Probabilistic Machine Learning Project
Final Presentation

Conditional Neural Adaptive Processes (CNAPs)

Bhavesh Jain (20110038)

Dhairya Shah (20110054)

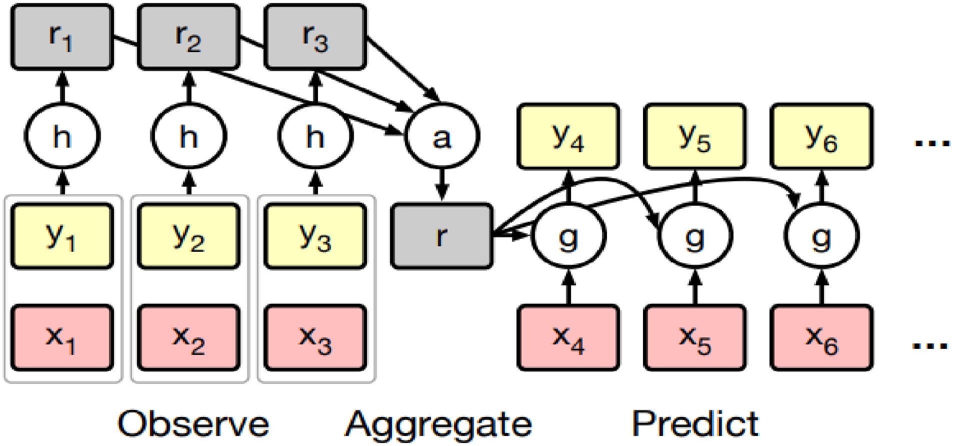
Lipika Rajpal (20110102)

R Yeeshu Dhurandhar (20110152)

Rahul Chembakasseril (20110158)

Mentors
Prof. Nipun Batra
Madhav Kanda
Sarth Dubey

Recap: Conditional Neural Adaptive Processes



Source: conditional_neural_processes.png

$$p\left(\boldsymbol{y}^{*} \middle| \boldsymbol{x}^{*}, \boldsymbol{\theta}, D^{\tau}\right) = p\left(\boldsymbol{y}^{*} \middle| \boldsymbol{x}^{*}, \boldsymbol{\theta}, \boldsymbol{\psi}^{\tau} = \boldsymbol{\psi}_{\boldsymbol{\phi}}\left(D^{\tau}\right)\right).$$

Recap: Hypernetwork

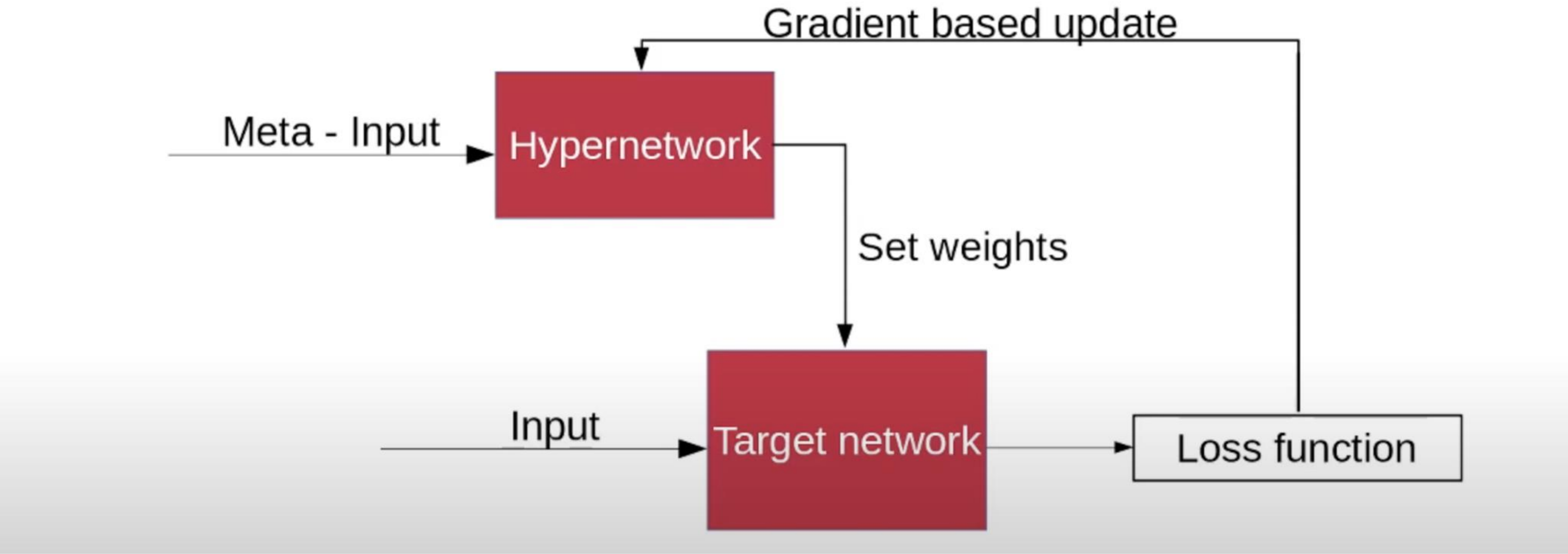
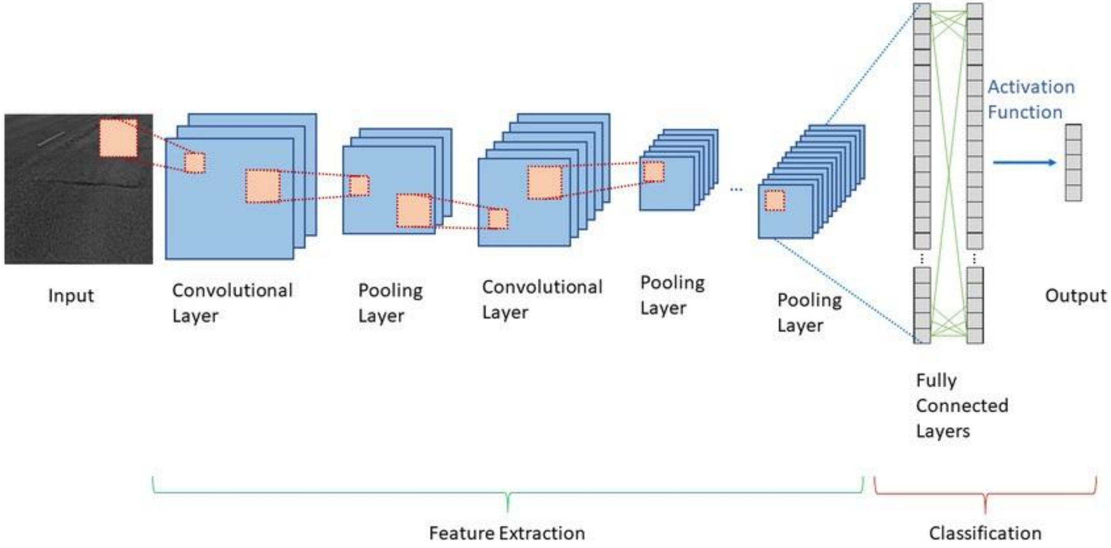


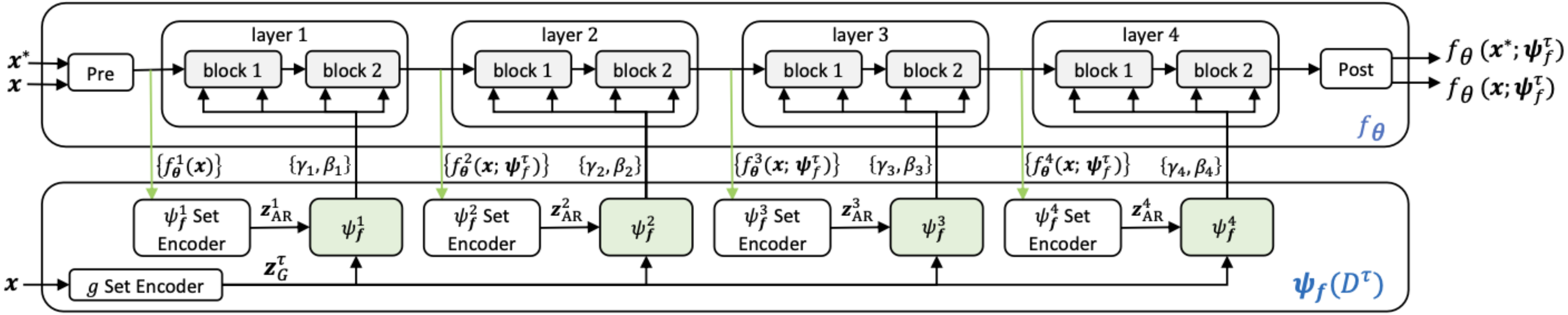
*Image is taken from @lipika

Traditional CNN



Source: CNN

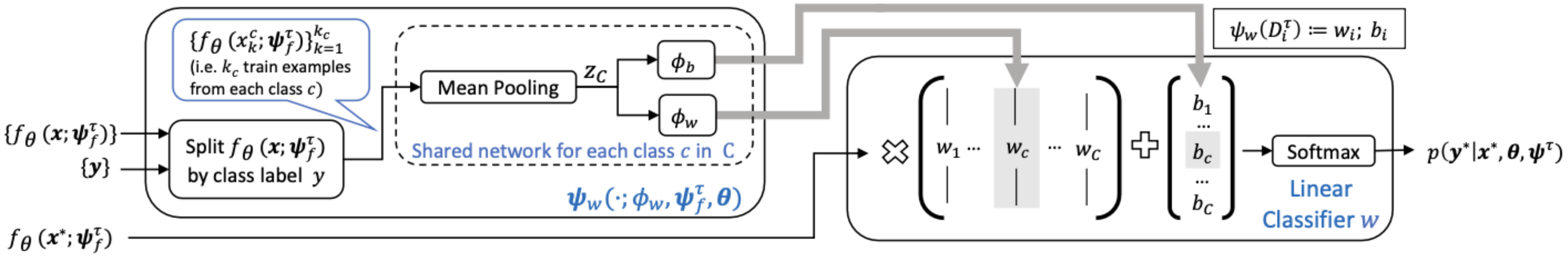
Recap: Conditional Neural Adaptive Processes (CNAPs)



FiLM layer parameters are obtained layer by layer

*Image is taken from the original paper

Recap: Conditional Neural Adaptive Processes (CNAPs)


$\{f_\theta(x_k^c;\psi_f^\tau)\}_{k=1}^{k_c}$
(i.e. k_c train examples from each class c)
Mean Pooling
z_C
ϕ_b
ϕ_w
Shared network for each class c in C
$\{f_\theta(\boldsymbol{x};\boldsymbol{\psi}_f^\tau)\}$
$\{\boldsymbol{y}\}$
Split $f_\theta(\boldsymbol{x};\boldsymbol{\psi}_f^\tau)$ by class label y
$\boldsymbol{\psi}_w(\cdot;\phi_w,\boldsymbol{\psi}_f^\tau,\boldsymbol{\theta})$
$f_\theta(\boldsymbol{x}^*;\boldsymbol{\psi}_f^\tau)$
$\psi_w(D_i^\tau) := w_i;\ b_i$
w_1 ··· w_c ··· w_C
b_1 ··· b_c ··· b_C
Softmax
Linear Classifier w
$p(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta},\boldsymbol{\psi}^\tau)$

Can support variable number of classes!

