthors:

Ting-Kam Leonard Wong

Ricky T. Q. Chen

David Duvenaud

[1] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[2] Hiroshi Kunita. *Stochastic Flows and Jump-Diffusions*. Springer, 2019.

Characterizing and Avoiding Problematic Global Optima of Variational Autoencoders

Yaniv Yacoby
Weiwei Pan
Finale Doshi-Velez

Harvard John A. Paulson
School of Engineering
and Applied Sciences

Background: Variational Autoencoders (VAEs)

VAEs: [2] latent variable models that capture a data distribution, $p(x)$, and infer latent representations.

Generative model:

$$z \sim \mathcal{N}(0, I)$$
$$x|z \sim \mathcal{N}\left(f_\theta(z), \sigma_\epsilon^2 \cdot I\right)$$

Inference:

$$\theta^*, \phi^* = \text{argmax}_{\theta,\phi}\text{ELBO}(\theta, \phi)$$

using an **inference model**, $q_\phi(z|x)$.

Our Contributions

Traditional VAE model & training exhibits pathologies (e.g. [1, 3]).

Contributions:

1. We characterize the pathologies under unifying framework.
2. We present a novel inference method that avoids them.

Now: walk-through of 1 example.

VAE Pathologies #1

Mismatch between aggregated posterior and prior (e.g. [3])

$$p(z) = \mathbb{E}_{p_{\text{data}}(x)}\left[p_\theta(z|x)\right] \approx \frac{1}{N}\sum_n q_\phi(z|x_n) \neq p(z)$$

Effect: poor quality generated data

Why does it happen? undesirable global optima of the ELBO.

A Toy Example

Example:

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

$$\epsilon \sim \mathcal{N}\left(0, I \cdot \sigma_\epsilon^2 - B\right)$$

$$x|z = \text{Cholesky}\left(AA^\intercal + B\right) z + \epsilon$$

- Fixed A
- Learned $\theta = B$

A Toy Example

Example:

$$z \sim \mathcal{N}(0, I)$$
$$\epsilon \sim \mathcal{N}\left(0, I \cdot \sigma_\epsilon^2 - B\right)$$
$$x|z = \text{Cholesky}\left(AA^\intercal + B\right) z + \epsilon$$

- Fixed A
- Learned $\theta = B$

Explanation:

$$-\text{ELBO} = \underbrace{\text{Objective \#1}}_{\text{match } p_\theta(x) \text{ with } p_{\text{data}}(x)} + \underbrace{\text{Objective \#2}}_{\text{match } q_\phi(z|x) \text{ with } p_\theta(z|x)} \quad [4]$$

A Toy Example

Example:

$$z \sim \mathcal{N}(0, I)$$
$$\epsilon \sim \mathcal{N}\left(0, I \cdot \sigma_\epsilon^2 - B\right)$$
$$x|z = \text{Cholesky}\left(AA^\intercal + B\right) z + \epsilon$$

- Fixed A
- Learned $\theta = B$

Explanation:

$$-\text{ELBO} = \underbrace{\text{Objective \#1}}_{\text{match } p_\theta(x) \text{ with } p_{\text{data}}(x)} + \underbrace{\text{Objective \#2}}_{\text{match } q_\phi(z|x) \text{ with } p_\theta(z|x)} \quad [4]$$

At ground truth:

$$-\text{ELBO}_{\text{GT}} = \underbrace{\text{Objective \#1}}_{=0} + \underbrace{\text{Objective \#2}}_{\text{very large}} = 0.532$$

A Toy Example

Example:

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

$$\epsilon \sim \mathcal{N}\left(0, I \cdot \sigma_\epsilon^2 - B\right)$$

$$x|z = \text{Cholesky}\left(AA^\intercal + B\right) z + \epsilon$$

- Fixed A
- Learned $\theta = B$

Explanation:

$$-\text{ELBO} = \underbrace{\text{Objective \#1}}_{\text{match } p_\theta(x) \text{ with } p_{\text{data}}(x)} + \underbrace{\text{Objective \#2}}_{\text{match } q_\phi(z|x) \text{ with } p_\theta(z|x)} \quad [4]$$

At ground truth:

$$-\text{ELBO}_{\text{GT}} = \underbrace{\text{Objective \#1}}_{=0} + \underbrace{\text{Objective \#2}}_{\text{very large}} = 0.532$$

At global optima of ELBO:

$$-\text{ELBO} = \underbrace{\text{Objective \#1}}_{\text{small}} + \underbrace{\text{Objective \#2}}_{=0} = 0.196$$

A Toy Example

Example:

$$z \sim \mathcal{N}(0, I)$$
$$\epsilon \sim \mathcal{N}\left(0, I \cdot \sigma_\epsilon^2 - B\right)$$
$$x|z = \text{Cholesky}\left(AA^\intercal + B\right) z + \epsilon$$

- Fixed A
- Learned $\theta = B$

Conclusion:

At global optima of ELBO, $\theta \neq \theta_{\text{GT}}$ (ground-truth)

This causes model to learn the wrong $p(x)$.

