Structured Meta-Learning for Cross-Domain Few-Shot Classification

Nicola De Angeli
nicola.deangeli@mail.polimi.it
Computer Science and Engineering Track
Advisors: Matteo Matteucci, Marco Ciccone

POLITECNICO MILANO 1863

HONOURS PROGRAMME HP-SR in Information Technology

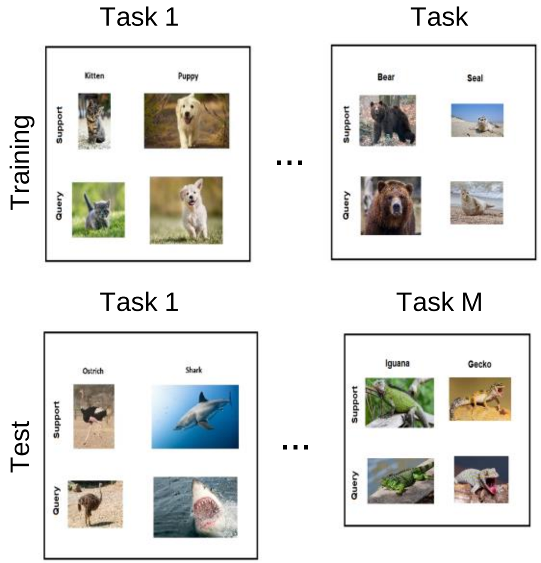
Few-Shot Learning (FSL)

Many-Shot

Few-Shot

- A *task* is a problem that a machine learning model solves by adapting to some data
- In *FSL*, **few training samples** from the task are available
- Incredibly difficult for **data-hungry models**
- Humans can easily deal with the problem thanks to **experience**

From “Learning” to “Learning to Learn”



Dataset

- A single task
- Objective: generalize to new examples within a task

Meta-Dataset (our experience!)

- Many FSL tasks
- Objective: generalize to new tasks

Meta-Learning

- The goal of Meta-Learning is *“learning to learn”*
  - Extracts and reuses knowledge by training on a **distribution of tasks** $p(\mathcal{T})$
- Characterized by the presence of a **learner** and a **meta-learner**
  - The learner follows an adaptation procedure to adapt to the task at hand
  - The meta-learner adjusts the adaptation procedure to improve performance over many tasks

MAML

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7: **end for**
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: **end while**

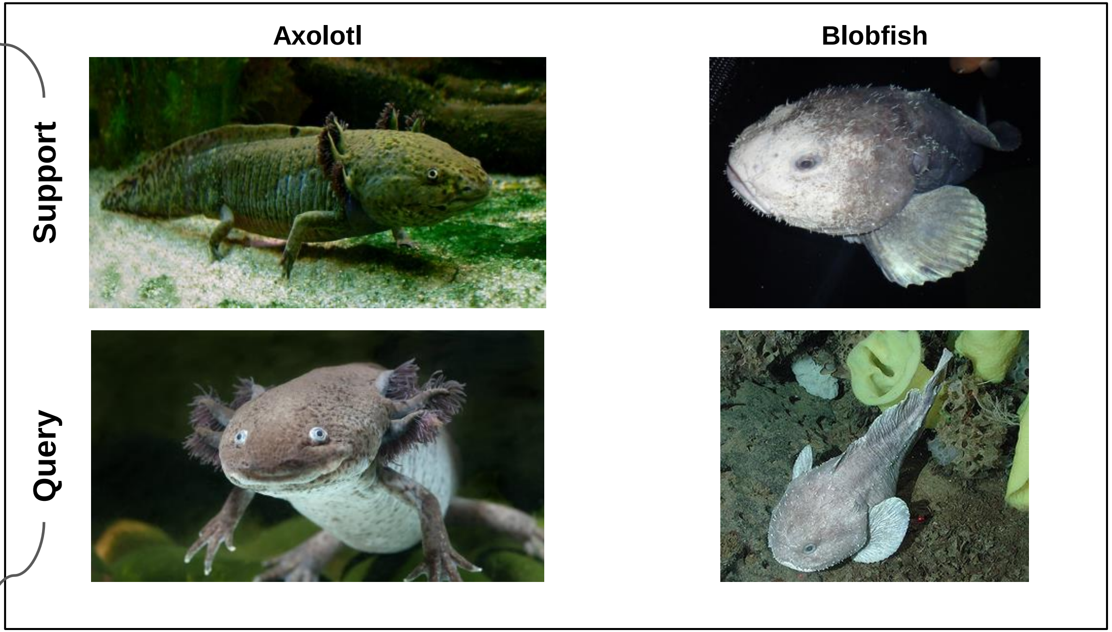
- Applies to a generic learner with parameters θ
- Only requirement for the learner is to be trainable via gradient descent
- Finds a single parameter initialization for the learner
- The algorithm features a nested loop
  - Inner loop: the learner
  - Outer loop: the meta-learner

[Finn et al. 2018]

One-Shot Classification

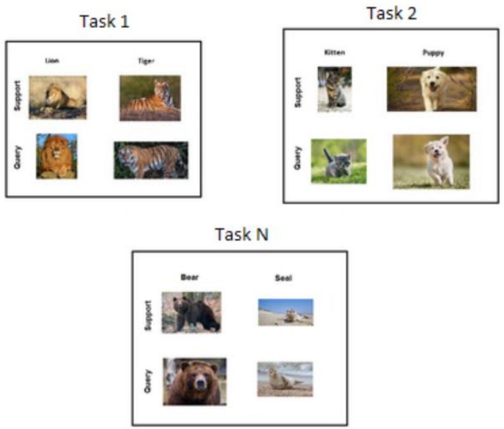
Images in the support set are labeled and used to adapt to the task

New Task



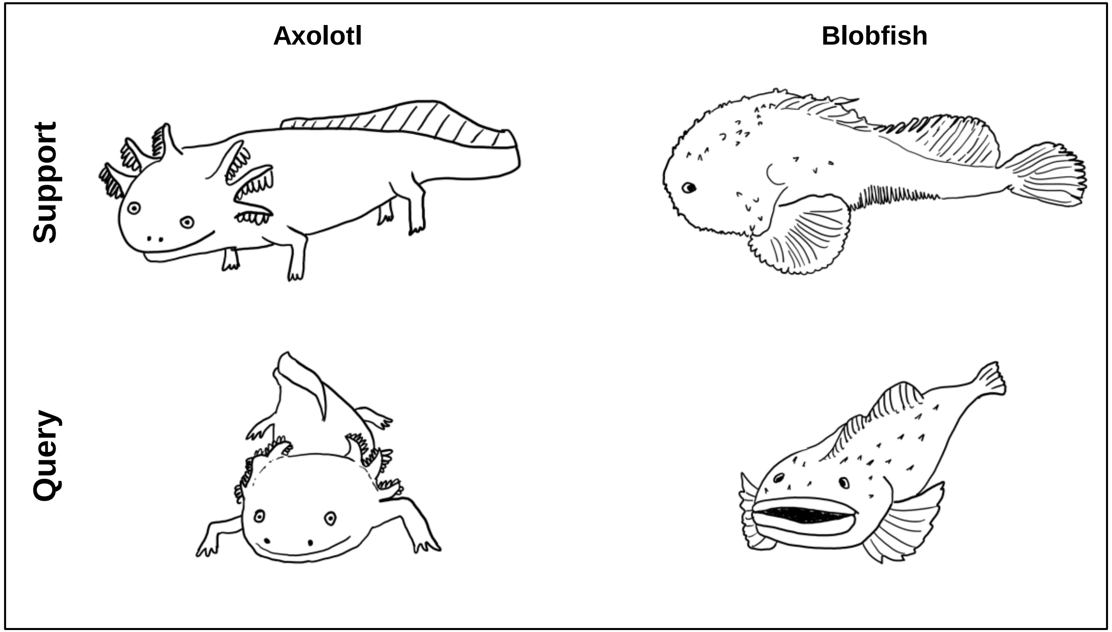
Images in the query set are unlabeled and the learner predicts their label