*Image is taken from the original paper

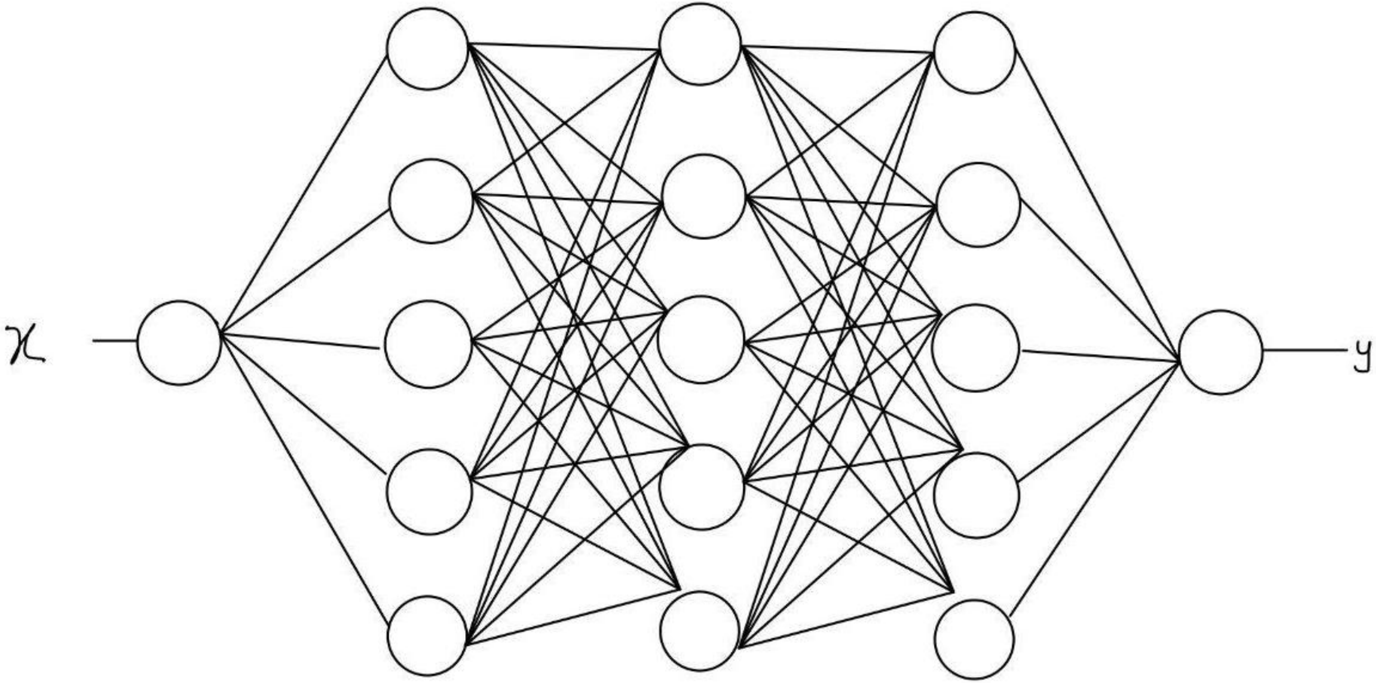
Feedbacks and Goals

- Experiment CNAPs on complex datasets with different context sizes. For example CelebA.
- Set some baseline for CNAPs, for example CNP, Hypernets, Fine-Tune.
- CNAPs on Regression with Autoregressive networks.

