Semi-Parametric Inducing Point Networks and Neural Processes

Richa Rastogi, Yair Schiff, Alon Hacohen, Zhaozhi Li, Ian Lee, Yuntian Deng, Mert R. Sabuncu, Volodymyr Kuleshov

ICLR
May 2023

Semi-parametric setup:

- We have access to training set at inference time:

$$\mathcal{D}_{train} = \{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^{n}$$

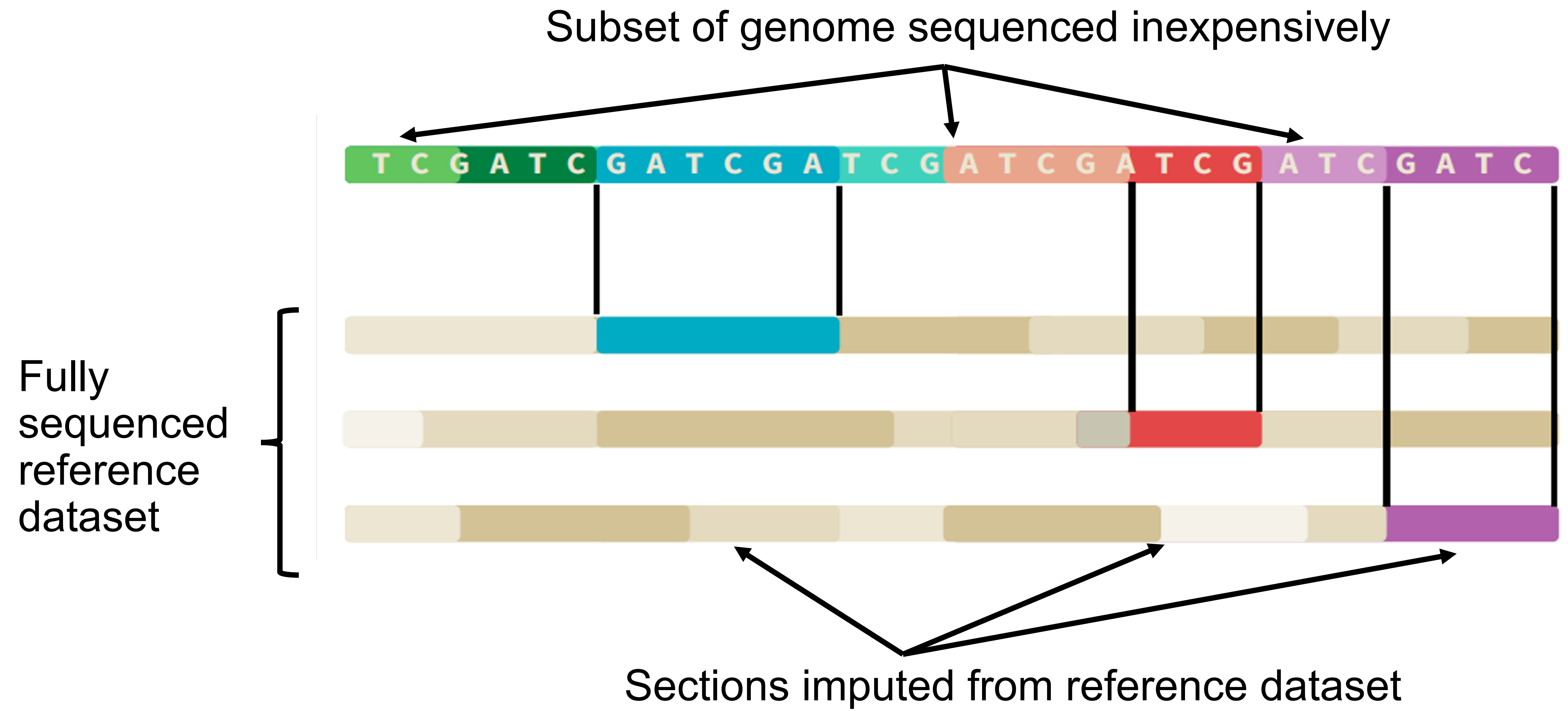
- Goal is to learn parametric mapping conditioned on this dataset:

$$\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \mathcal{D}_{train})$$

- Most **parametric models scale superlinearly** in size of dataset (e.g., Transformers[1] scale quadratically).

