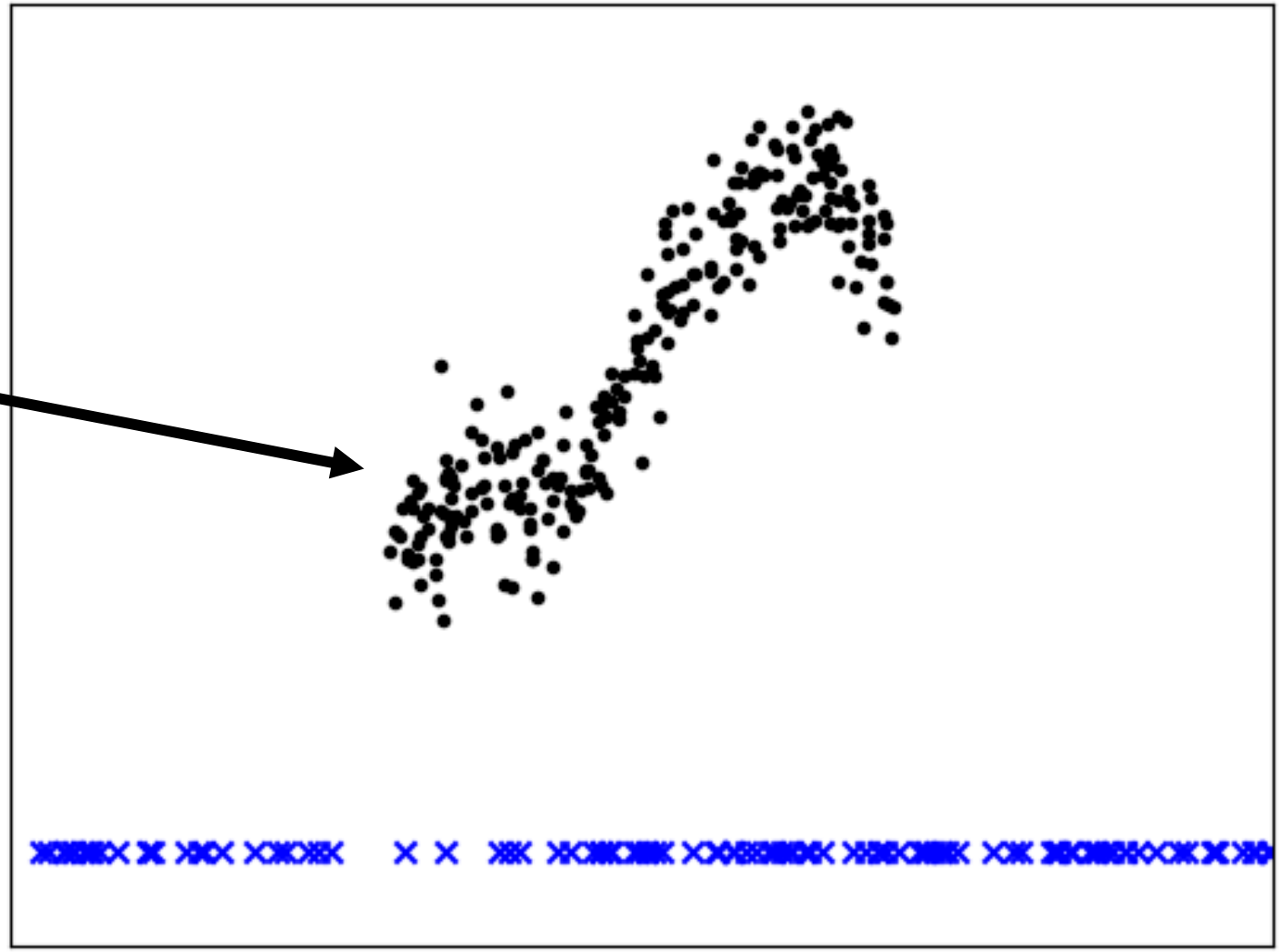
Unifying Transformers and Convolutional Neural Processes