Meta-Training

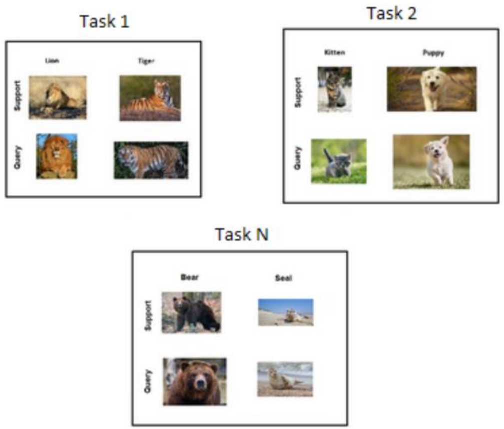


One-Shot Classification

New Task: Sketch

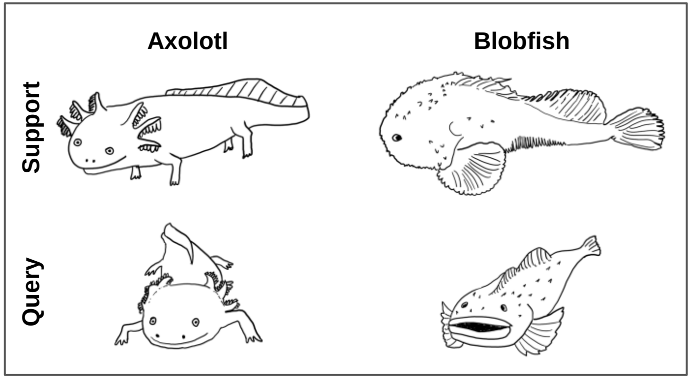


Meta-Training: Natural

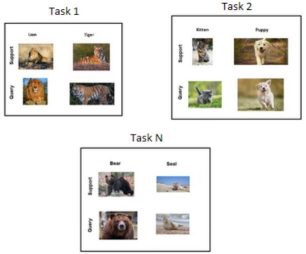


Cross-Domain Few-Shot Learning (CDFSL)

New Task: Sketch

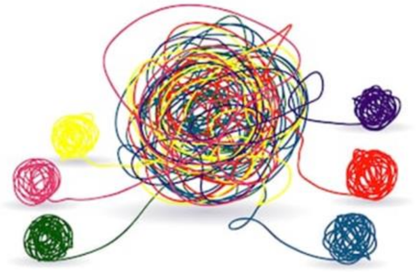


Meta-Training: Natural

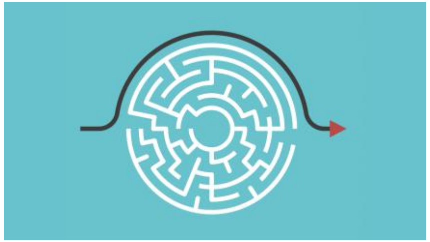


- The above is an example of *CDFSL*, easy for humans and **hard for meta-learning**
- CDFSL deals with data coming from *heterogeneous domains*
  - Examples: different source cameras or different light conditions
- In our case:
  - the domain does not change same within a task
  - test tasks feature data from unseen domains
- We want to refine meta-learning techniques to deal with CDFSL

Disentangled and Simple Embedding



vs



"[W]e would like our representations to disentangle the factors of variation"
Bengio et al. 2013

- *Disentanglement* captures different independent properties of data in different units
- Many works argue that a **disentangled embedding** could prove useful in general
- Others believe training a **simple embedding** with few regularizations is enough
- We believe disentanglement to be the better solution for CDFSL:
  - Previous works have leveraged disentanglement to boost performance in standard FSL
  - Capturing domain information may be useful

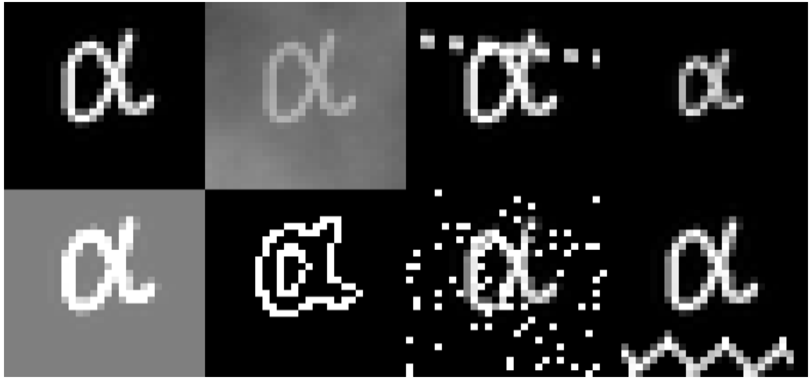
[Bengio et al. 2013, Van Steenkiste et al. 2019, Chen et al. 2019]

Our Contribution

Benchmark: Corrupted-Omniglot

Training Domains

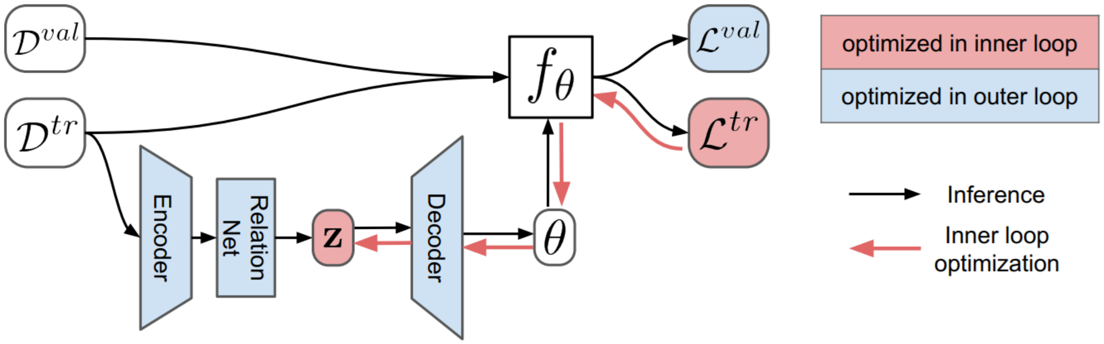
Test Domains



- We obtain *Corrupted-Omniglot* by augmenting the dataset of characters *Omniglot* with image corruptions
- **Each corruption is a domain**
- 16 domains are split into *training* and *test domains*
- *20-way, 1-shot tasks* (20 classes per task, 1 example per class)