This causes mismatch:

$$\begin{aligned} p(z) &= \mathbb{E}_{p_{\text{data}}(x)}\left[p_{\theta_{\text{GT}}}(z|x)\right] \\ &\neq \mathbb{E}_{p_{\text{data}}(x)}\left[p_\theta(z|x)\right] \end{aligned}$$

A Toy Example

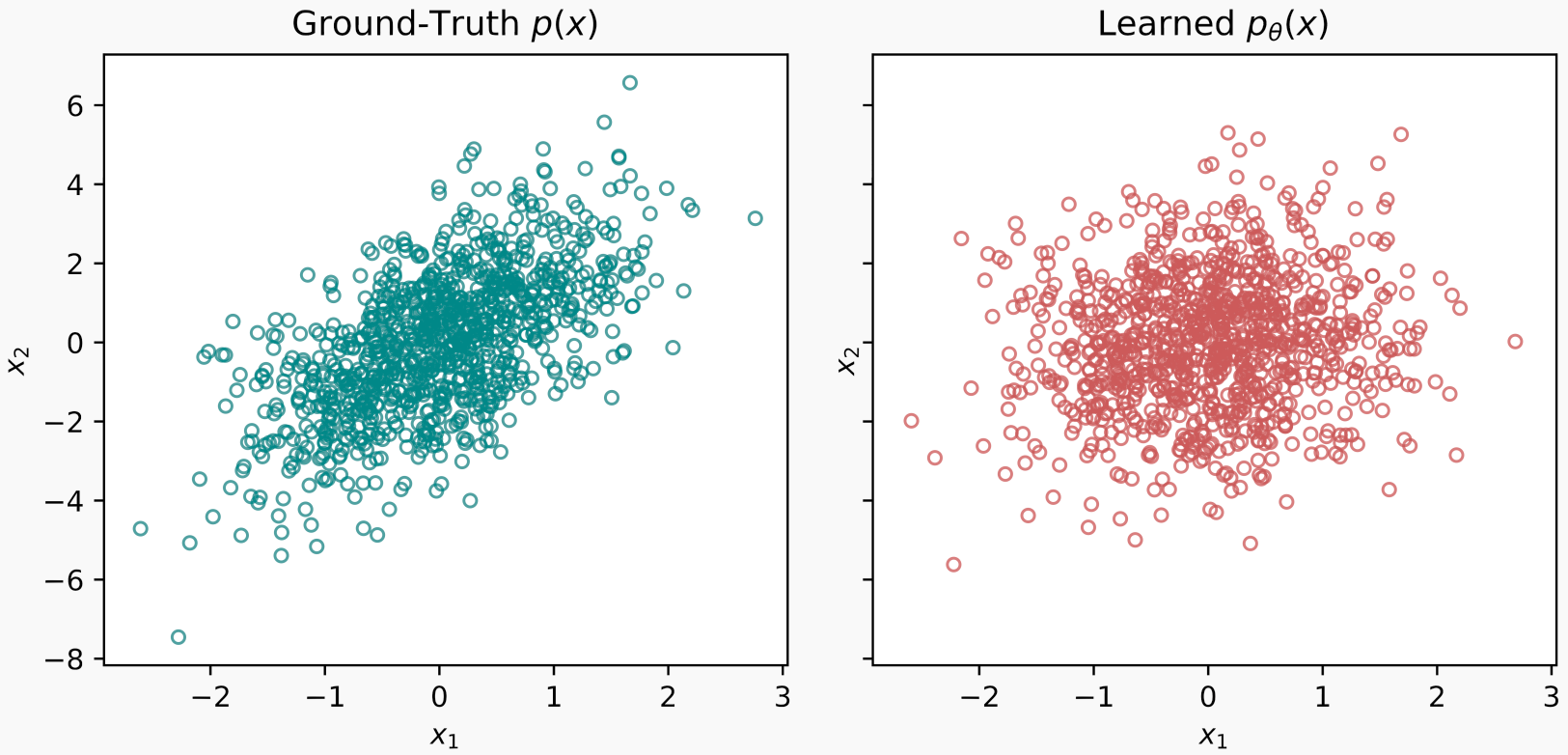
Example:

