

# Scalable Gaussian Processes with Billions of Inducing Inputs via Tensor Train Decomposition

Pavel Izmailov  
pi49@cornell.edu

Alexander Novikov  
sasha.v.novikov@gmail.com

Dmitry Kropotov  
dmitry.kropotov@gmail.com



## Summary

- Gaussian processes are powerful and elegant models, but exact inference requires  $\mathcal{O}(n^3)$  computations, where  $n$  is the number of training data
- We propose the Tensor Train GP (TT-GP) framework with linear complexity  $\mathcal{O}(n)$
- TT-GP allows to build flexible posterior approximations and train expressive deep kernels by using billions of inducing points for datasets containing millions of data points of dimensionality up to 10
- TT-GP achieves state-of-the-art results on several important benchmarks both with RBF and deep kernels

## Inducing Inputs and Structured Kernel Interpolation

- Inducing points  $(Z, u)$  are  $m$  imaginary points that compress information in data. The joint distribution of labels  $y$ , process values  $f \in \mathbb{R}^n$  at data points and  $u \in \mathbb{R}^m$  at inducing points is given by

$$p(y, f, u) = \prod_i p(y_i|f) \cdot p(f|u) \cdot p(u),$$

where  $p(f|u) = \mathcal{N}(f|K_{nm}K_{mm}^{-1}u, K_{nn} - K_{nm}K_{mm}^{-1}K_{nm}^T)$ ,  $p(u) = \mathcal{N}(u|0, K_{mm})$ . Here  $K_{mm} \in \mathbb{R}^{m \times m}$ ,  $K_{nm} \in \mathbb{R}^{n \times m}$ ,  $K_{nn} \in \mathbb{R}^{n \times n}$  are covariance matrices computed between  $Z$  and  $Z$ ,  $X$  and  $Z$ ,  $X$  and  $X$  respectively

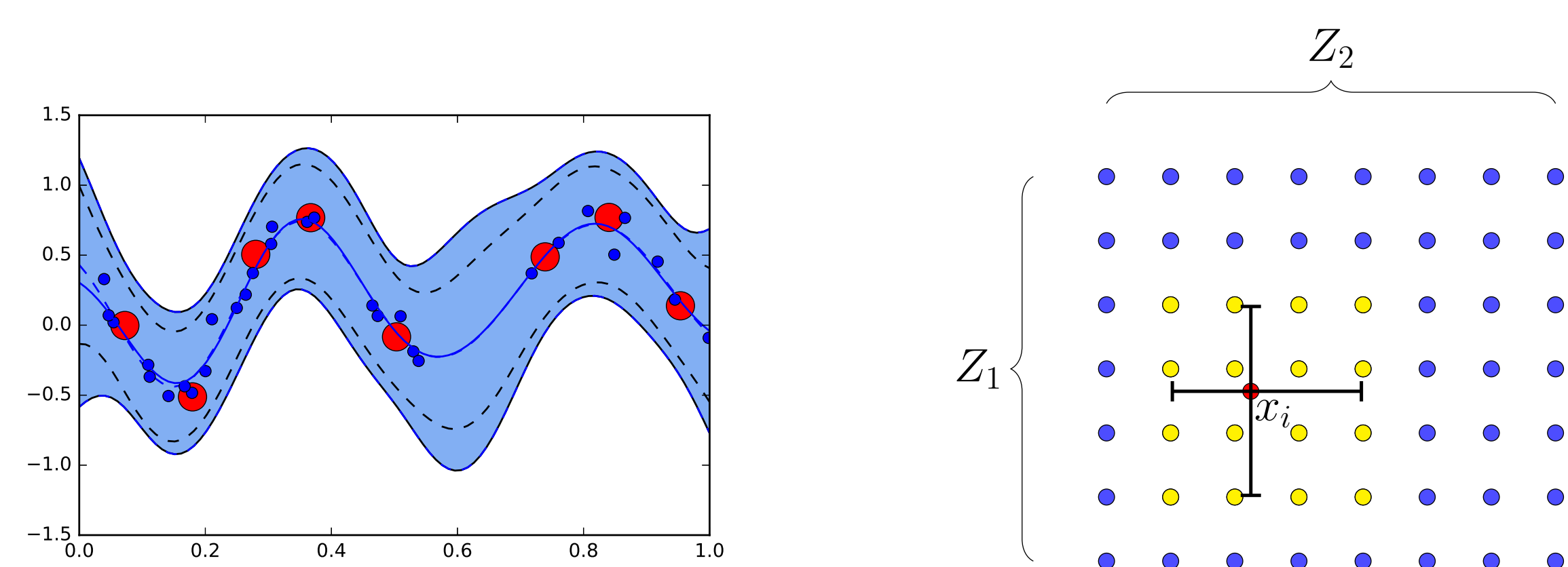
- Using the variational distribution of the form  $q(f, u) = p(f|u)q(u)$  with  $q(u) = \mathcal{N}(u|\mu, \Sigma)$  we can derive the variational lower bound