[Lake et al. 2019, Mu et al. 2019]

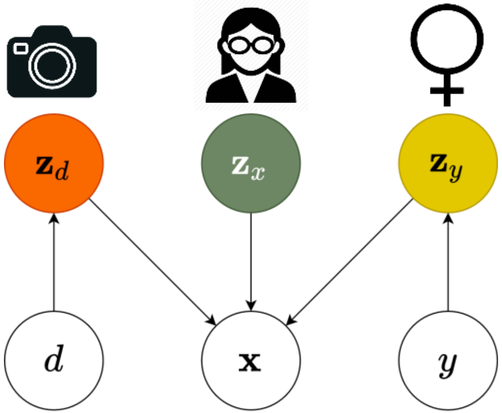
Meta-Learning Model: LEO



- Features inner task-specific loop and outer across-task loop, **like MAML**
- Encodes the task into a latent code
- Decodes the parameters of the learner from the latent code
- Performs gradient descent in the latent code

[Rusu et al. 2019]

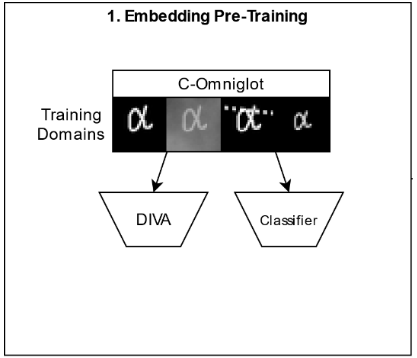
Disentanglement Model: DIVA



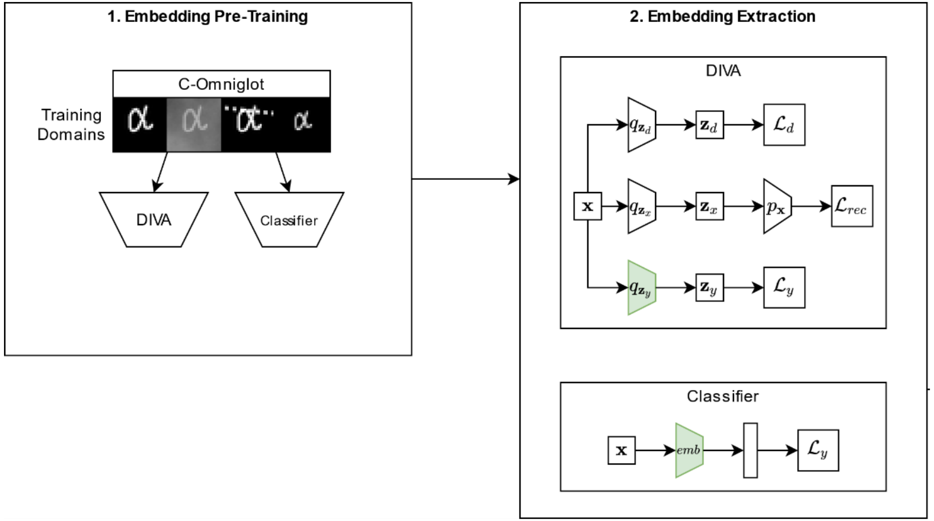
- Encodes the input in a disentangled representation
- The latent space is divided in three subspaces
    - Domain, residual, and class information
- Latent space in continuous and stochastic
    - Can potentially generalize to unseen domains

[Ilse et al. 2018]