Lakee Sivaraya

Supervisor: Prof Rich Turner

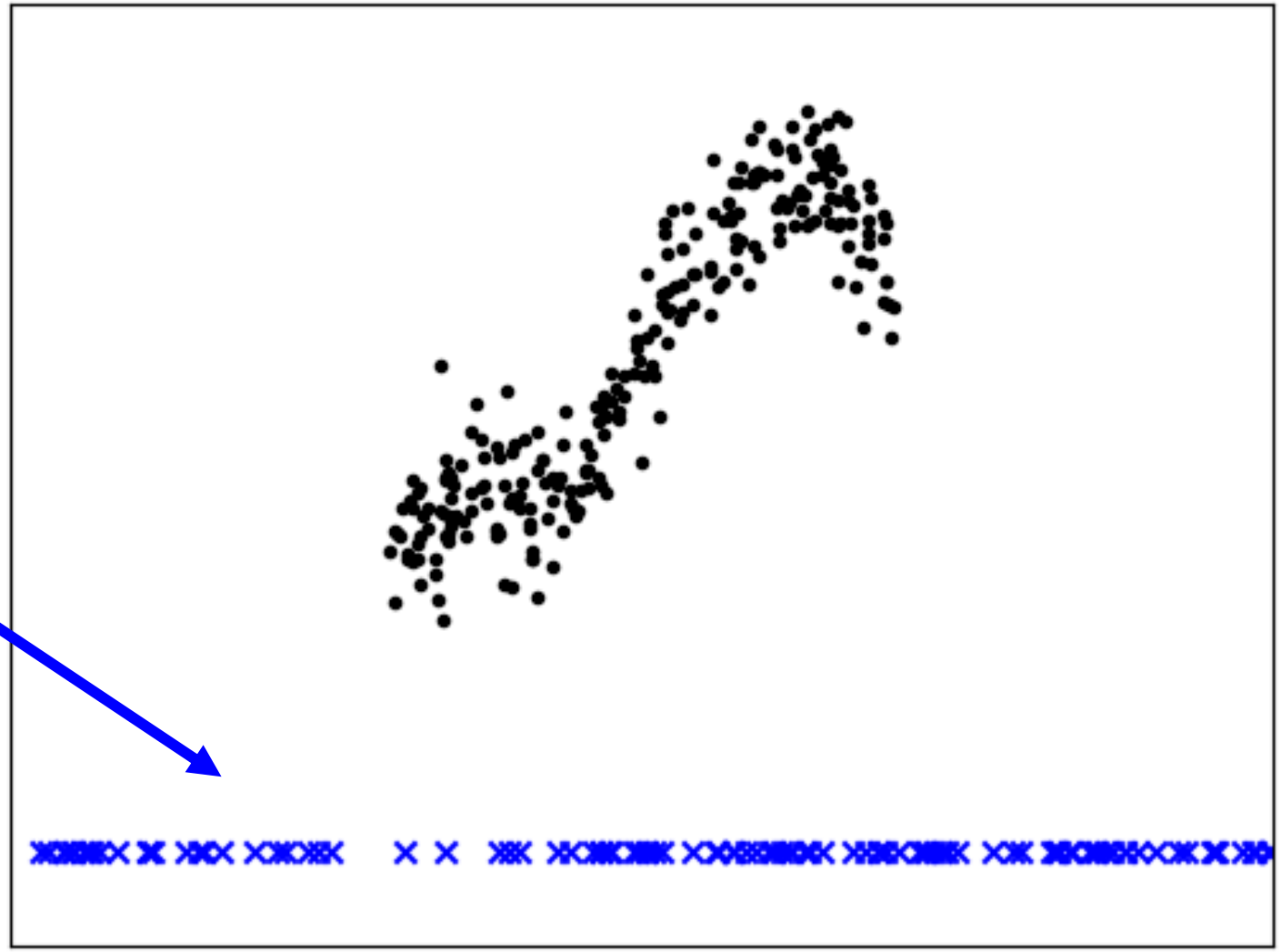
Regression in Neural Networks

Our data

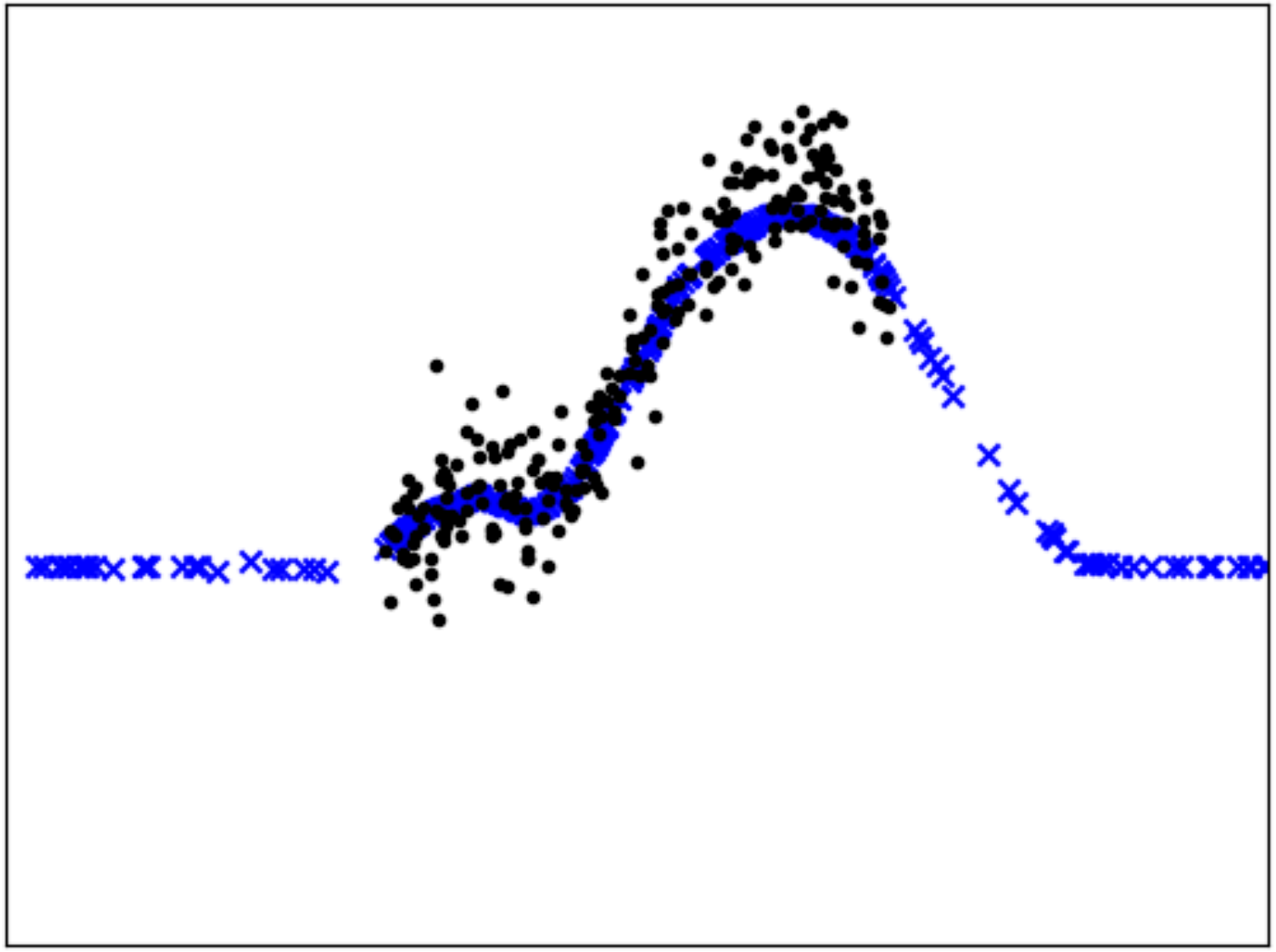


Regression in Neural Networks

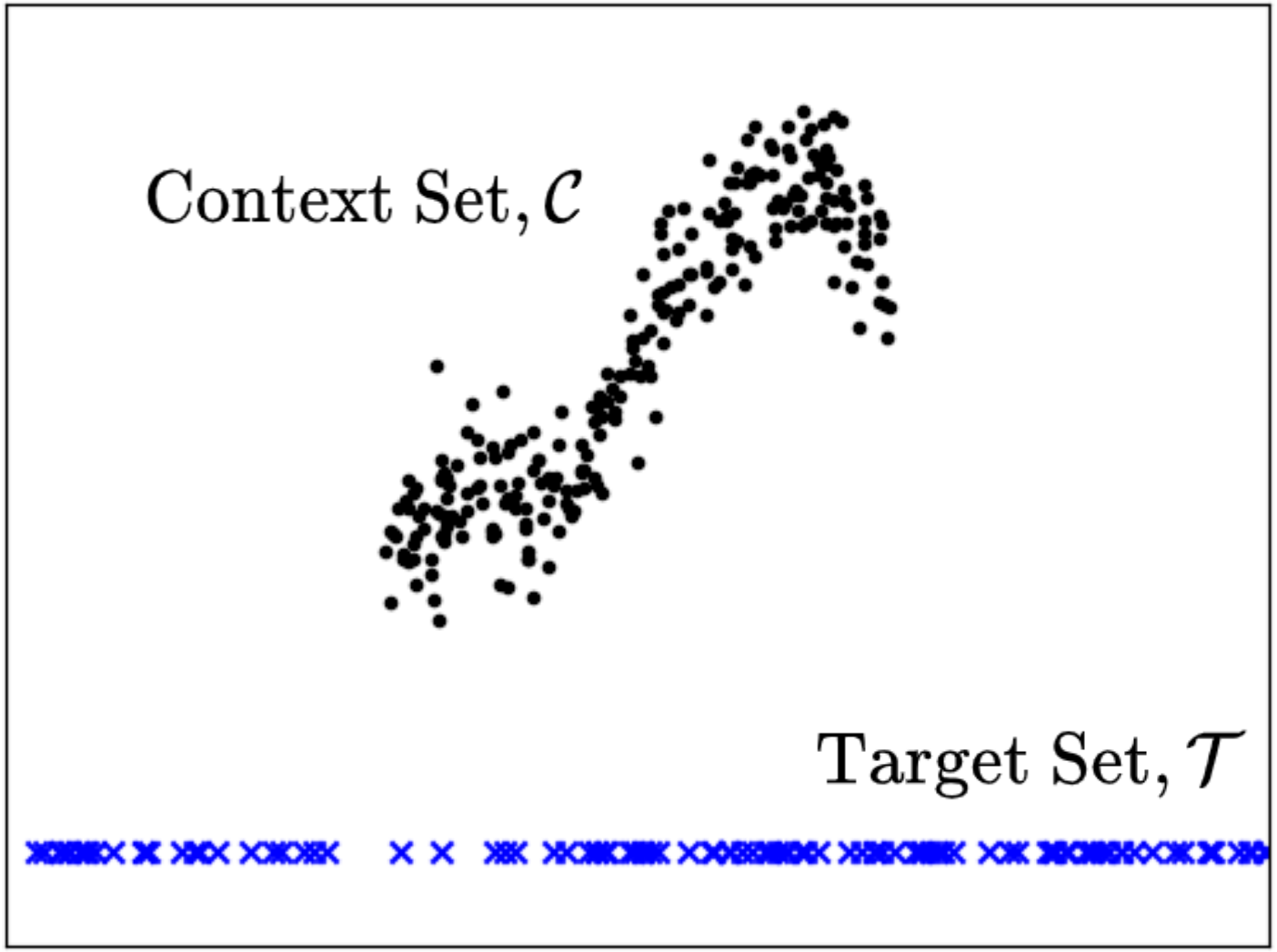
Predict Samples at these x values



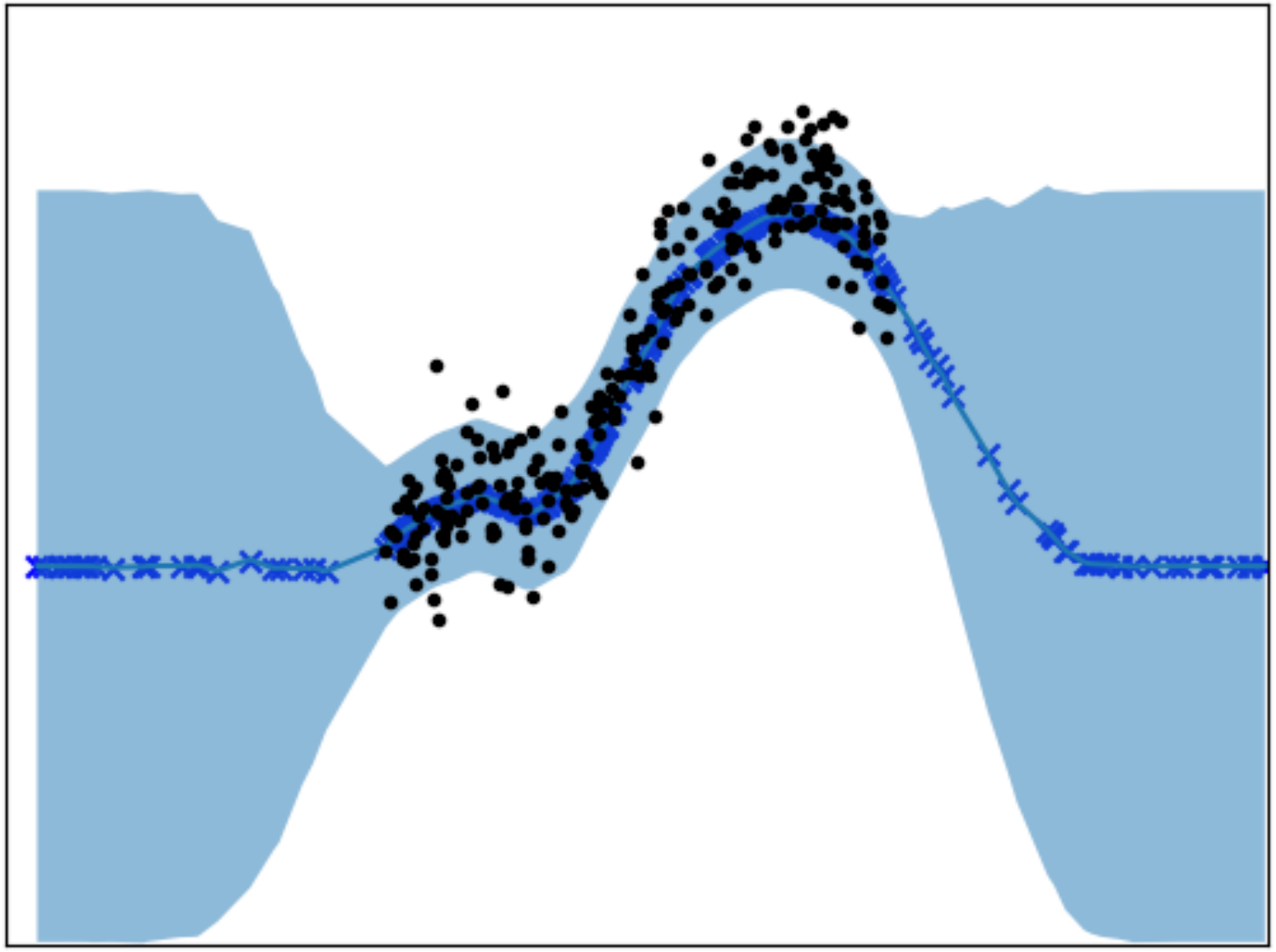
Regression in Neural Networks



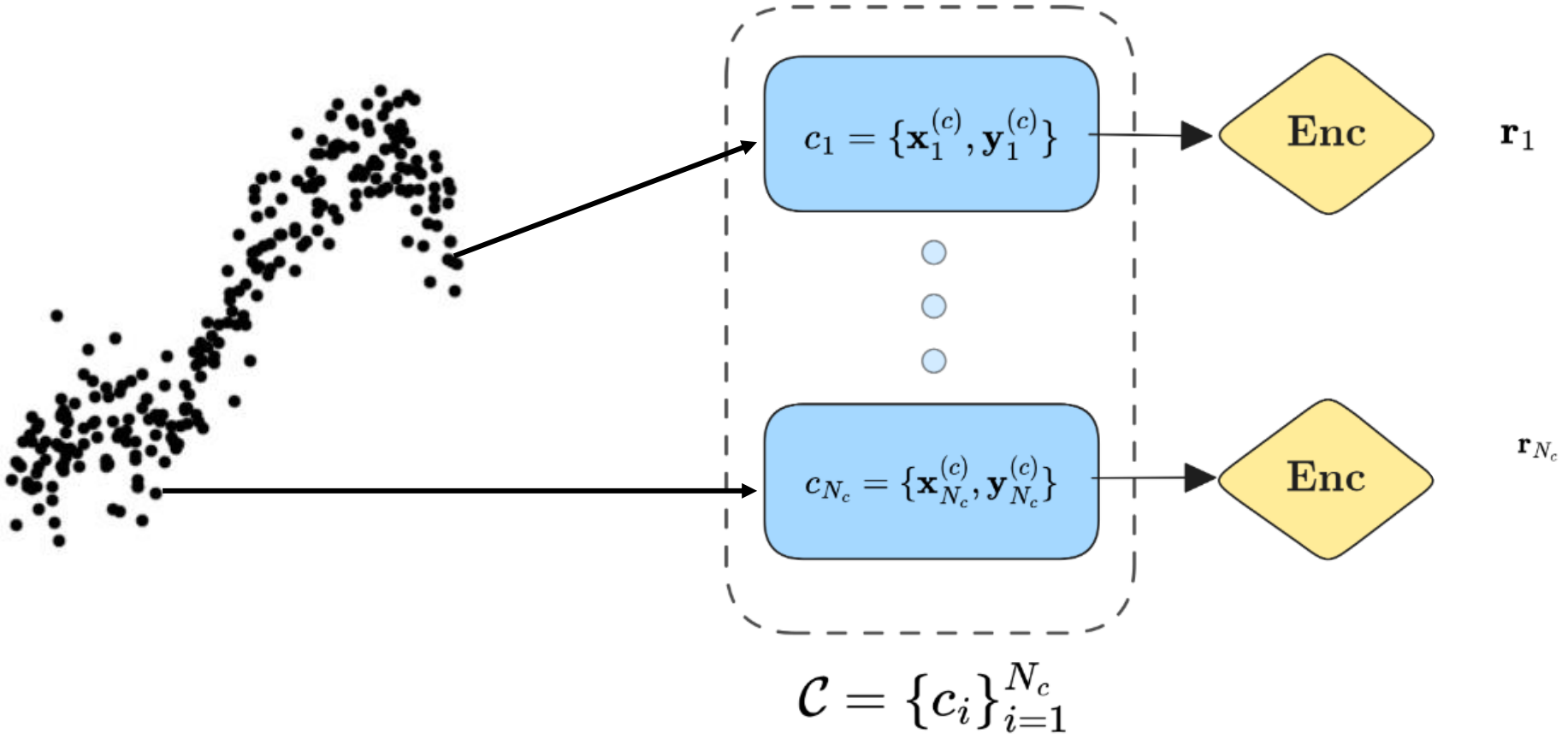
Regression in Neural Processes (NP)



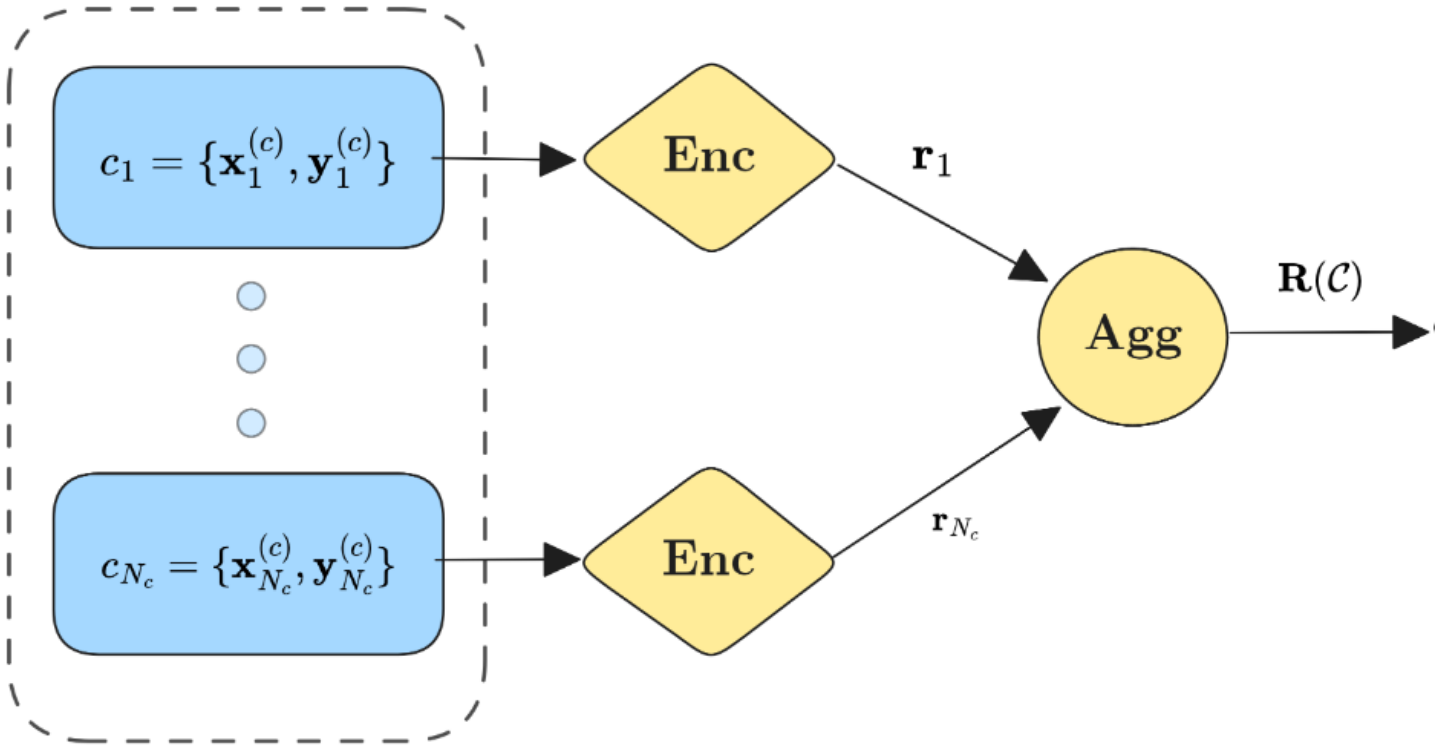
Regression in Neural Processes (NP)



