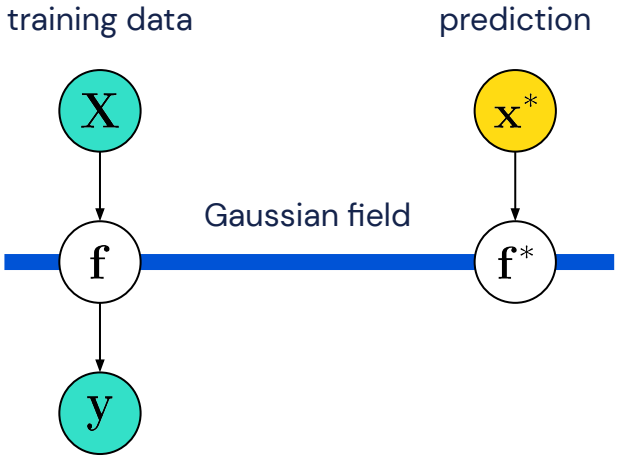
Sparse Orthogonal Variational Inference for Gaussian Processes

Jiaxin Shi (Tsinghua University)

Michalis K. Titsias, **Andriy Mnih** (DeepMind)

Gaussian Processes (GP)



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

Definition

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

mean function

covariance function / kernel

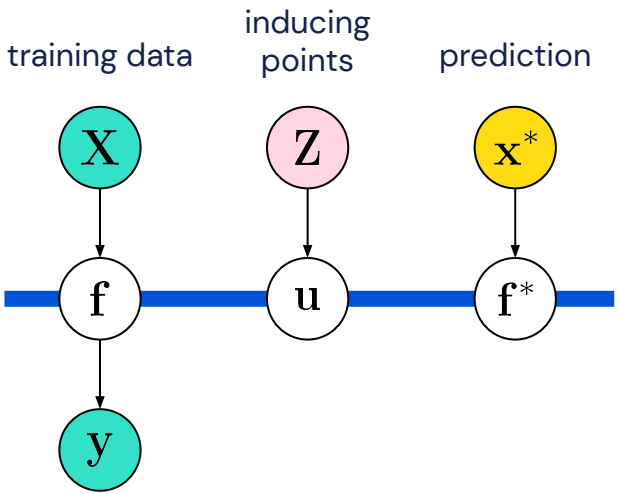
$$p(\mathbf{f}, \mathbf{f}^*) := \mathcal{N}\left(\begin{array}{c}\mathbf{f} \\ \mathbf{f}^*\end{array} \middle| \left[\begin{array}{c} m(\mathbf{X}) \\ m(\mathbf{x}^*)\end{array}\right], \left[\begin{array}{cc}\mathbf{K}_{\mathbf{ff}} & \mathbf{K}_{\mathbf{f}*} \\ \mathbf{K}_{*\mathbf{f}} & \mathbf{K}_{**}\end{array}\right]\right)$$

Exact Inference

$$p(\mathbf{f}, \mathbf{f}^*|\mathbf{y}) \propto p(\mathbf{f}, \mathbf{f}^*)p(\mathbf{y}|\mathbf{f})$$

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

Inducing Points



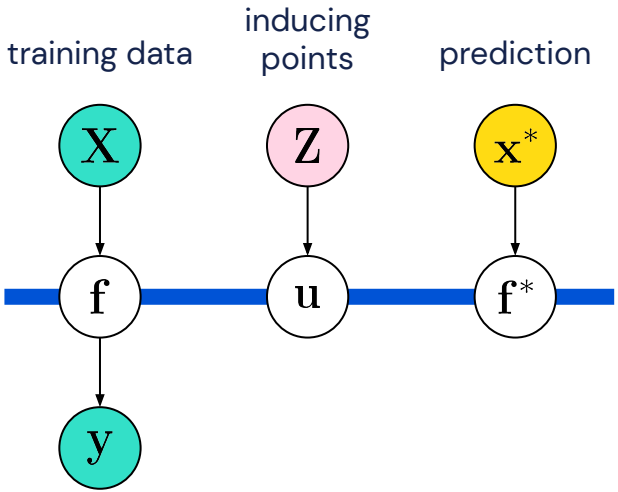
We need approximations for big data.

Solution: Summarize f with a small number of inducing variables u.

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

(Titsias, 09; Hensman et al., 13)

Inducing Points



We need approximations for big data.

Solution: Summarize f with a small number of inducing variables u.

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

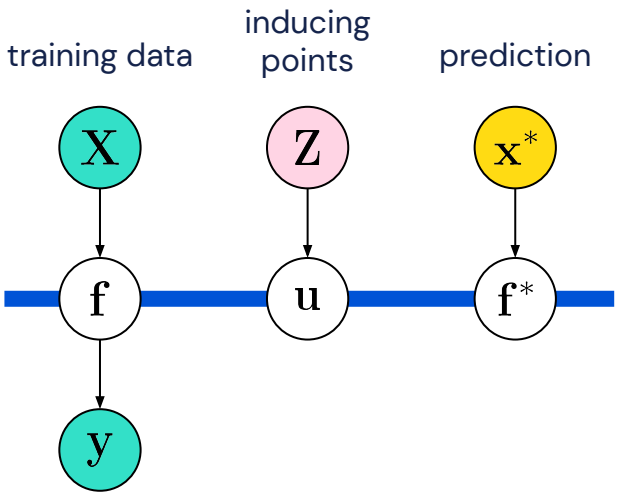
The augmented joint distribution is

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{array}{c} \mathbf{f} \\ \mathbf{u} \end{array} \middle| \, \mathbf{0}, \begin{bmatrix} \mathbf{K_{ff}} & \mathbf{K_{fu}} \\ \mathbf{K_{uf}} & \mathbf{K_{uu}} \end{bmatrix} \right)$$

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

(Titsias, 09; Hensman et al., 13)

Inducing Points



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

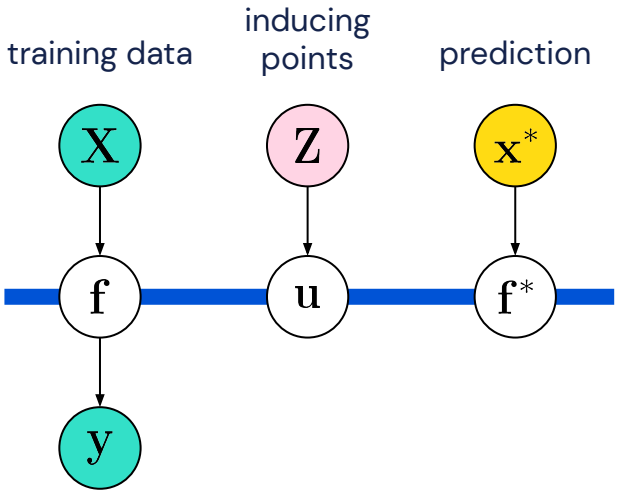
Sparse variational GP methods (SVGP)

- Augmented joint distribution:

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

(Titsias, 09; Hensman et al., 13)

Inducing Points



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

Sparse variational GP methods (SVGP)

- Augmented joint distribution:

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

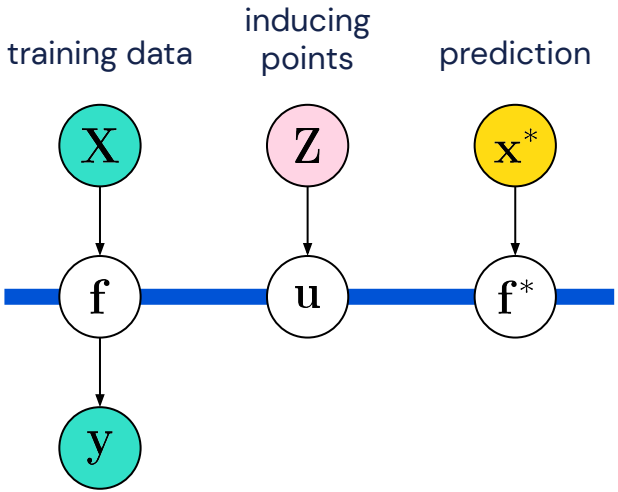
- Variational distribution:

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

$$\mathrm{KL}\left[q(\mathbf{f}, \mathbf{u})||p(\mathbf{f}, \mathbf{u}|\mathbf{y})\right]$$

(Titsias, 09; Hensman et al., 13)

Inducing Points



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

Sparse variational GP methods (SVGP)

- Augmented joint distribution:

