Kernel-Based Multi-channel PolyCovNet

AI Hackathon Challenge I

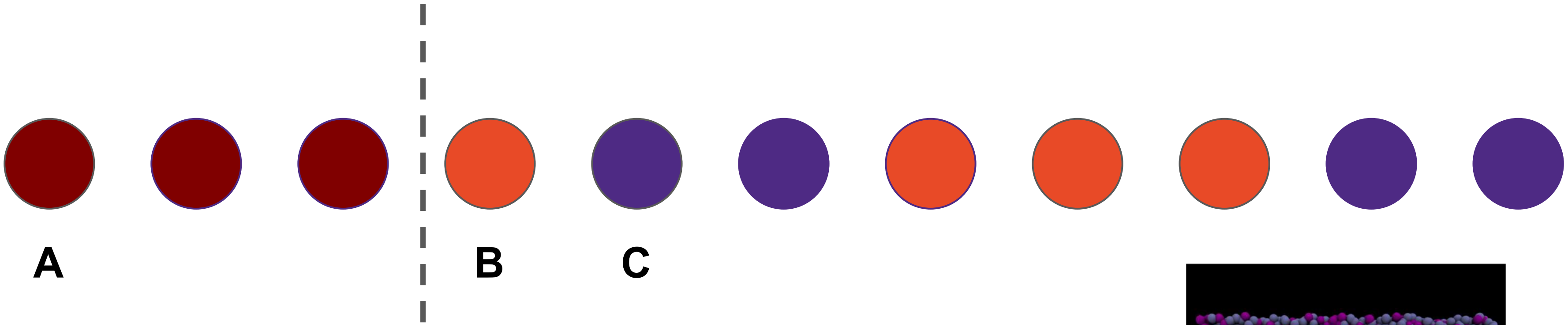
Team Borides

Kastan Day, Ruijie Zhu, Aria Coraor, Seonghwan Kim, Jiahui Yang

Contents

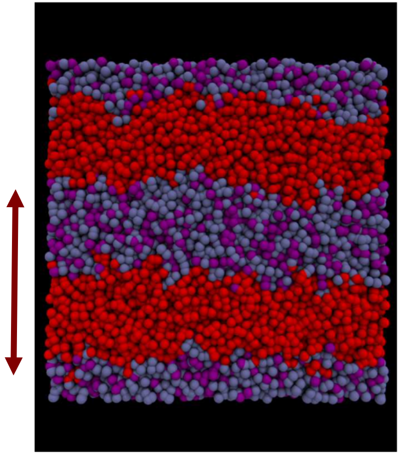
Problem Restatement

Predicting the lamellar period using monomer sequences and interaction parameters