$$\log p(y) \geq \sum_i \mathbb{E}_q(f_i) \log p(y_i|f_i) - \text{KL}(q(u)||p(u))$$

- SKI (Wilson and Nickisch, 2015): set inducing points on a multi-dimensional grid

$$Z = Z^1 \times Z^2 \times \dots \times Z^D$$

Assume the kernel decomposes as  $k(x, x') = k^1(x^1, x'^1) \cdot k^2(x^2, x'^2) \cdot \dots \cdot k^D(x^D, x'^D)$ . Covariance matrix  $K_{mm}$  takes form  $K_{mm} = K_{m_1 m_1}^1 \otimes K_{m_2 m_2}^2 \otimes \dots \otimes K_{m_D m_D}^D$



- $\det(K_{mm})$  and  $K_{mm}^{-1}$  can be computed efficiently
- Inducing points can be considered as interpolation points for the kernel

$$k_i \approx K_{mm} w_i,$$

where  $k_i \in \mathbb{R}^m$  is the vector of covariances between the  $i$ -th training object and the inducing points,  $w_i \in \mathbb{R}^m$  is the vector of interpolation coefficients

- KISS-GP uses cubic convolutional interpolation for which

$$w_i = w_i^1 \otimes w_i^2 \otimes \dots \otimes w_i^D$$

## Gaussian Process ELBO

Evidence Lower Bound (Hensman et al., 2013) with KISS-GP approximation of  $k_i$ :

$$\log p(y) \geq \sum_{i=1}^n \left( \log \mathcal{N}(y_i | w_i^T \mu, \sigma^2) - \frac{1}{2\sigma^2} (\delta - w_i^T K_{mm} w_i) - \frac{1}{2\sigma^2} \text{tr}(w_i^T \Sigma w_i) \right) - \frac{1}{2} \left( \log \frac{\det(K_{mm})}{\det(\Sigma)} - m + \text{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right)$$