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

$$\epsilon \sim \mathcal{N}\left(0, I \cdot \sigma_\epsilon^2 - B\right)$$

$$x|z = \text{Cholesky}\left(AA^\intercal + B\right) z + \epsilon$$

- Fixed A
- Learned $\theta = B$



A Toy Example

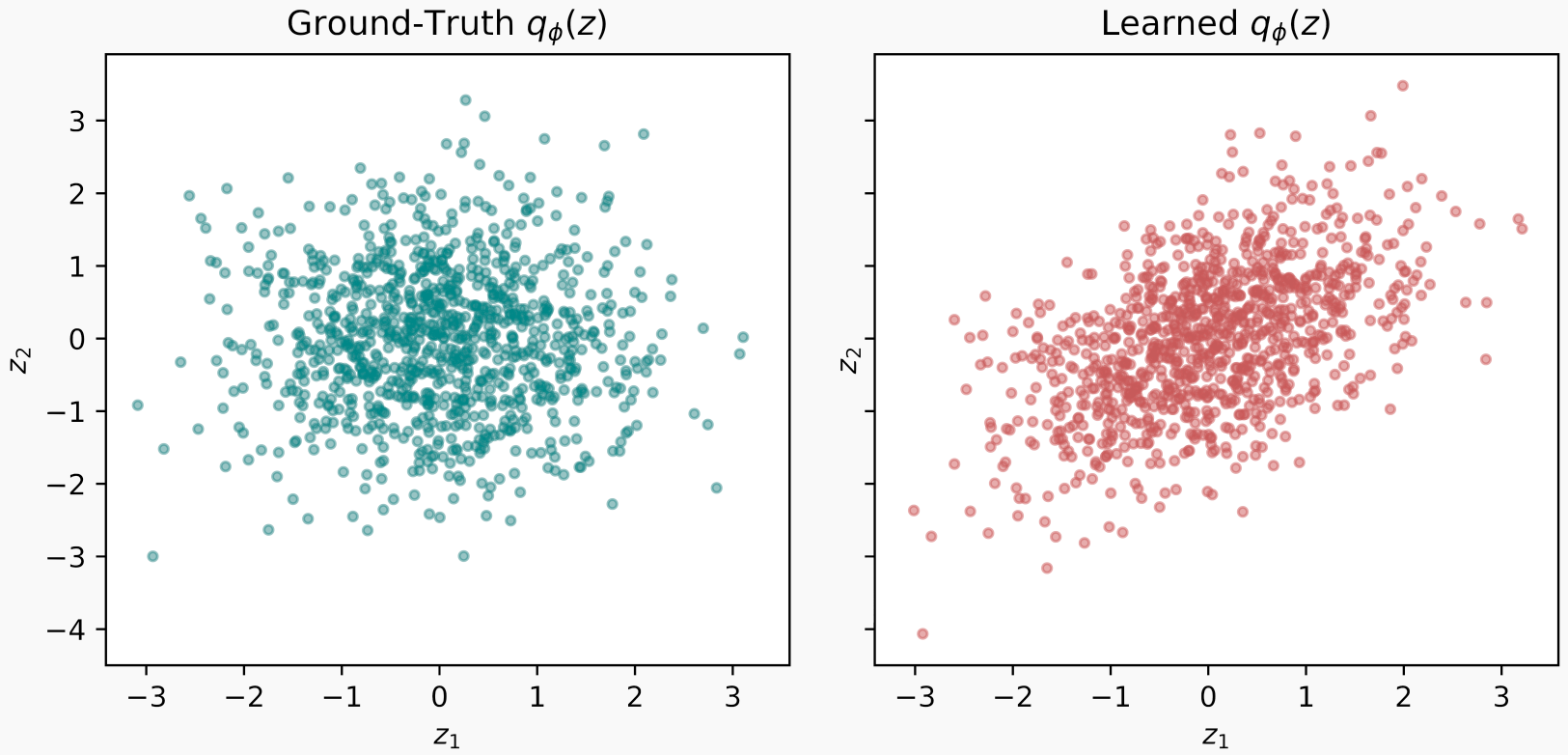
Example:

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

$$\epsilon \sim \mathcal{N}\left(0, I \cdot \sigma_\epsilon^2 - B\right)$$

$$x|z = \text{Cholesky}\left(AA^\intercal + B\right) z + \epsilon$$

- Fixed A
- Learned $\theta = B$



Our Contributions

Our Contributions:

1. Unifying analysis of source of pathologies:
   (a) Ex. 1: VAE model non-identifiability $\rightarrow$ posterior collapse
   (b) **Ex. 2: VAE inference compromise quality of generative model $\rightarrow$ aggregated posterior mismatch**
   (c) Ex. 3: VAE objective prefers models with undesireable properties
2. LiBI: a novel **inference method** to avoid pathologies

Thank you!