- Meta-learning tasks benefit from conditioning on **larger contexts.**

1. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

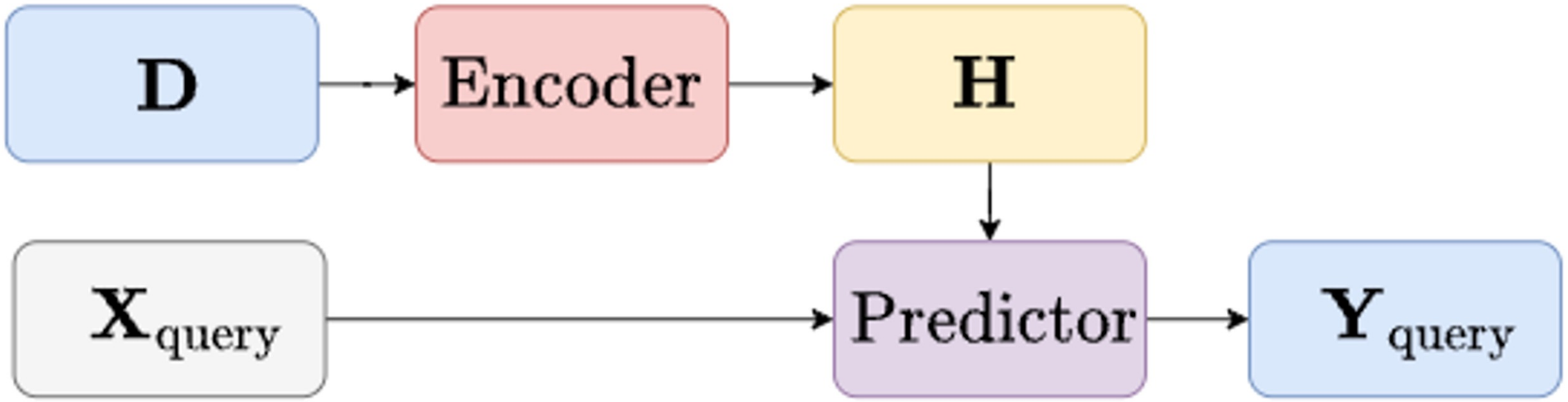


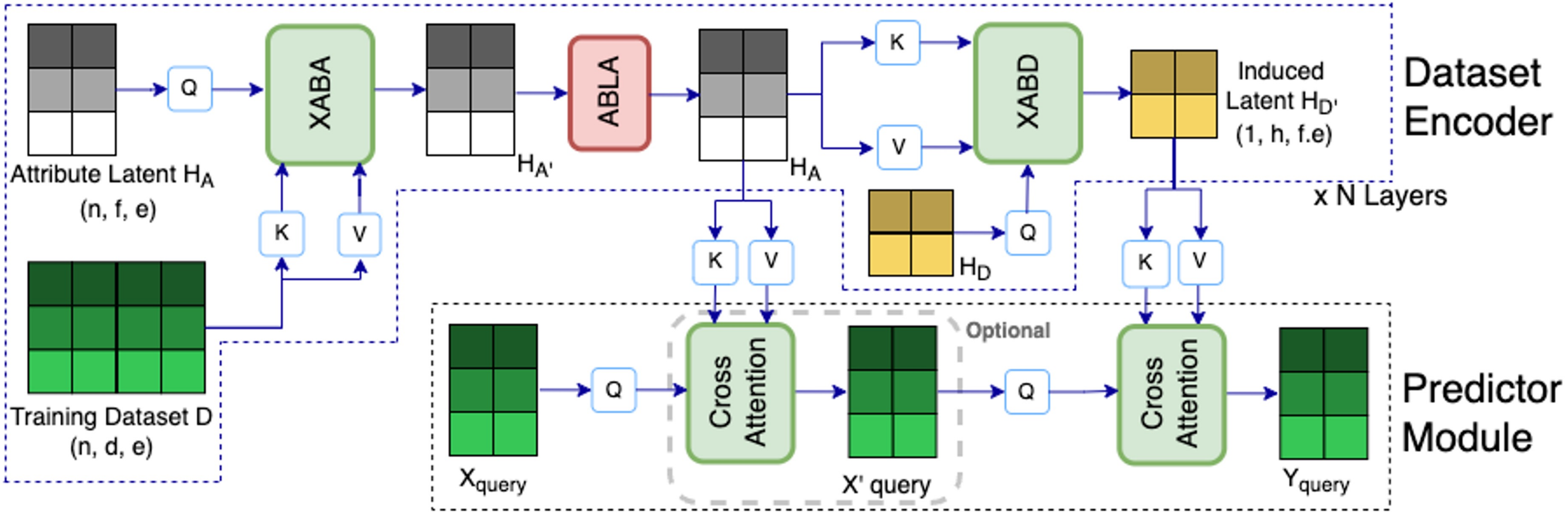
Motivating example: Parametric models are poor fit for large genome sequence imputation and cannot scale to larger reference datasets

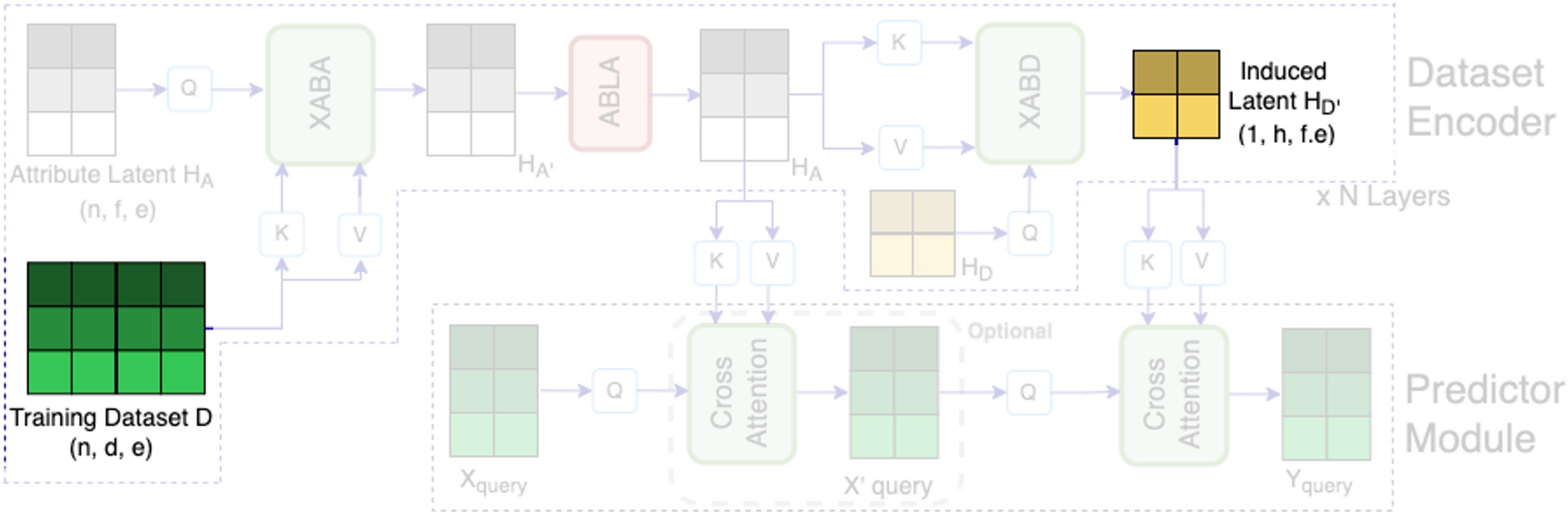
Semi-
Parametric
Inducing Point
Networks

- **Linear time and space complexity** in the size and the dimension of the data during training.
- Neural Processes architecture that supports **larger context sizes**.
- State-of-the-art results on **genotype imputation** task.