Neural Processes

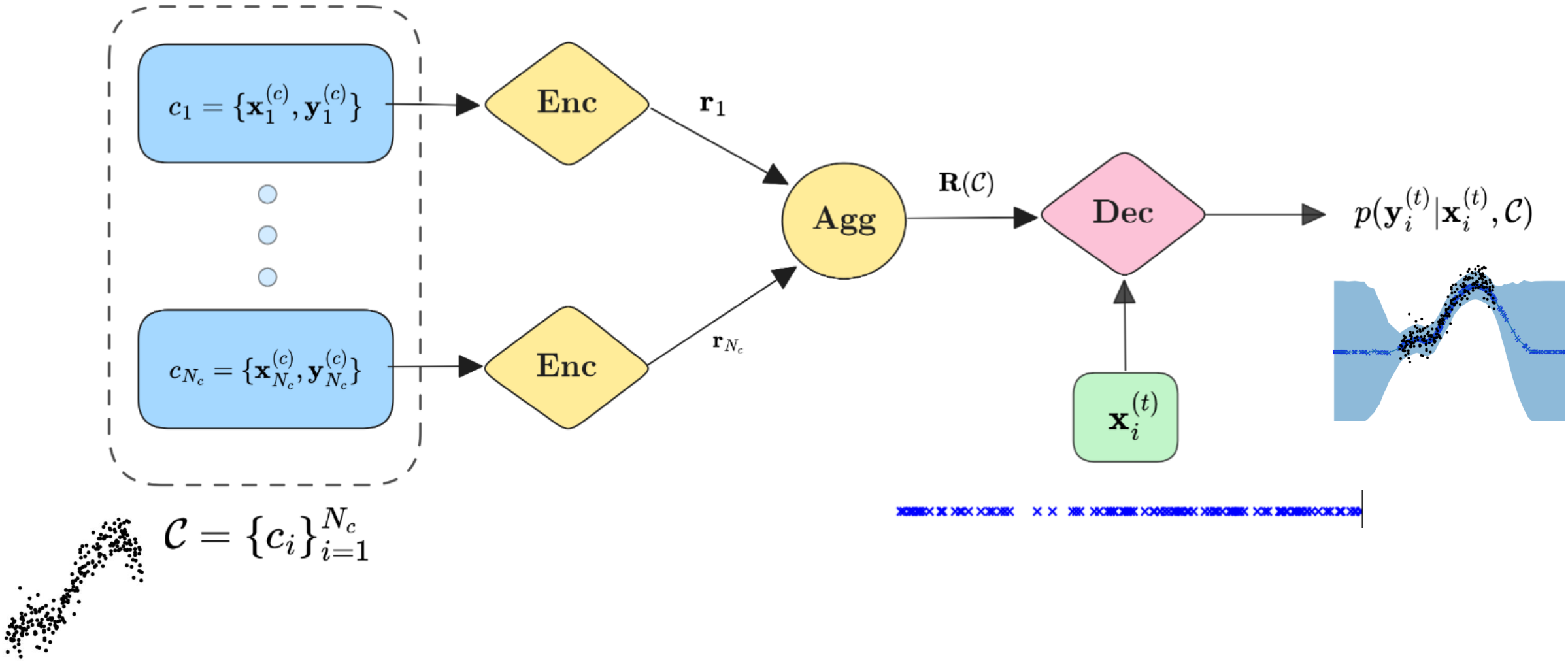


Neural Processes

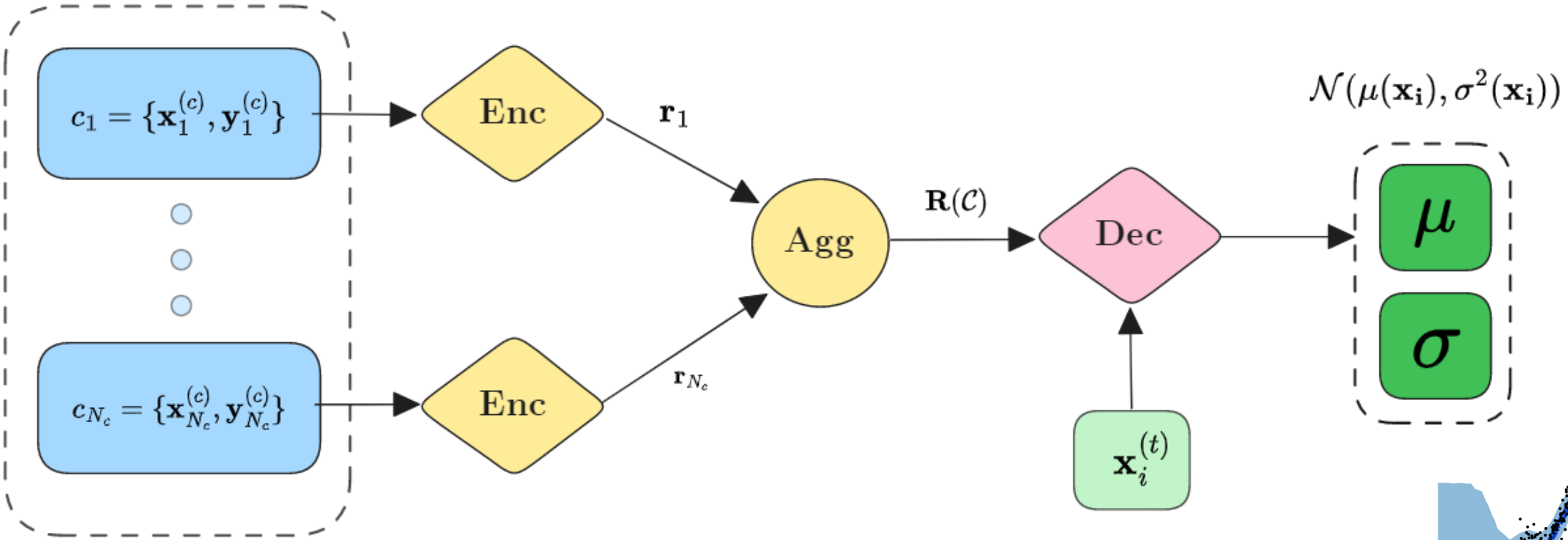


$$\mathcal{C} = \{c_i\}_{i=1}^{N_c}$$

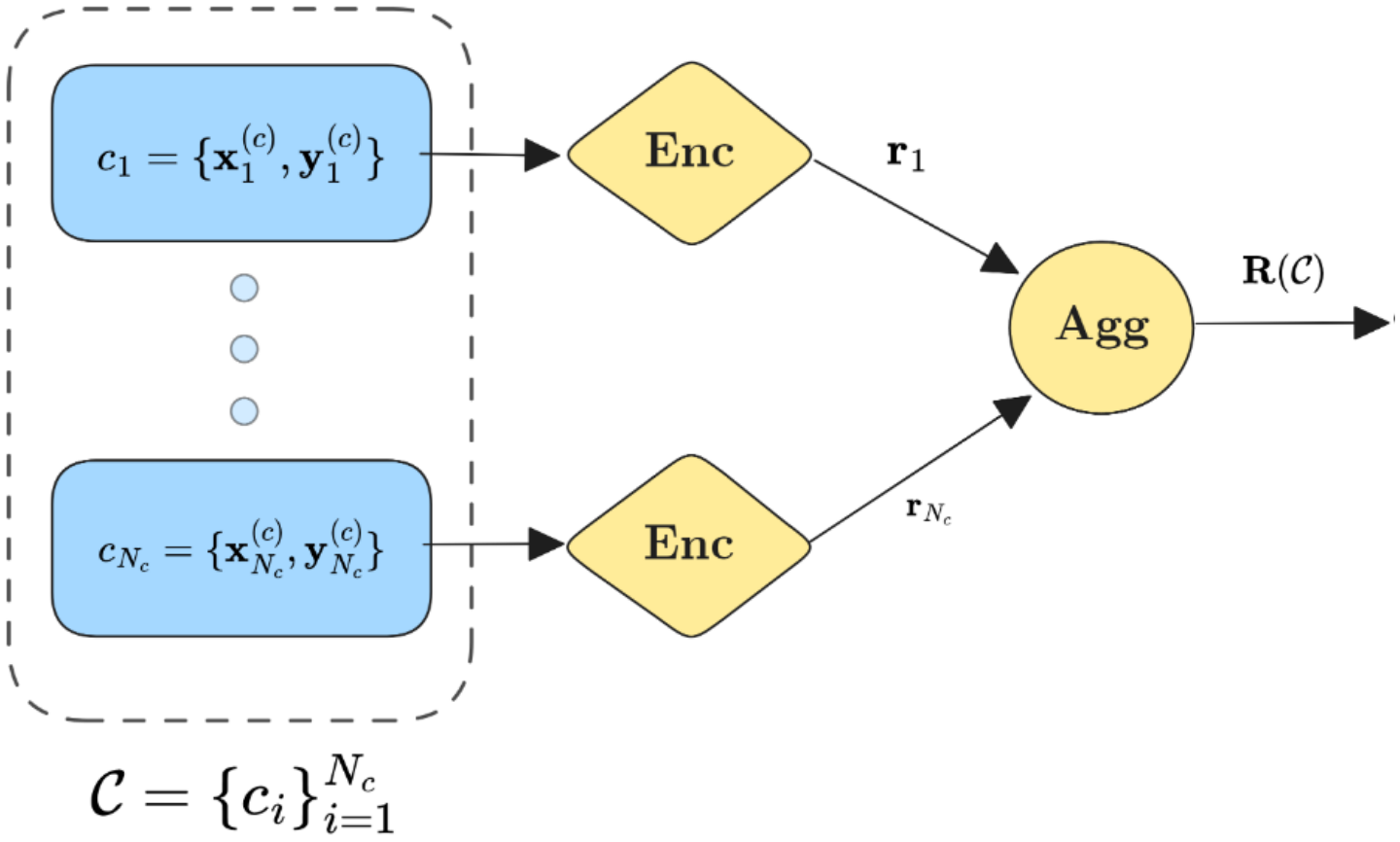
Neural Processes



Neural Processes



Neural Processes Enc + Agg



Transformer Based

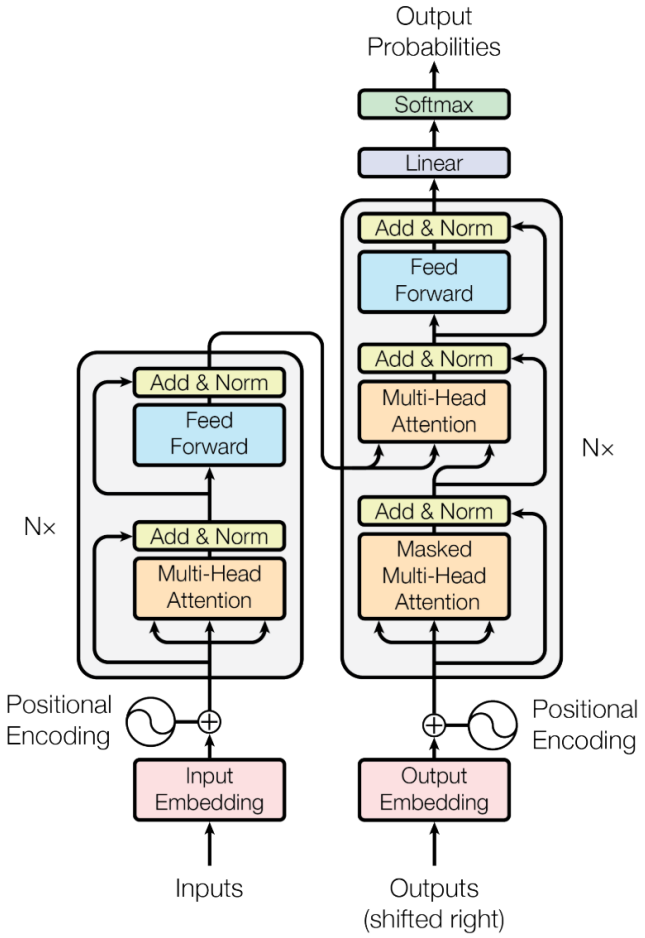
Enc & Agg = Transformer

Convolution Based

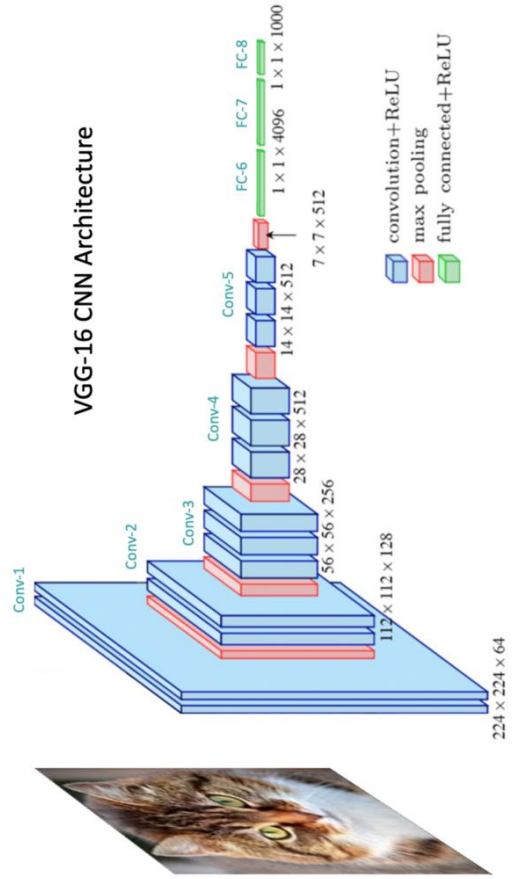
Enc = CNN
Agg = SetConv

Neural Processes Enc + Agg

Enc & Agg =

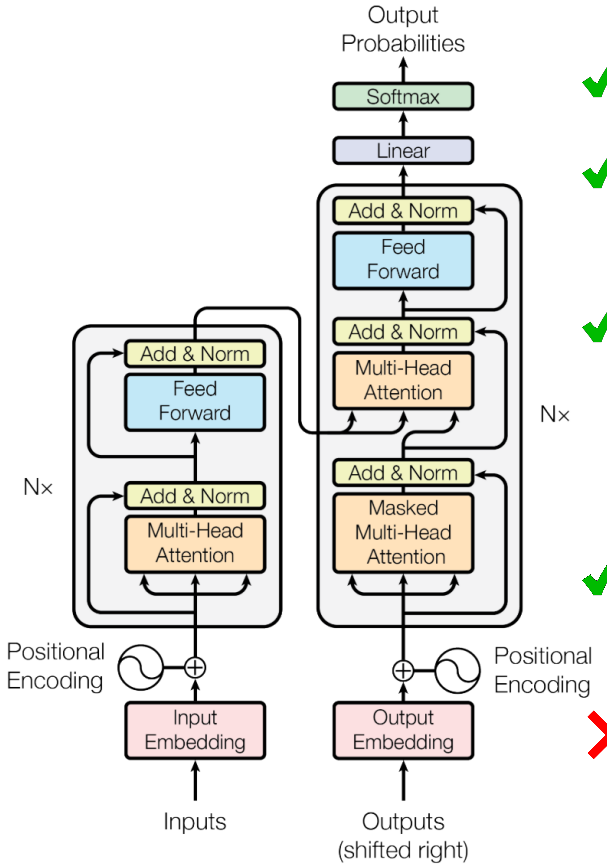


Enc =

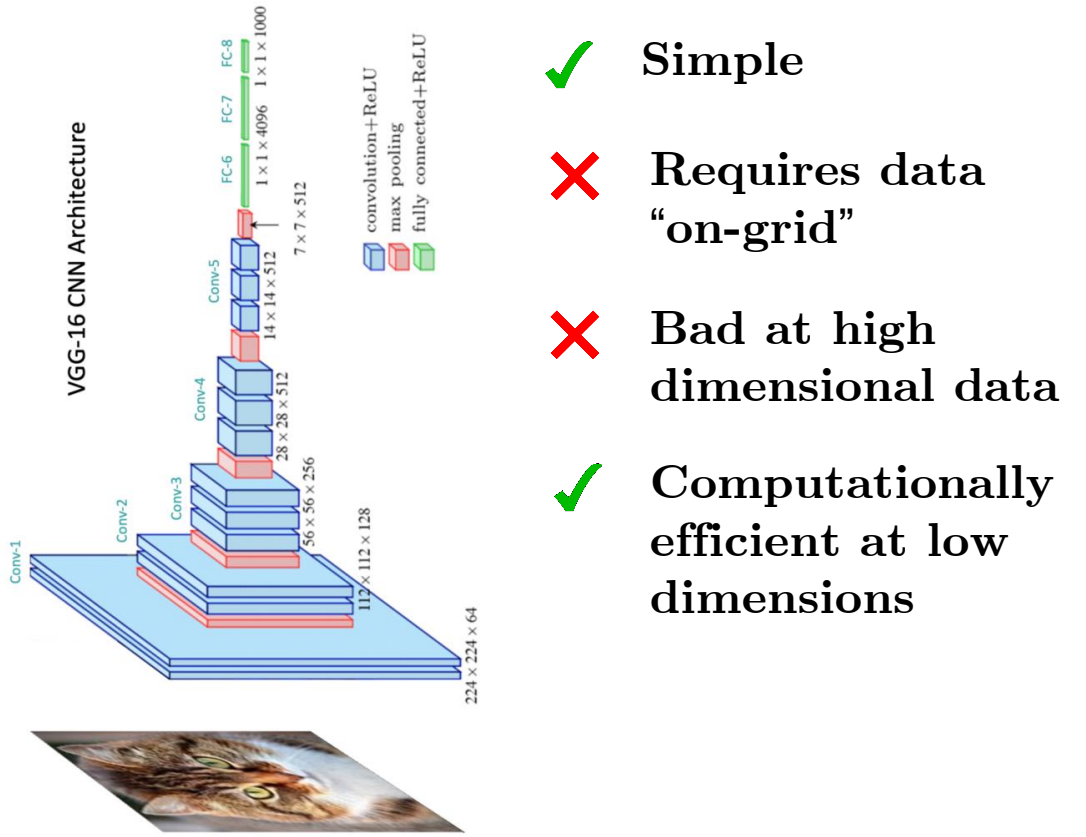


Agg = SetConv

Transformers vs CNNs



- ✓ **Proven to scale**
- ✓ **Works on "off grid" data**
- ✓ **Works on high dimensional data like spatio-temporal**
- ✓ **Flexible and general purpose**
- ✗ **High computational complexity**



- ✓ **Simple**
- ✗ **Requires data "on-grid"**
- ✗ **Bad at high dimensional data**
- ✓ **Computationally efficient at low dimensions**

Which model is better?