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

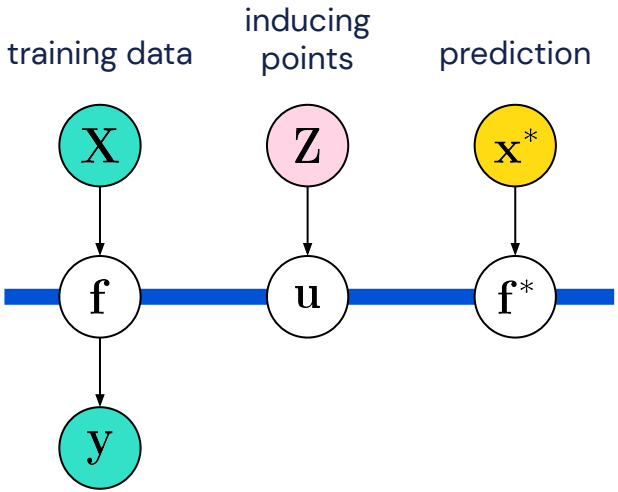
- Variational distribution:

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

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

(Titsias, 09; Hensman et al., 13)

Inducing Points



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

Sparse variational GP methods (SVGP)

- Augmented joint distribution:

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

- Variational distribution:

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

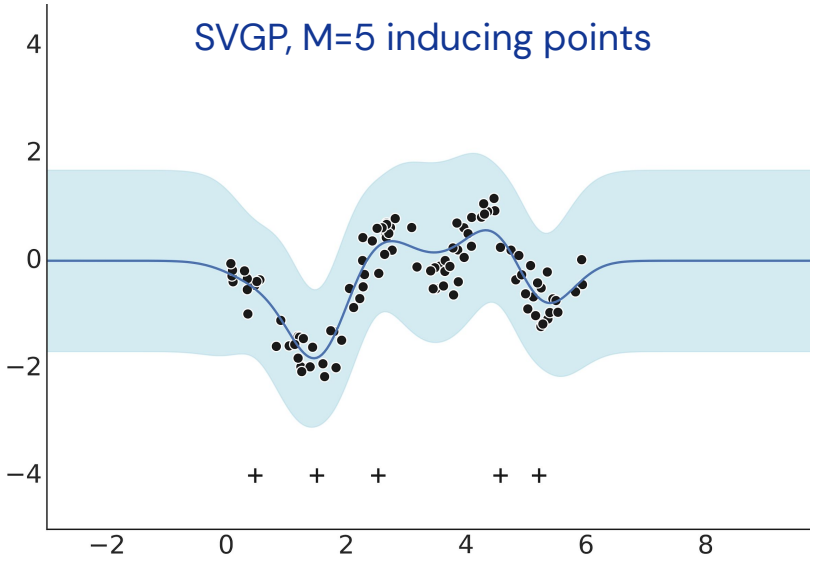
$$\mathrm{KL}\left[q(\mathbf{f}, \mathbf{u}) \| p(\mathbf{f}, \mathbf{u}|\mathbf{y})\right] = \mathbb{E}_q \log \frac{p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \cdot p(\mathbf{y})}{p(\mathbf{y}|\mathbf{f}) \cdot p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}$$

$$\mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}[q(\mathbf{u}) \| p(\mathbf{u})]$$

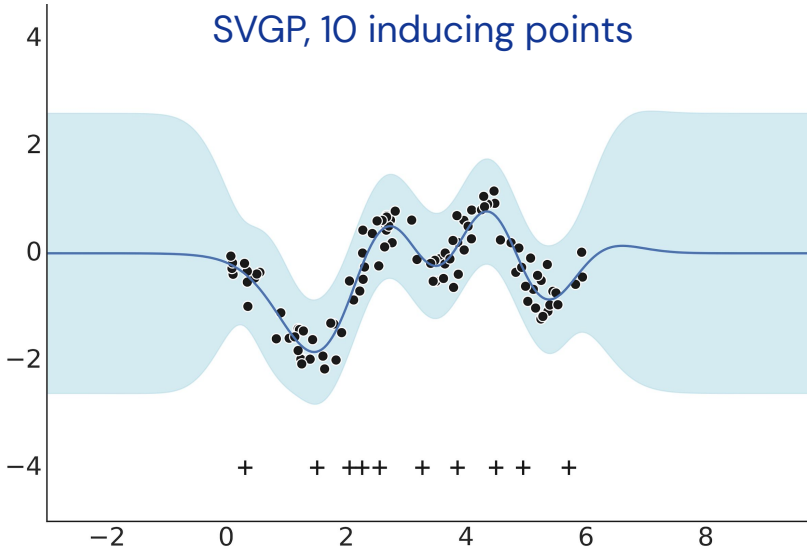
$\mathcal{O}(M^3)$ complexity per update

(Titsias, 09; Hensman et al., 13)

Challenge: Inducing Points Are Expensive



Cholesky: cM^3



Cholesky: $8cM^3$

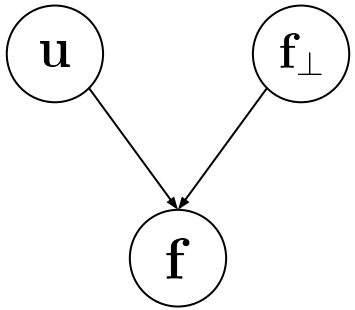
Can we do better?

+ Inducing points

Reinterpreting SVGP

Reparameterization

$$\mathbf{f} \sim p(\mathbf{f}) \begin{cases} \mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{uu}}) & p(\mathbf{u}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) & \begin{cases} \mathbf{f}_\perp \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{K_{uf}}) & p_\perp(\mathbf{f}_\perp) \\ \mathbf{f} = \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{u} + \mathbf{f}_\perp \end{cases} \end{cases}$$

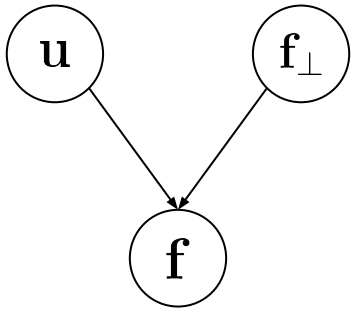


Reinterpreting SVGP

Reparameterization

$$p(\mathbf{u}) \qquad p_\perp(\mathbf{f}_\perp)$$

$$\mathbf{f} \sim p(\mathbf{f}) \begin{cases} \mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{uu}}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) \end{cases} \quad \begin{cases} \mathbf{f}_\perp \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{K_{uf}}) \\ \mathbf{f} = \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{u} + \mathbf{f}_\perp \end{cases}$$



$$\mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})]$$

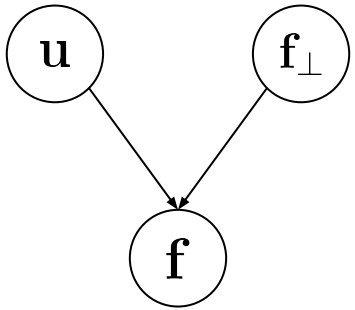
$$\mathbb{E}_{q(\mathbf{u})p_\perp(\mathbf{f}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

The variational posterior distribution over $\mathbf{f}_\perp$ is simply chosen to be the prior distribution.

Reinterpreting SVGP

Reparameterization

$$\mathbf{f} \sim p(\mathbf{f}) \begin{cases} \mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{uu}}) & p(\mathbf{u}) \\ \mathbf{f} \sim p(\mathbf{f}|\mathbf{u}) & p_\perp(\mathbf{f}_\perp) \end{cases} \begin{cases} \mathbf{f}_\perp \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{K_{uf}}) \\ \mathbf{f} = \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{u} + \mathbf{f}_\perp \end{cases}$$

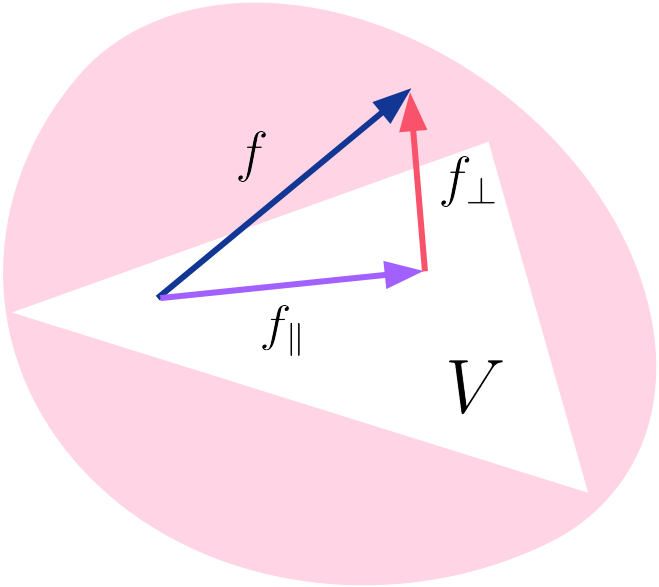


$$\mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})]$$

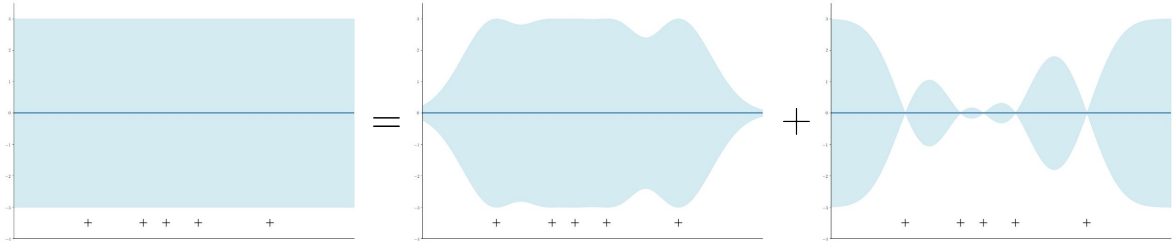
$$\mathbb{E}_{q(\mathbf{u})p_\perp(\mathbf{f}_\perp)}\left[\log p(\mathbf{y}|\mathbf{f}_\perp + \mathbf{K_{fu}}\mathbf{K_{uu}}^{-1}\mathbf{u})\right] - \mathrm{KL}\left[q(\mathbf{u})\|p(\mathbf{u})\right]$$