SPIN Overview




Dataset Encoder
Attribute Latent H_A
(n, f, e)
Q
K
V
XABA
$H_{A'}$
ABLA
H_A
K
V
XABD
Q
H_D
Induced Latent $H_{D'}$
(1, h, f.e)
x N Layers
Training Dataset D
(n, d, e)
K
V
K
V
Optional
X_{query}
Q
Cross Attention
X' query
Q
Cross Attention
Y_{query}
Predictor Module



Attribute Latent H_A
(n, f, e)
Q
XABA
K
V
$H_{A'}$
ABLA
H_A
K
V
XABD
Q
H_D
Induced
Latent $H_{D'}$
(1, h, f.e)
Dataset
Encoder
x N Layers
Training Dataset D
(n, d, e)
K
V
K
V
Q
Cross
Attention
Optional
Q
Cross
Attention
X_{query}
X' query
Y_{query}
Predictor
Module

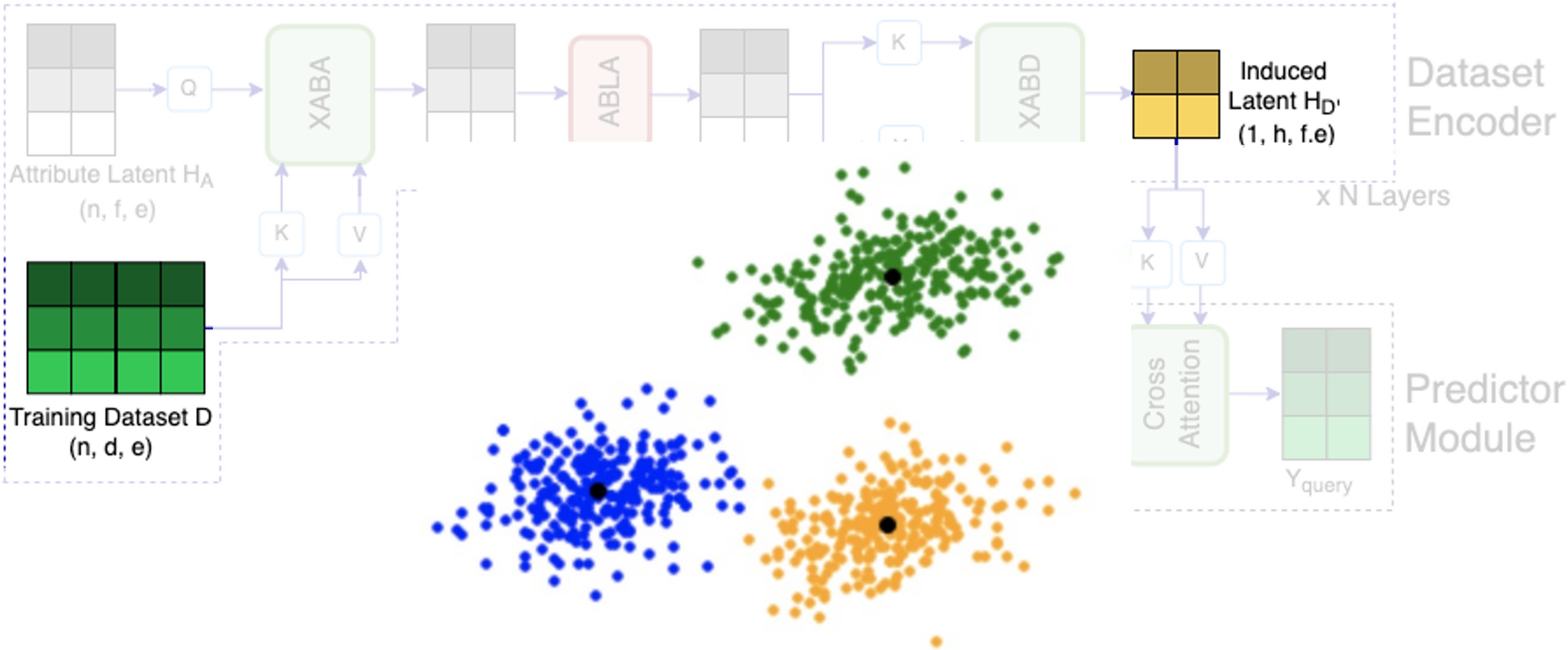
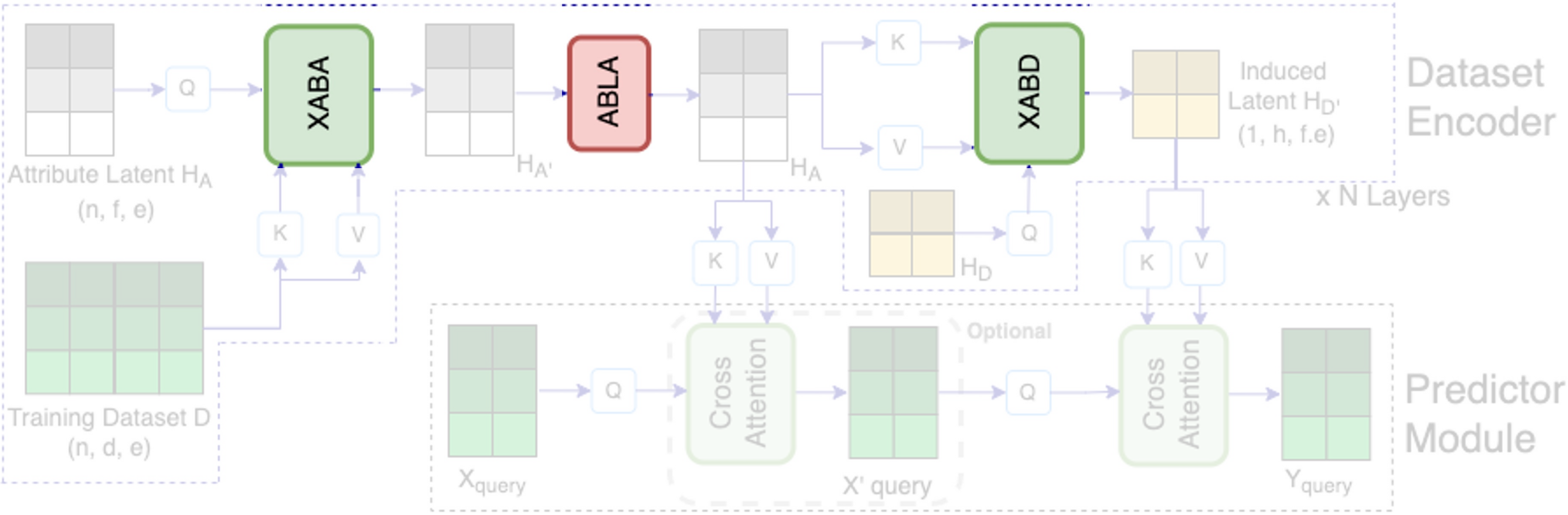
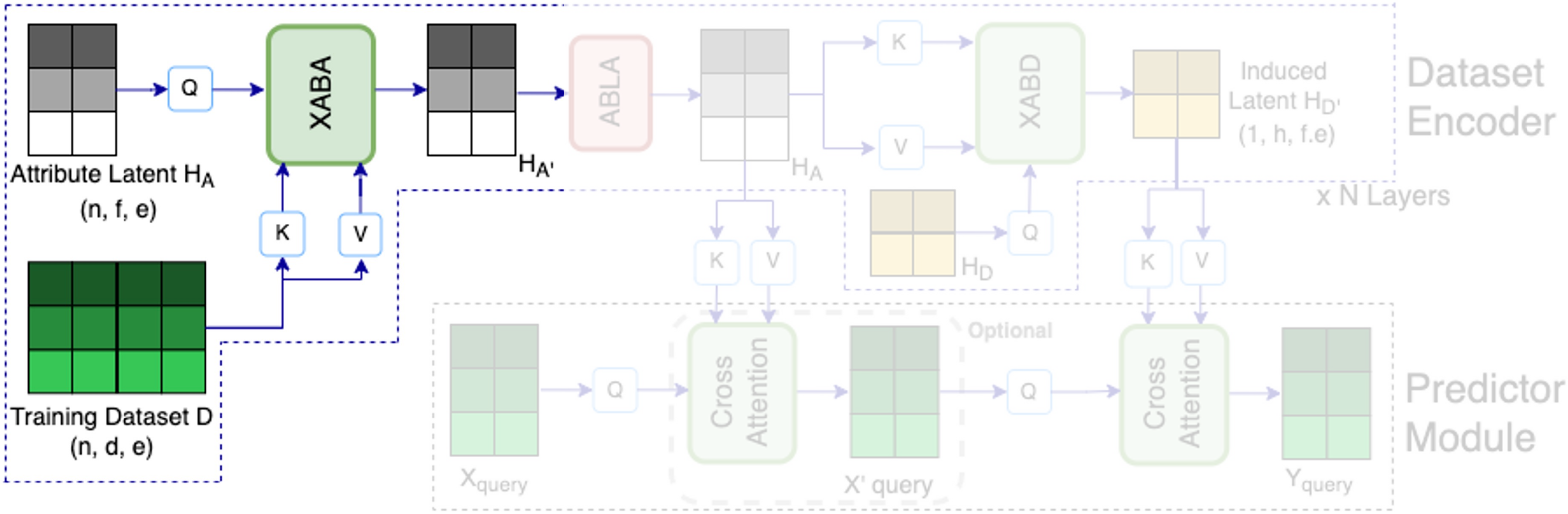
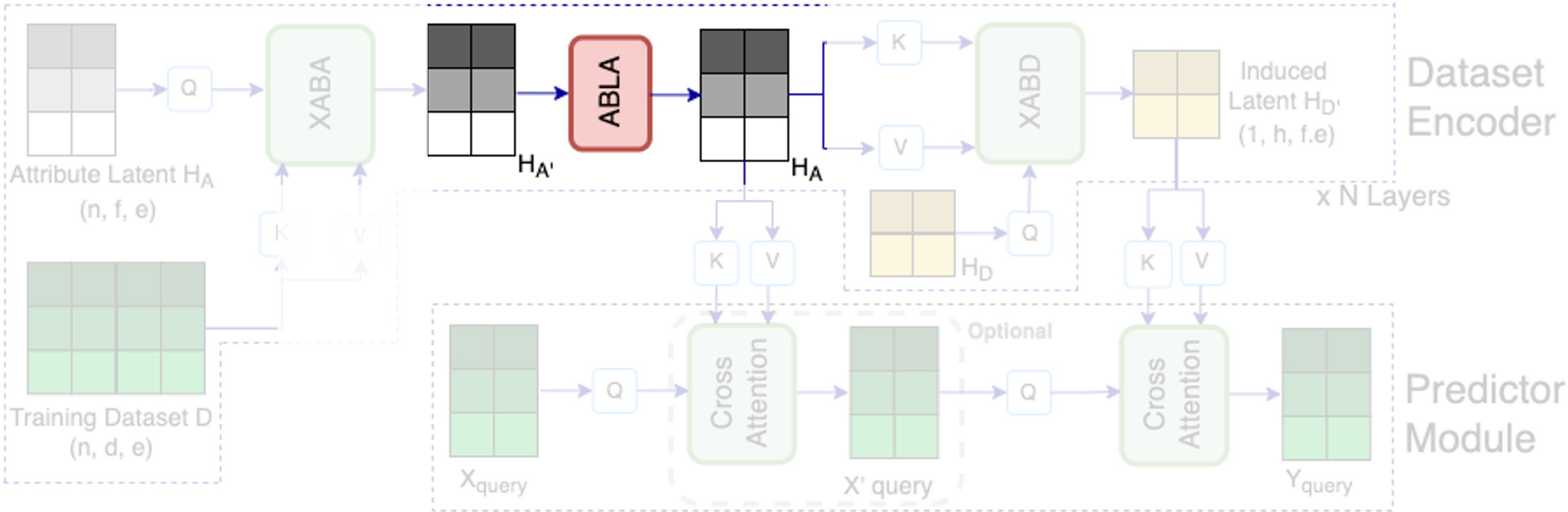