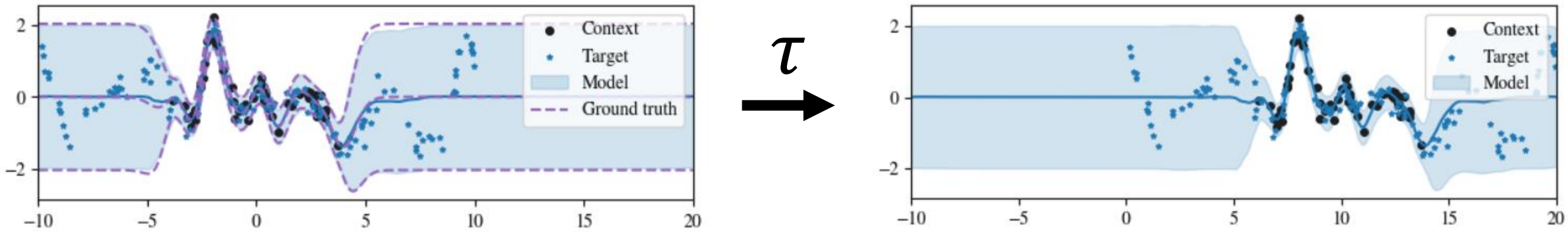
Our Aim

To Compare Transformer and CNN models in a Neural Process (NP) setting

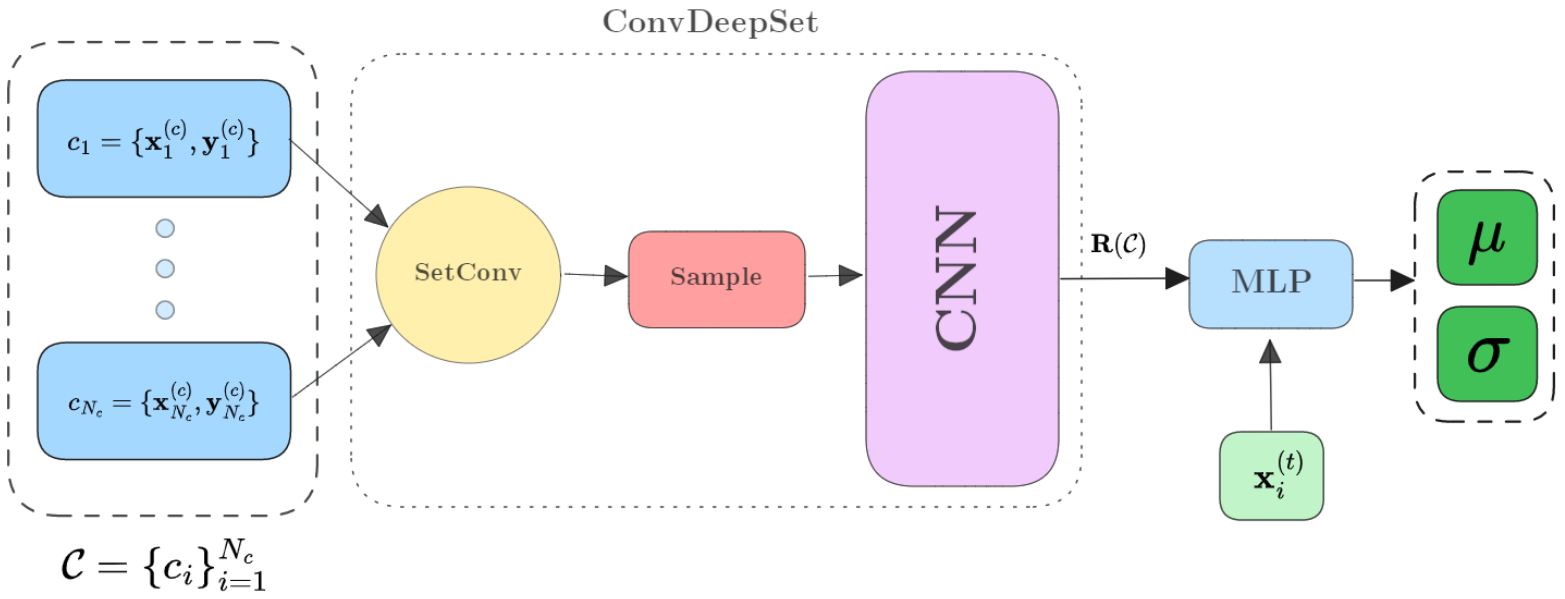
- **Michaelmas term focused on building in desirable properties into the Transformer based architecture**
- **Then investigating hyperparameters of the model**
- **Perform systematic and fair comparison of models**

Key Properties of NPs

- **Permutation Invariant (PE):**
  Invariant to order of context datapoints
- **Translation Equivariant (TE):**
  Translating context points should result in a translated predictions



Convolutional NP (ConvNP)

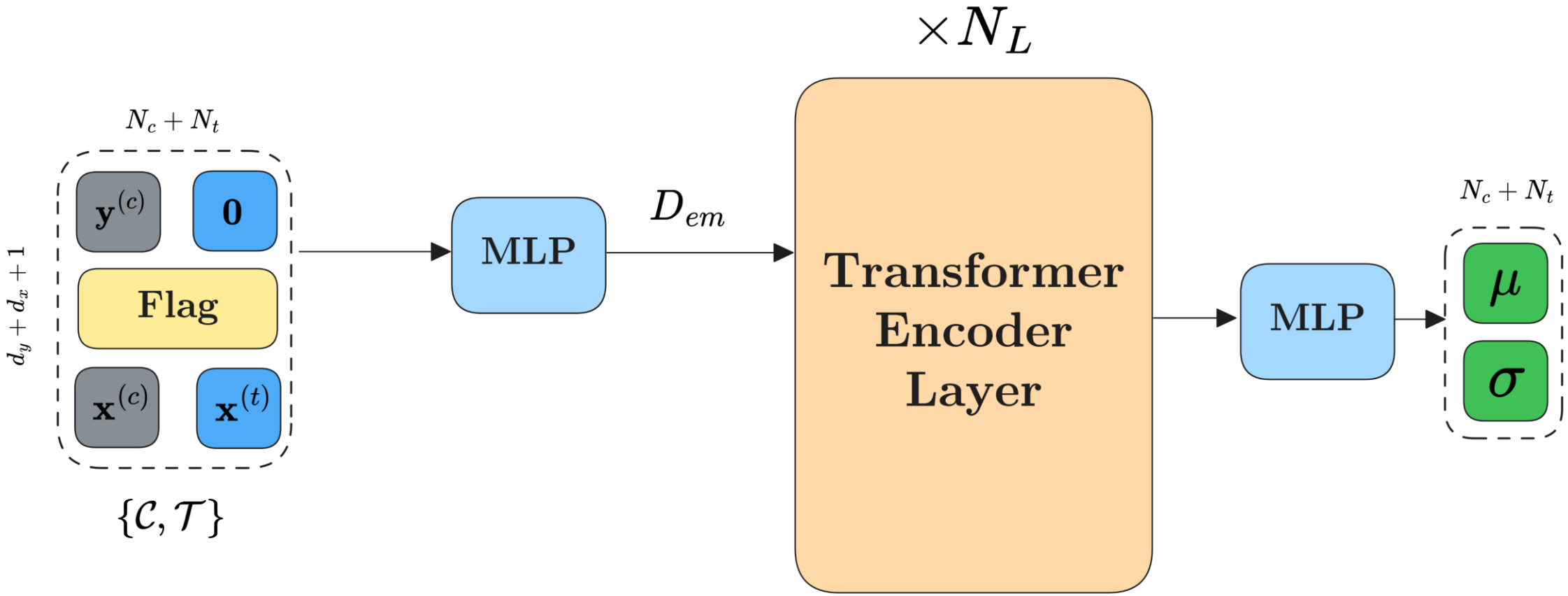


- ConvNP uses a ConvDeepSet to encoder the data

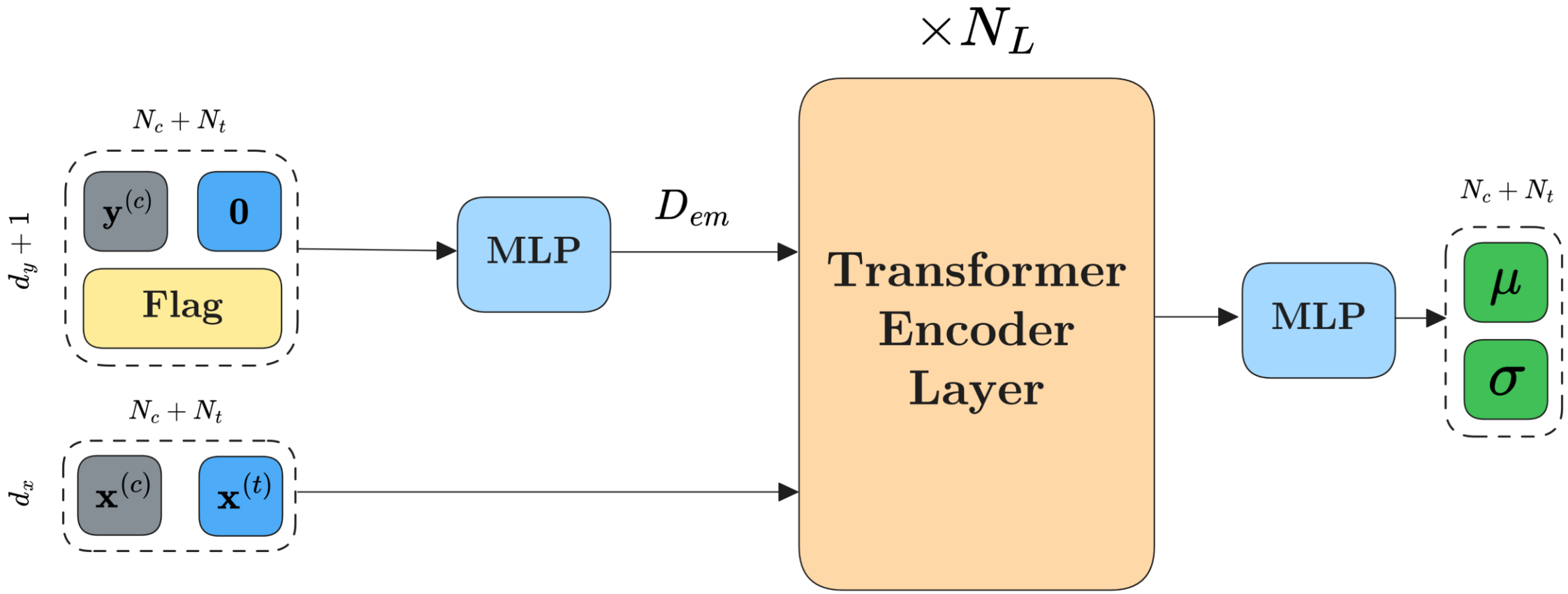
- PI by SetConv
- TE by CNN

Detail of ConvNP is not discuss in detail for the sake of time

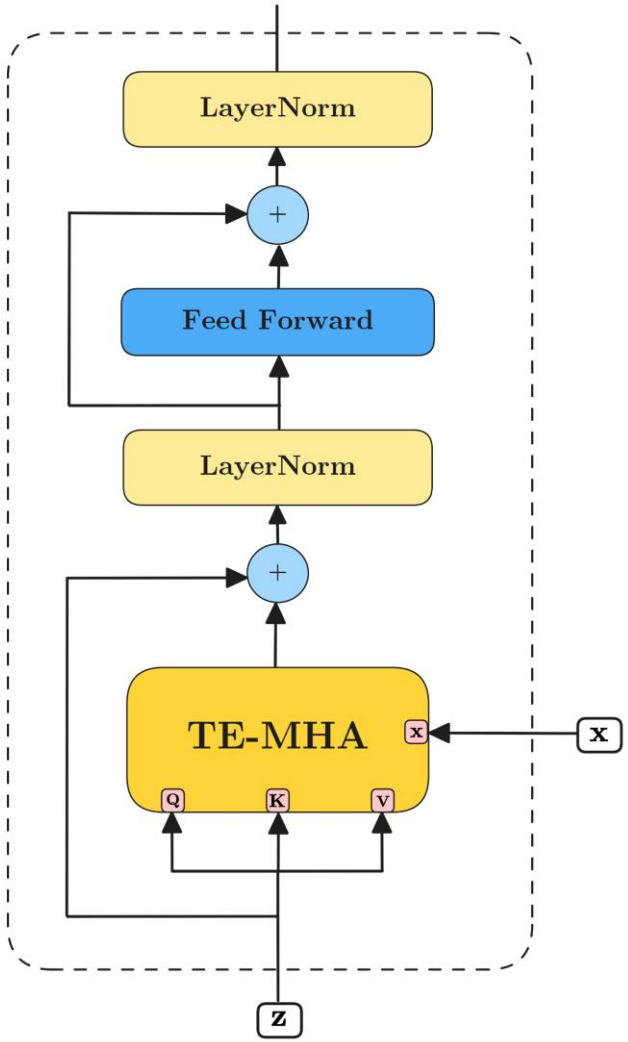
Transformer NP (TNP)



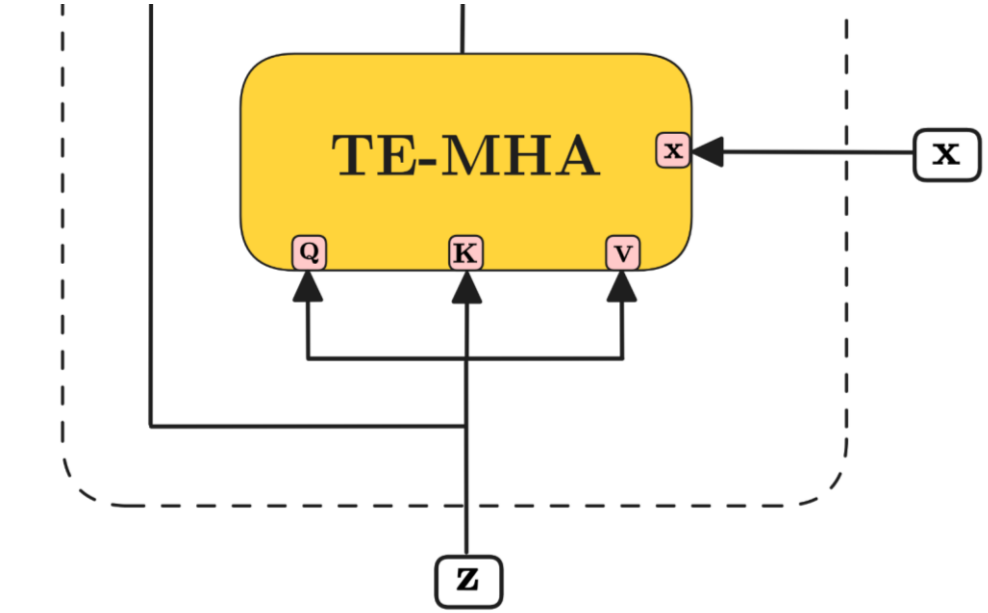
TE Transformer NP (TE-TNP)



Transformer Encoder Layer



TE Multi-Head Attention



$$\mathbf{Attn(z,x) = DotProdAttn(z) + TE\text{-}Term(x) + Mask}$$

Relative Attention

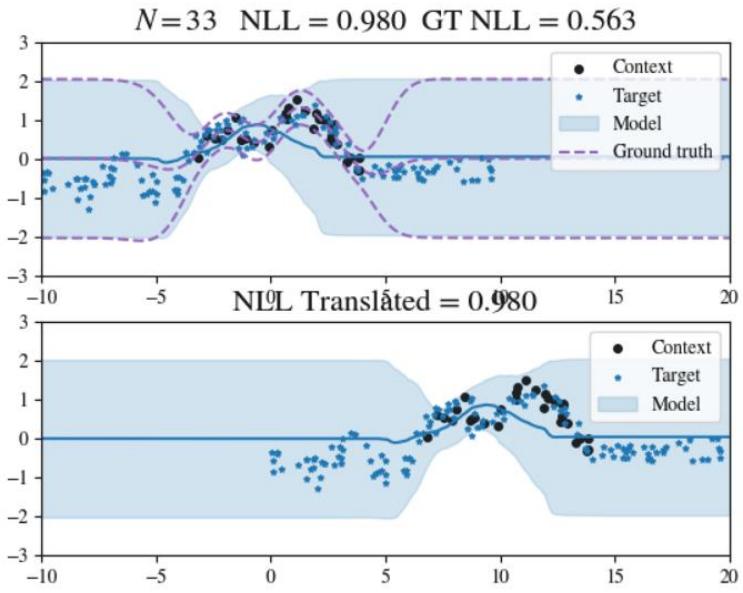
$$\boldsymbol{A} = \mathrm{softmax}(\boldsymbol{E})$$

$$\boldsymbol{e}_{ij} = \frac{\boldsymbol{q}_i \boldsymbol{k}_j^T}{\sqrt{d}} + \underbrace{F(\boldsymbol{x}_i - \boldsymbol{x}_j)}_{\text{TE Term}} + \mathrm{Mask}_{ij}$$