- Can we improve the variational approximation for $\mathbf{f}_\perp$?
- Full Gaussian parameterization of $q(\mathbf{f}_\perp)$ has $\mathcal{O}(N^3)$ cost.

Orthogonal Decomposition



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



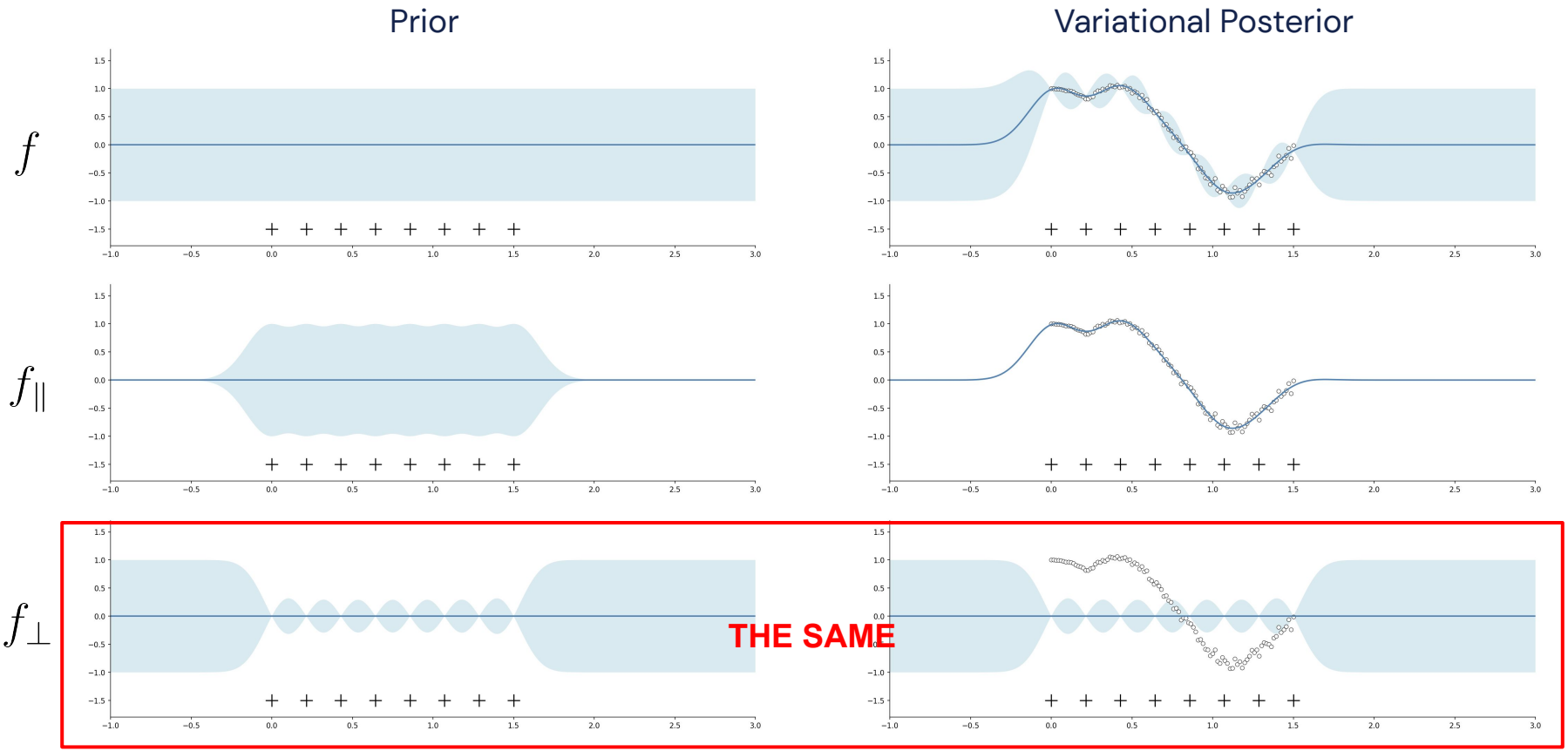
$$p : f = f_\perp + f_\parallel \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$

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

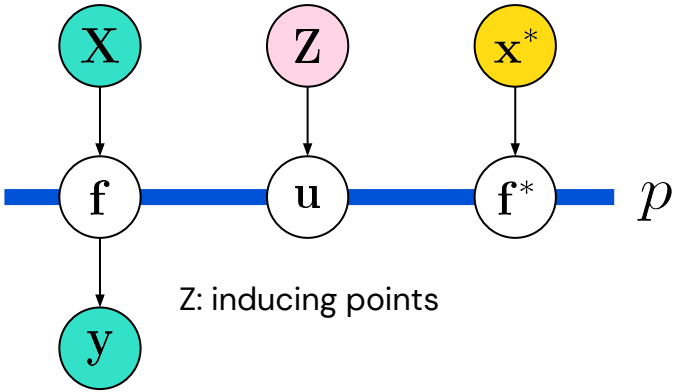
$$p_\perp : f_\perp \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}') - \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{k_u}(\mathbf{x}'))$$