Image credit: https://www.geeksforgeeks.org/ml-k-means-algorithm/

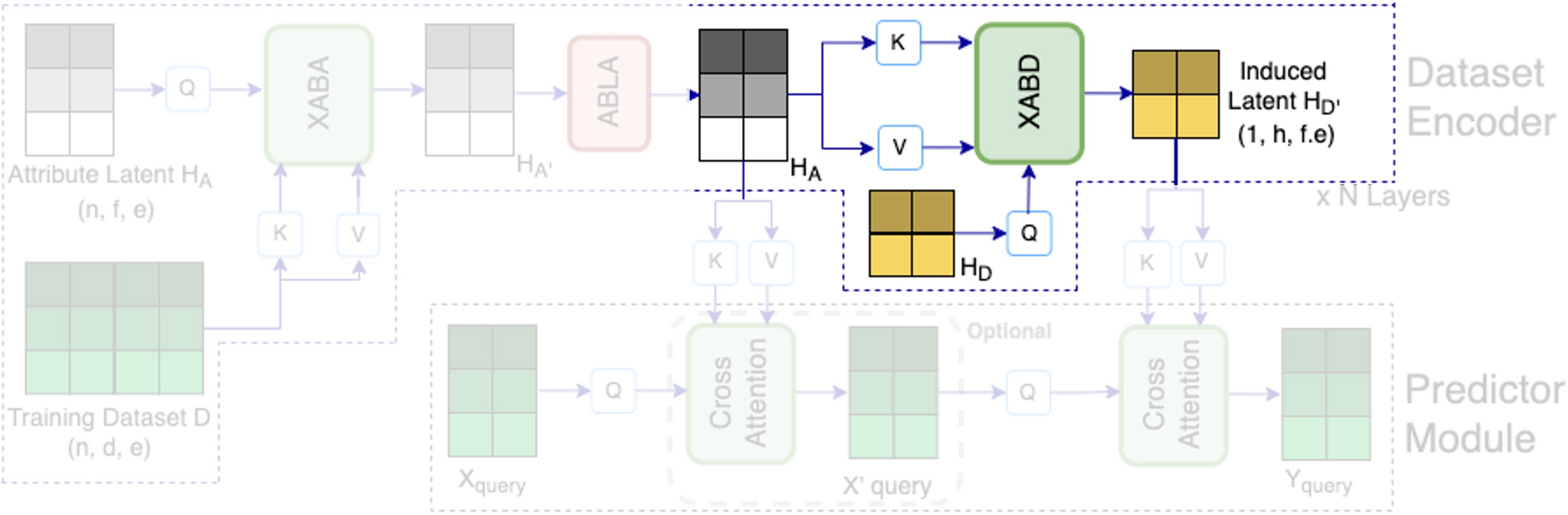

Attribute Latent H_A
(n, f, e)
Q
XABA
K
V
Training Dataset D
(n, d, e)
$H_{A'}$
ABLA
H_A
K
V
XABD
Q
H_D
Induced
Latent $H_{D'}$
(1, h, f.e)
Dataset
Encoder
x N Layers
K
V
X_{query}
Q
Cross
Attention
Optional
X' query
Q
Cross
Attention
K
V
Y_{query}
Predictor
Module