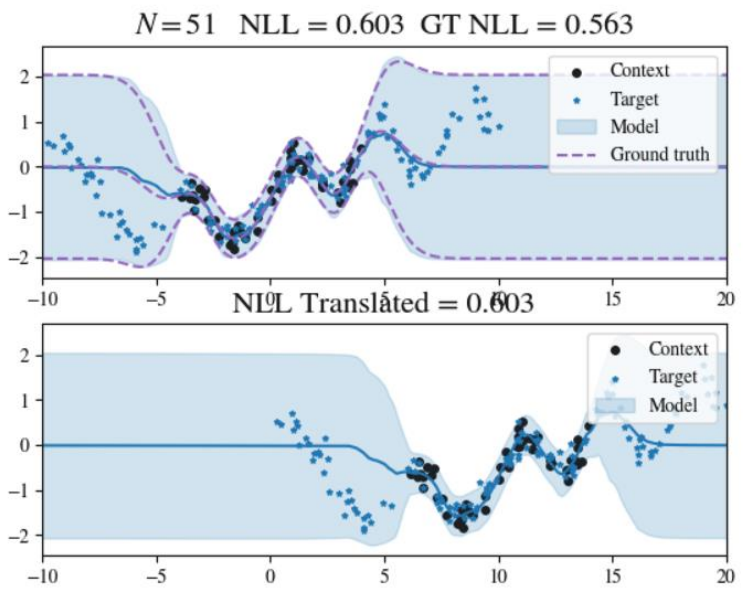
$$F(\Delta) = \{\mathrm{MLP}(\Delta), \quad \mathrm{RBF}(\Delta), \quad \Delta\}$$

TE Properties

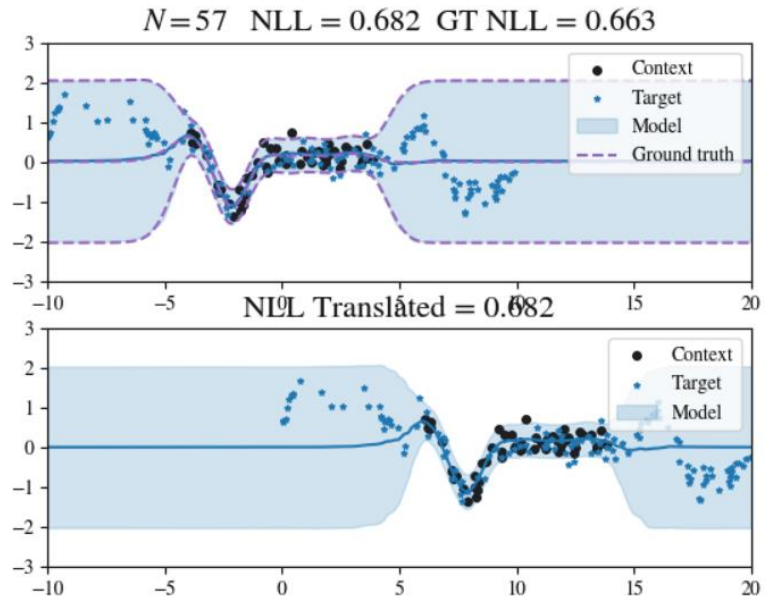
Nothing



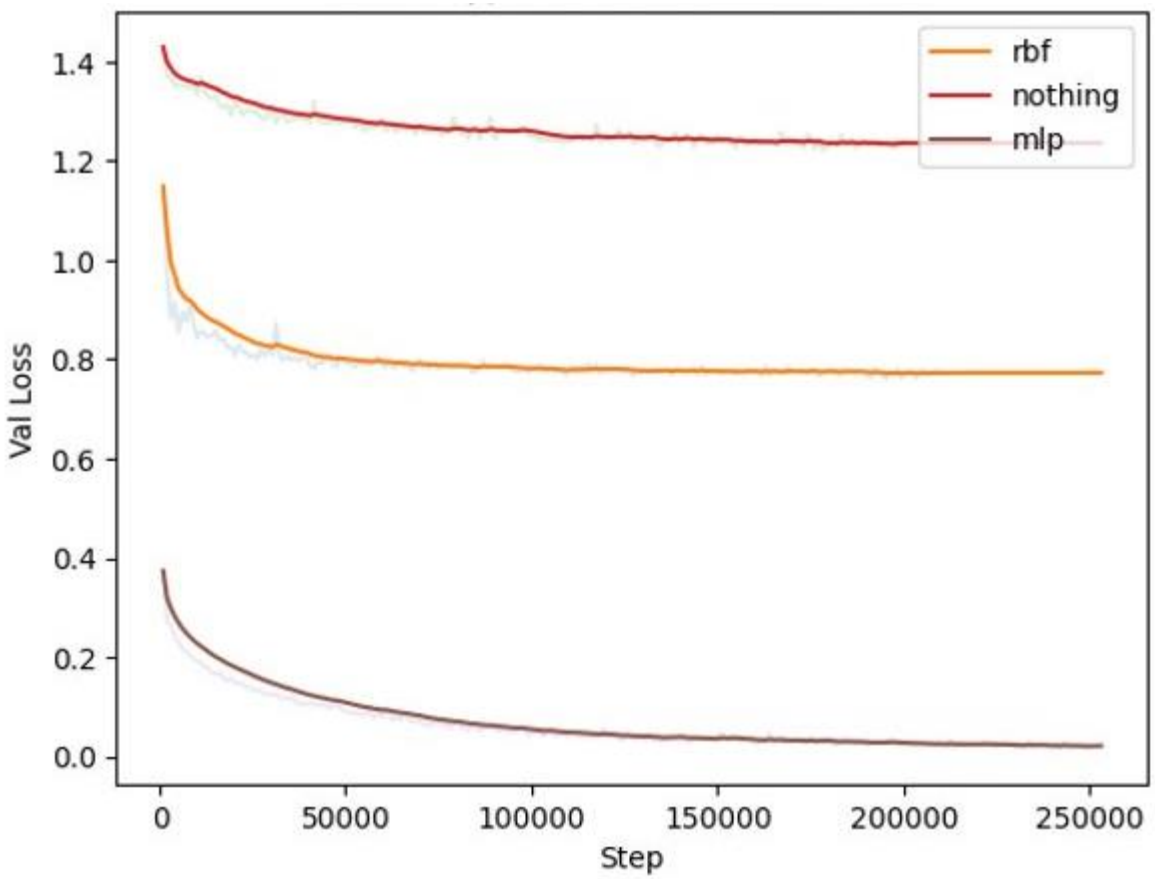
RBF



MLP



Relative Attention Results



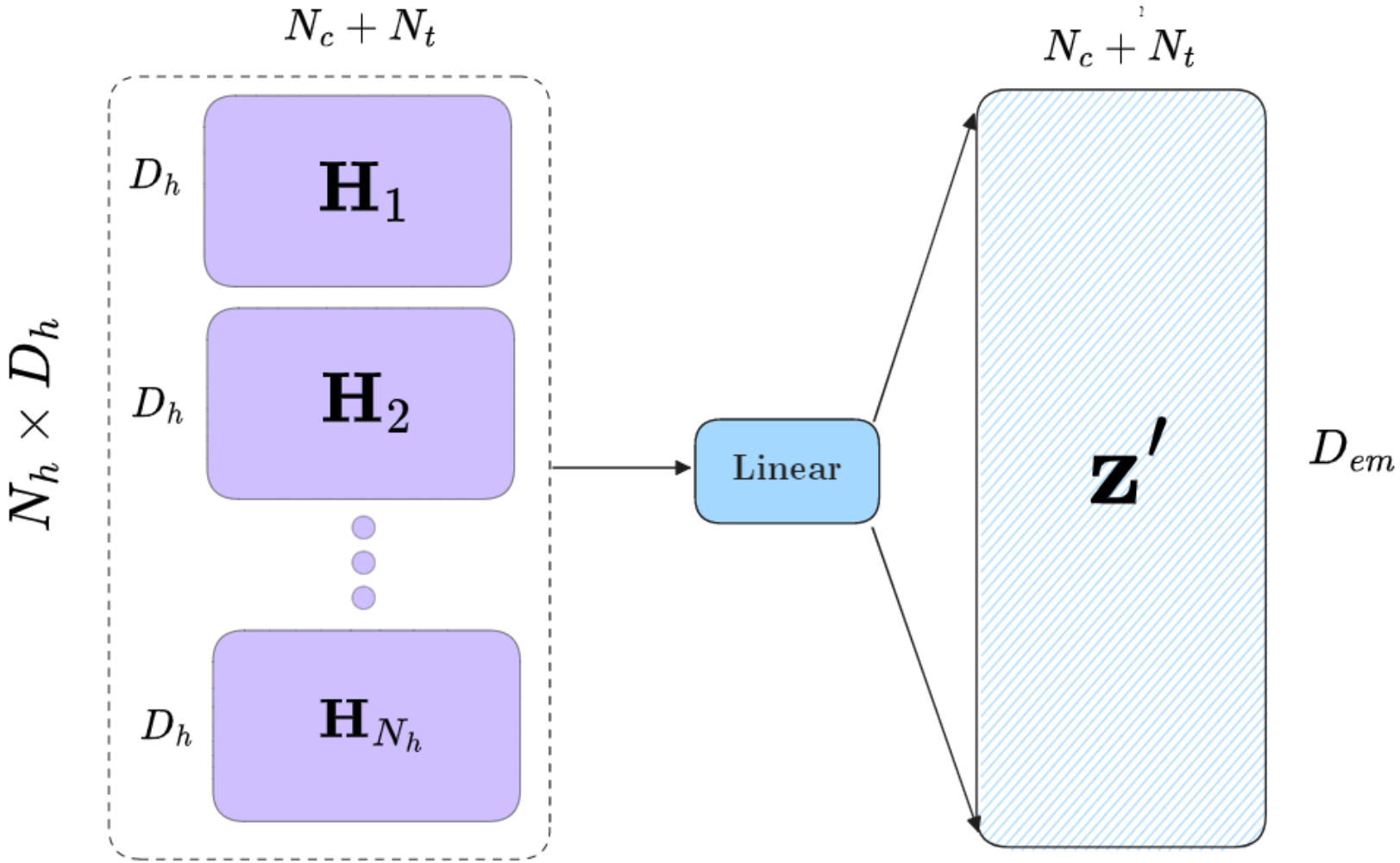
- Nothing effects attention too much, thus struggles to learn

- RBF performs ok

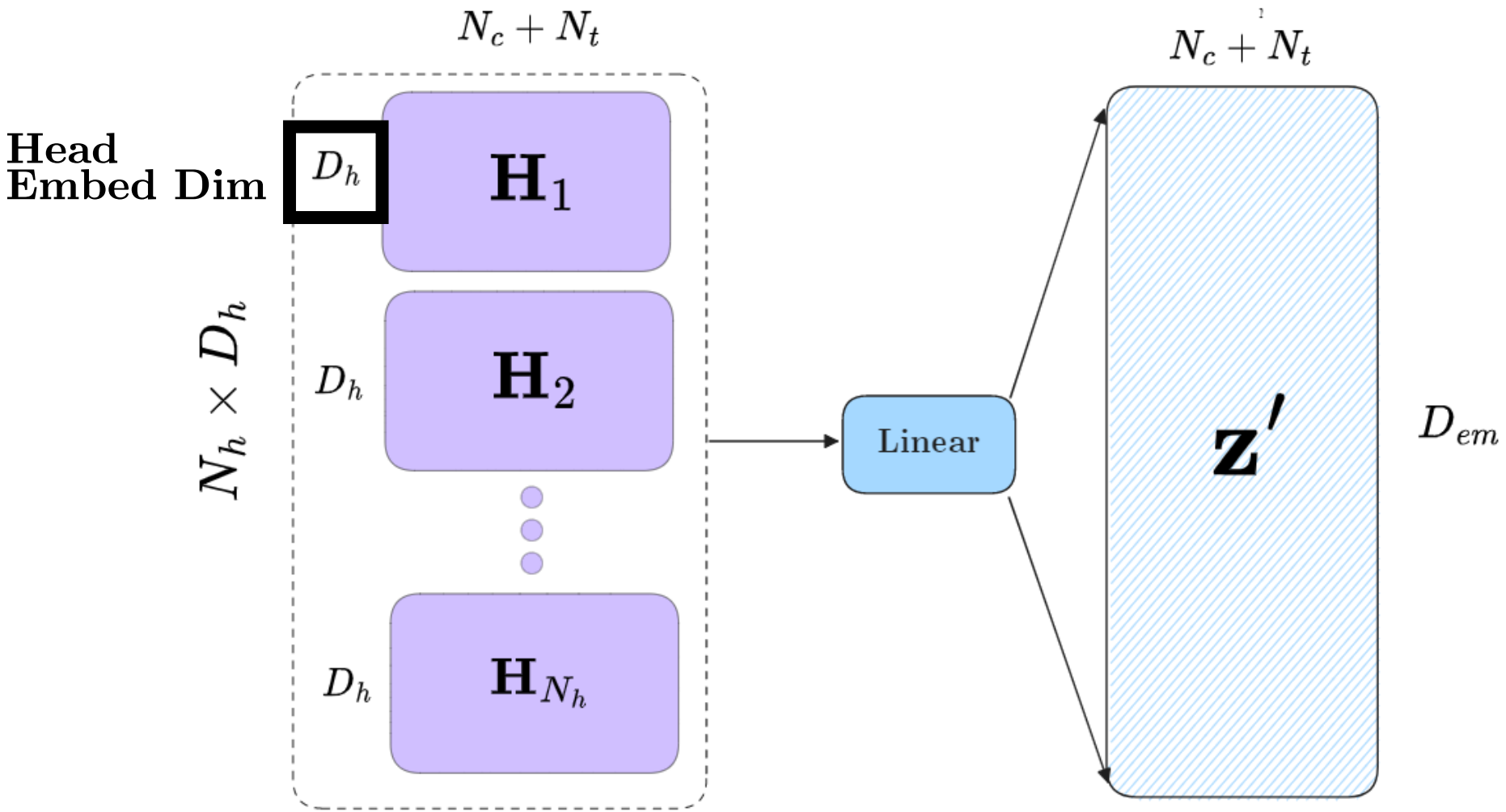
- MLP performs the best, it generalizes well is quite cheap to compute

Attention is Multi-Headed!