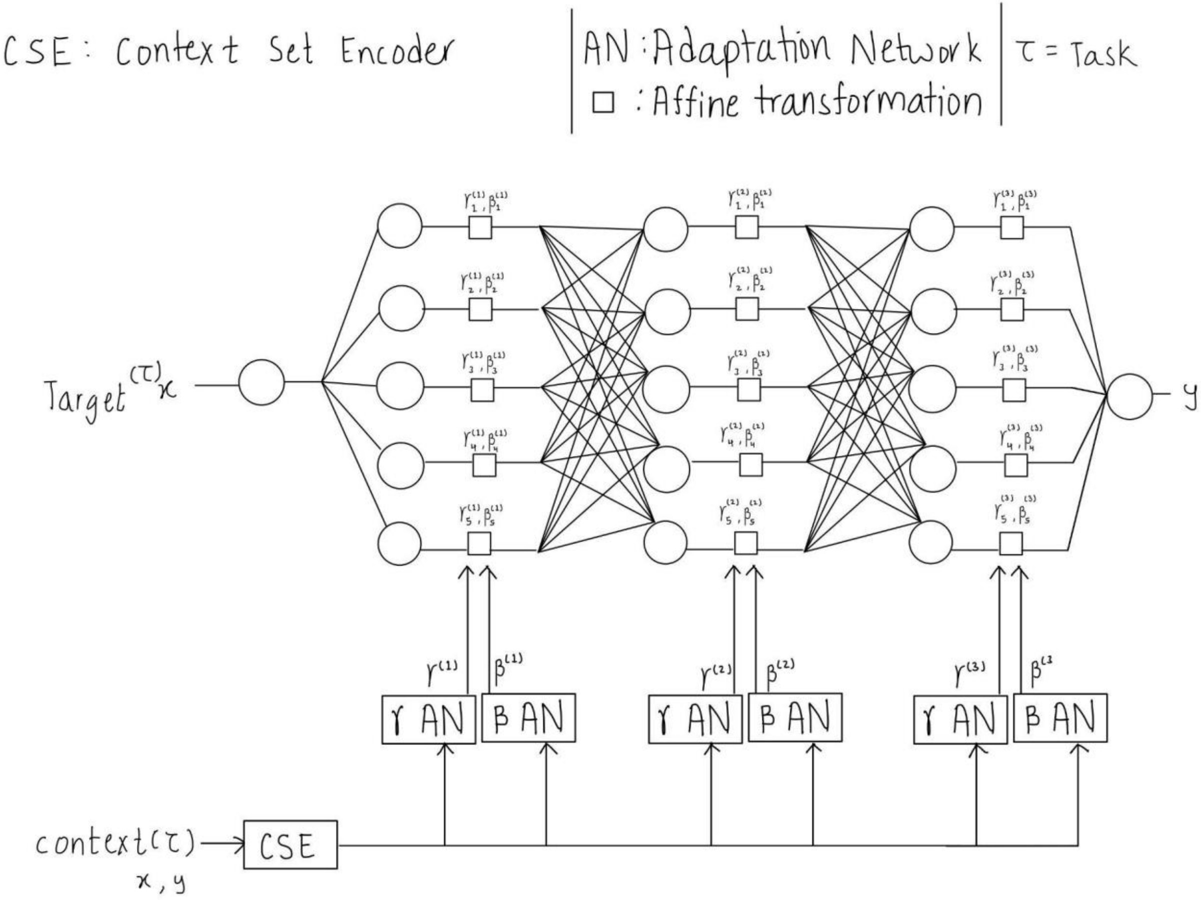
CNAPs for Regression Problem

1. Base Model
2. CNAPs without Autoregressive
3. CNAPs with Autoregressive

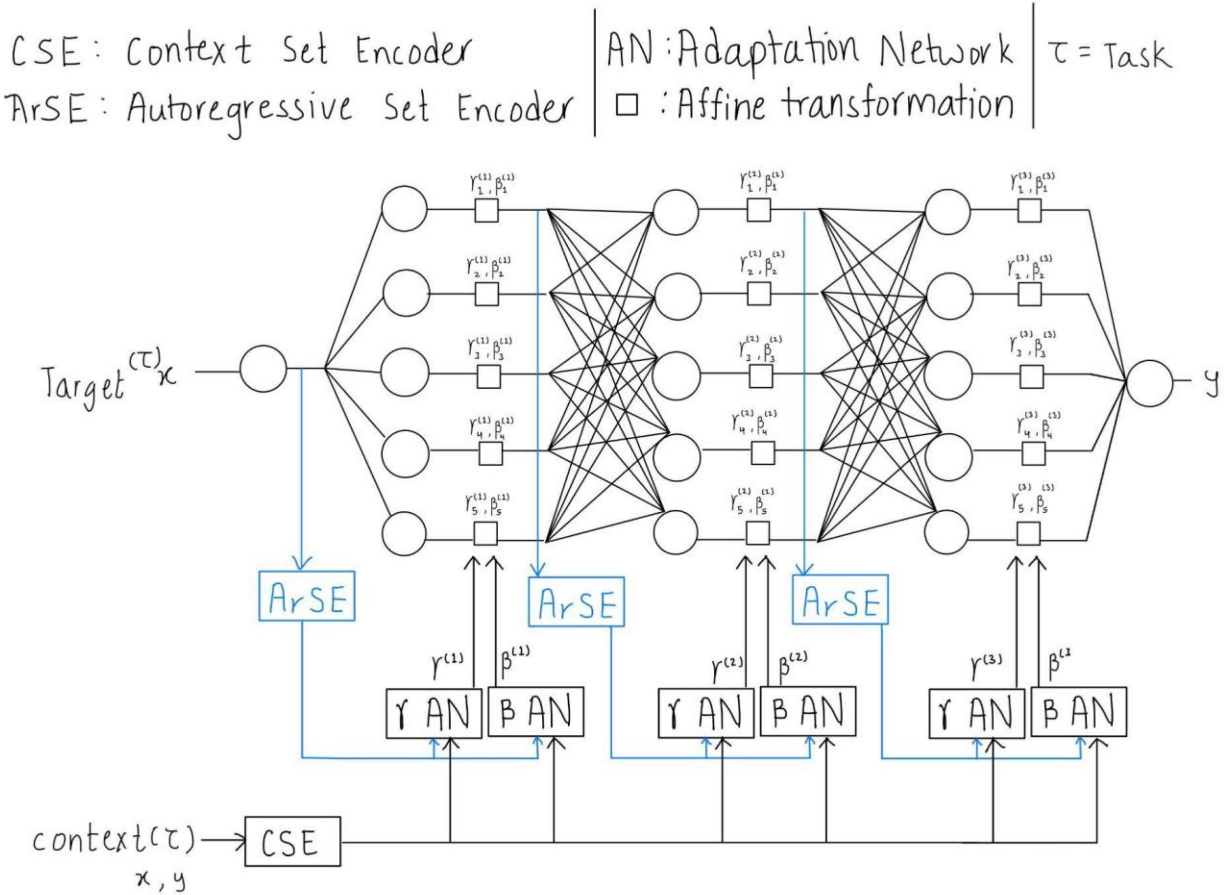
1. Base Model



2. CNAPs without Autoregressive

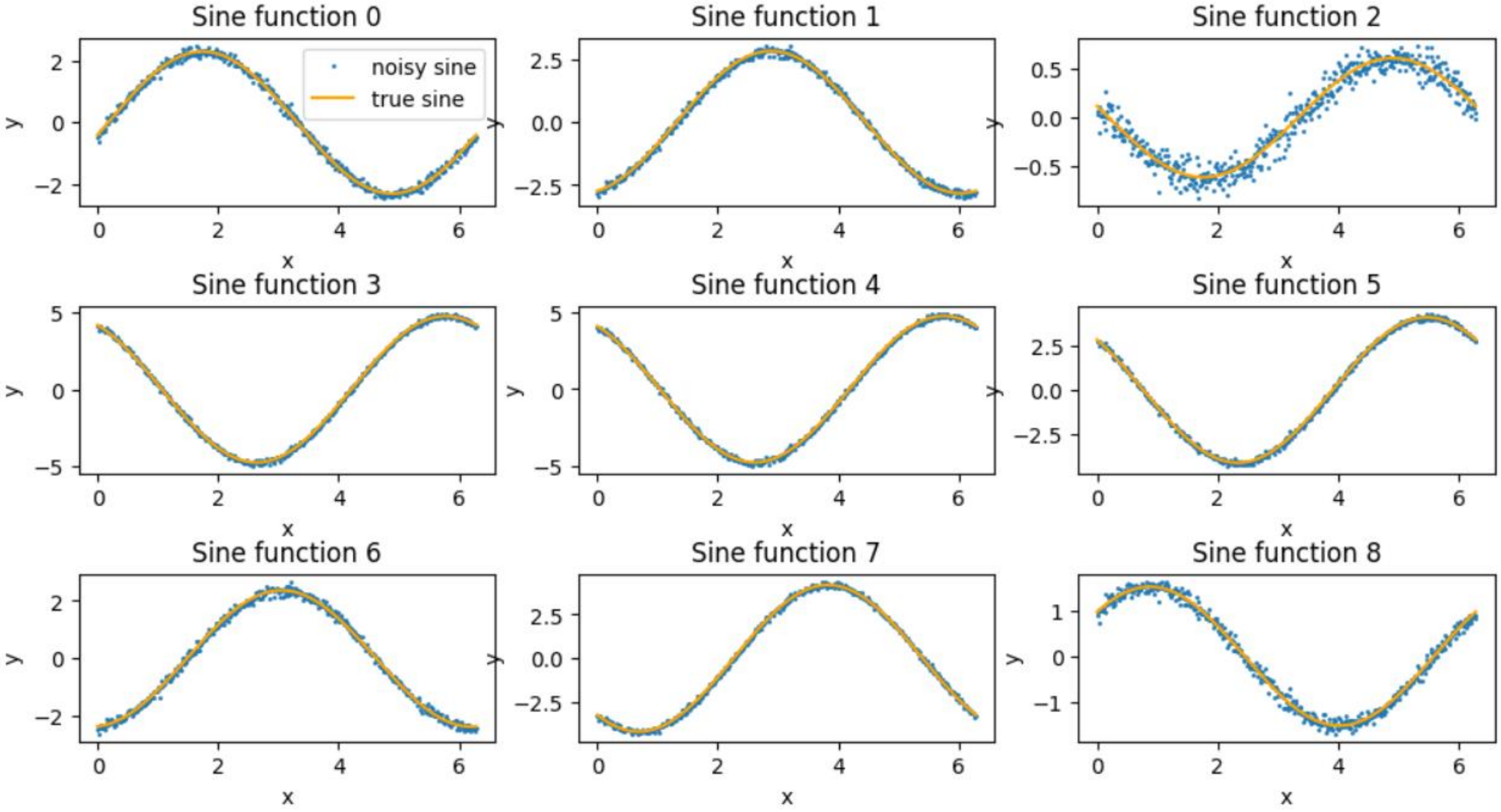

CSE : Context Set Encoder
AN : Adaptation Network
□ : Affine transformation
τ = Task
Target$^{(\tau)}$ x
$\gamma_1^{(1)}, \beta_1^{(1)}$
$\gamma_2^{(1)}, \beta_2^{(1)}$
$\gamma_3^{(1)}, \beta_3^{(1)}$
$\gamma_4^{(1)}, \beta_4^{(1)}$
$\gamma_5^{(1)}, \beta_5^{(1)}$
$\gamma_1^{(2)}, \beta_1^{(2)}$
$\gamma_2^{(2)}, \beta_2^{(2)}$
$\gamma_3^{(2)}, \beta_3^{(2)}$
$\gamma_4^{(2)}, \beta_4^{(2)}$
$\gamma_5^{(2)}, \beta_5^{(2)}$
$\gamma_1^{(3)}, \beta_1^{(3)}$
$\gamma_2^{(3)}, \beta_2^{(3)}$
$\gamma_3^{(3)}, \beta_3^{(3)}$
$\gamma_4^{(3)}, \beta_4^{(3)}$
$\gamma_5^{(3)}, \beta_5^{(3)}$
y
$\gamma^{(1)}$
$\beta^{(1)}$
$\gamma^{(2)}$
$\beta^{(2)}$
$\gamma^{(3)}$
$\beta^{(3}$
γ AN
β AN
γ AN
β AN
γ AN
β AN
context(τ)
x, y
CSE

3. CNAPs with Autoregressive

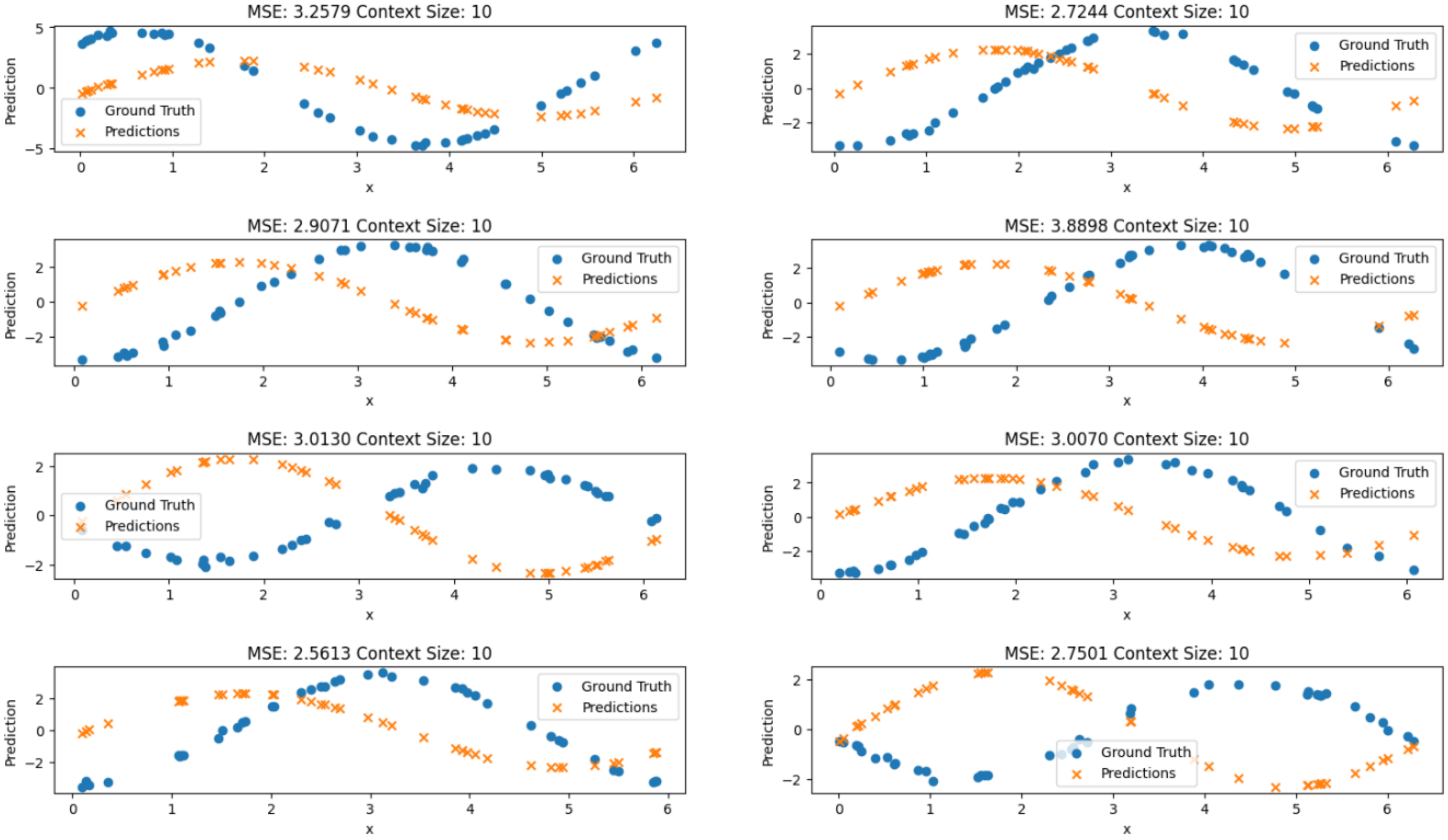

CSE : Context Set Encoder
ArSE : Autoregressive Set Encoder
AN : Adaptation Network
□ : Affine transformation
τ = Task
$\gamma_1^{(1)}, \beta_1^{(1)}$
$\gamma_2^{(1)}, \beta_2^{(1)}$
$\gamma_3^{(1)}, \beta_3^{(1)}$
$\gamma_4^{(1)}, \beta_4^{(1)}$
$\gamma_5^{(1)}, \beta_5^{(1)}$
$\gamma_1^{(2)}, \beta_1^{(2)}$
$\gamma_2^{(2)}, \beta_2^{(2)}$
$\gamma_3^{(2)}, \beta_3^{(2)}$
$\gamma_4^{(2)}, \beta_4^{(2)}$
$\gamma_5^{(2)}, \beta_5^{(2)}$
$\gamma_1^{(3)}, \beta_1^{(3)}$
$\gamma_2^{(3)}, \beta_2^{(3)}$
$\gamma_3^{(3)}, \beta_3^{(3)}$
$\gamma_4^{(3)}, \beta_4^{(3)}$
$\gamma_5^{(3)}, \beta_5^{(3)}$
Target$^{(\tau)}$ x
y
ArSE
ArSE
ArSE
$\gamma^{(1)}$
$\beta^{(1)}$
$\gamma^{(2)}$
$\beta^{(2)}$
$\gamma^{(3)}$
$\beta^{(3}$
γ AN
β AN
γ AN
β AN
γ AN
β AN
context(τ)
x , y
CSE