Reinterpreting SVGP

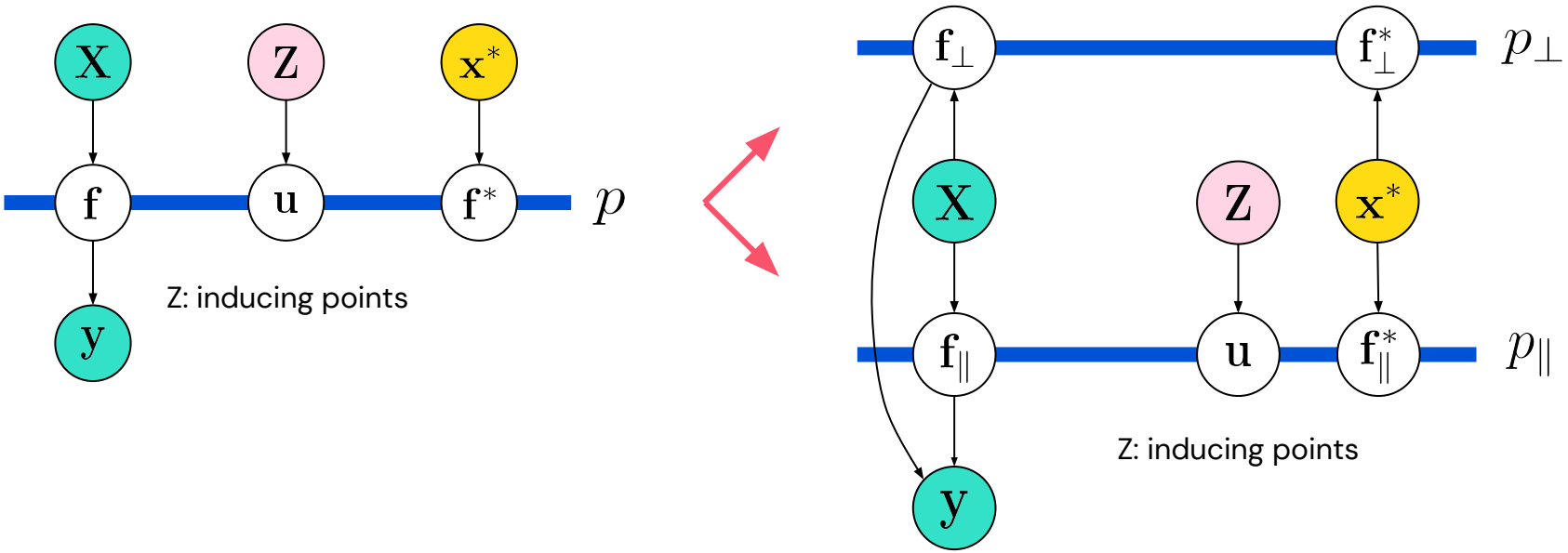
+ Inducing points



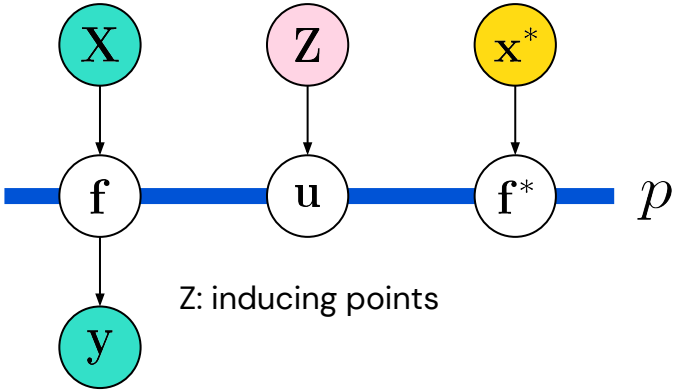
Orthogonal Inducing Points

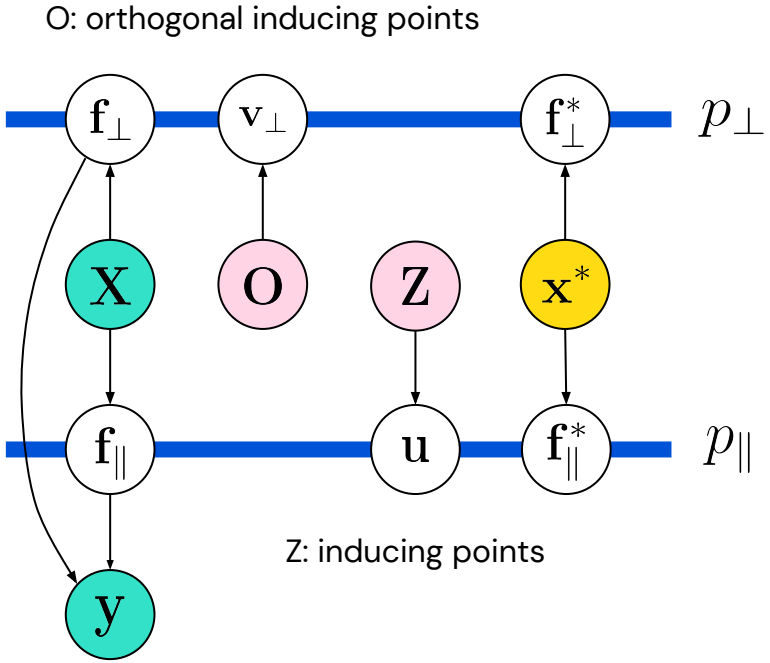


Orthogonal Inducing Points



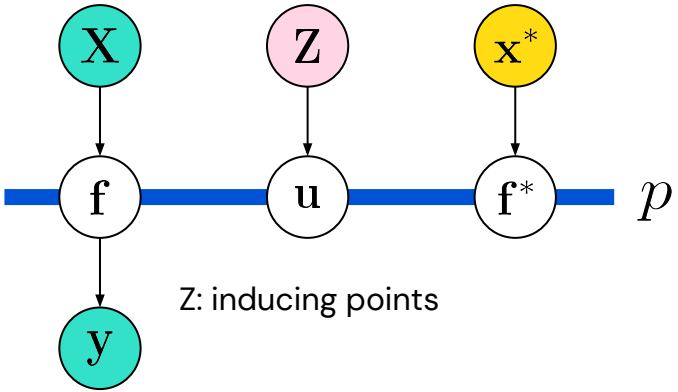
Orthogonal Inducing Points

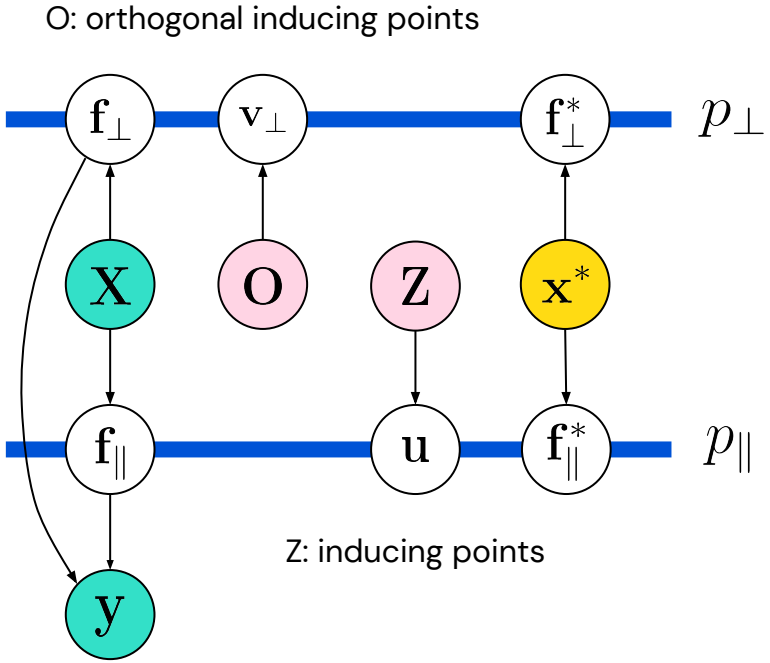




Idea: introducing an additional set of inducing variables $\mathbf{v}_\perp$ to summarize $p_\perp$