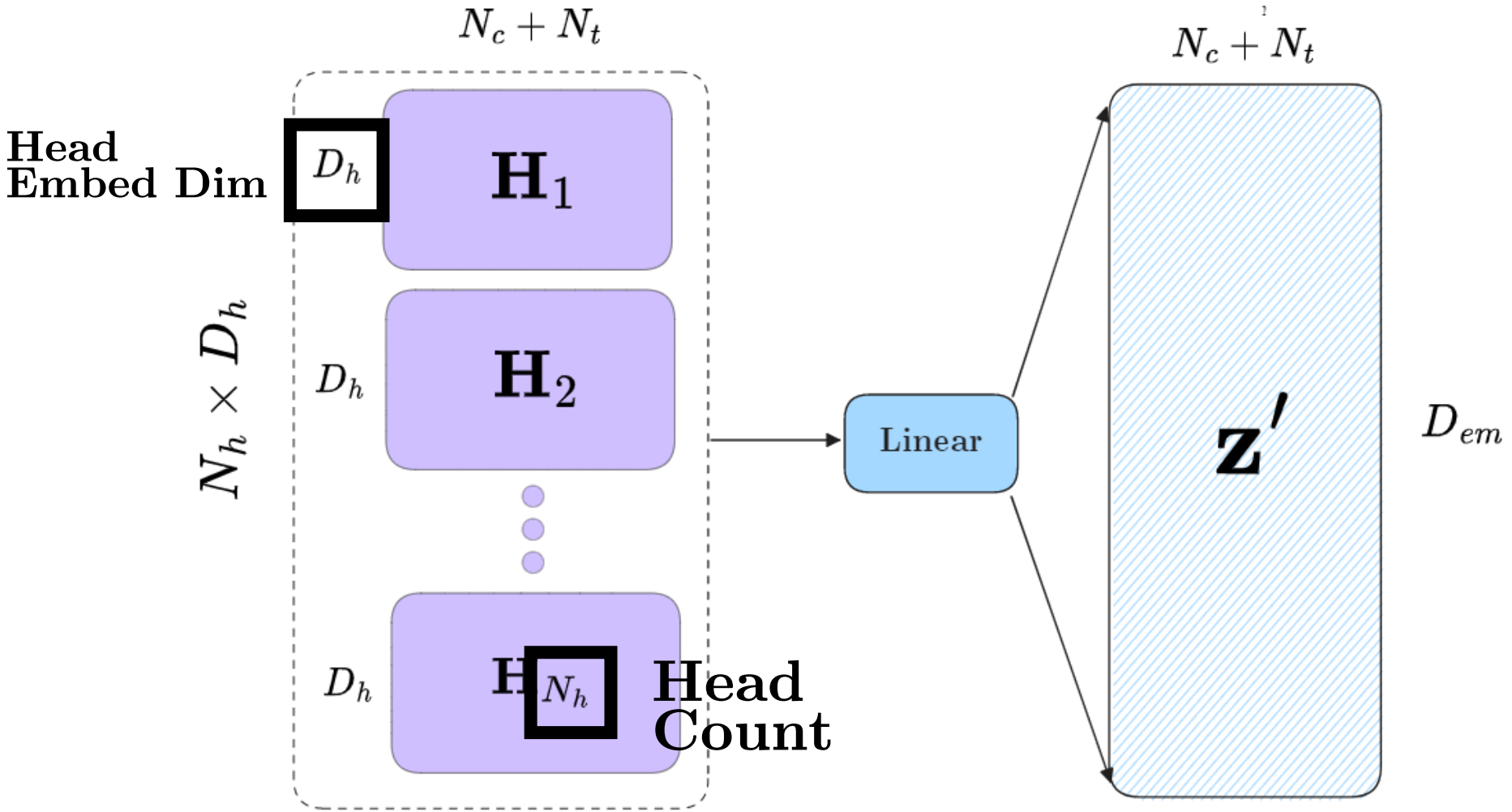
Multi Head Output



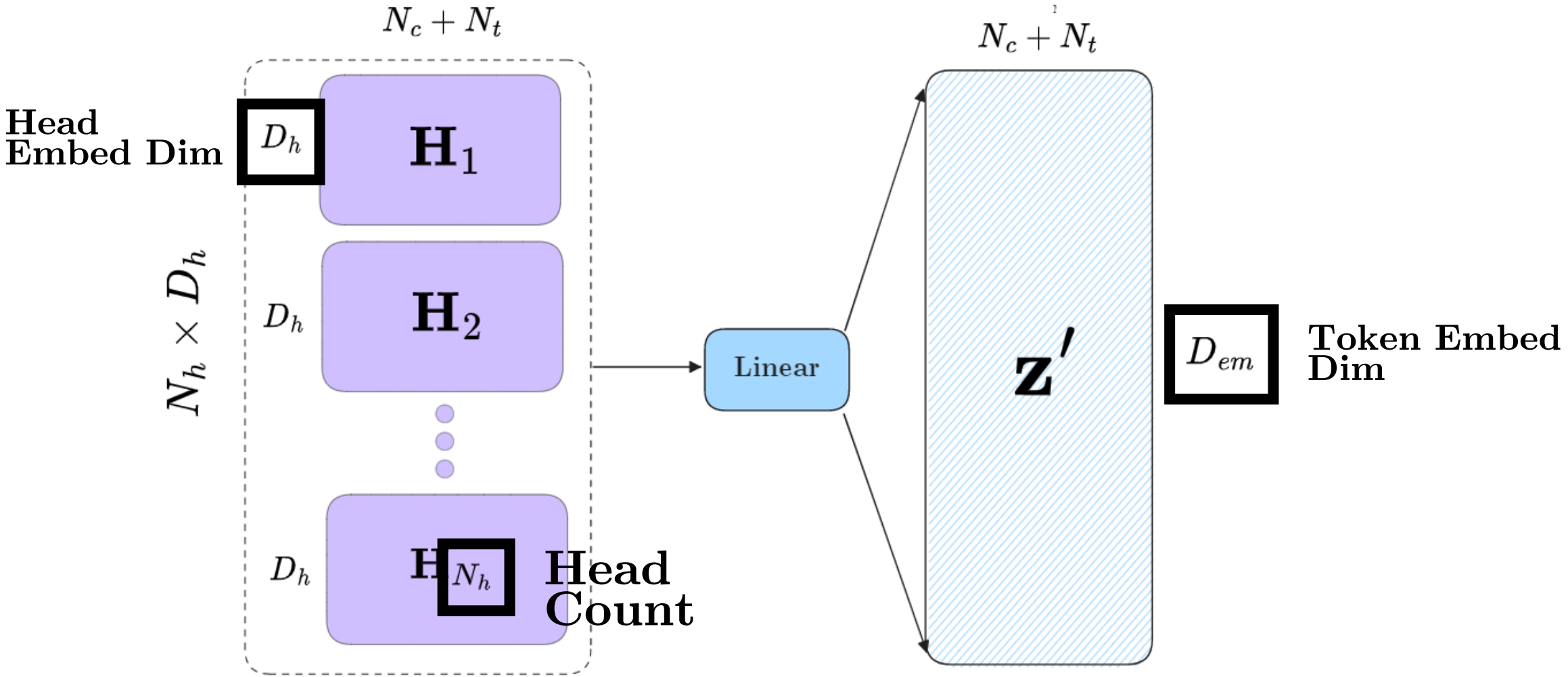
Multi Head Output



Multi Head Output



Multi Head Output



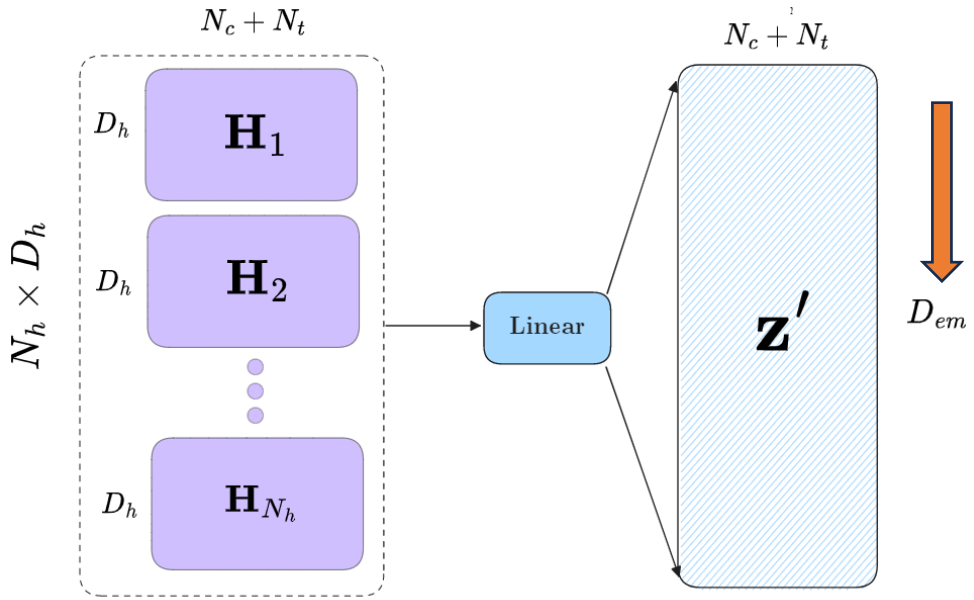
Investigating TNP Hyperparameters

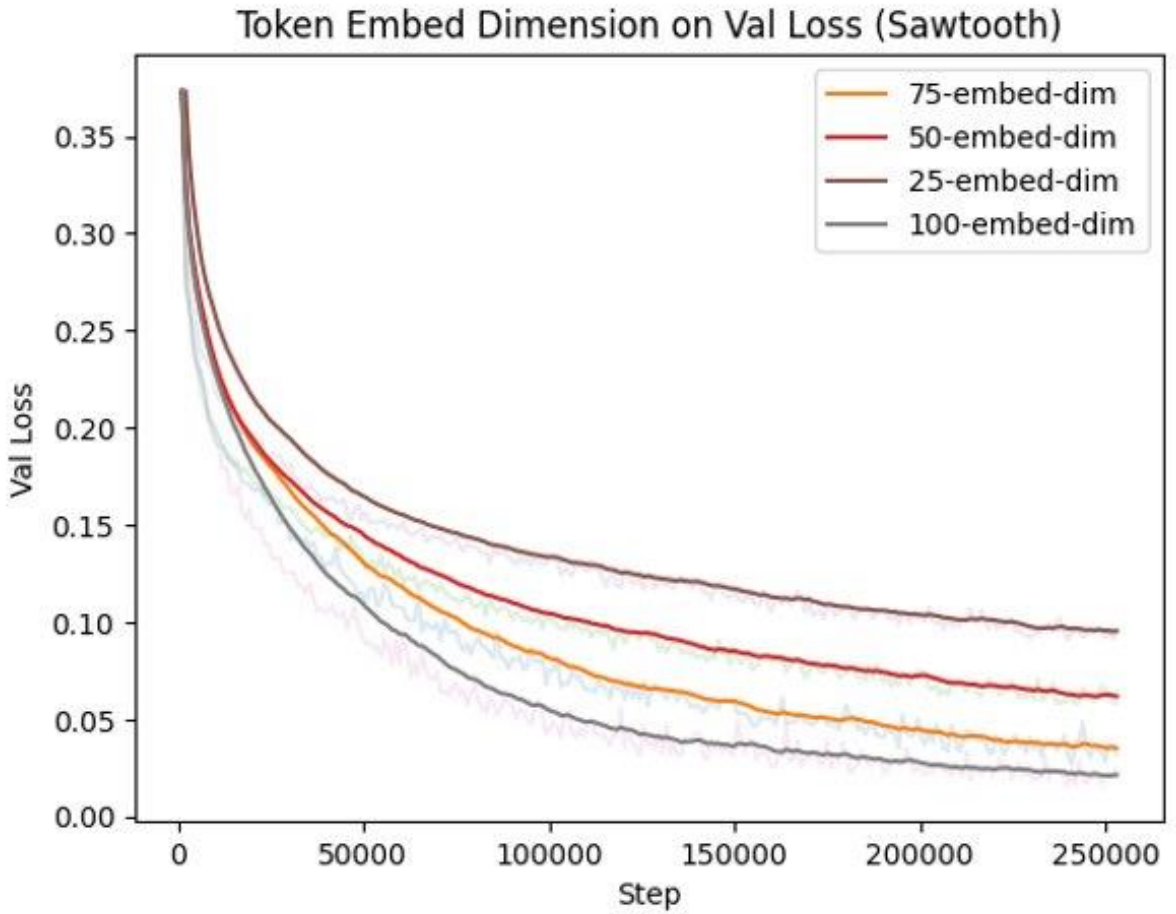
Take 1M Parameter Model and "back down on"