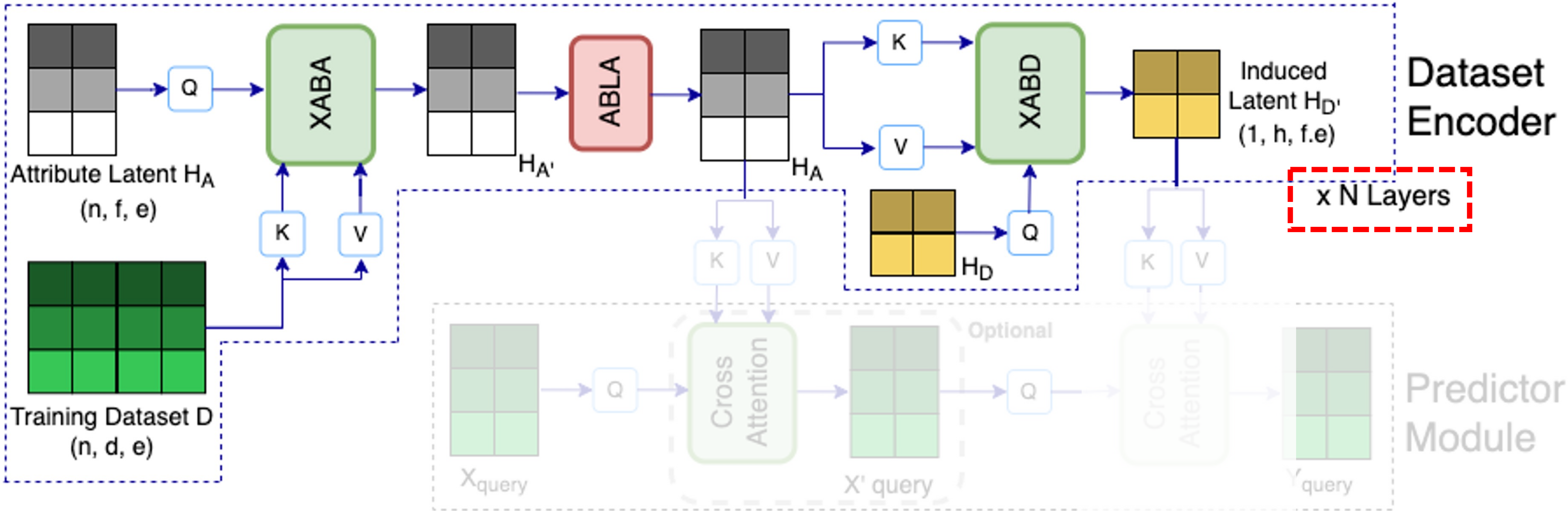
<u>Cross</u>-<u>A</u>ttention <u>B</u>etween <u>A</u>ttributes (XABA): Reduce dimensionality of datapoints



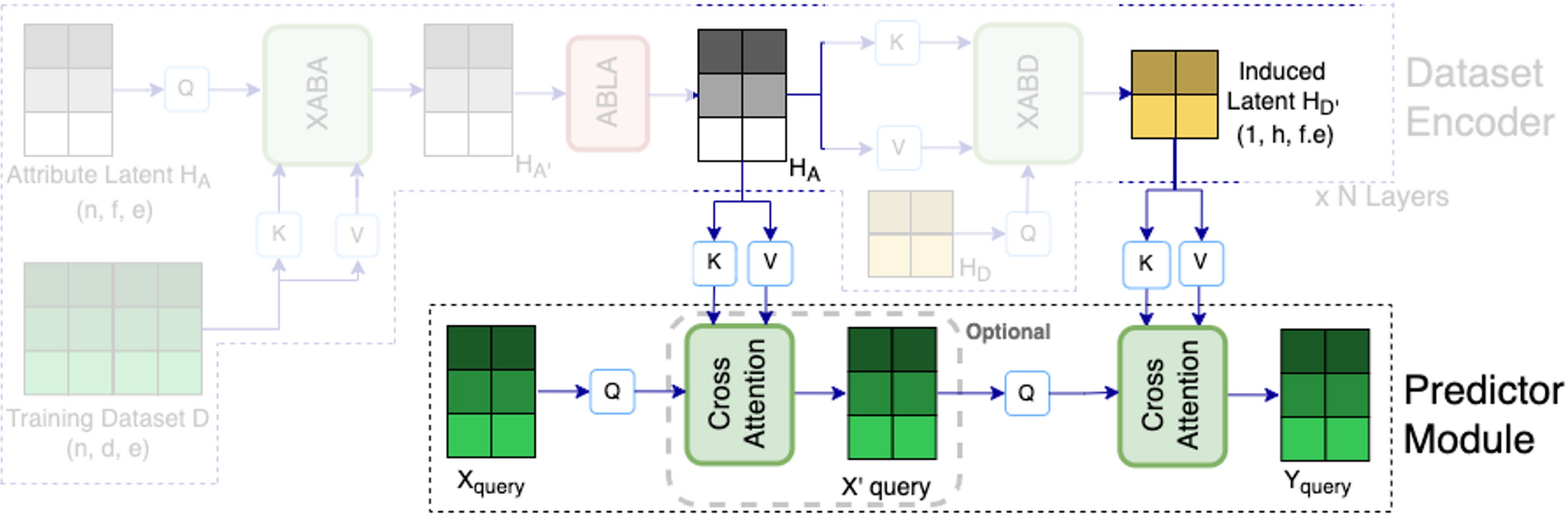
(Self-)<u>A</u>ttention <u>B</u>etween <u>L</u>atent <u>A</u>ttributes (ABLA): Enables inducing points to refine internal representations



Cross-Attention Between Datapoints (XABD): Generate inducing points that reduce context size



We stack multiple SPIN layers to form the complete **Dataset Encoder**



Predictor Module: Query refined inducing points; computation is *constant time* with respect to reference dataset size