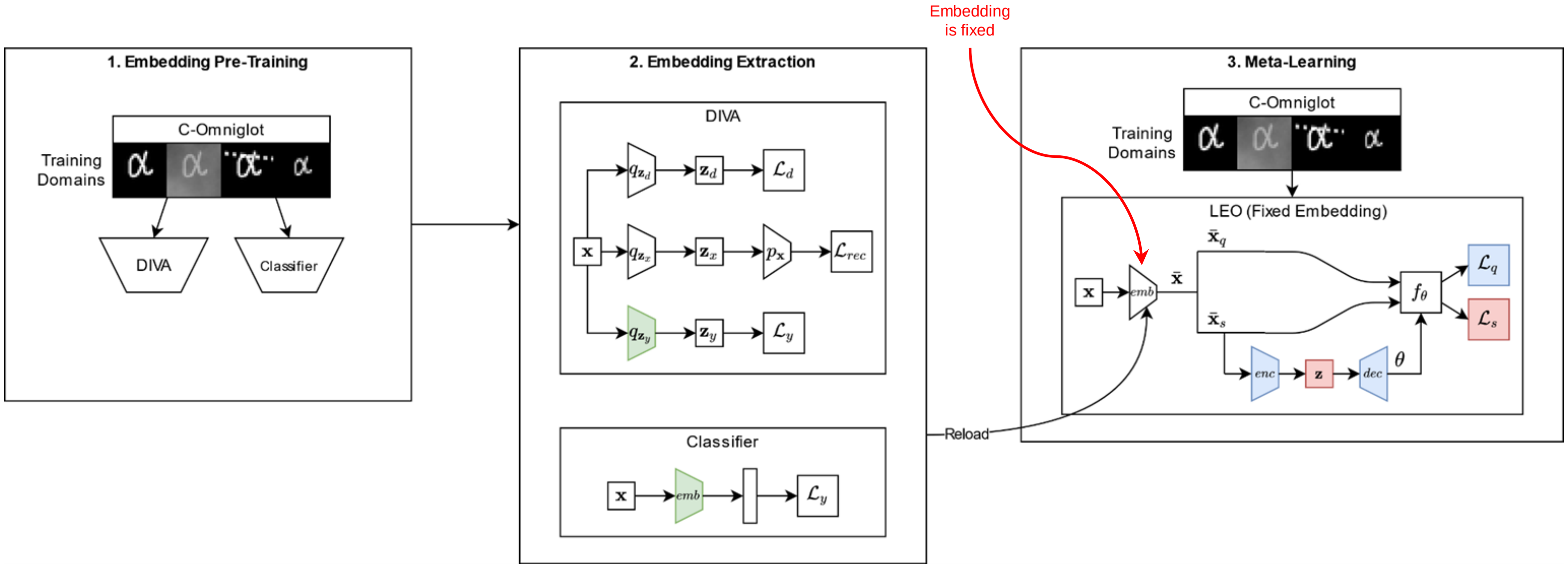
Pipeline of the Models



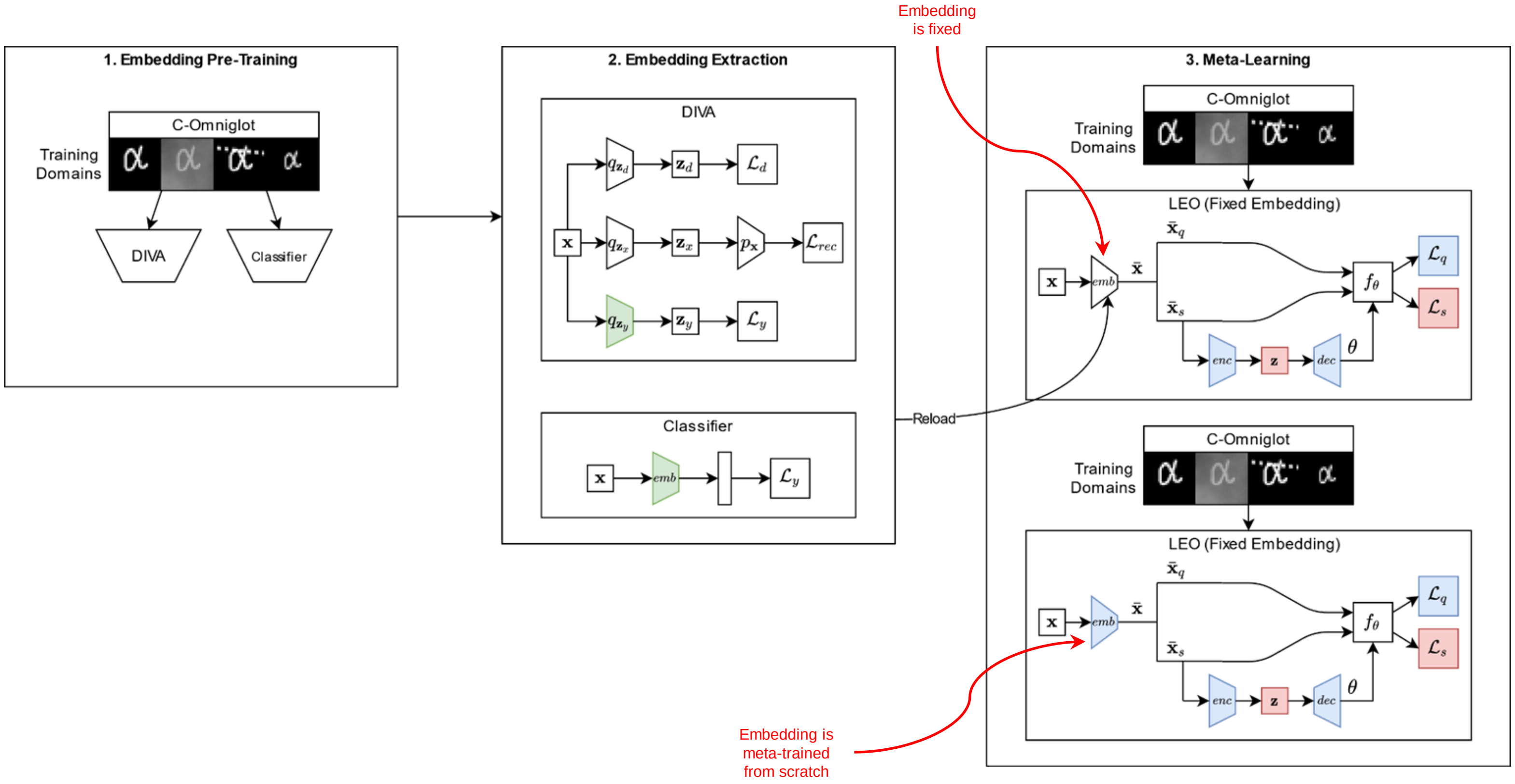
Pipeline of the Models



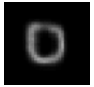
Pipeline of the Models



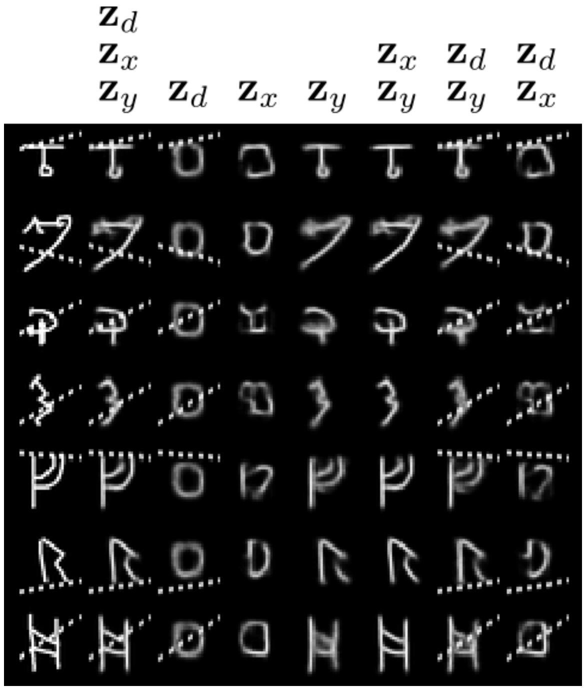
Pipeline of the Models



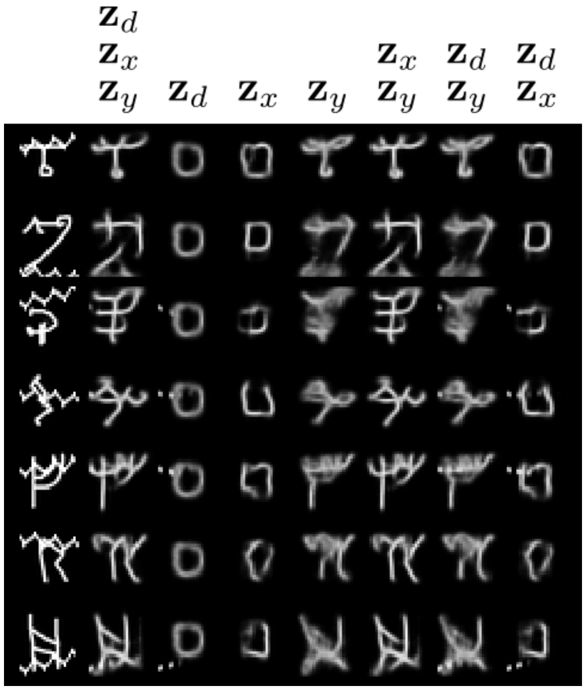
Pre-Trained DIVA Reconstructions



Very general character shows up when not considering z_y during reconstruction

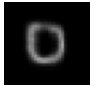


Training Domain
Dotted Line

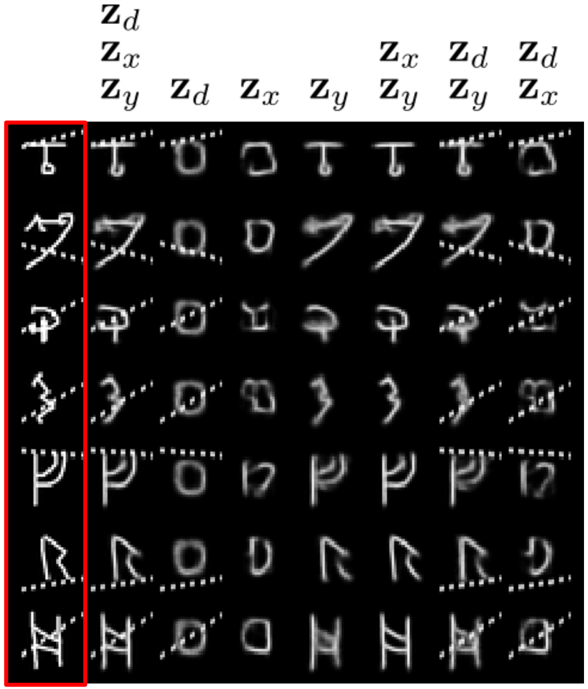


Test Domain
Zigzag

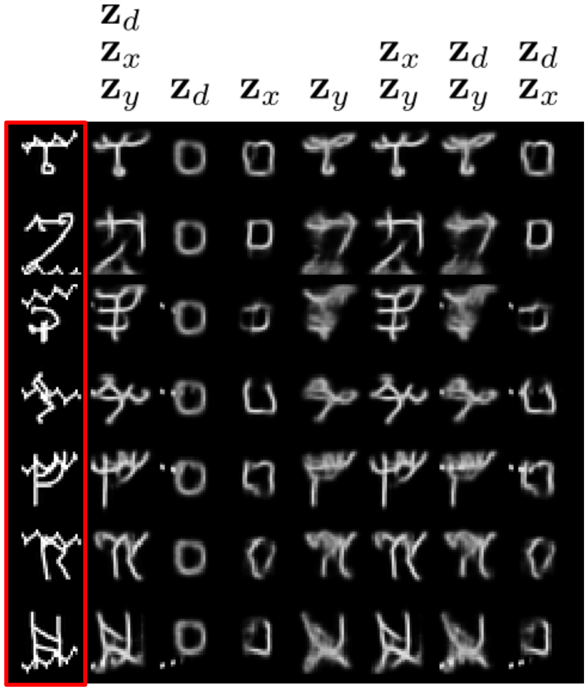
Pre-Trained DIVA Reconstructions



Very general character shows up when not considering z_y during reconstruction