- **Token Embed Dim**
- **Head Embed Dim**
- **Number of Heads**

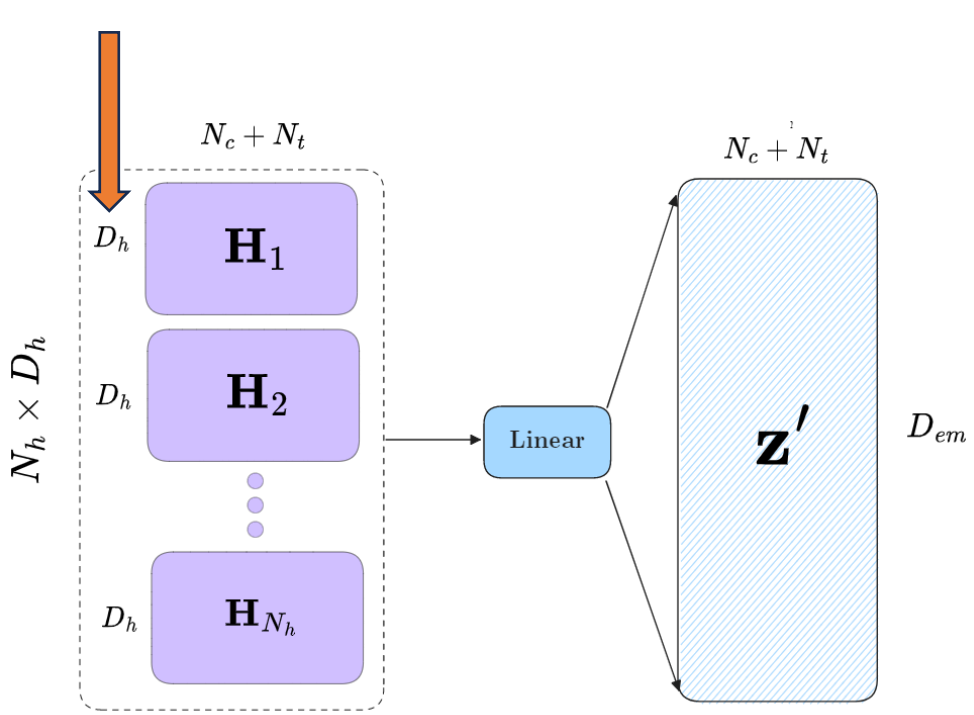
Explore effects on the Val Loss

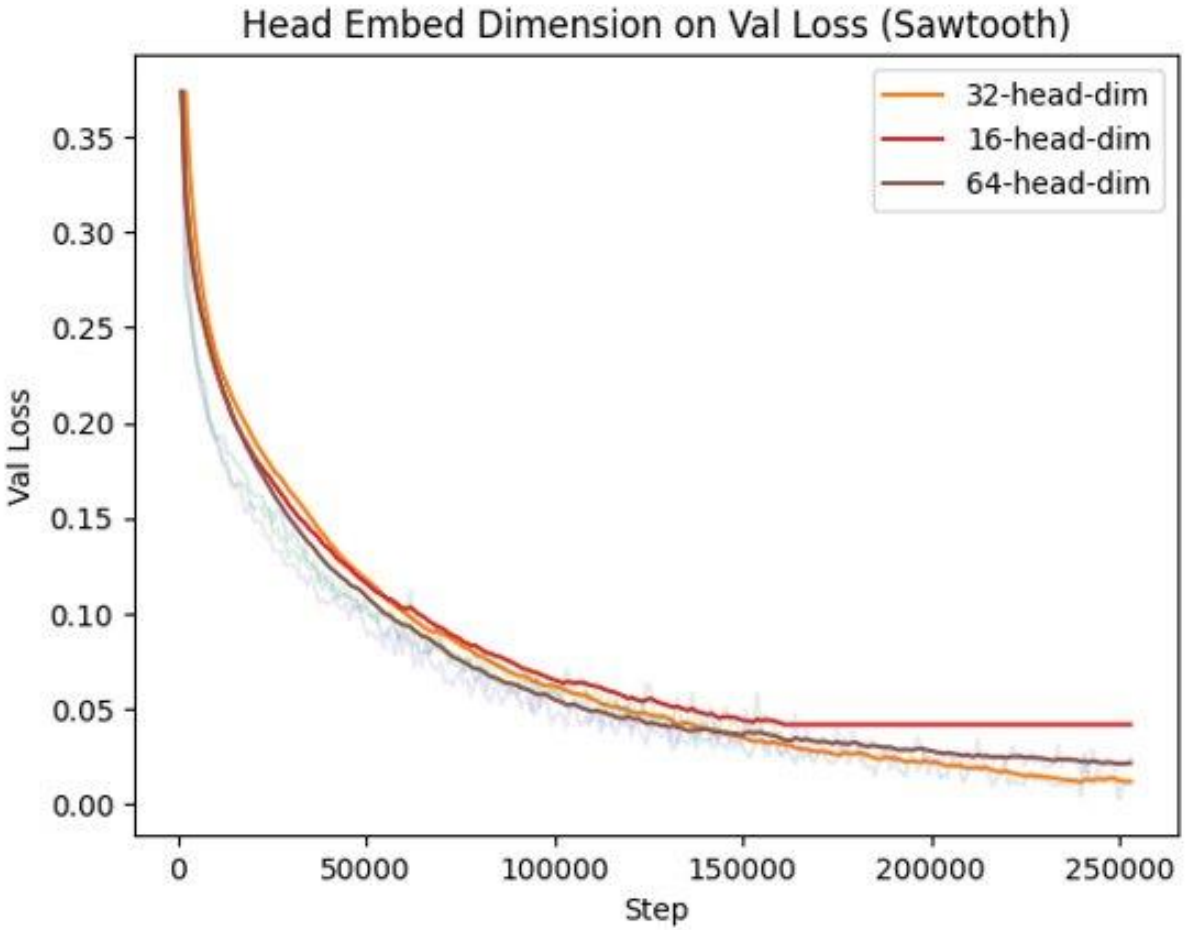
Token Embed Dim



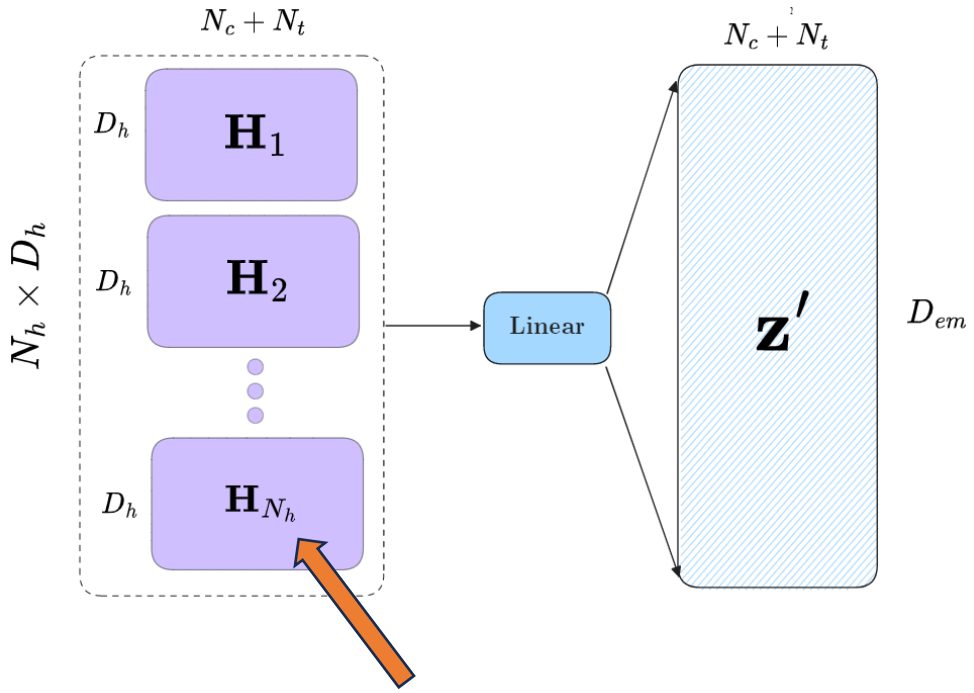


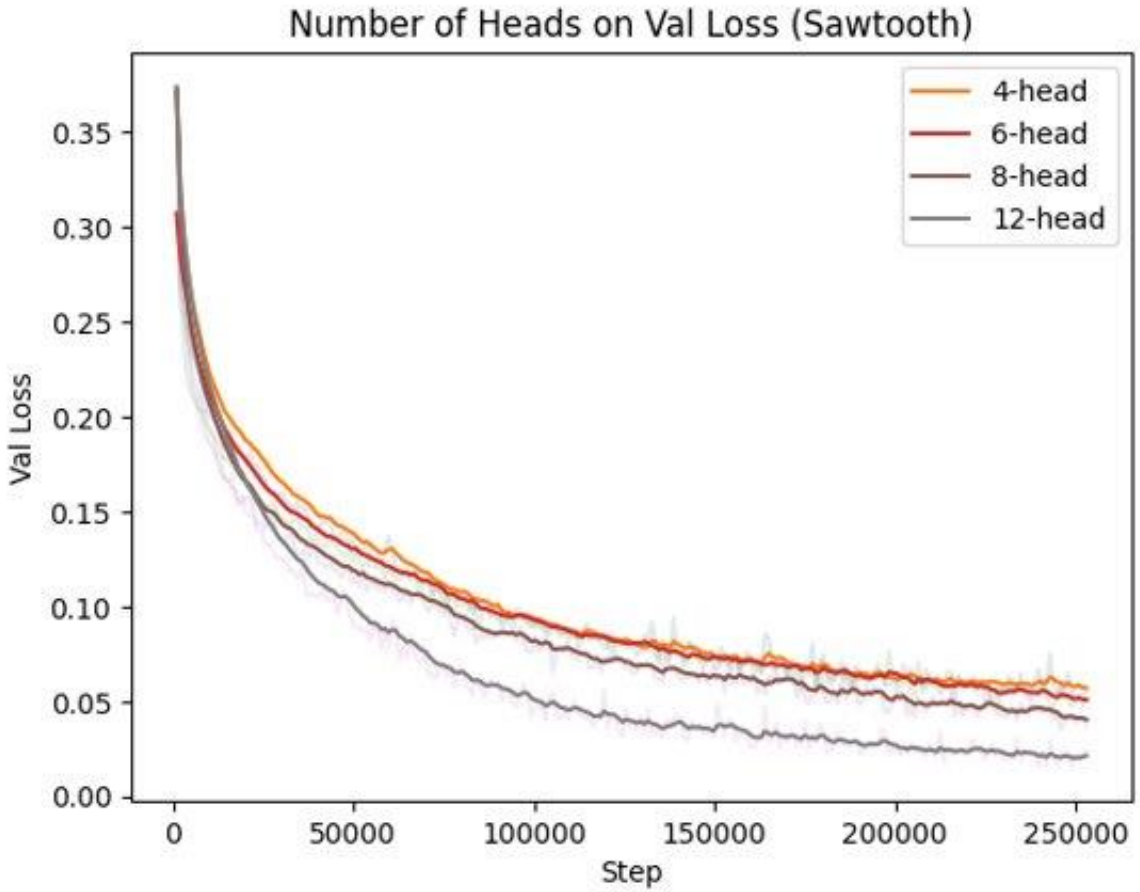
Head Embed Dim



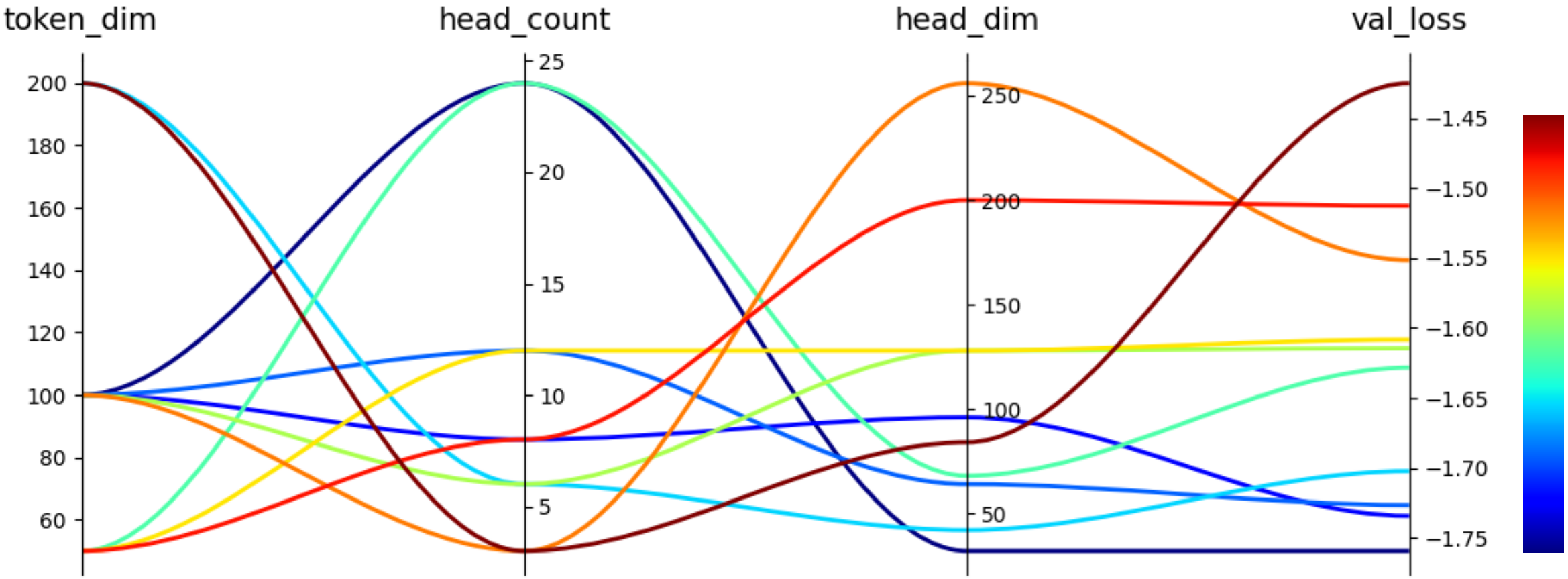


Number of Heads

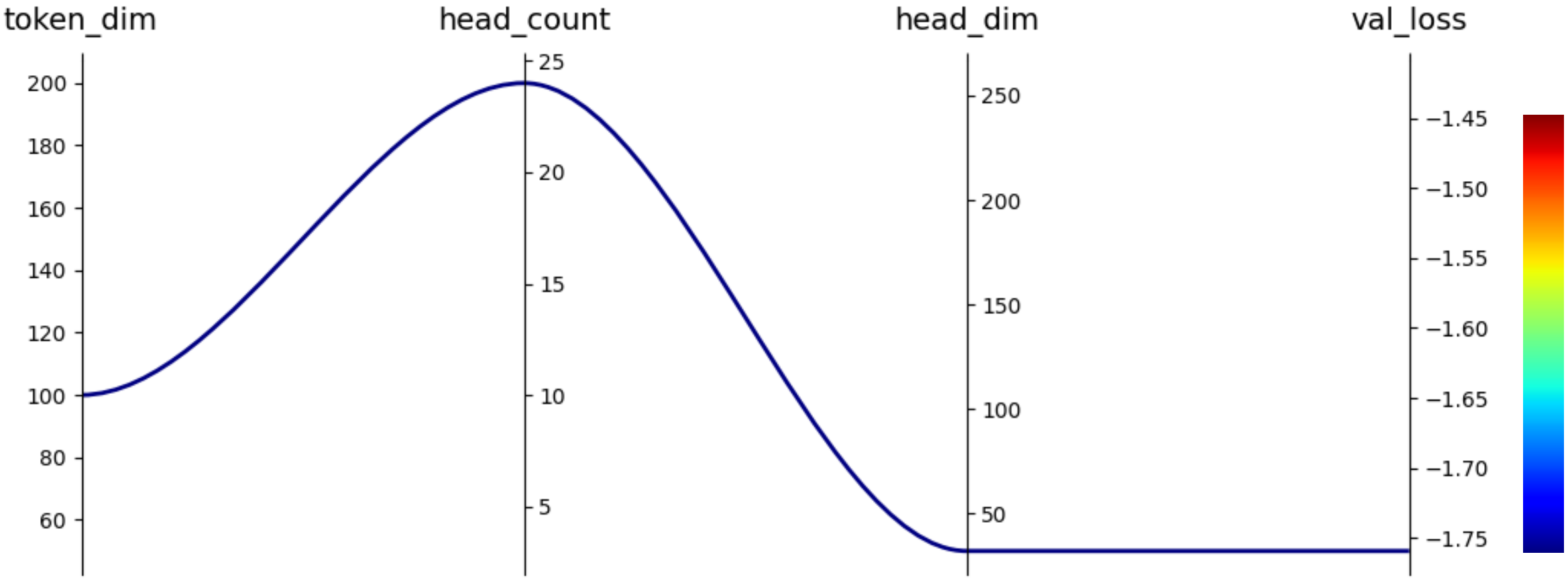




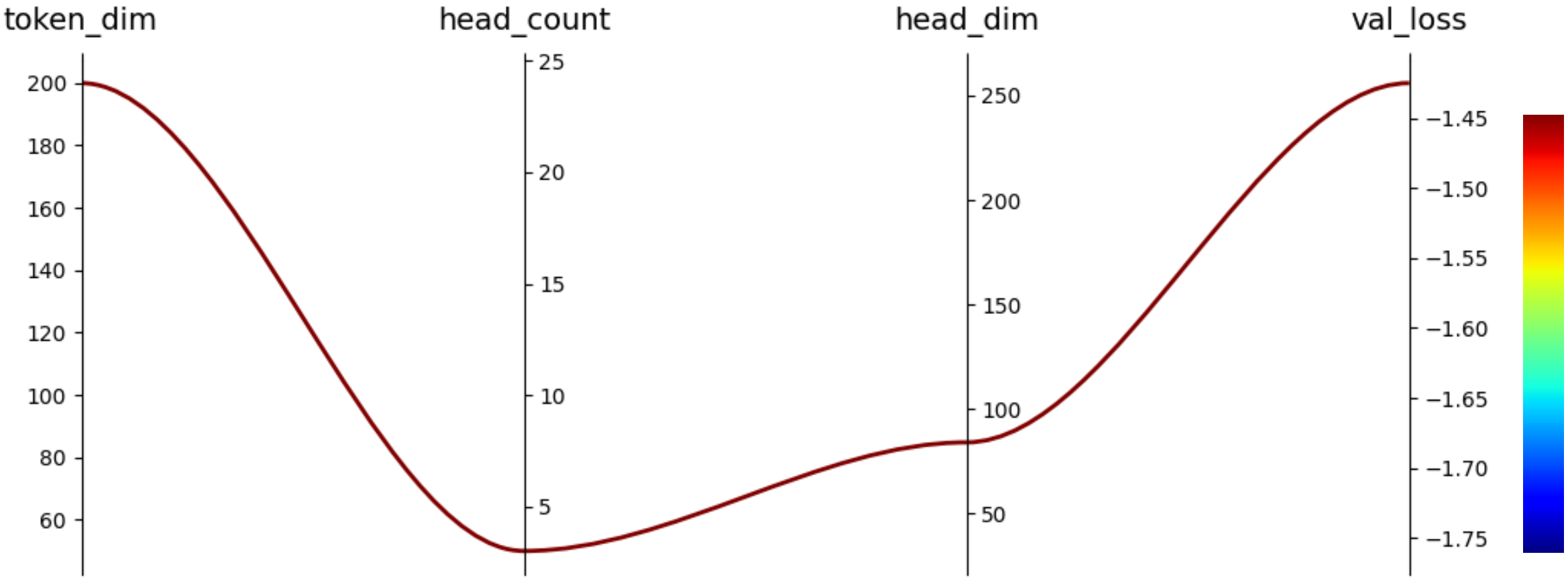
1M Param Models



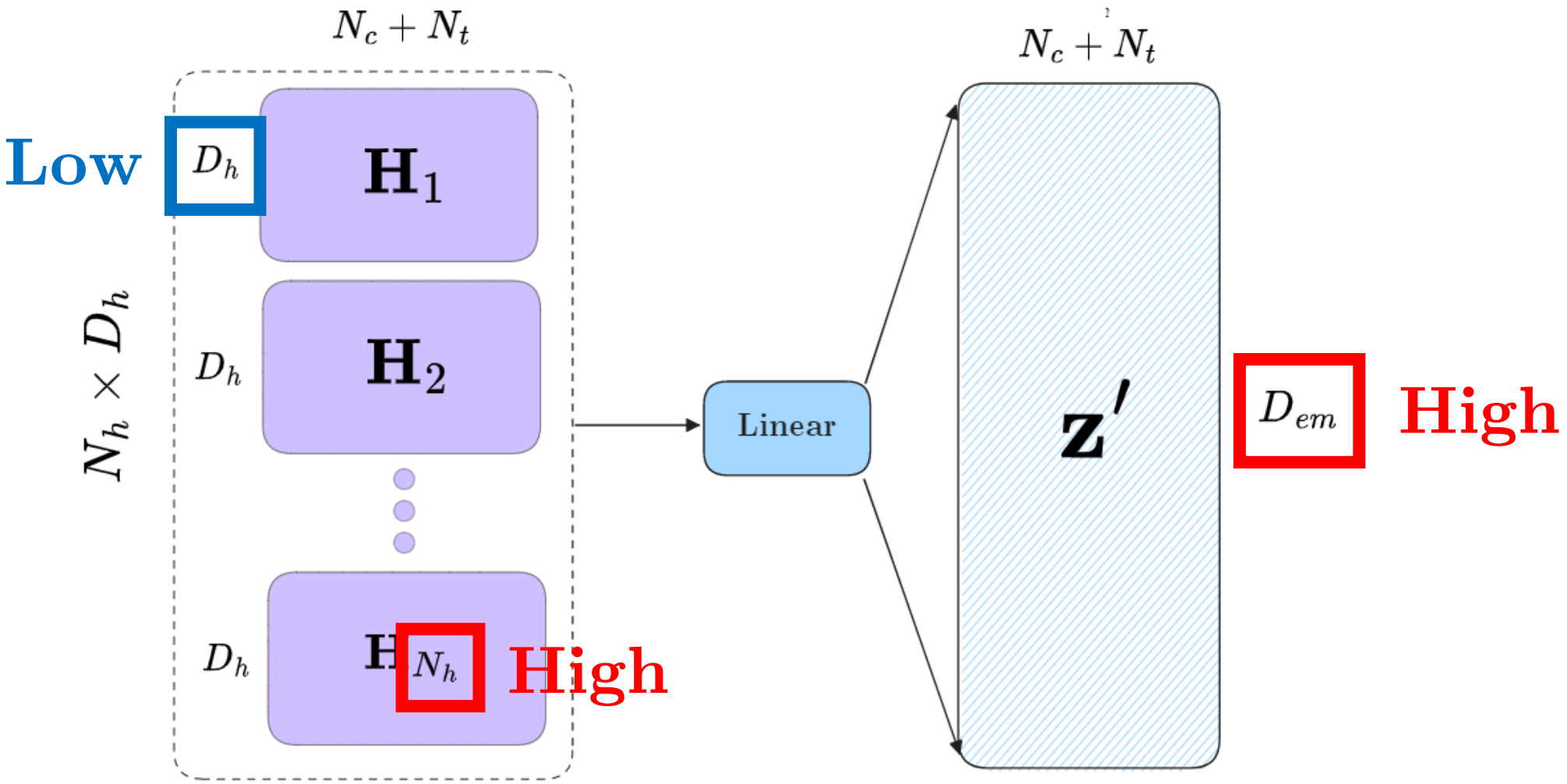
1M Param Models (Best)



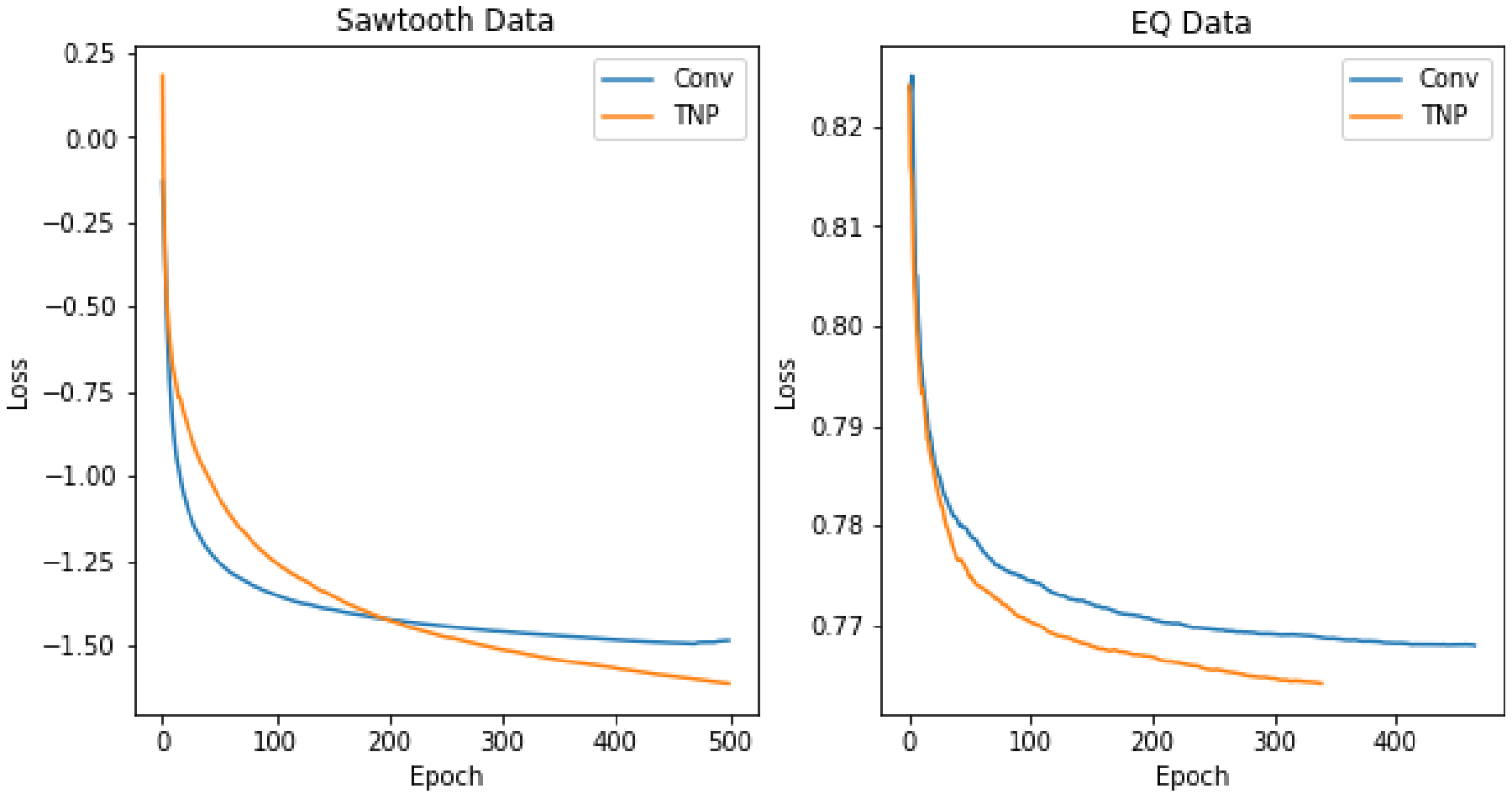
1M Param Models (Worst)



Optimal Hyperparameter Selection

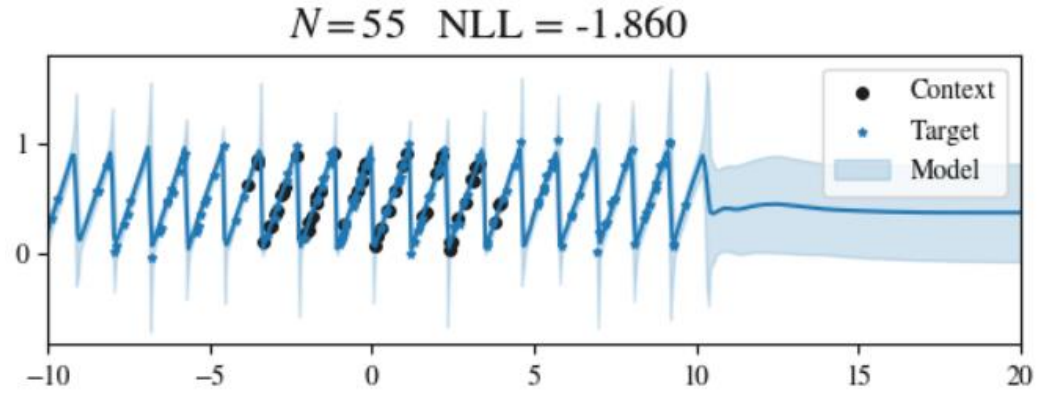


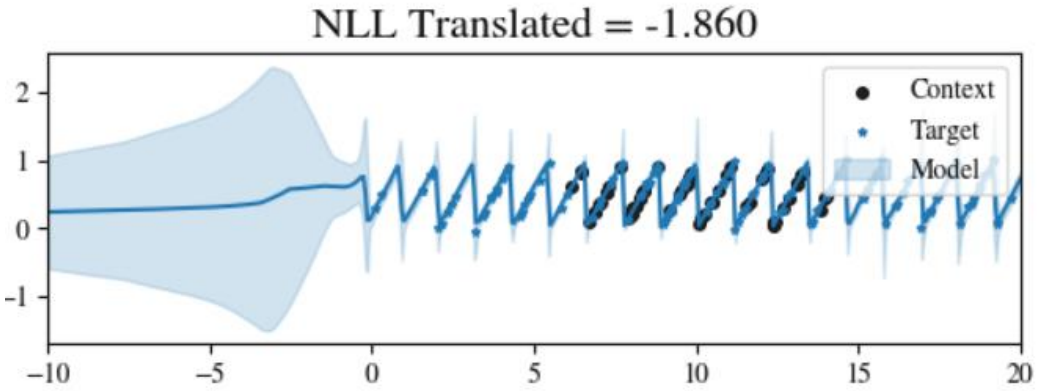