Applying SPIN to Neural Processes…

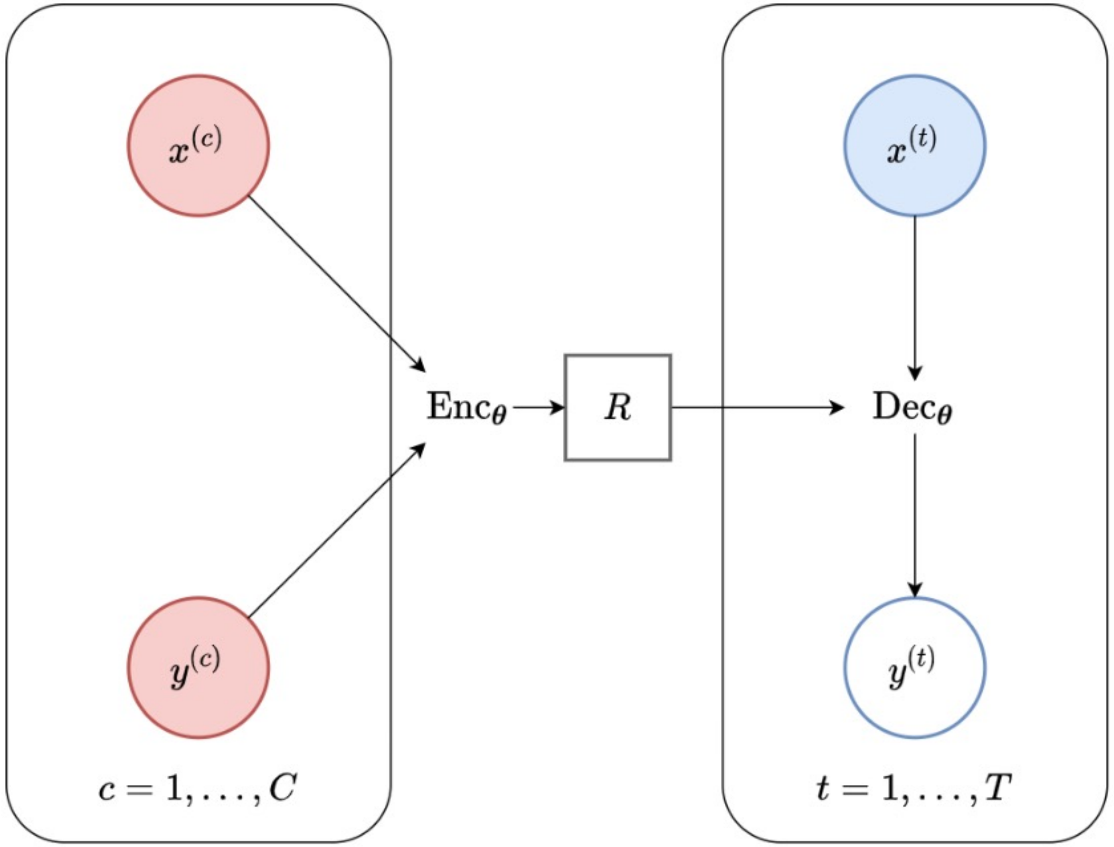
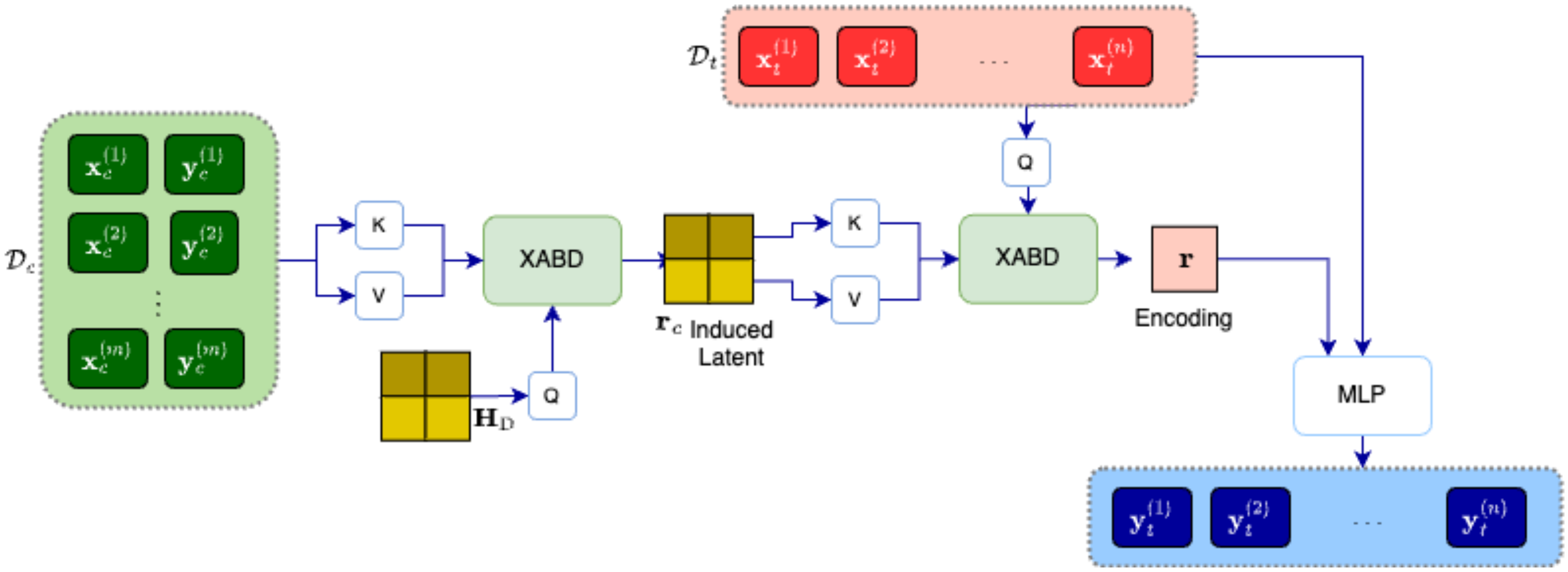