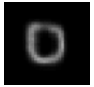


Training Domain
Dotted Line

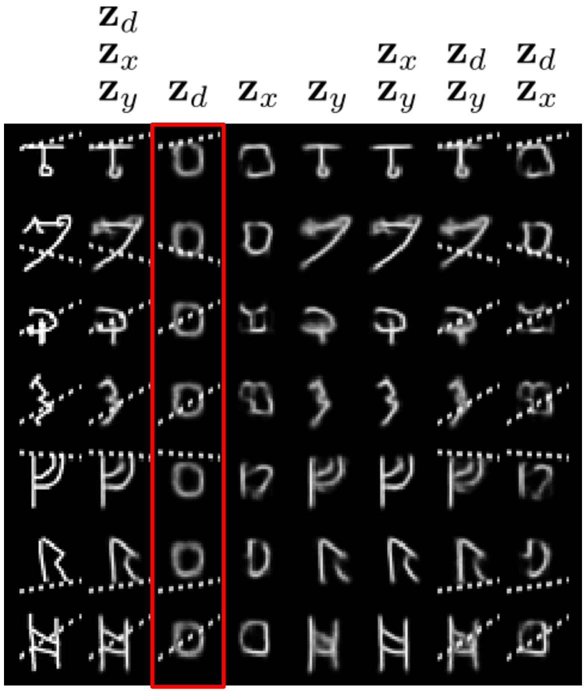


Test Domain
Zigzag

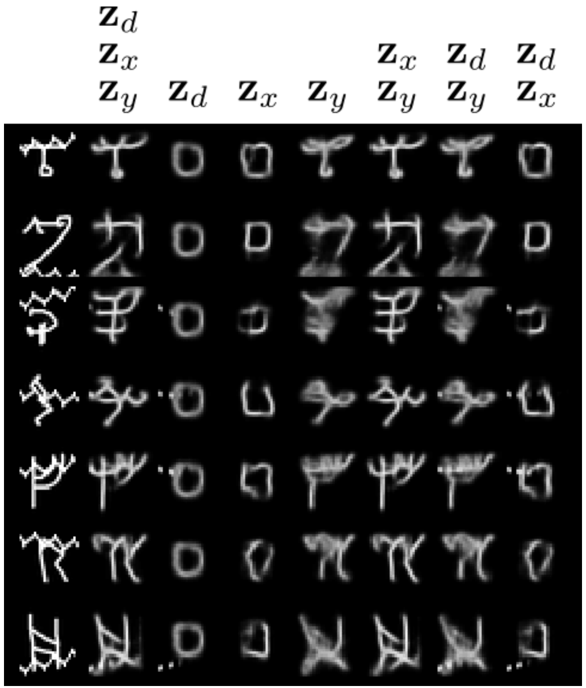
Pre-Trained DIVA Reconstructions



Very general character shows up when not considering z_y during reconstruction

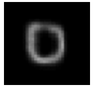


Training Domain
Dotted Line

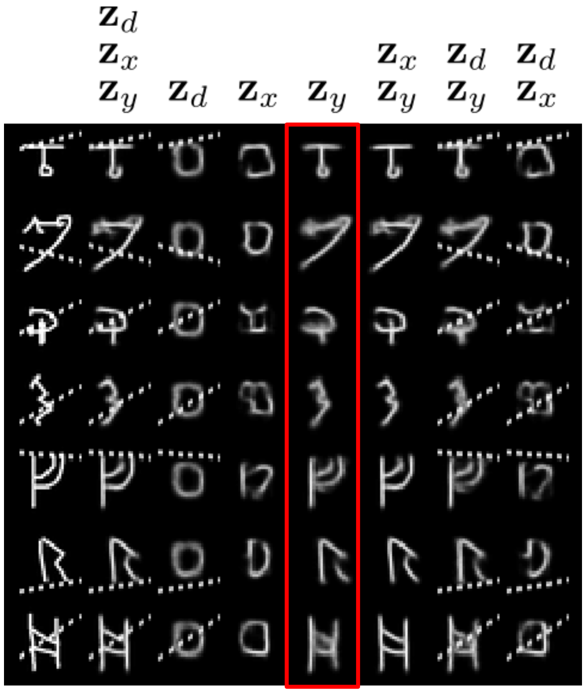


Test Domain
Zigzag

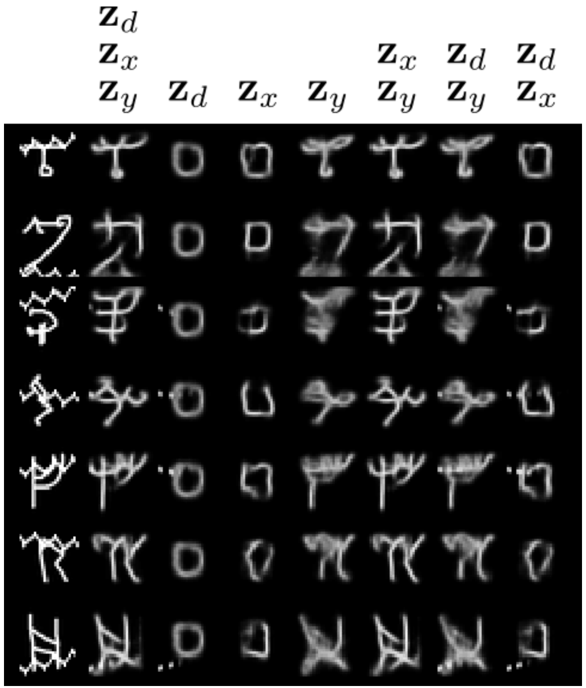
Pre-Trained DIVA Reconstructions



Very general character shows up when not considering z_y during reconstruction

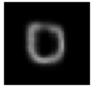


Training Domain
Dotted Line

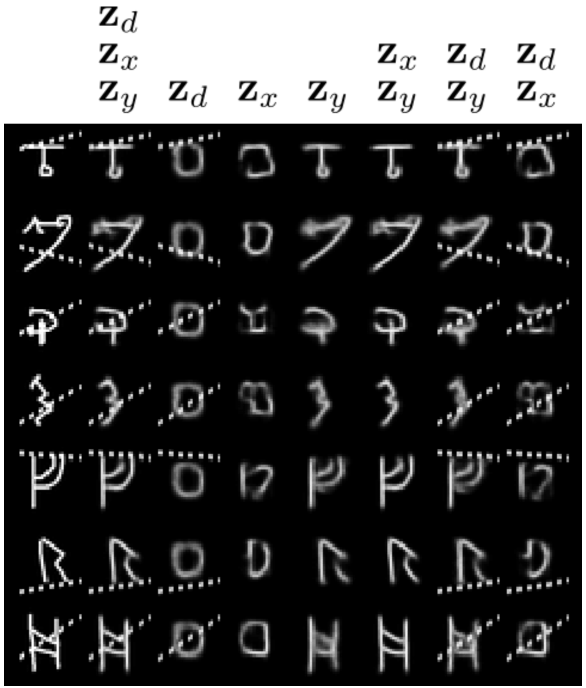