ConvNP vs TNP in Training



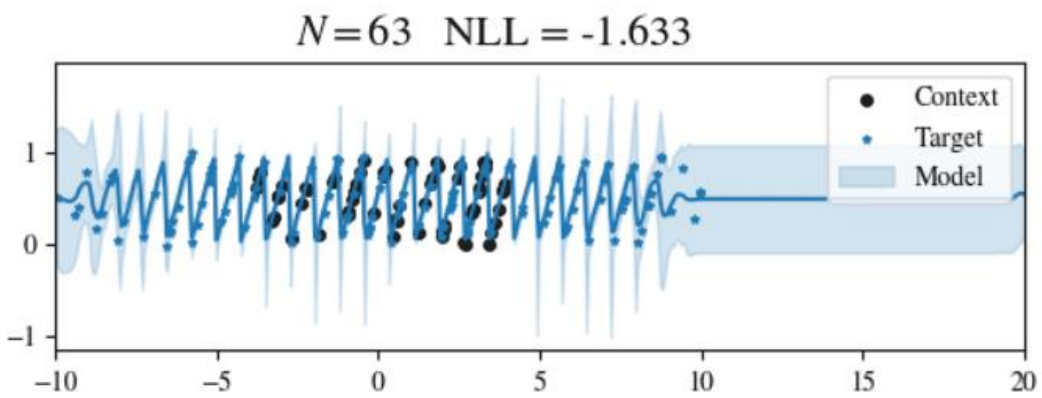
ConvNP vs TNP Sawtooth Fits

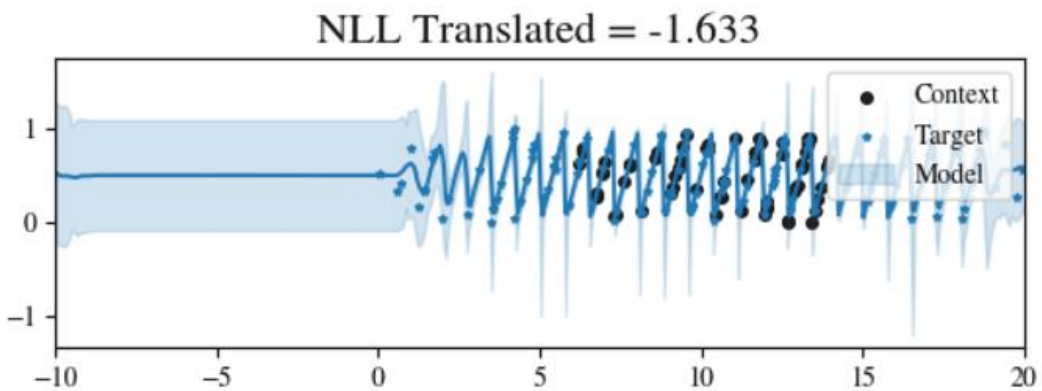
TNP





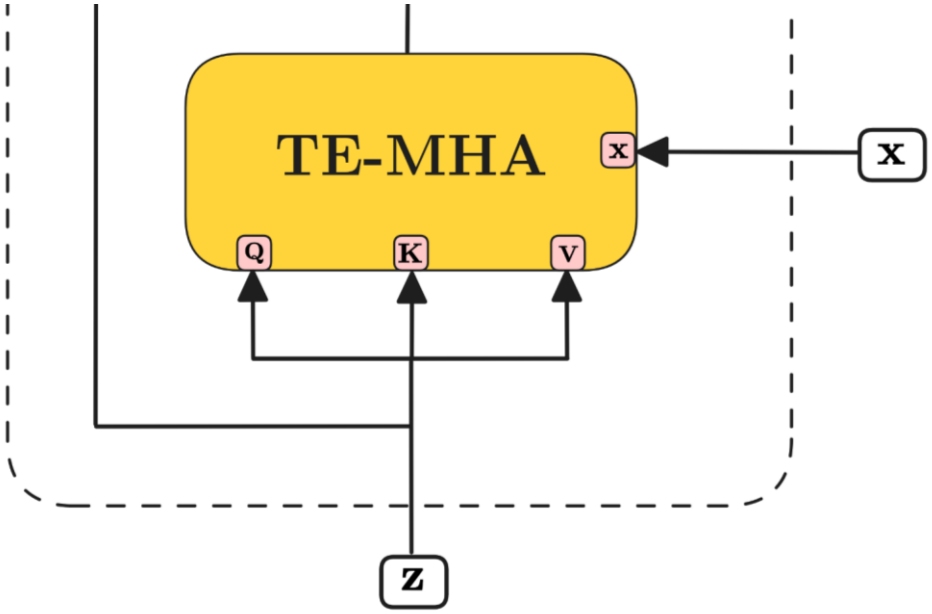
ConvNP





ConvNP vs TNP in Inference

- **Graphs/Metrics for Inference still in WIP**
- **However, it was clear that TNP uses a lot more memory!**



$$\mathcal{O}((N_c + N_t)^2)$$

Future Aims

- **Implement a memory efficient transformers using pseudotokens**
  - ***Constant Memory Attention Block, Feng et al***
  - ***Latent Bottlenecked Attentive Neural Processes, Feng et al***
  - ***Set Transformer, Lee et al***

- **Compare Models that have the same inference complexity**
- **Explore other datasets, especially in higher dimensions**