(Some) underlying Sine functions- act as datasets

y = A sin(x + phi), 0.5<= A <=5, 0<= phi <=2*pi



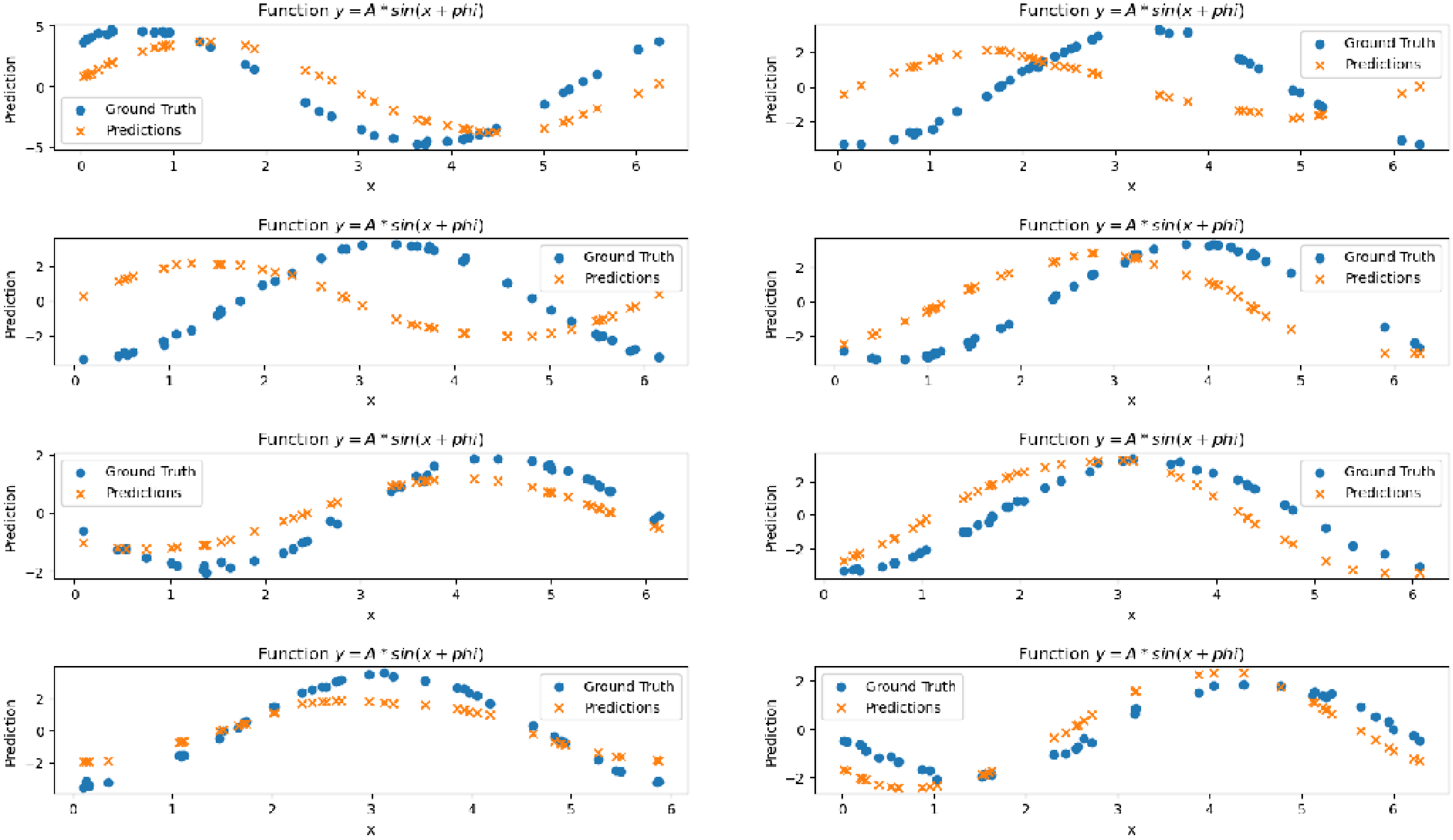
1. Results: Base Model



2. Results: CNAPs without Autoregressive

Context Set- 10

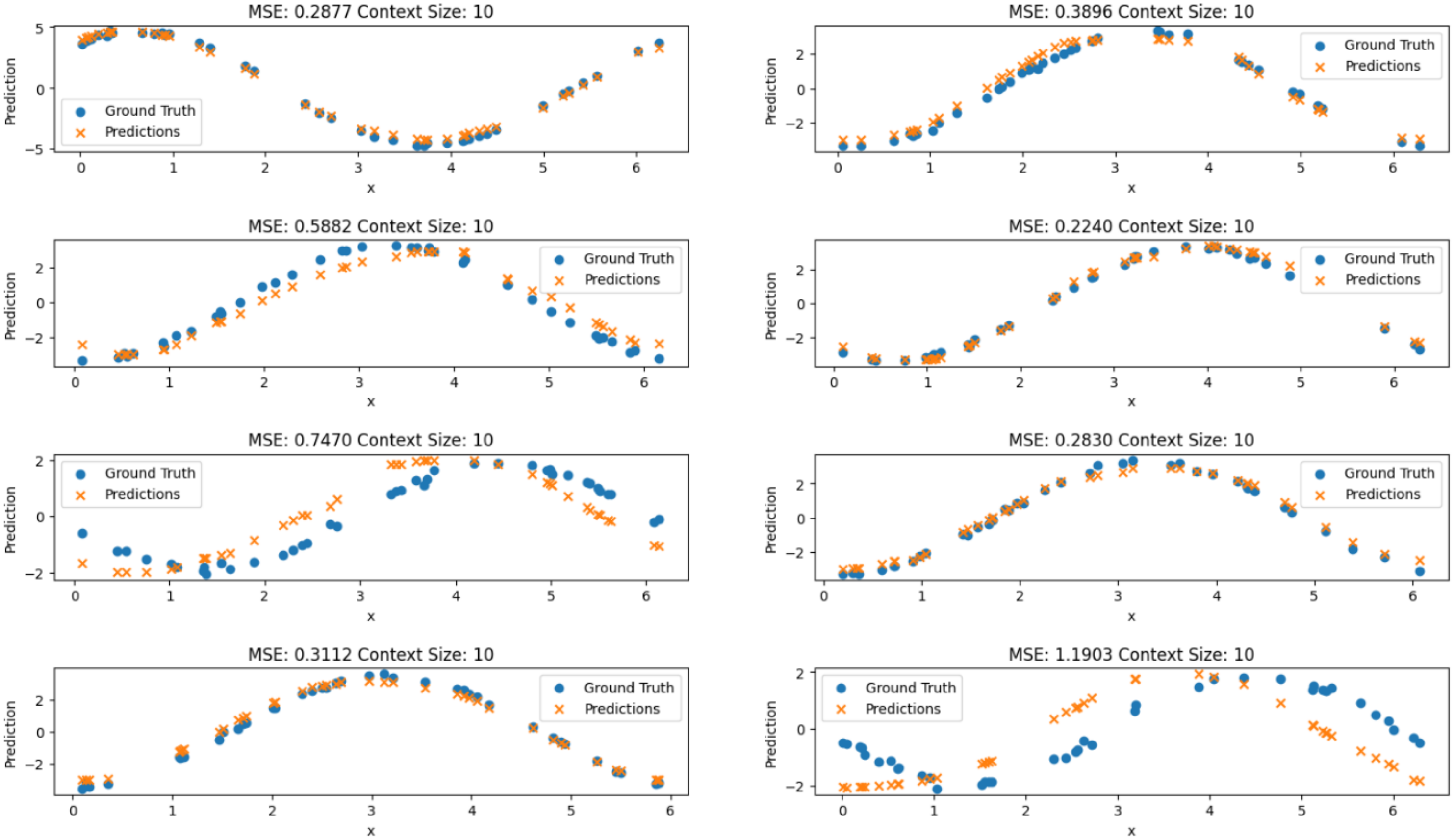
Target Set- 40



3. CNAPs with Autoregressive

Context Set- 10

Target Set- 40



MSE: 0.2877 Context Size: 10

MSE: 0.3896 Context Size: 10

MSE: 0.5882 Context Size: 10

MSE: 0.2240 Context Size: 10

MSE: 0.7470 Context Size: 10

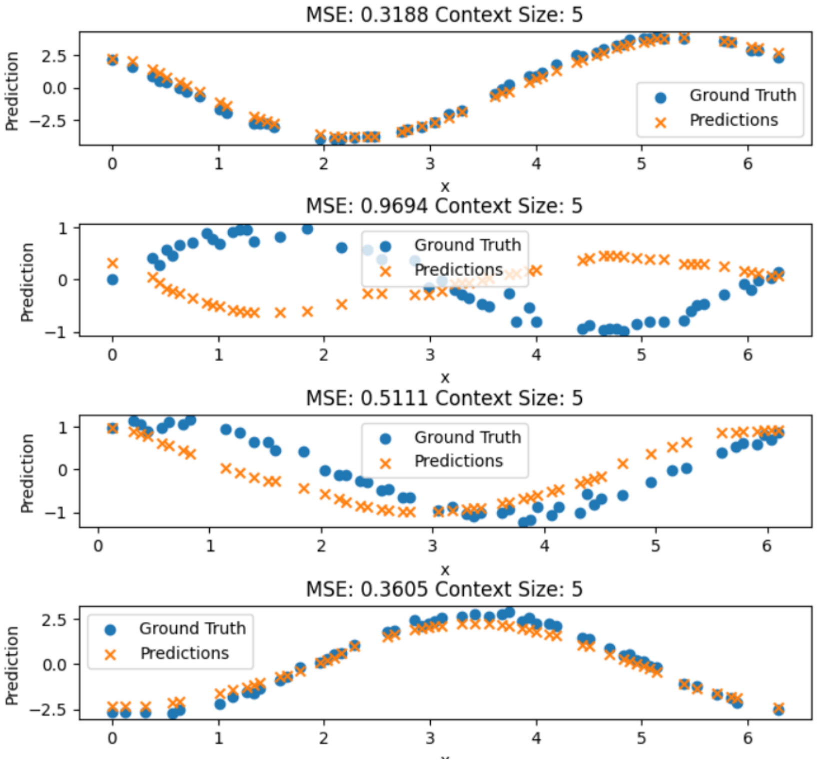
MSE: 0.2830 Context Size: 10

MSE: 0.3112 Context Size: 10

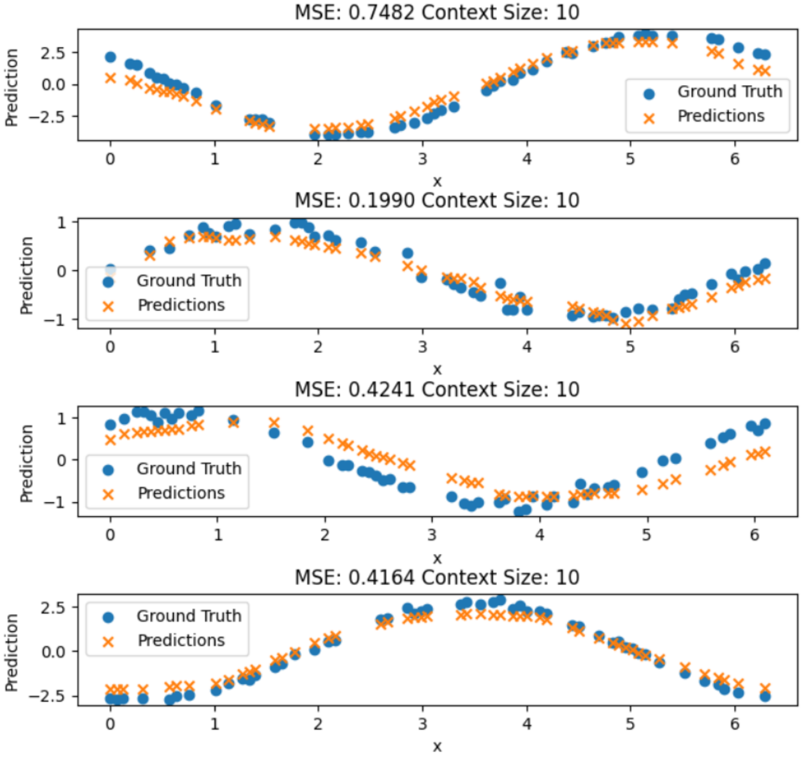
MSE: 1.1903 Context Size: 10

Effect of Increasing Number of Context Points

Context size: 5

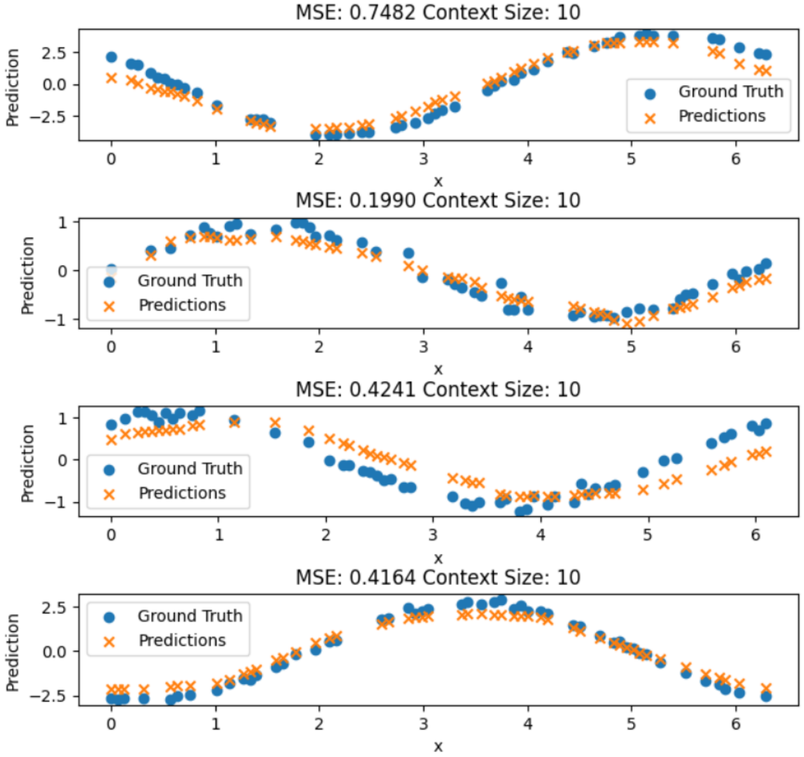


Context size: 10



Effect of Increasing Context Points

Context size: 10



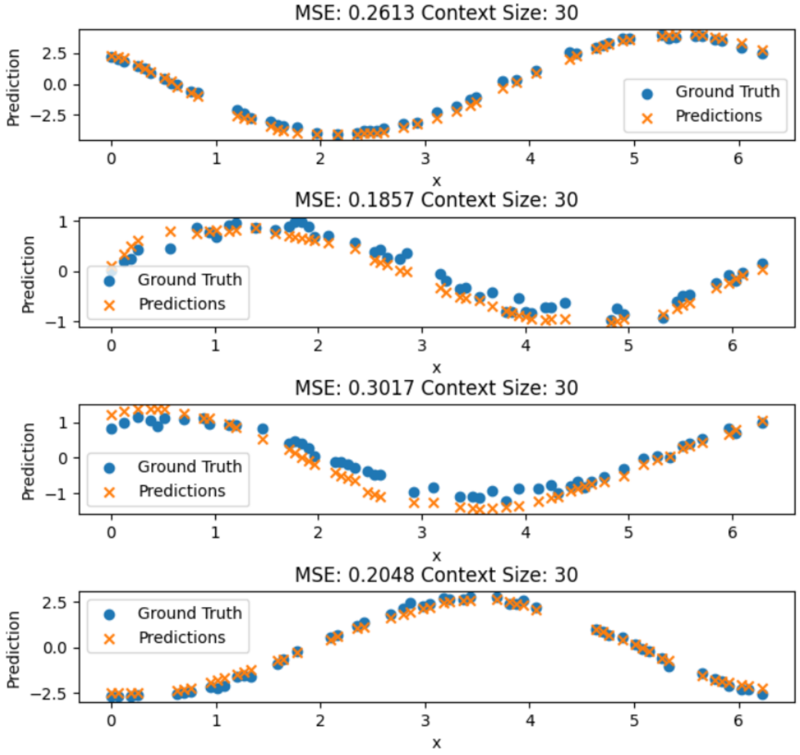
MSE: 0.7482 Context Size: 10

MSE: 0.1990 Context Size: 10

MSE: 0.4241 Context Size: 10

MSE: 0.4164 Context Size: 10

Context size: 30



MSE: 0.2613 Context Size: 30

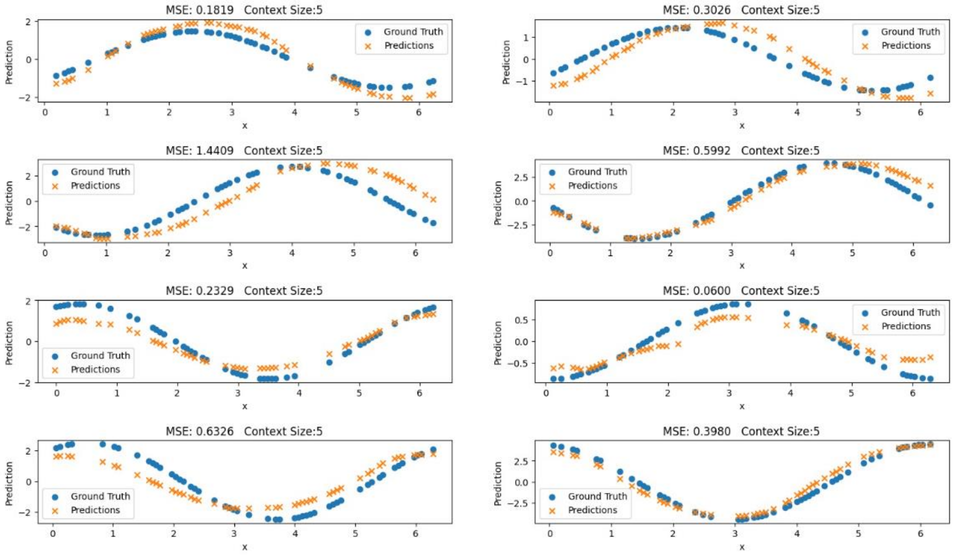
MSE: 0.1857 Context Size: 30

MSE: 0.3017 Context Size: 30