Test Domain
Zigzag

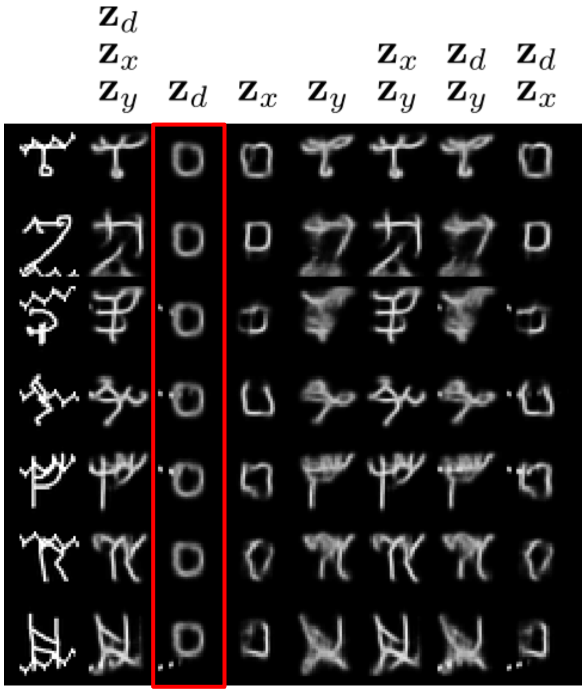
Pre-Trained DIVA Reconstructions



Very general character shows up when not considering z_y during reconstruction

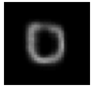


Training Domain
Dotted Line

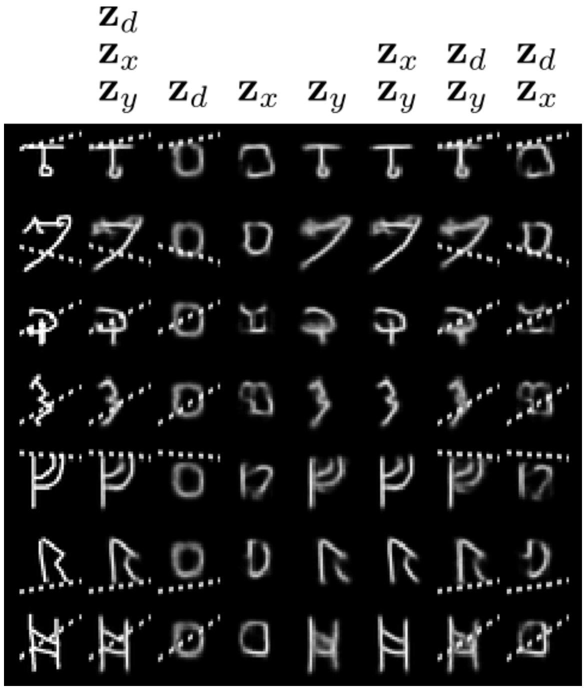


Test Domain
Zigzag

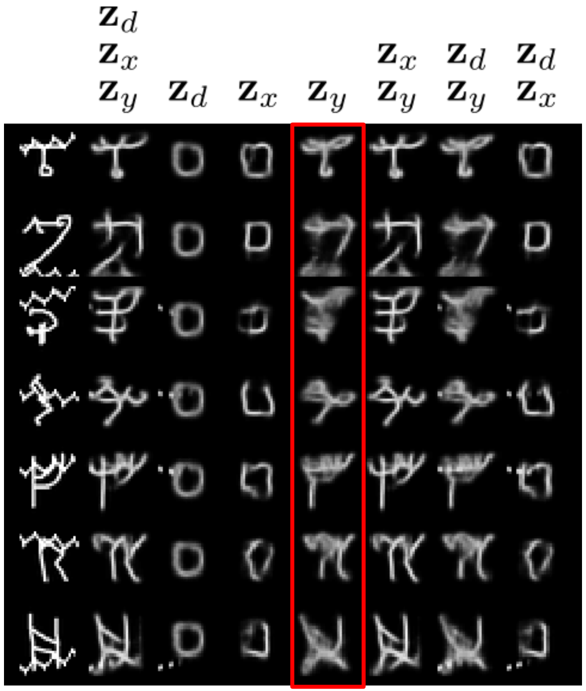
Pre-Trained DIVA Reconstructions



Very general character shows up when not considering z_y during reconstruction



Training Domain
Dotted Line



Test Domain
Zigzag

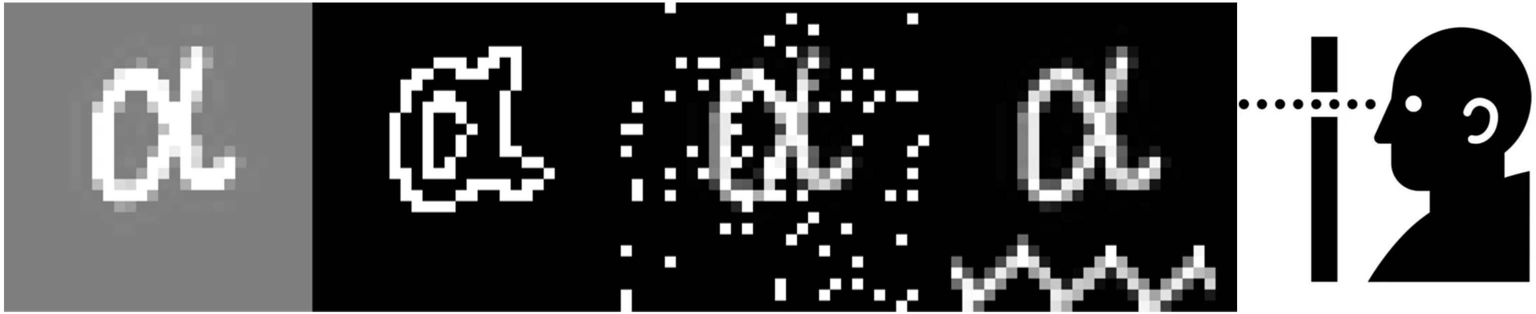
Meta-Learning Results

	Class Accuracy	
Embedding	Training Domains	Test Domains
DIVA	0.91	0.72
Classifier	0.88	0.64
Meta-Trained	**0.95**	**0.76**