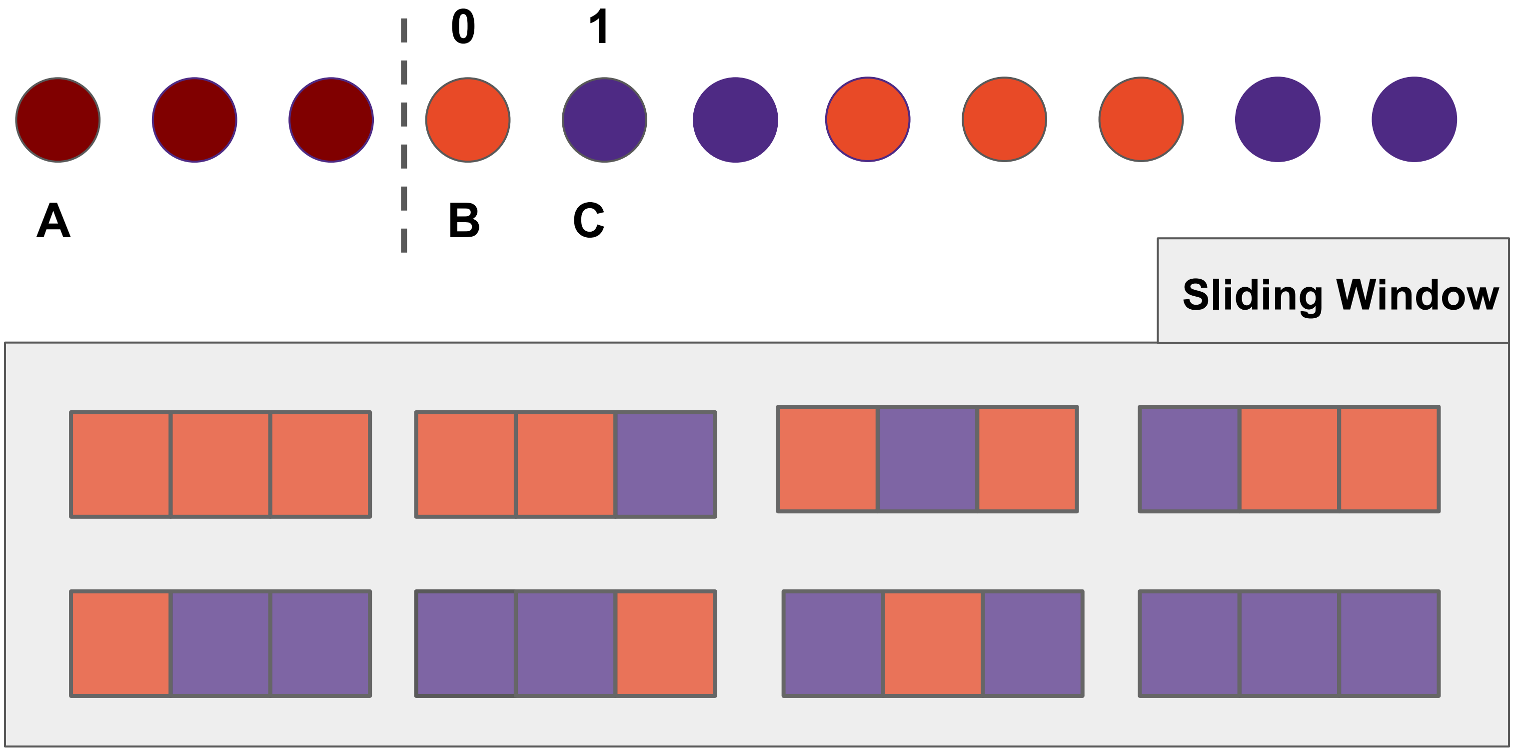


Interaction Parameter

Lamellar Period



Sliding window - extract monomer sequence features



Sliding Window

Features

window size

2

[0,0]
[0,1]
[1,0]
[1,1]

$2^2 = 4$

3

[0,0,0]
[0,0,1]
[0,0,0]
[0,0,0]
[0,0,1]
[0,0,0]
[0,0,0]
[0,0,1]

$2^3 = 8$

4

[0, 0, 0, 0]
[0, 0, 0, 1]
[0, 0, 1, 0]
[0, 0, 1, 1]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 1, 0]
[0, 1, 1, 1]
[1, 0, 0, 0]
[1, 0, 0, 1]
[1, 0, 1, 0]
[1, 0, 1, 1]
[1, 1, 0, 0]
[1, 1, 0, 1]
[1, 1, 1, 0]
[1, 1, 1, 1]