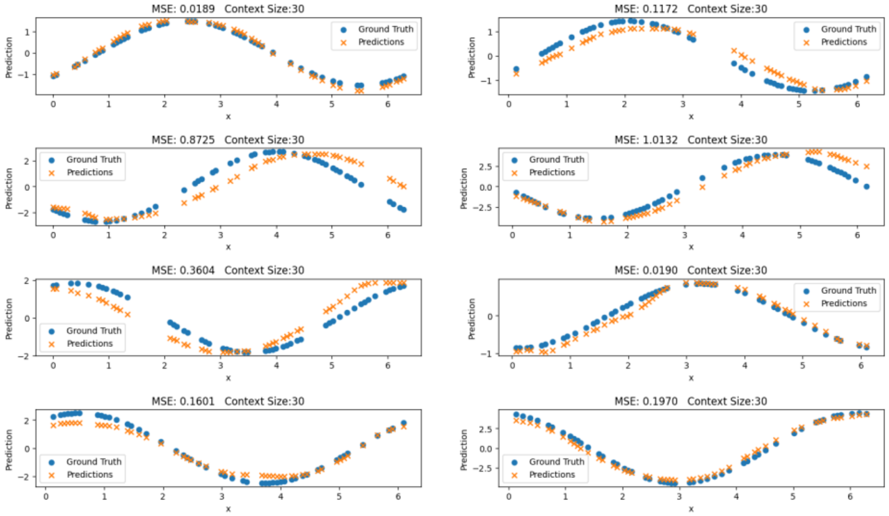
MSE: 0.2048 Context Size: 30

Comparison with baseline models: CNPs

Context size: 5

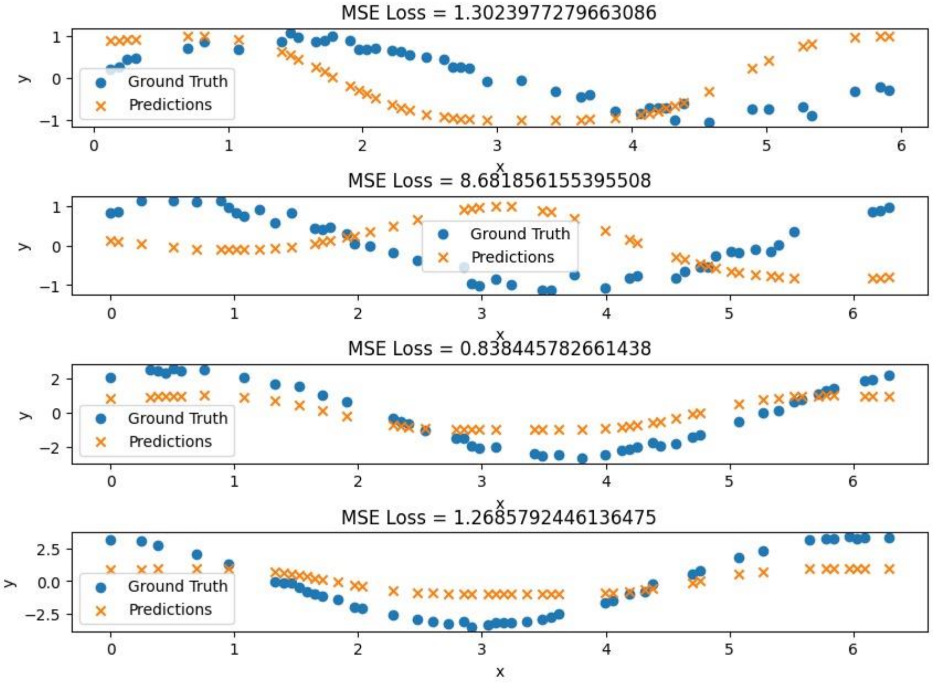


Context size: 30

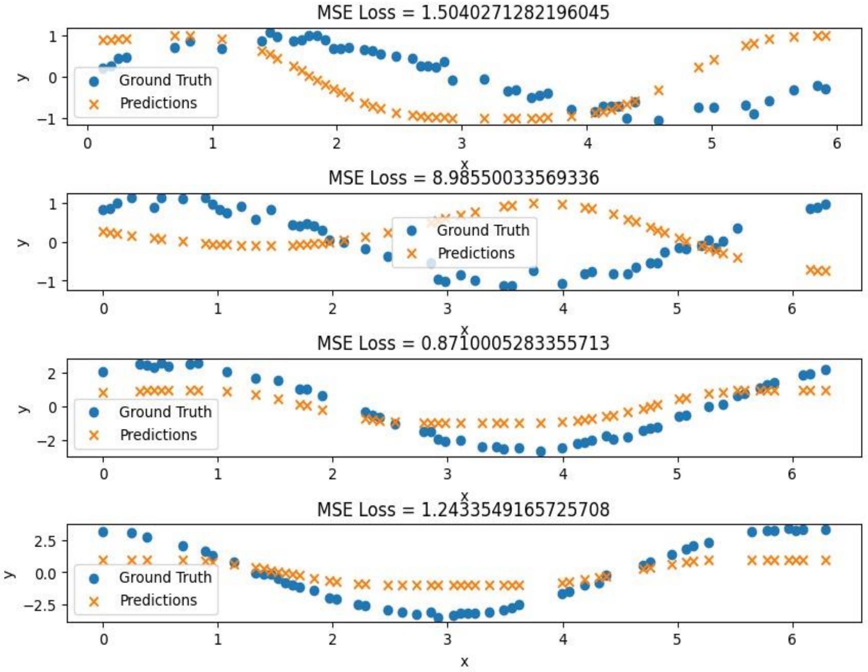


Comparison with baseline models: HyperNetworks

Context size: 5

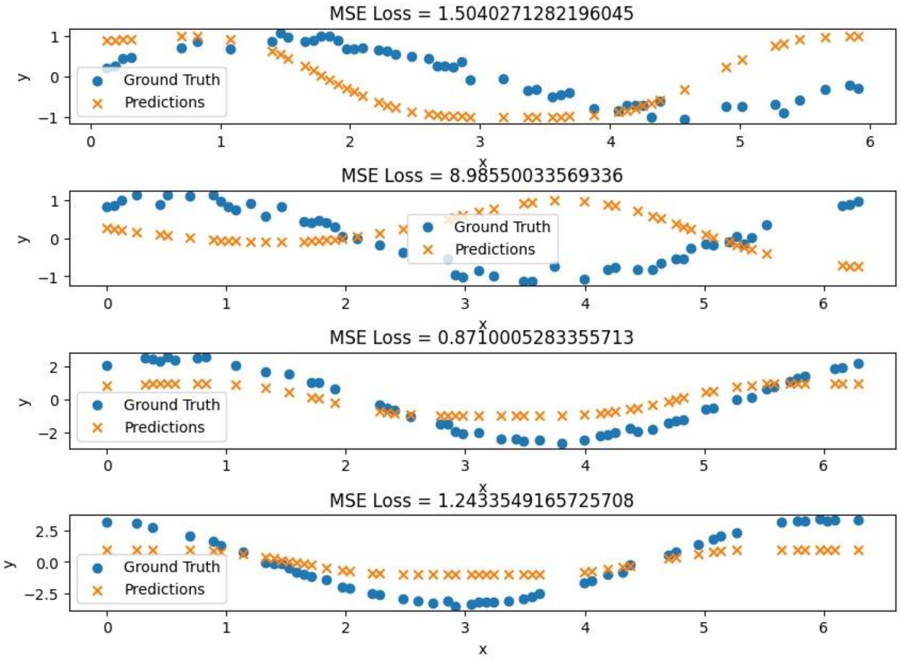


Context size: 10

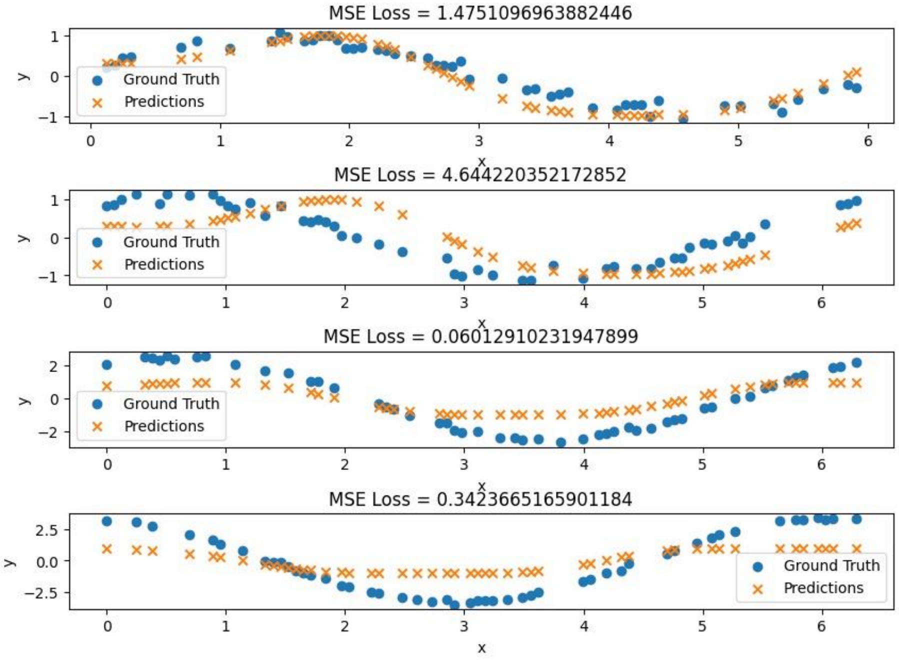


Comparison with baseline models: HyperNetworks

Context size: 10



Context size: 30



Observations

1. As we move from base model to CNAPs without Autoregressive to CNAPs with Autoregressive, the model is able to perform better for the meta learning task. This is due to the increase in ability to represent each of the context points as well as target points.
2. As the number of context points increases, the quality of results also increases. The increase in number of data points increases the information about the task in hand. Therefore, more the information, better the representation, better will be the result.
3. Results obtained from the CNAPs regression analysis are comparable to those obtained via CNPs, but far outperform the results from Hypernetworks.