Sparse Orthogonal Variational Inference for Gaussian Processes

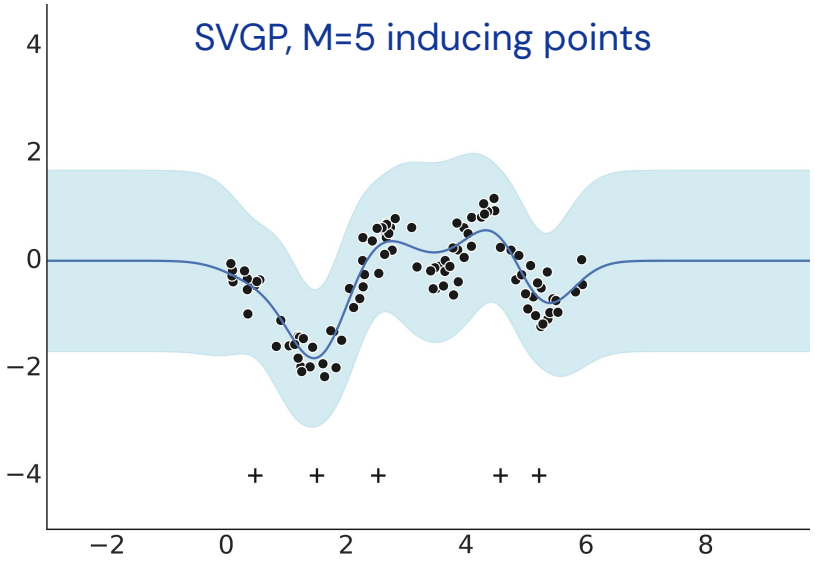




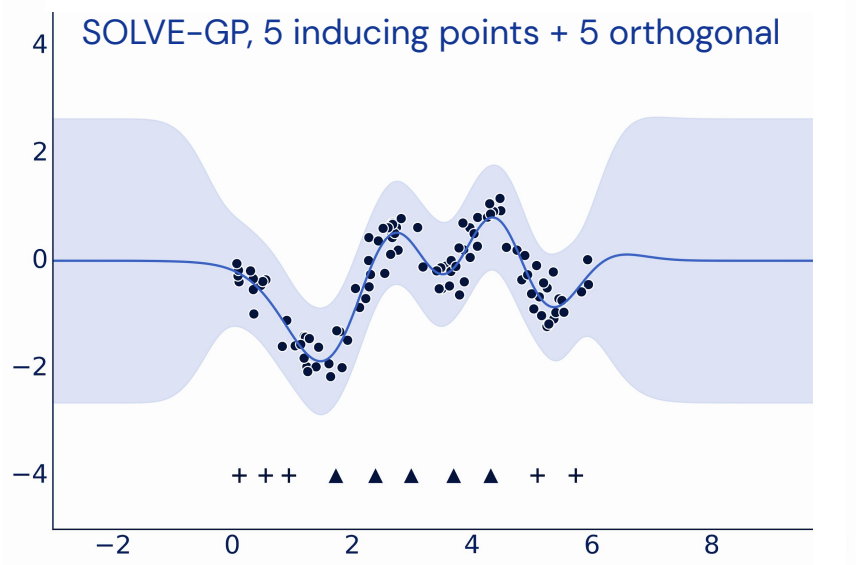
SOLVE-GP lower bound

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

SOLVE-GP: Orthogonal Inducing Points Are Cheaper



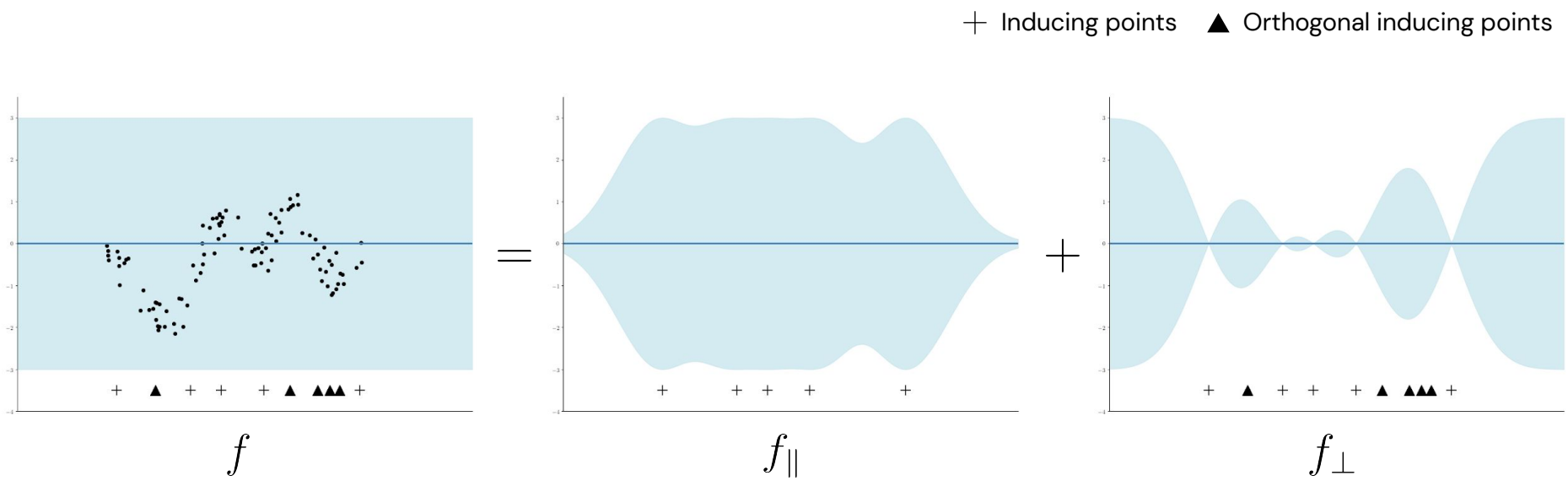
Cholesky: cM^3



Cholesky: $\mathbf{2}cM^3$

+ Inducing points ▲ Orthogonal inducing points

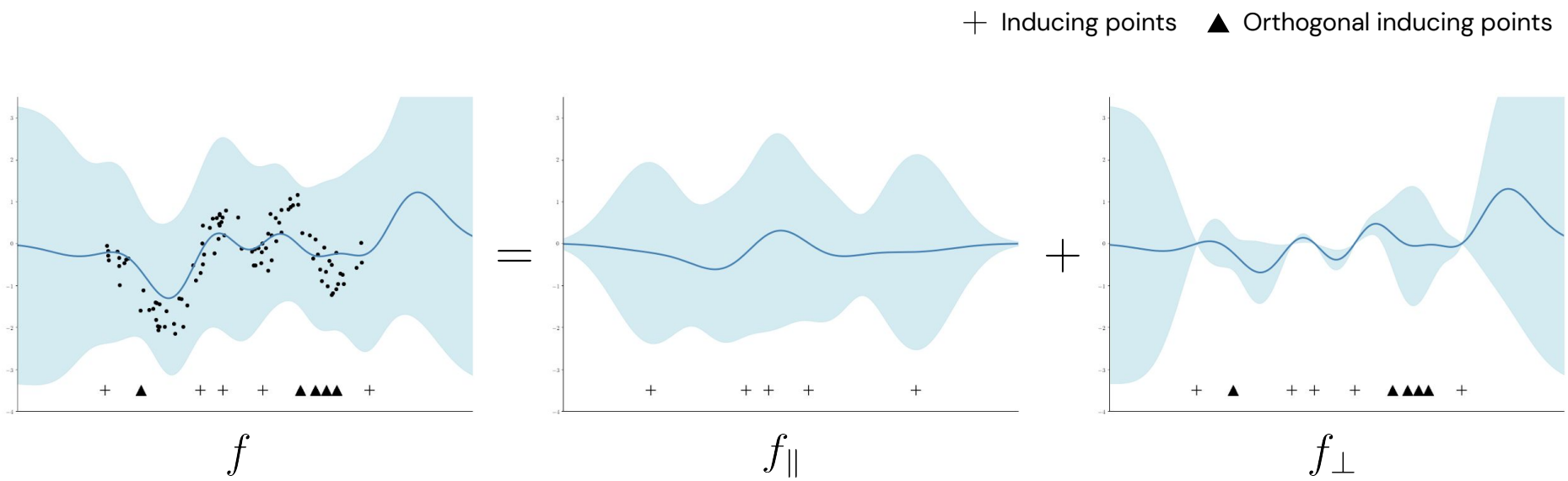
Posterior Approximation during Training



SOLVE-GP lower bound

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

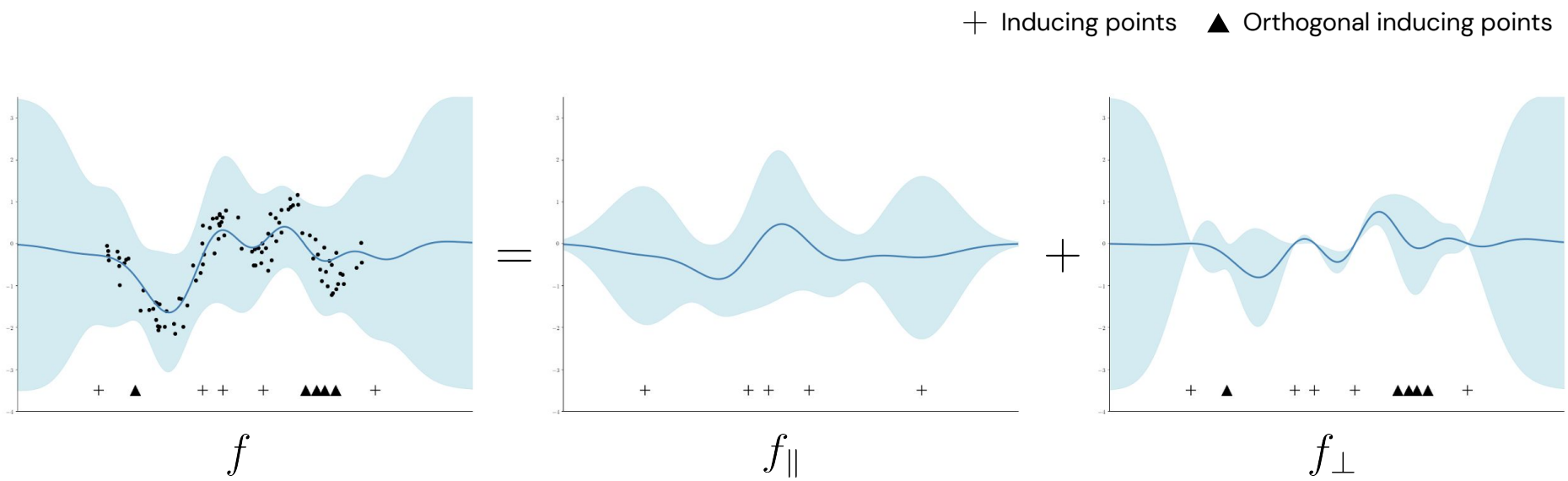
Posterior Approximation during Training



SOLVE-GP lower bound