- **DIVA is much better than Classifier** in test domains
  - Promising!
- **Meta-trained embedding outperforms both** pre-trained embeddings
  - Not so promising...
  - Meta-training may play an important role in generalizing to unseen domains

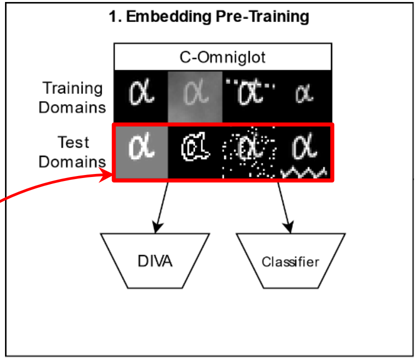
Introducing Oracle Models

Test Domains



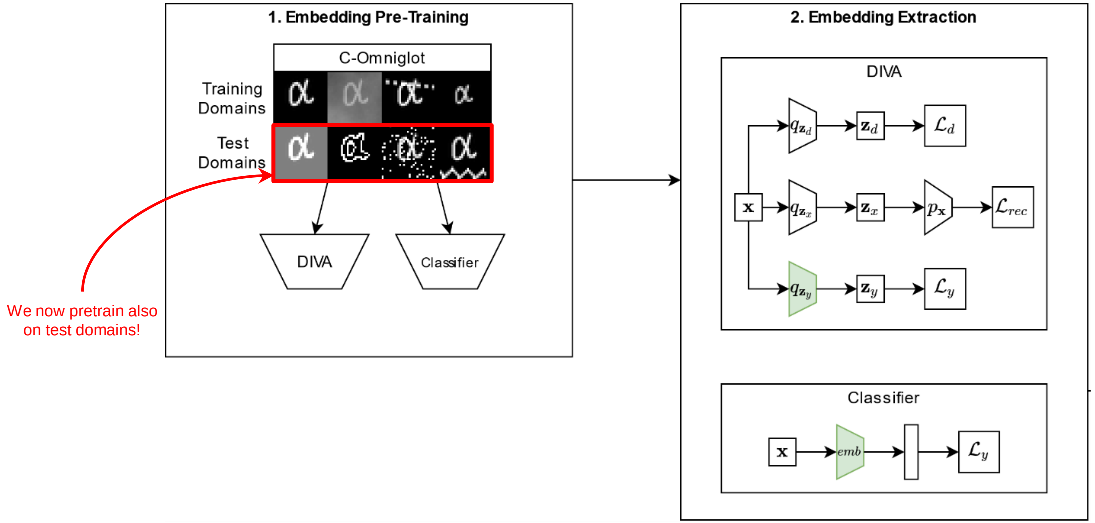
- What if we had a disentangling model that can generalize to unseen domains?
- We can position ourselves in this “what if” scenario by leveraging *oracle models*
    - Oracle models are trained on images from **both training and test domains**

Pipeline of the Oracle Models

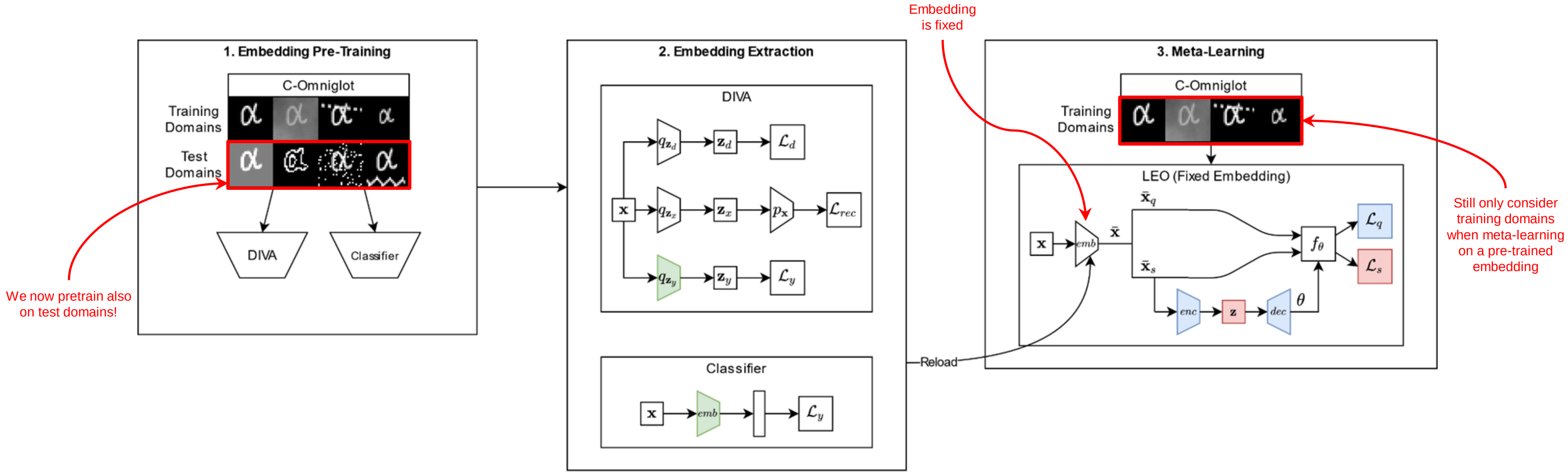


We now pretrain also on test domains!