Try CNAPs on CelebA

Why?

CNAPs on the CelebA Dataset

- 2,00,000+ images
- 10,000+ celebrities
- 40 facial attributes each (big nose, bald, blonde, etc.)

Not an obvious multitask or multi-class classification problem.

Diverse and challenging dataset…

CelebA vs Cifar10, Cifar100, MNIST

Earlier -

Classifying objects, digits.

Now -

Has features like hair color, eyeglasses etc from faces of celebrities.

Harder task, more fine grained attribute recognition.

Try CNAPs on CelebA

Curated 3 completely new datasets using CelebA and Stanford Cars dataset:

- Celeb10
- CelebFace150
- CelebCars

Try CNAPs on Celeb10

Splits and the process of training:

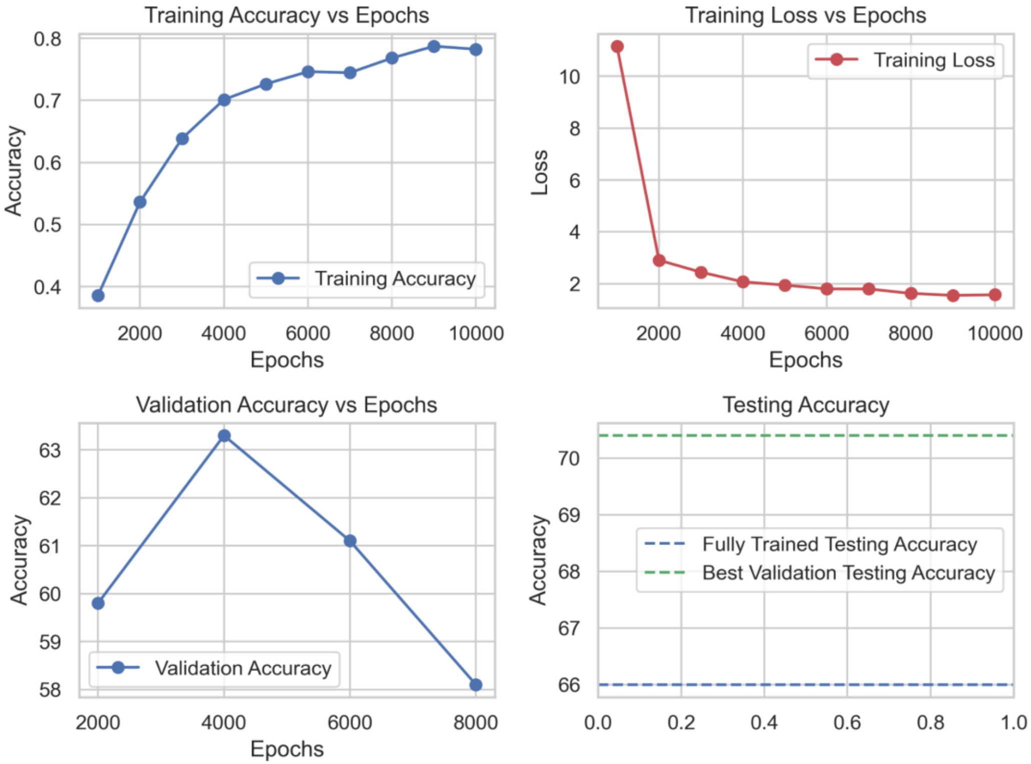
- Celeb10
  - Train: "Bald", "Big_Lips", "Receding_Hairline", "Wearing_Hat", "Brown_Hair", "Pale_Skin".
  - Test / Validation: "Bangs", "Blond_Hair", "Black_Hair", "Eyeglasses". Each class having 1600 images for train, test, and validation.

Process?

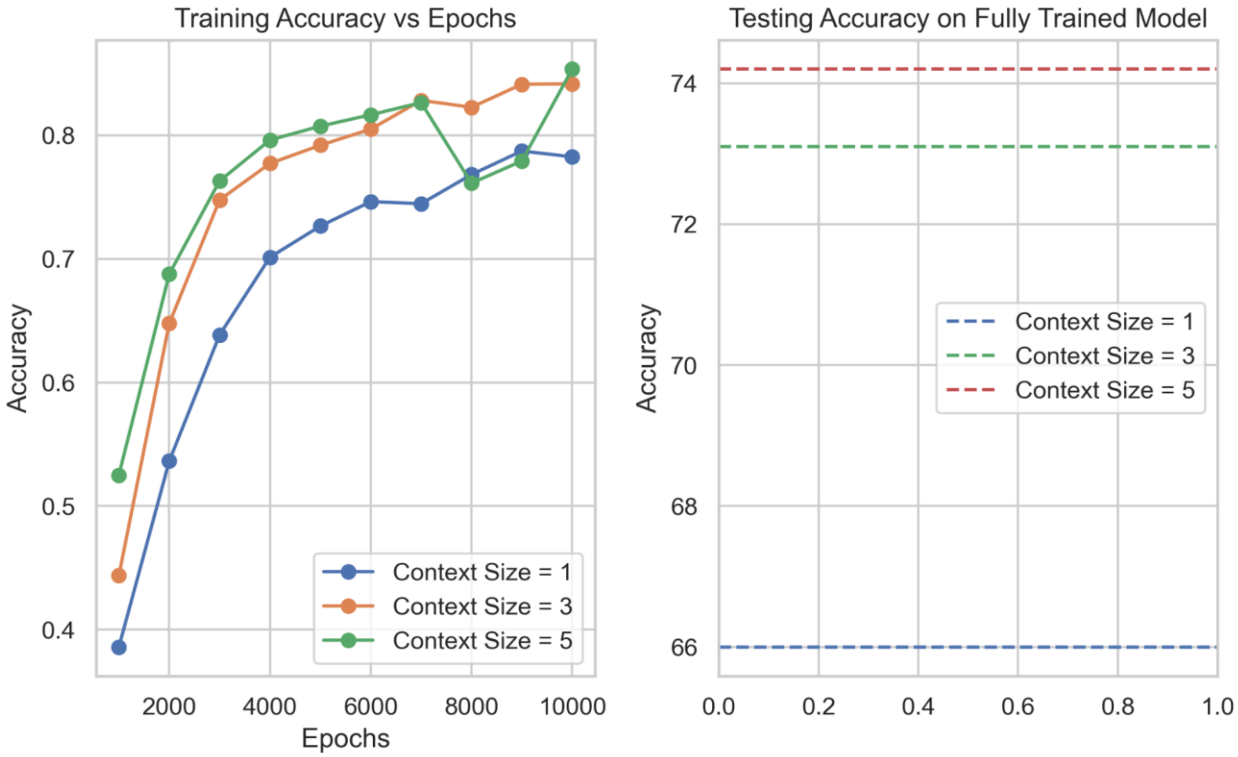
Train

Validation / Test

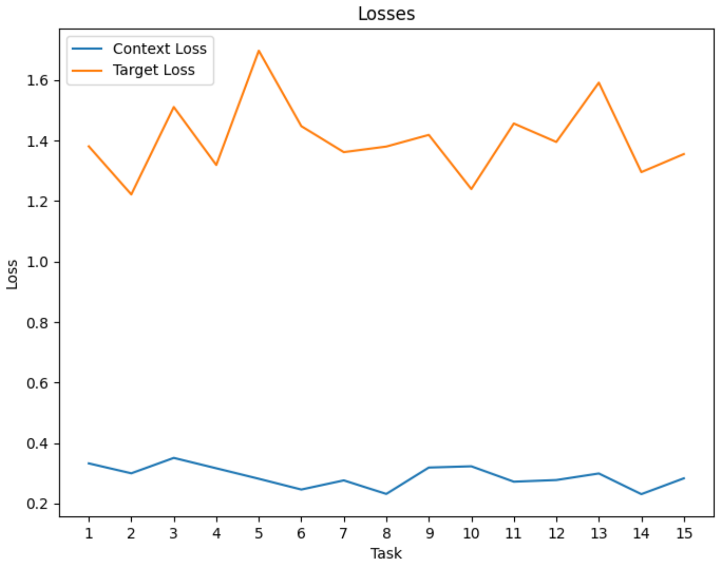
Results on Celeb10 (Context = 1,Target = 10, Way = 5)

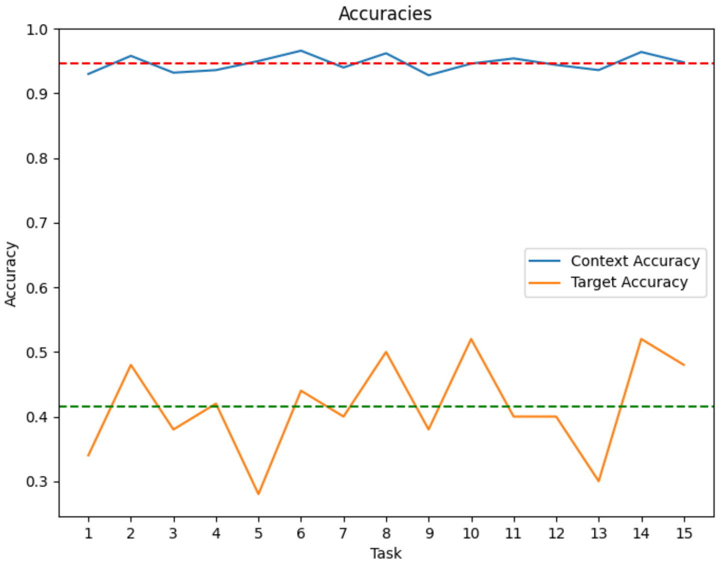


For different Context Sizes

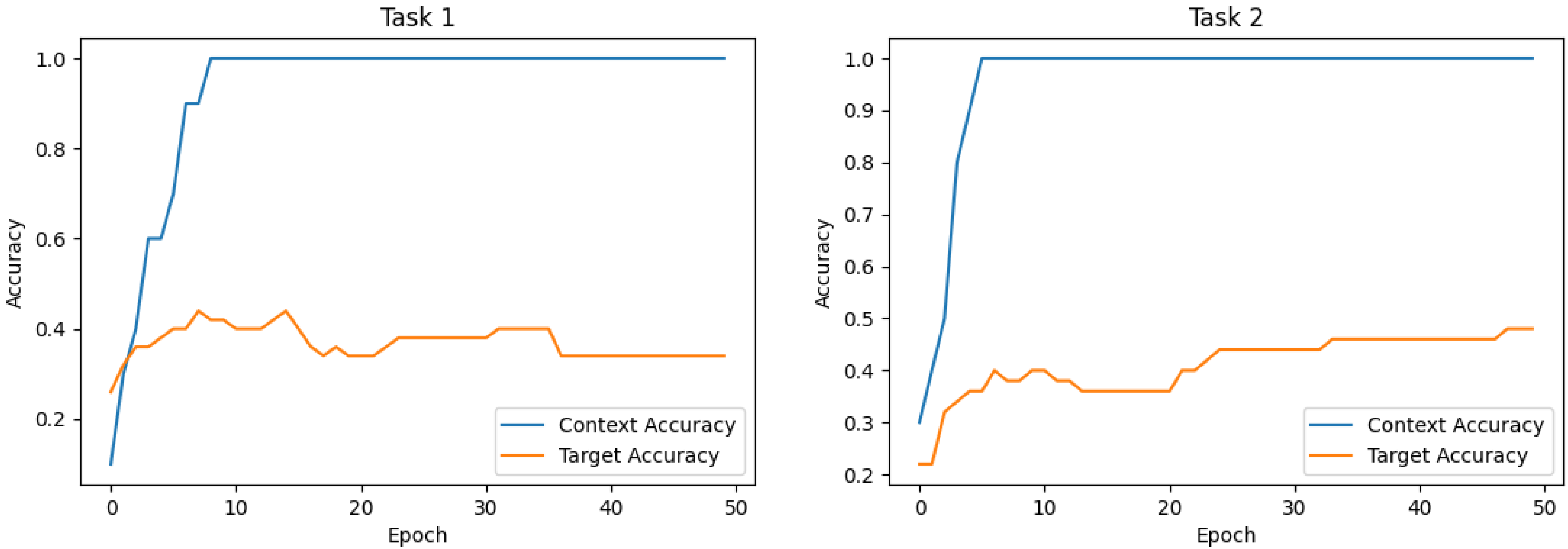


Celeb10- Fine Tuning (Context = 2, Target = 10, Way = 5)