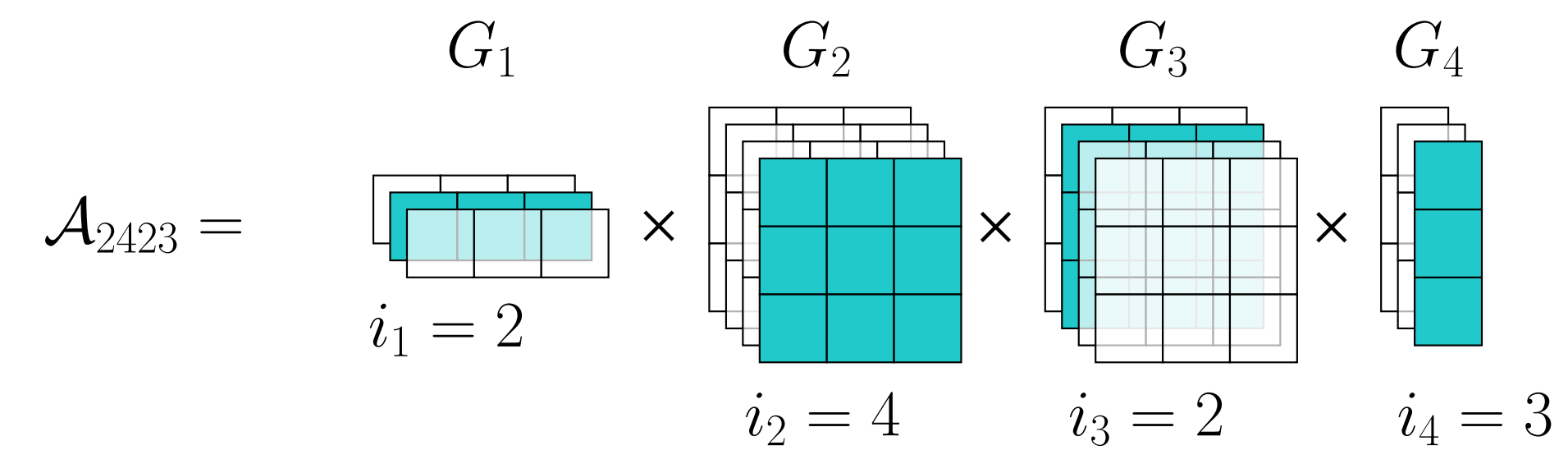
where  $\sigma^2$  is the noise variance,  $\delta$  is the prior variance of the process at any point and  $\mu \in \mathbb{R}^m$ ,  $\Sigma \in \mathbb{R}^{m \times m}$  are variational parameters

## Tensor Train Format

Tensor  $\mathcal{A}$  is said to be represented in Tensor Train (Oseledets, 2011) format if:

$$\mathcal{A}_{i_1 \dots i_d} = \underbrace{G_1[i_1]}_{1 \times r} \underbrace{G_2[i_2]}_{r \times r} \dots \underbrace{G_d[i_d]}_{r \times 1}$$

An example of computing one element of a 4-dimensional tensor:



- TT-format uses  $\mathcal{O}(dnr^2)$  memory to approximate a tensor with  $n^d$  elements
- Allows for efficient implementation of linear algebra operations

## TT-GP

- Set inducing points  $Z$  on a grid in the feature space
- Restrict  $\Sigma$  to be in a Kronecker product format  $\Sigma = \Sigma^1 \otimes \Sigma^2 \otimes \dots \otimes \Sigma^D$
- Represent  $\mu$  as a  $d$ -dimensional tensor, restrict to be in TT format with TT-ranks  $r$
- Maximize ELBO with respect to TT-cores of  $\mu$ , Kronecker factors of  $\Sigma$  using stochastic optimization