$2^4 = 16$

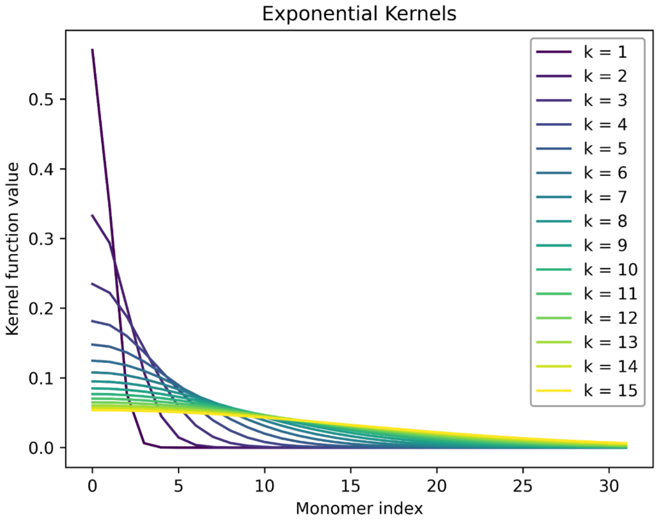
5

[0, 0, 0, 0, 0]
[0, 0, 0, 0, 1]
[0, 0, 0, 1, 0]
[0, 0, 0, 1, 1]
[0, 0, 1, 0, 0]
[0, 0, 1, 0, 1]
[0, 0, 1, 1, 0]
[0, 0, 1, 1, 1]
[0, 1, 0, 0, 0]
[0, 1, 0, 0, 1]
[0, 1, 0, 1, 0]
[0, 1, 0, 1, 1]
[0, 1, 1, 0, 0]
[0, 1, 1, 0, 1]
[0, 1, 1, 1, 0]
[0, 1, 1, 1, 1]
[1, 0, 0, 0, 0]
[1, 0, 0, 0, 1]
[1, 0, 0, 1, 0]
[1, 0, 0, 1, 1]
[1, 0, 1, 0, 0]
[1, 0, 1, 0, 1]
[1, 0, 1, 1, 0]
[1, 0, 1, 1, 1]
[1, 1, 0, 0, 0]
[1, 1, 0, 0, 1]
[1, 1, 0, 1, 0]
[1, 1, 0, 1, 1]
[1, 1, 1, 0, 0]
[1, 1, 1, 0, 1]
[1, 1, 1, 1, 0]
[1, 1, 1, 1, 1]