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

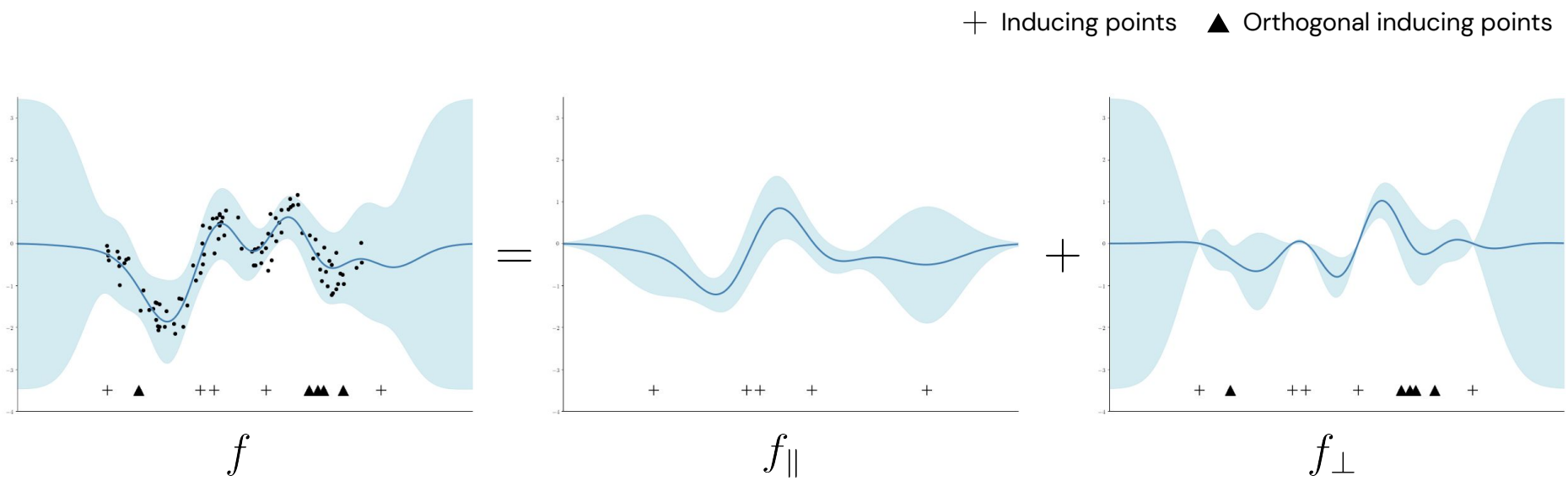
Posterior Approximation during Training



SOLVE-GP lower bound

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

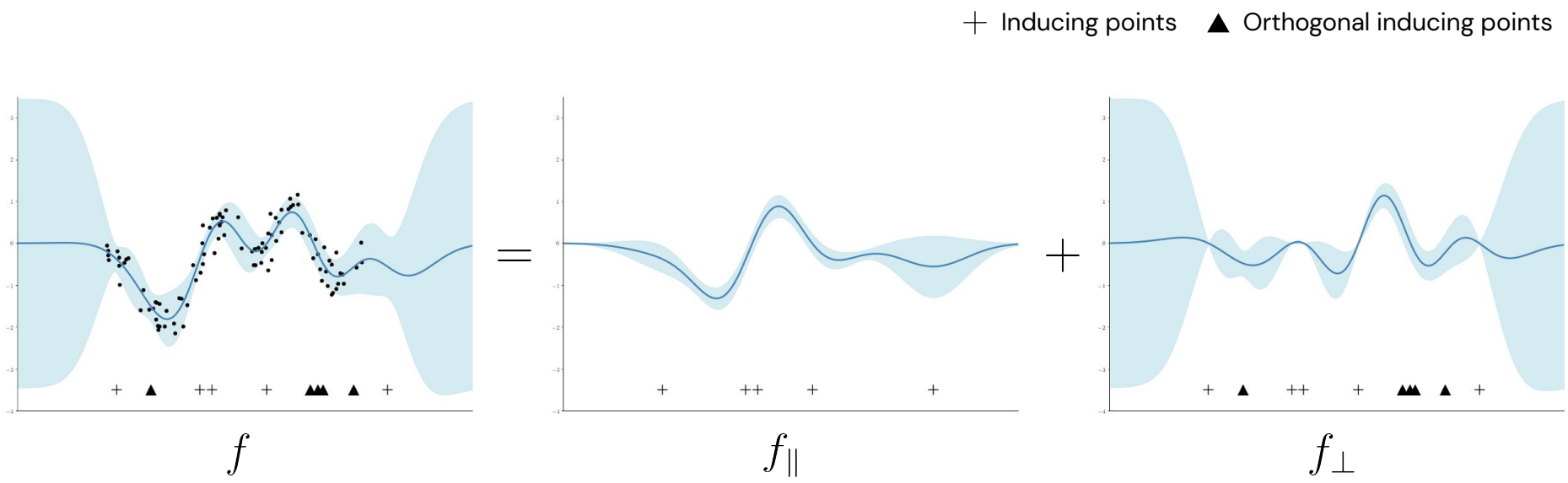
Posterior Approximation during Training



SOLVE-GP lower bound

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

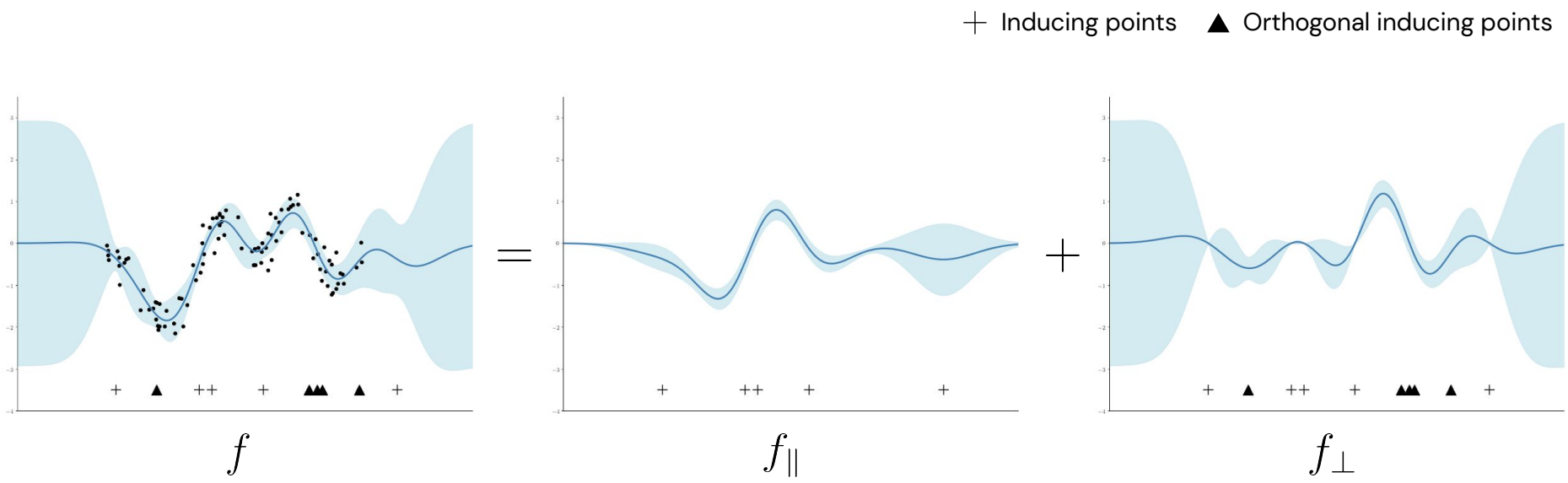
Posterior Approximation during Training



SOLVE-GP lower bound

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

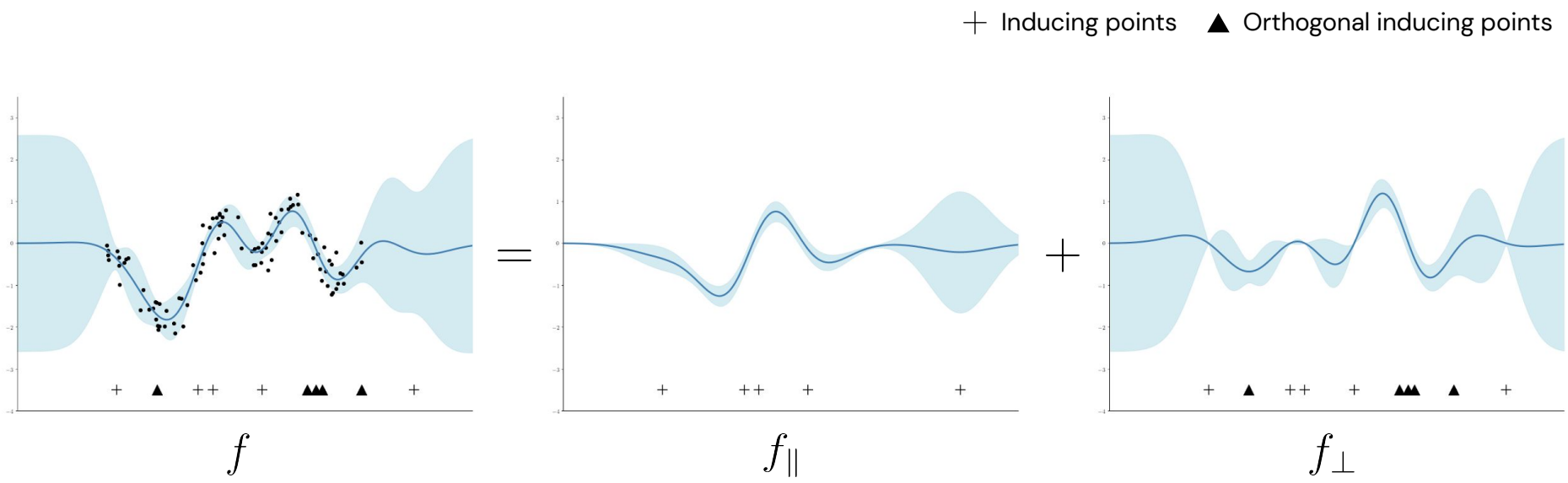
Posterior Approximation during Training



SOLVE-GP lower bound

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