Celeb10- Fine Tuning (Context = 2, Target = 10, Way = 5)



Try CNAPs on Celeb150

Splits and the process of training:

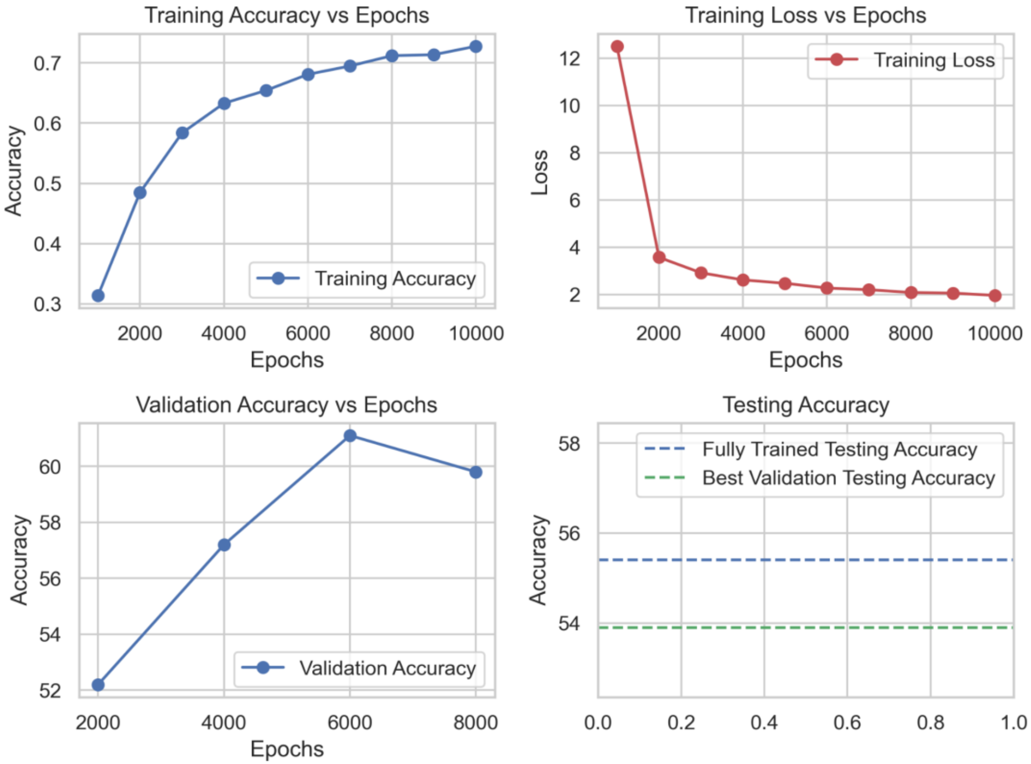
- CelebFace150
  - Train: 30 images for each celebrity. A total of 100 celebrities.
  - Test / Validation: 30 images for each celebrity. A total of 25 completely different celebrities both in validation and test set.

Why?

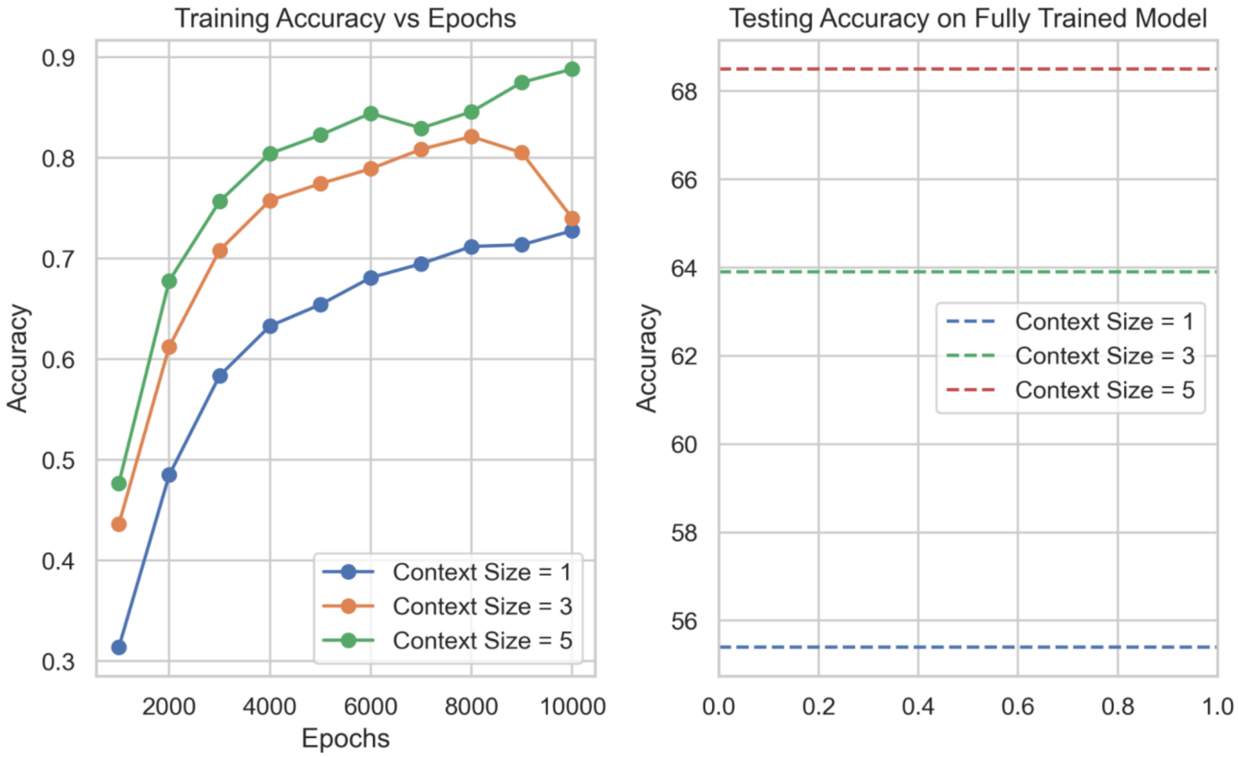
Train

Validation / Test

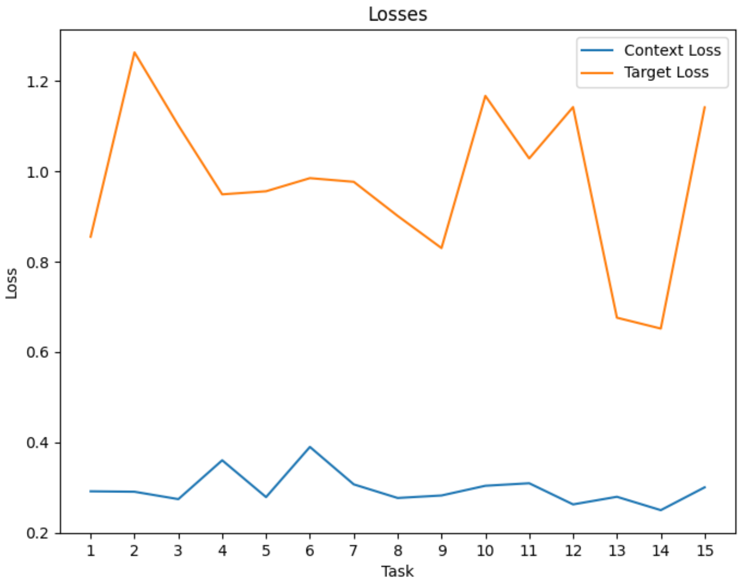
Results on CelebFace150 (Context = 1, Target = 10, Way = 5)

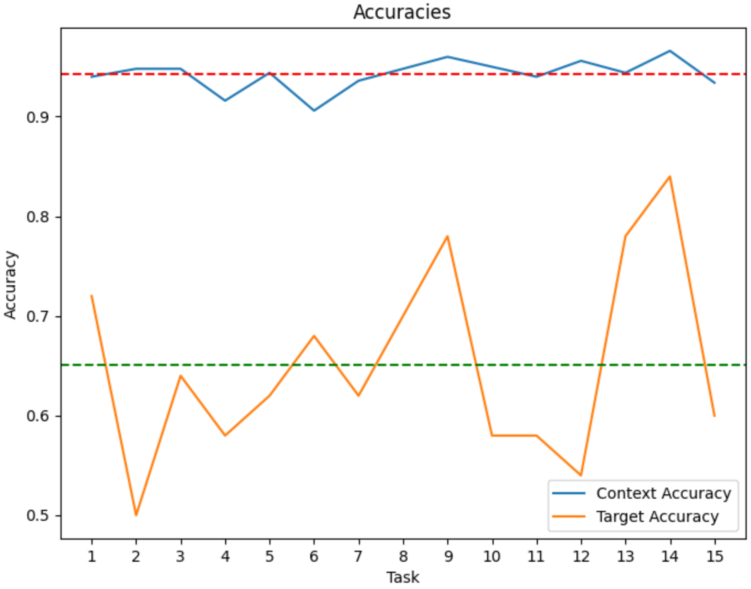


For different Context Sizes

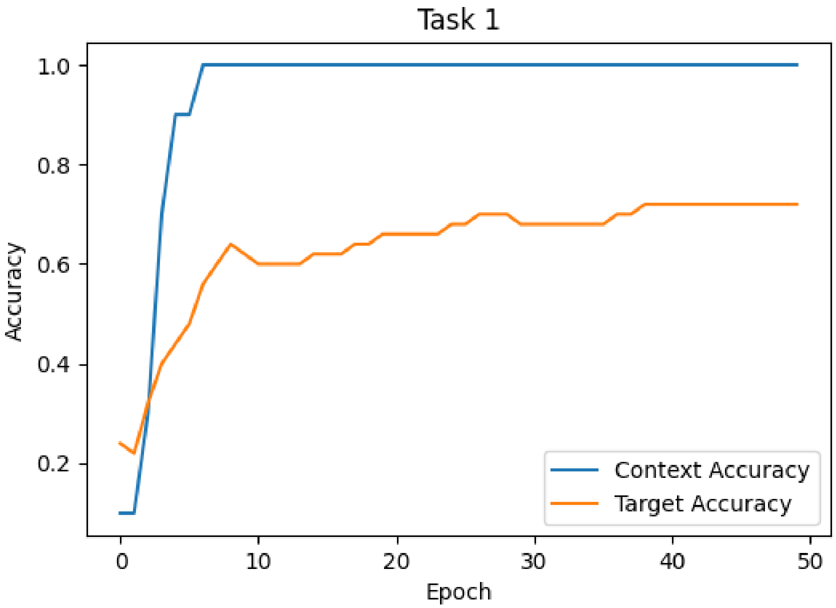


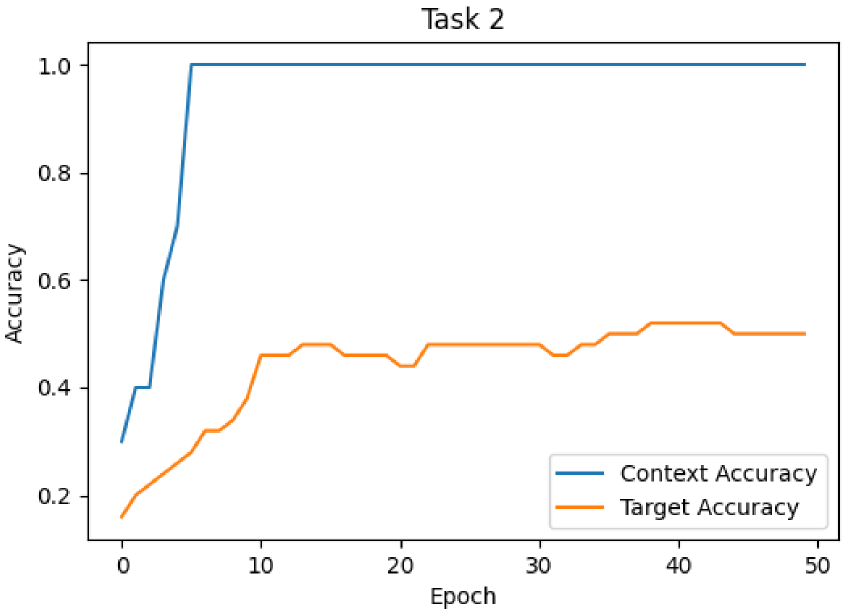
CelebFace150- Fine Tuning (Context = 2, Target = 10, Way = 5)