More details @ poster #55!

References:

[1] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick. Lagging Inference Networks and Posterior Collapse in Variational Autoencoders. *arXiv:1901.05534 [cs, stat]*, Jan. 2019. arXiv: 1901.05534.

[2] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114, Dec 2013.

[3] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial Autoencoders. *arXiv:1511.05644 [cs]*, Nov. 2015. arXiv: 1511.05644.

[4] S. Zhao, J. Song, and S. Ermon. Towards Deeper Understanding of Variational Autoencoding Models. *arXiv:1702.08658 [cs, stat]*, Feb. 2017. arXiv: 1702.08658.

Information in Infinite Ensembles of Infinitely-Wide Neural Networks

Ravid Shwartz Ziv[1]

Alex Alemi[2]

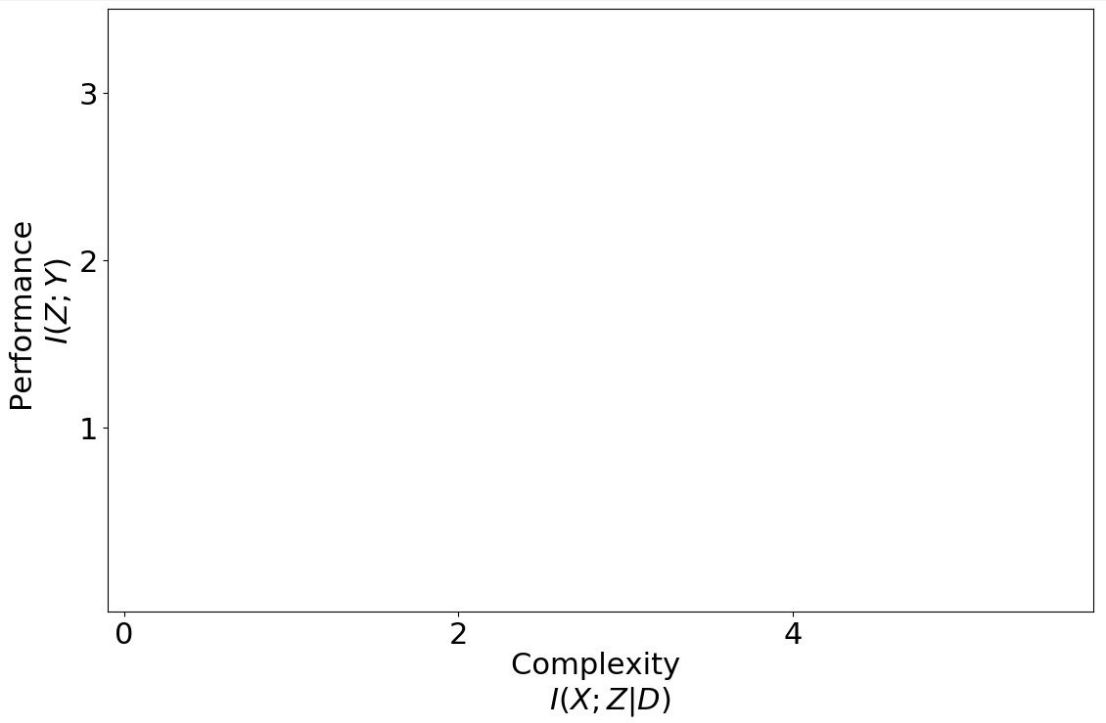
[1] האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