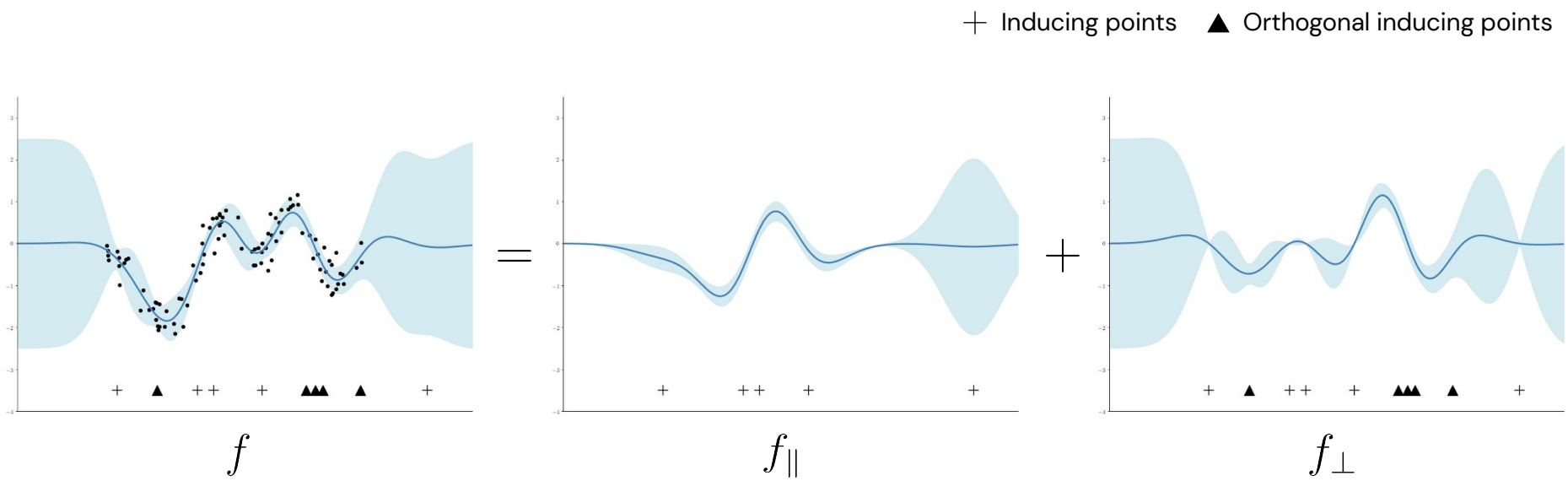
Posterior Approximation during Training



SOLVE-GP lower bound

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

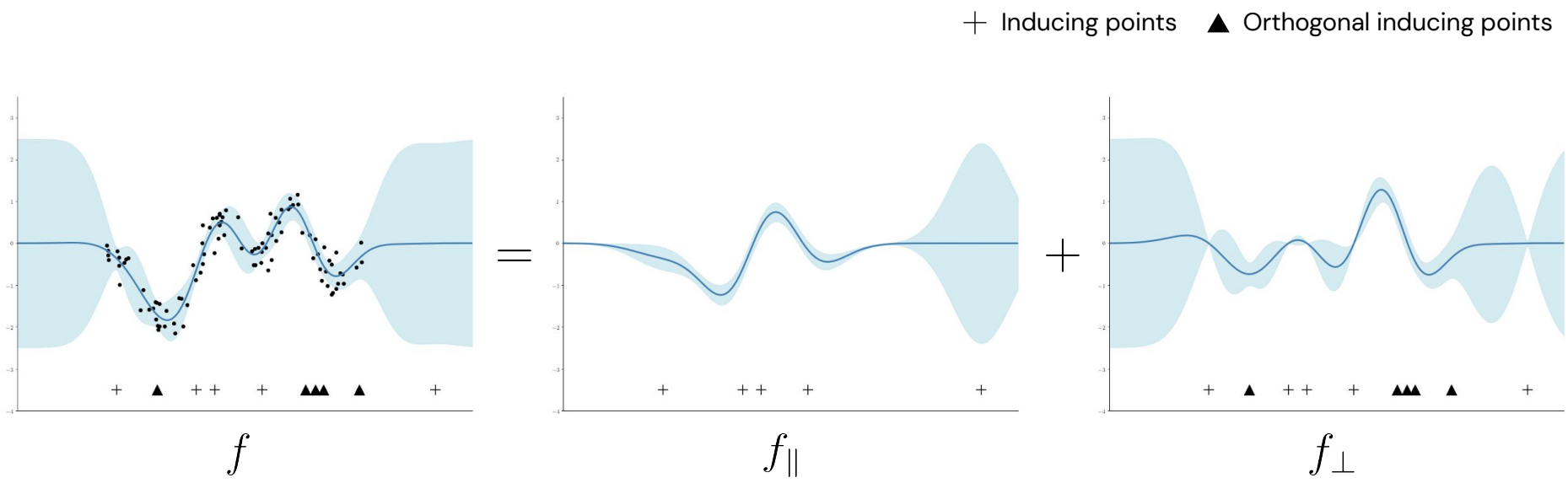
Posterior Approximation during Training



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

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

Understanding SOLVE-GP - Structured Covariance

We can express our variational approximation w.r.t. the original GP.

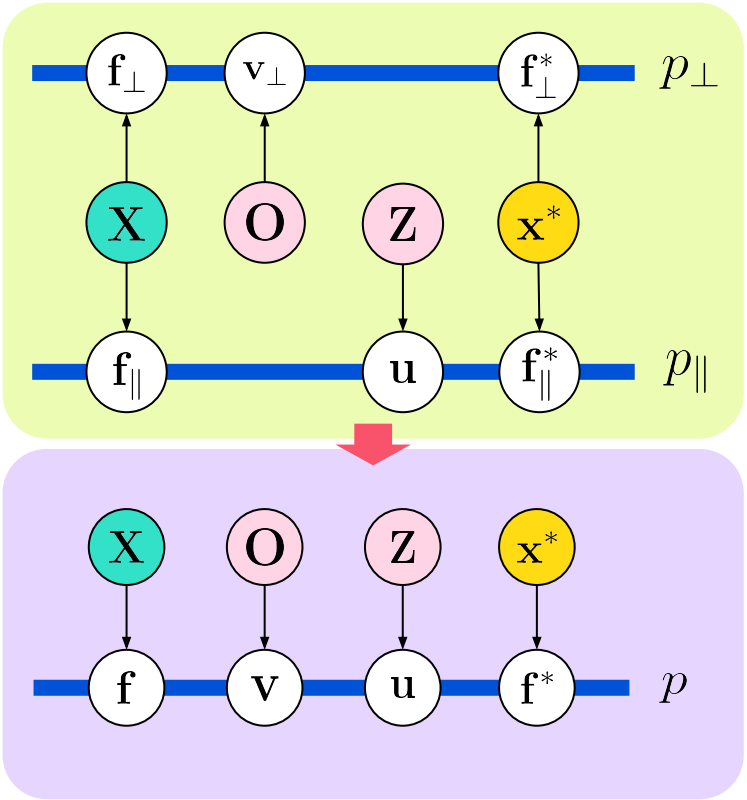
Applying change-of-variable:

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{K_{vu}}\mathbf{K_{uu}^{-1}} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}_\perp\end{bmatrix}$$