## Properties

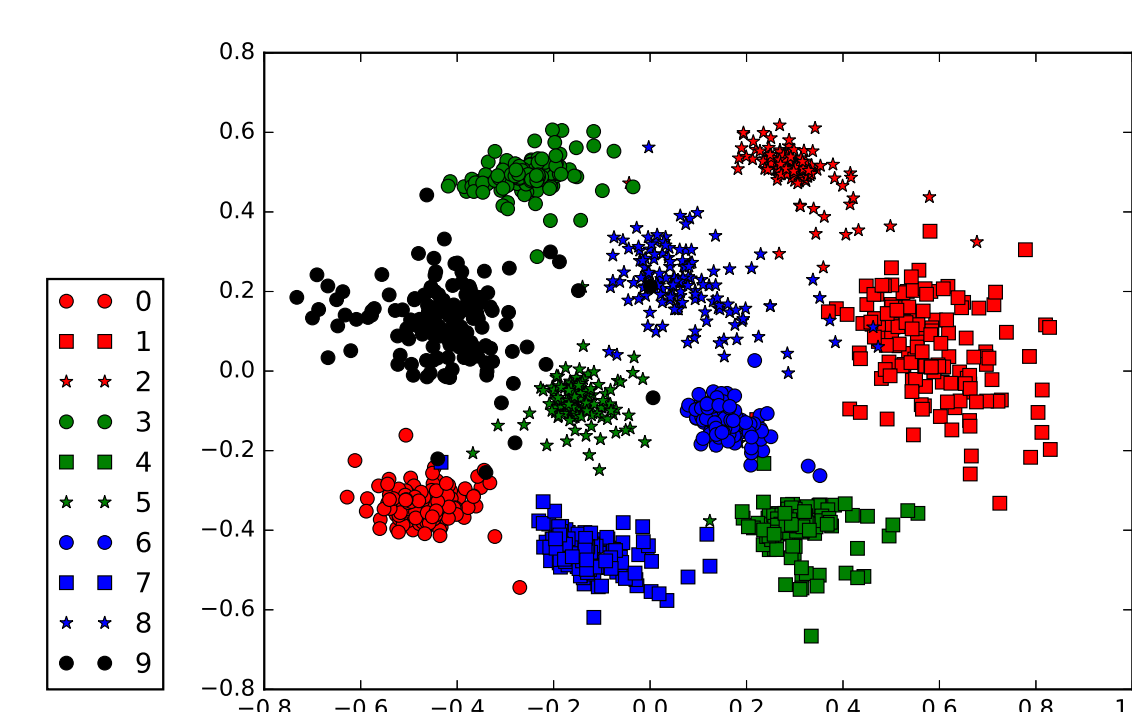
- Linear computational complexity in the size of the data  $\mathcal{O}(nDm^{1/D}r^2 + Dm^{1/D}r^3 + Dm^{3/D})$ . TT-ranks are in general on the scale of  $r \approx 10$ . Here  $m = m_0^D$ , where  $m_0$  is the number of inducing points per dimension
- TT-GP can be applied for  $n \approx 10^6$  and  $m \approx 10^{10}$

## RBF Kernel Experiments

Dataset		SVI-GP / KLSP-GP				TT-GP			
Name	$n$	$D$	acc.	$m$	$t$ (s)	acc.	$m$	$d$	$t$ (s)
Powerplant	7654	4	0.94	200	10	<b>0.95</b>	$35^4$	-	5
Protein	36584	9	0.50	200	45	<b>0.56</b>	$30^9$	-	40
YearPred	463K	90	0.30	1000	597	<b>0.32</b>	$10^6$	6	105
Airline	6M	8	0.665*	-	-	<b>0.694</b>	$20^8$	-	5200
svmguide1	3089	4	0.967	200	4	<b>0.969</b>	$20^4$	-	1
EEG	11984	14	<b>0.915</b>	1000	18	0.908	$12^{10}$	10	10
covtype bin	465K	54	0.817	1000	320	<b>0.852</b>	$10^6$	6	172

Comparison with SVI-GP (Hensman et al., 2013) on regression and classification tasks

## Deep Kernel Experiments



Embedding learned by TT-GP with a deep kernel on *digits* dataset

Dataset		SV-DKL	DNN		TT-GP		
Name	n	acc.	acc.	$t$ (s)	acc.	$d$	$t$ (s)
Airline	6M	0.781	0.780	1055	<b>0.788 ± 0.002</b>	2	1375
CIFAR-10	50K	—	<b>0.915</b>	166	0.908 ± 0.003	9	220
MNIST	60K	—	0.993	23	<b>0.9936 ± 0.0004</b>	10	64

Comparison with SV-DKL (Wilson et al., 2016) and stand-alone DNN

## References

- J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290. AUAI Press, 2013.
- I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.
- A. G. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015.
- A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016.