[2] Google Research

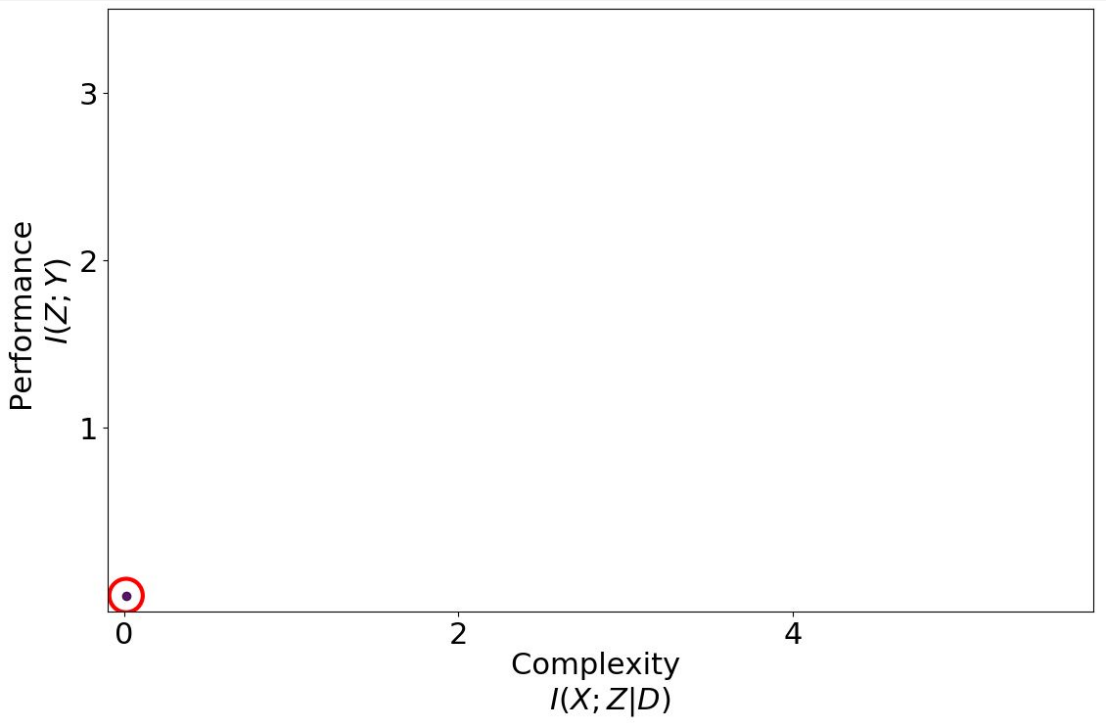
arXiv:1911.09189

- Better understanding of generalization.
- Mutual information is hard to quantify.
- Ensembling over the initial parameters.
- Infinitely wide neural networks -
  - Tracatable distributions.
  - Information theoretic quantities.

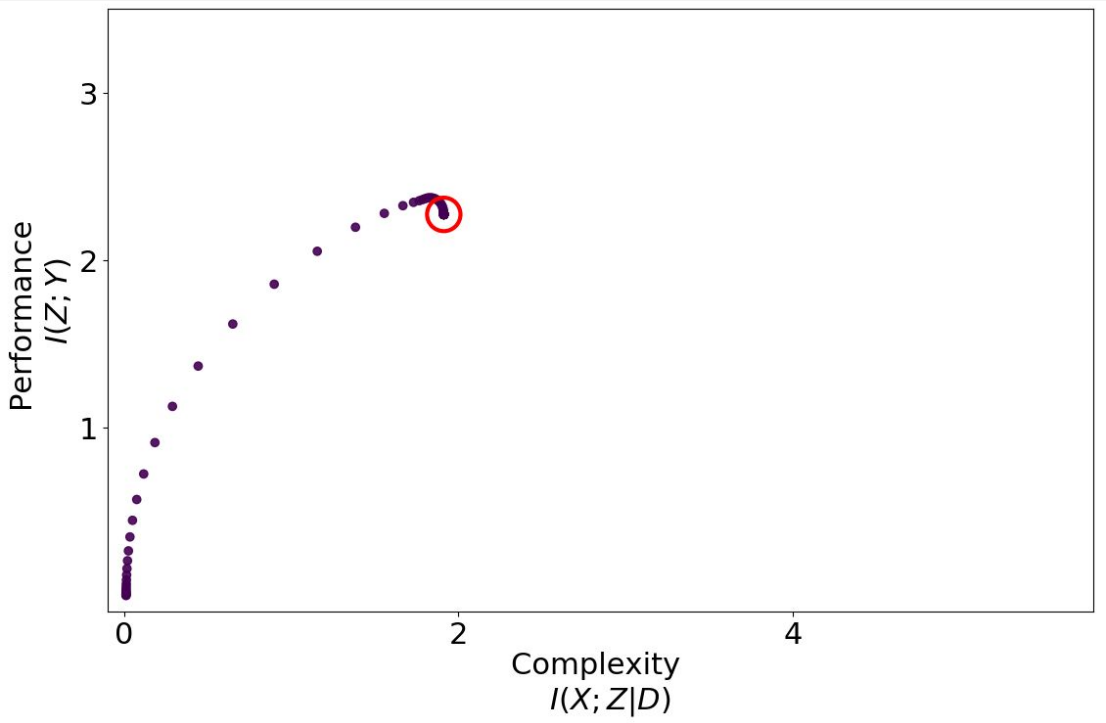
Generalization and dataset size



Generalization and dataset size



Generalization and dataset size



Generalization and dataset size