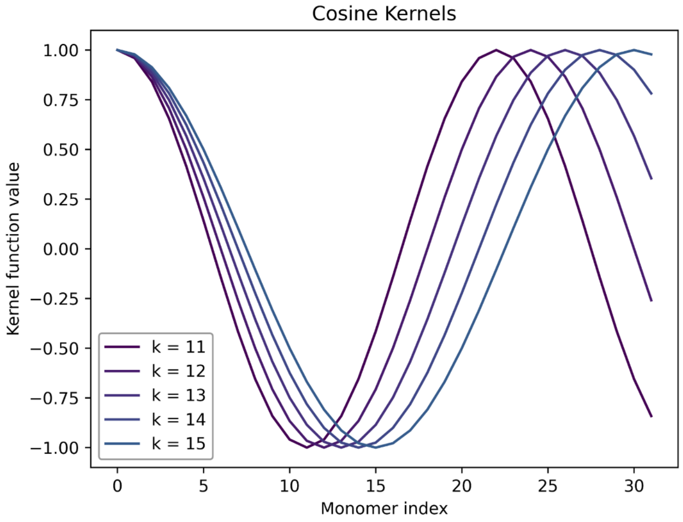
$2^5 = 32$

Non-linear kernel functions - preprocess monomer sequences

1. Exponential kernels $\exp(\frac{x^2}{2k^2})$



2. Cosine kernels $\cos(\frac{\pi x}{k})$

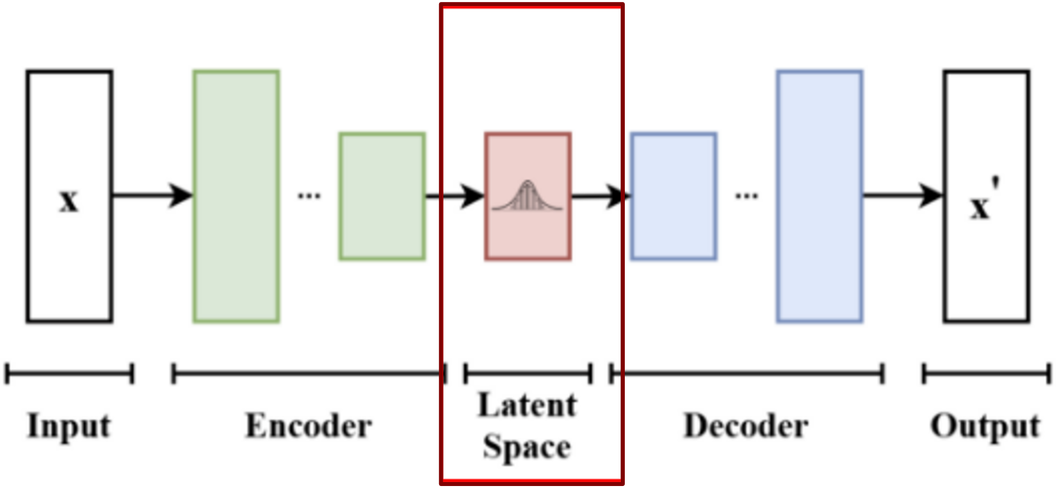


Applying the above kernels on monomer sequences ➡ Non-linearity

Variational autoencoder - extract features from monomer sequences

Monomer sequences

BBBBBCCCCC

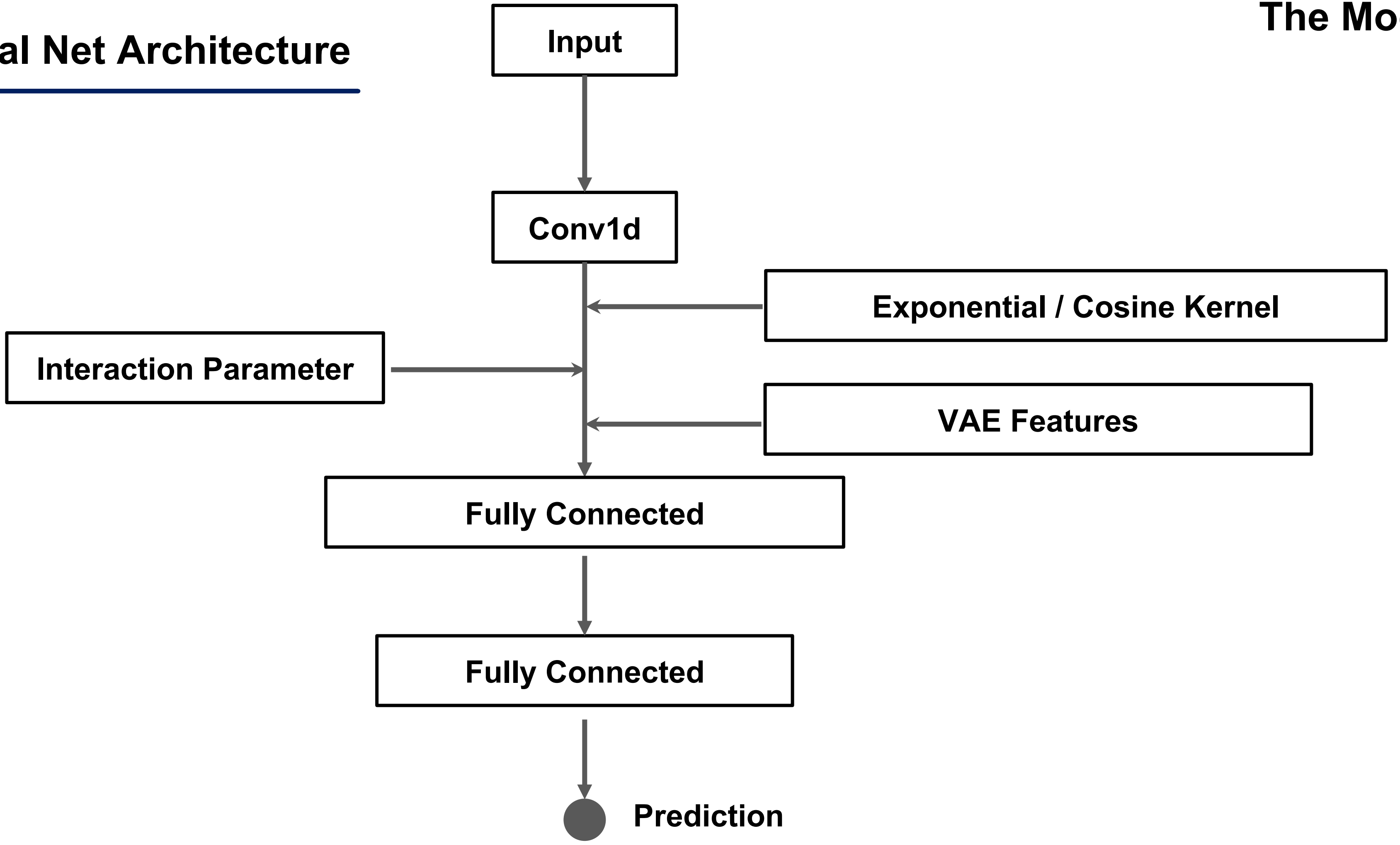


Monomer sequences

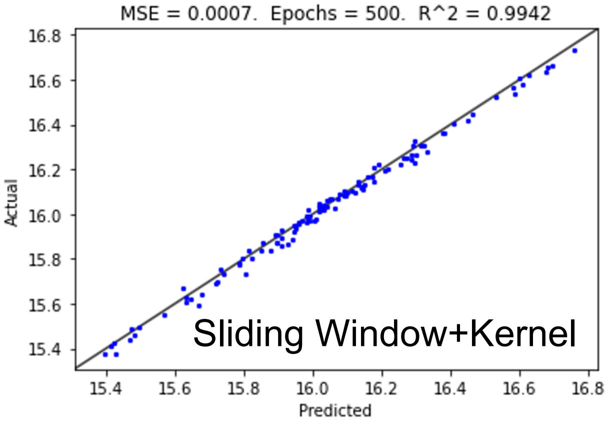
BBBBBCCCCC

Reduced feature vectors

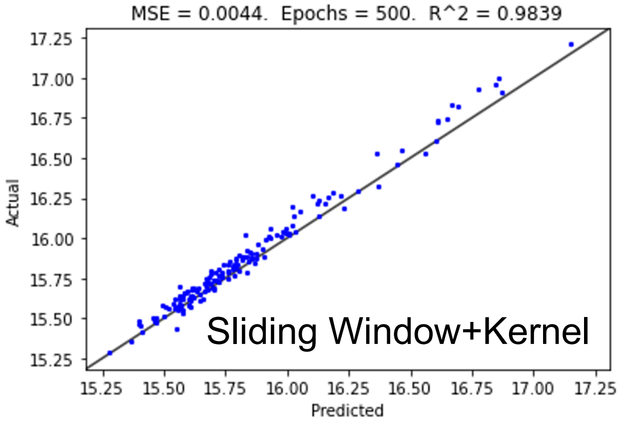
Neural Net Architecture



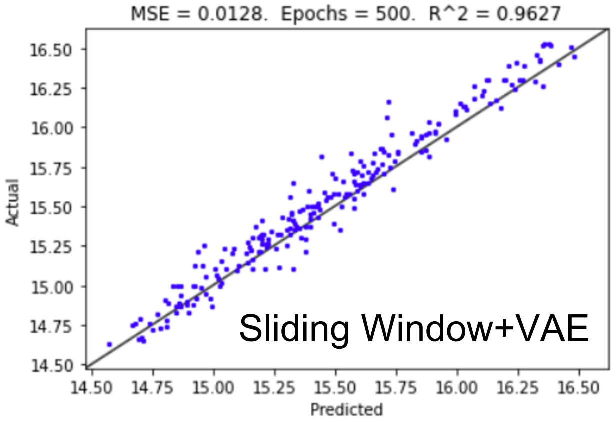