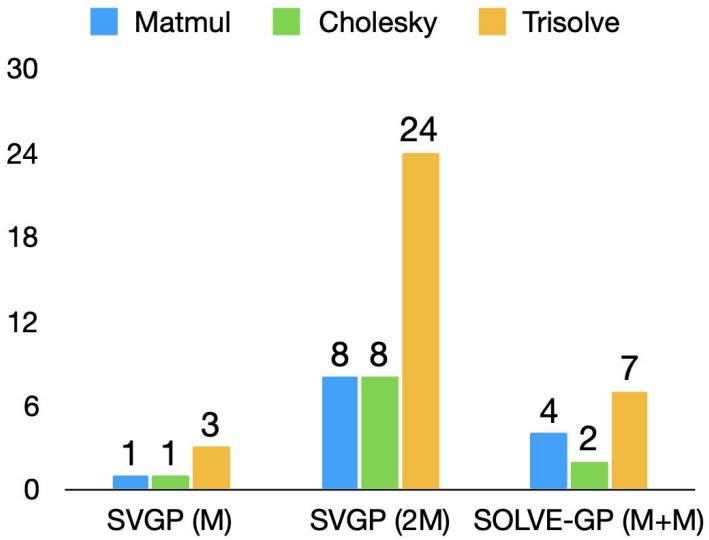
The covariance of $q(\mathbf{u}, \mathbf{v})$ induced by the orthogonal parameterization $q(\mathbf{u})q(\mathbf{v}_\perp)$ is

$$\mathbf{S}_{u,v} = \begin{bmatrix}\mathbf{S}_u & \mathbf{S}_u\mathbf{K_{uu}^{-1}}\mathbf{K_{uv}}\\ \mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{S}_u & \mathbf{S}_v + \mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{S}_u\mathbf{K_{vu}^{-1}}\mathbf{K_{uv}}\end{bmatrix}$$

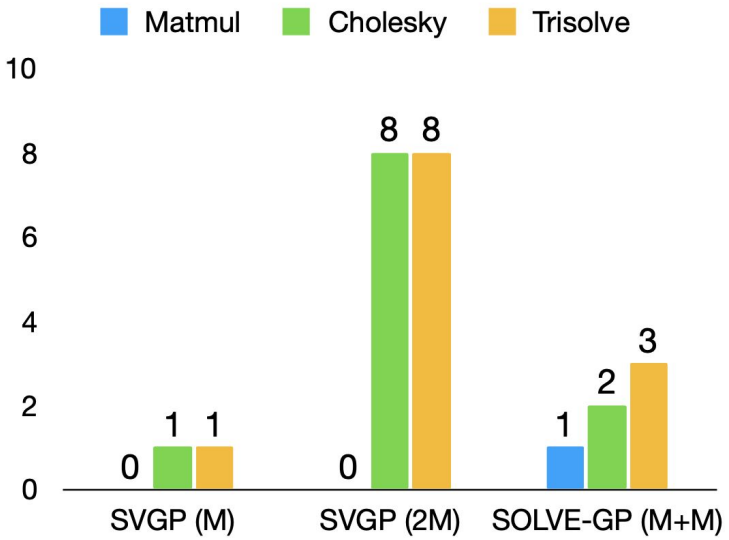
Decoupled inducing points (Salimbeni et al., 18) is equivalent to enforcing prior covariance in $q(\mathbf{v}_\perp)$



Understanding SOLVE-GP - Computational Benefits



(a) $N \approx M$

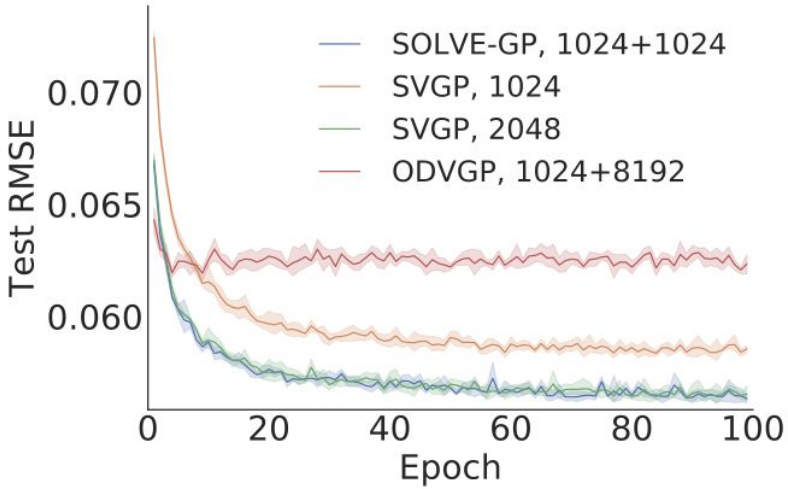


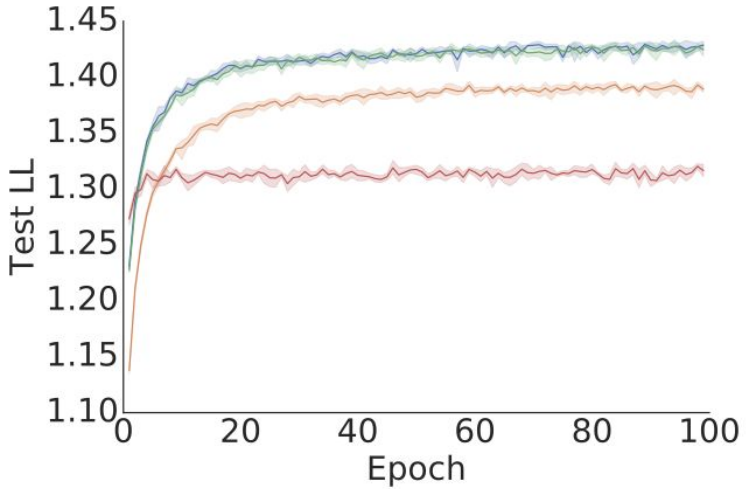
(b) $M \gg N$

N: mini-batch size; M: number of inducing points

Experiments - Large-scale Regression

HouseElectric (N=1,311,539, D=9)





- SOLVE-GP (1024+1024) has no performance loss compared to SVGP (2048).
- Both outperforms SVGP (1024).

Experiments - CIFAR 10 Classification

Convolutional Gaussian Processes

(Van der Wilk et al., 2017)

Previous 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 Processes

(Blomqvist et al., 2018)

Methods	SVGP	SOLVE-GP	SVGP
M	384, 384, 1K	384+384, 384 +384, 1K+1K	768,768,2 K*
Test LL	-0.88	-0.79	-0.82
Test Acc	78.76%	**80.3%**	**80.33%**
Time /iter	0.418	0.752	1.246

Experiments - CIFAR 10 Classification

Convolutional Gaussian Processes

(Van der Wilk et al., 2017)

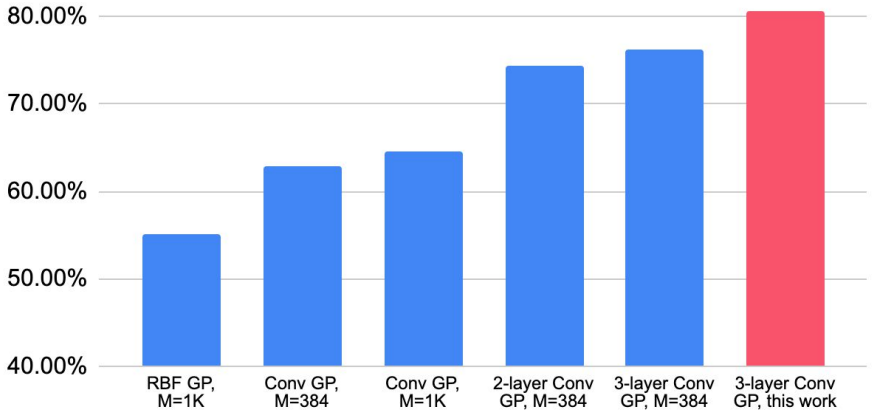
Previous SOTA: 64.6% -> 68.2%

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 Processes

(Blomqvist et al., 2018)

Previous 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).

Takeaways

- We introduce the idea of **orthogonal inducing points** to efficiently parameterize Gaussian process approximations.
- This leads to more **scalable** variational inference algorithms for GPs (SOLVE-GP).
- We report **state-of-the-art results** in training **large, hierarchical** GP models such as deep convolutional Gaussian processes.

Code: github.com/thjashin/solvegp
Paper, slides & video: jiaxins.io

References

- Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. AISTATS 2009.
- James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. AISTATS 2013.
- Ching-An Cheng and Byron Boots. Variational inference for Gaussian process models with linear complexity. NIPS 2017.
- Hugh Salimbeni, Ching-An Cheng, Byron Boots, and Marc Deisenroth. Orthogonally decoupled variational Gaussian processes. NeurIPS 2018.
- Mark van der Wilk, Carl Edward Rasmussen, and James Hensman. Convolutional Gaussian processes. NIPS 2017.
- Kenneth Blomqvist, Samuel Kaski, and Markus Heinonen. Deep convolutional Gaussian processes. arXiv:1810.03052, 2018.