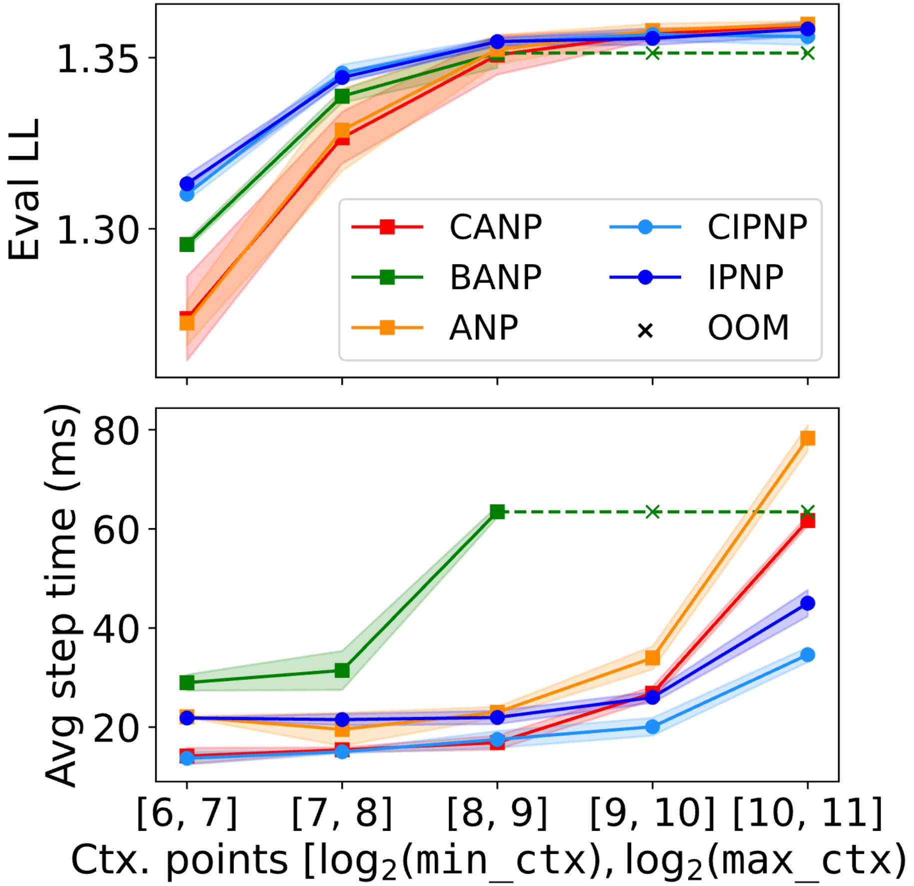
Fig. 5 High level computational graph of the Neural Process Family.

Image credit: Dubois, Yann and Gordon, Jonathan and Foong, Andrew YK. "Neural Process Family." (2020). http://yanndubs.github.io/Neural-Process-Family

Inducing Point Neural Processes (IPNP)



IPNP better scales to larger contexts



SOTA results on genome imputation

Table 3: Performance Summary on Genomic Sequence Imputation. (∗) represents parametric models. A difference of 0.5% is statistically significant at pvalue 0.05.

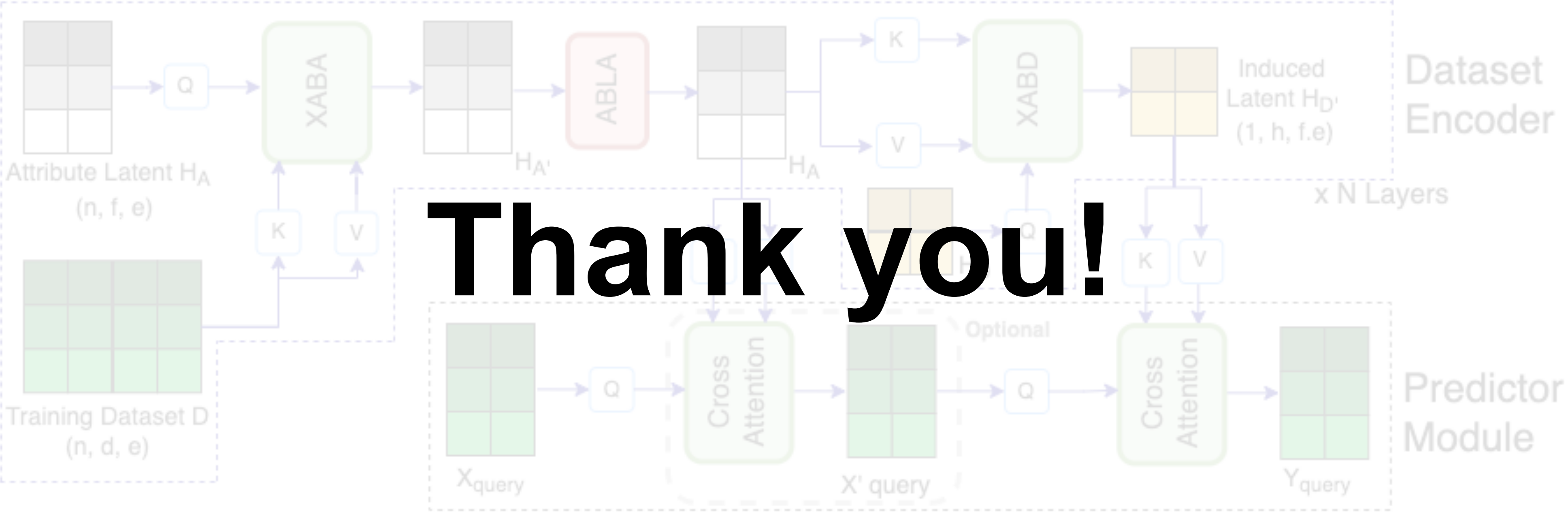
	GBT*	MLP*	KNN	Beagle	NPT-16	Set-TF-16	SPIN-16
Pearson R^2 ↑	87.63	95.31	89.70	95.64	95.84 ±0.06	**95.97±0.09**	95.92 ±0.12
Param Count↓	-	65M	-	-	16.7M	33.4M	**8.1M**

SPIN **outperforms state-of-the-art**, widely adopted software (Beagle) and is **more efficient** than alternative Transformer-based approaches (NPT, Set-TF)

Summary

- **SPIN is linear time and space complexity** in the size and the dimension of the data.
- **IPNP** is uncertainty aware, meta-learning algorithm that **scales to larger context sizes**.
- SPIN achieves **state-of-the-art results on genome imputation task.**

Thank you!



May 2023