Results



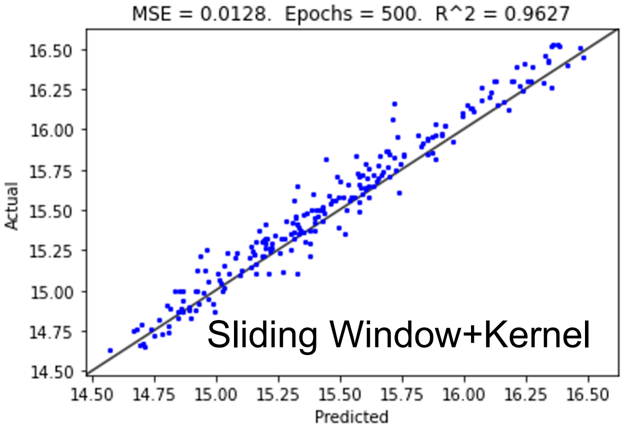
DataSetA



DataSetB



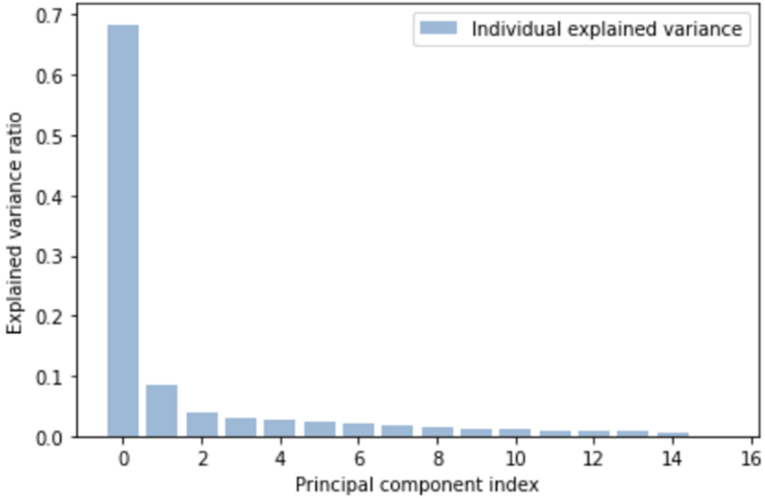
DataSetC



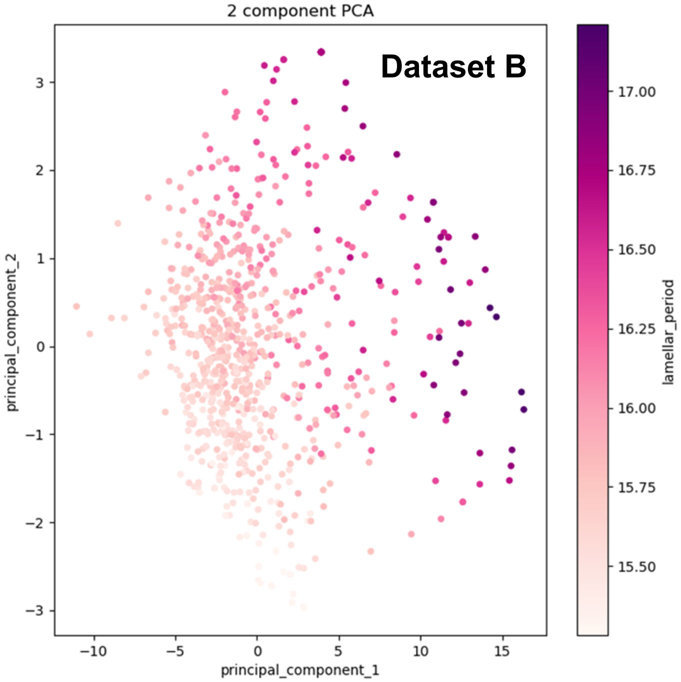
DataSetD

500 epoch
0.01 learning rate

Principal component analysis on the latent space



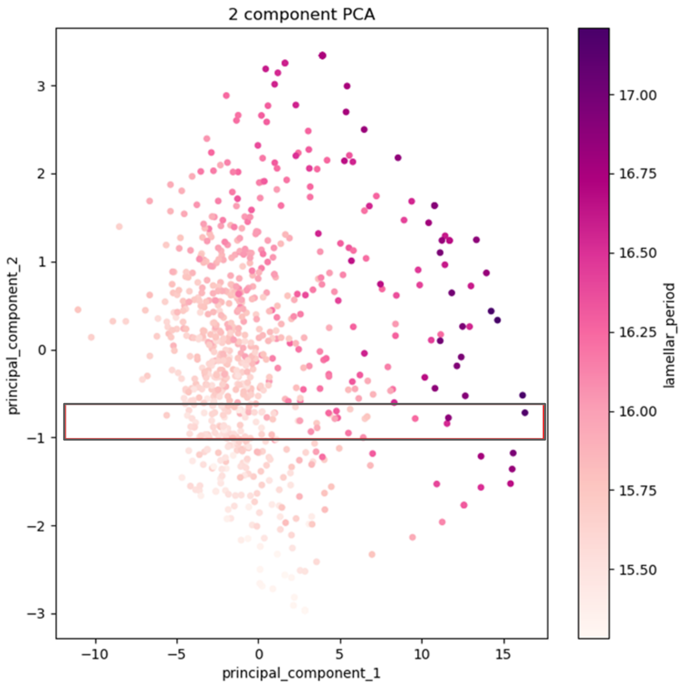
Principal component 1 dominates



Larger principal component 1

Larger lamellar period

Blockiness is important to result in a high lamellar period



Higher principal component 1

Sequence	Blockiness
11111111111111110000000000000000	0.875000
11111111111111100000000000000000	0.874510
11111111111111111100000000000000	0.873016
11111111111111111110000000000000	0.870445
11111111111111111111000000000000	0.866667

Lower principal component 1

Sequence	Blockiness
11000011001110000110000000111111	0.498039
00000011111110000000000000111111	0.740891
11000001110110110000000000111110	0.372549
01000111110110010001101000100111	0.000000
10100001110001000100100100011011	-0.036437

Computational Efficiency (500 epochs)

Feature Generation	Time (min)
2-channel Sliding Window Features	0.5
Exponential / Cosine Kernel Features	0.08
VAE Features	30

Model Training / Validation	Time (min)
Training	1
Validation	0.02

* All runtimes are reported using ThetaGPU

Thank you!