CelebFace150- Fine Tuning (Context = 2, Target = 10, Way = 5)


Task 1


Task 2

Try CNAPs on CelebCars

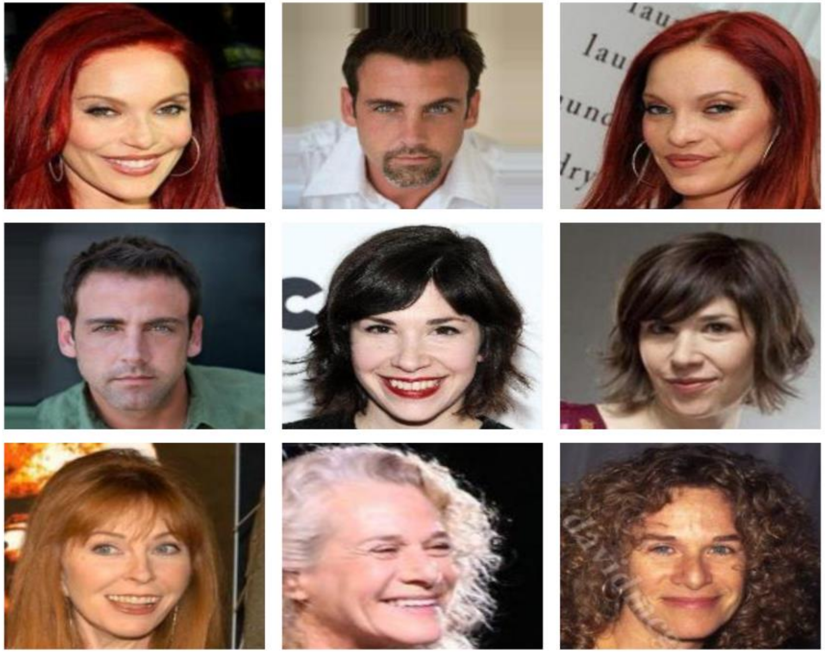
Splits and the process of training:

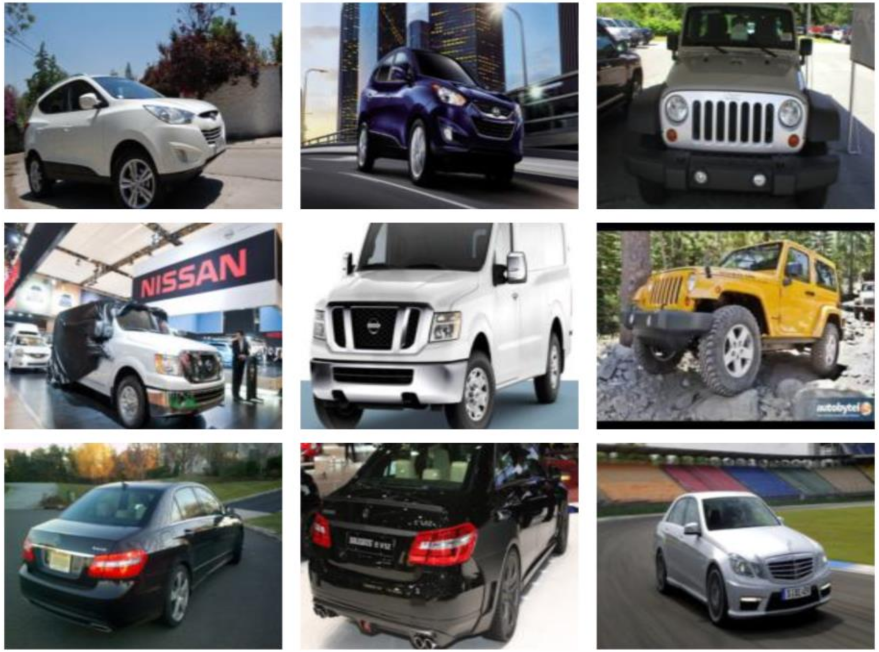
- CelebCars
  - Train: 30 images for each celebrity. A total of 100 celebrities.
  - Test / Validation: 30 images for each car model. A total of 25 car models both in validation and test set.

Why?

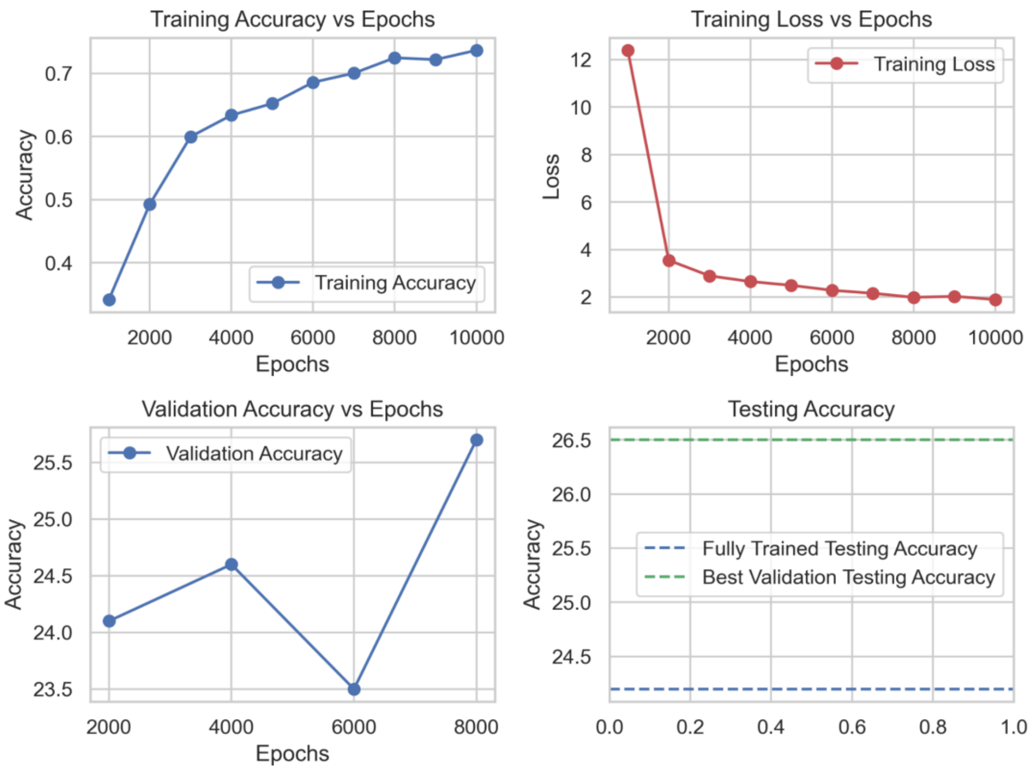
Train

Validation / Test

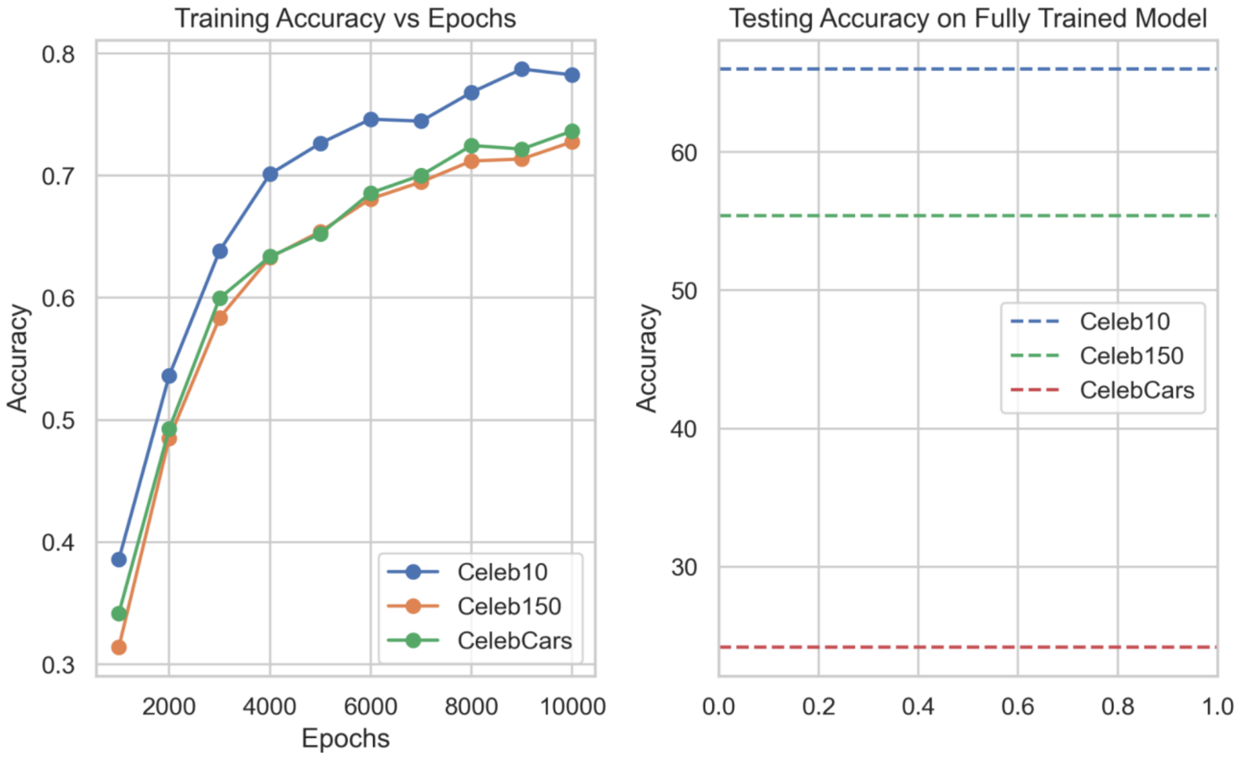




Results on CelebCars (Context = 1, Target = 10, Way = 5)



Comparison Across Datasets



Results and Observations

- CNAPs performs better on Celeb10 compared to Celeb150 as recognizing similarity in faces is tougher than in specific feature like Big Nose.
- CNAPs performs better on increasing context sizes as it will adapt better.
- CNAPs performs as good as random for the case of CelebCars because we are giving completely out of distribution data (Cars) in validation and test.

Achievements and Future Scope

- Initially CNAPs codebase was working for meta-dataset, MNIST, Cifar10, and Cifar100. We have added a new pipeline through which you can insert any dataset (even your own just like us) and test CNAPs with just a few commands.
- We can work on it further to make it modular and raise a PR.
- We thought of a new way to look at CelebA dataset and converting it into a multi-task classification setting.
- Conceptualising adaptive and autoregressive networks for regression problems.

Thank you!