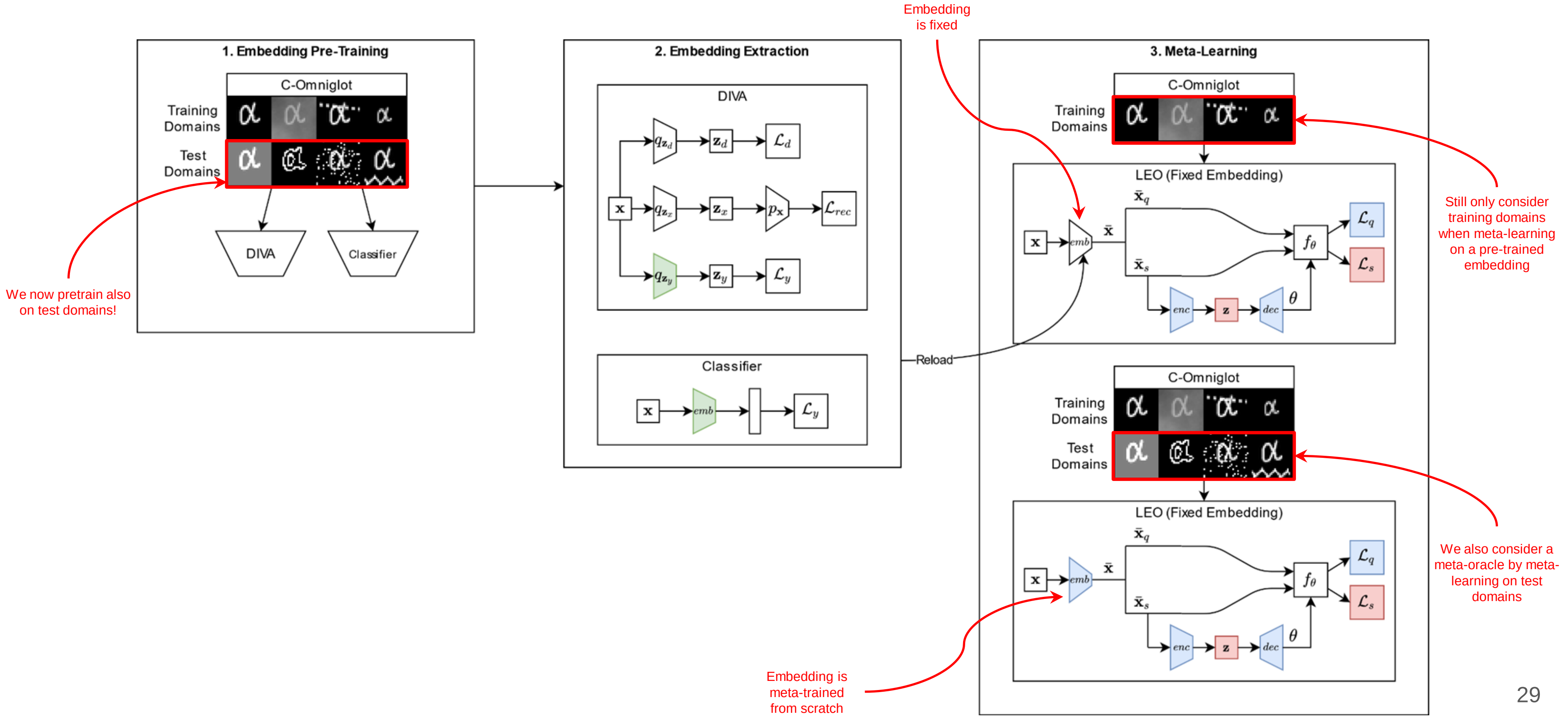
Pipeline of the Oracle Models



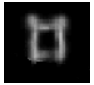
Pipeline of the Oracle Models



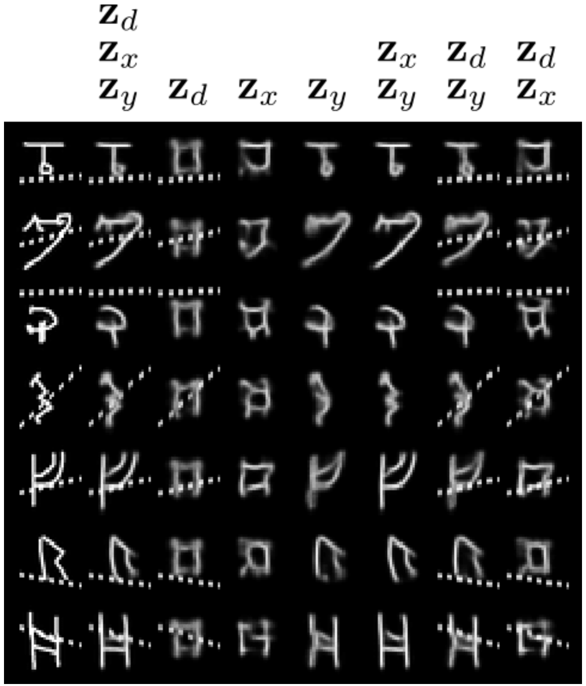
Pipeline of the Oracle Models



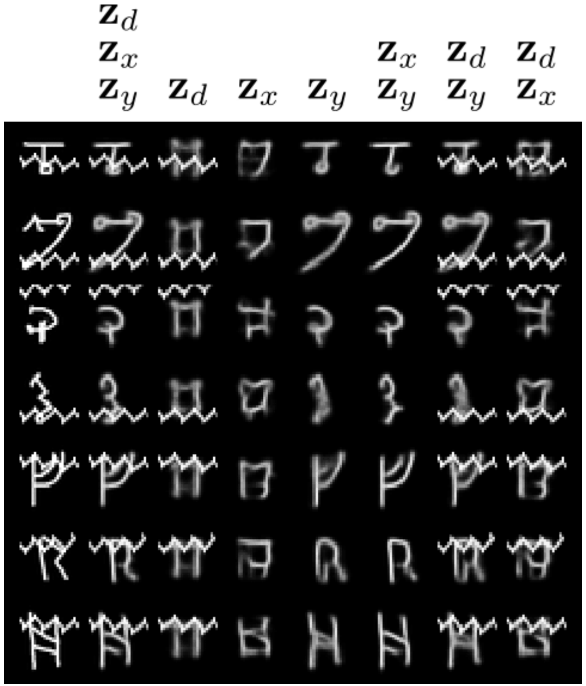
Pre-Trained Oracle DIVA Reconstructions



Very general character shows up when not considering z_y during reconstruction

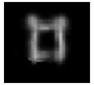


Training Domain
Dotted Line

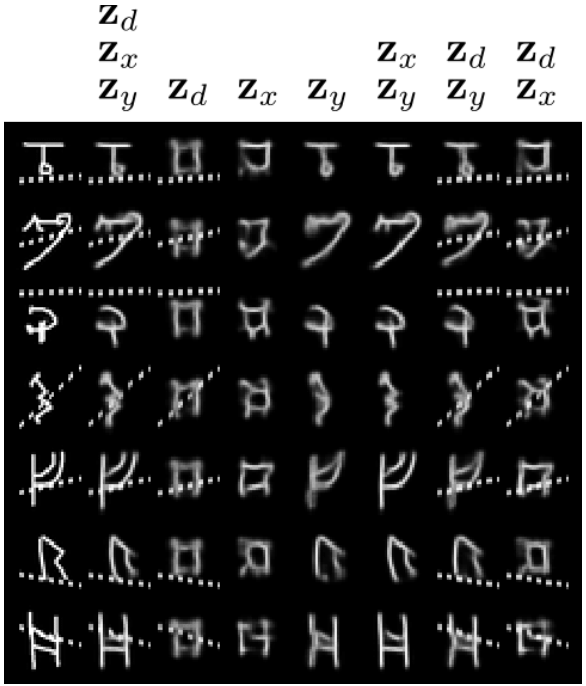


Test Domain
Zigzag

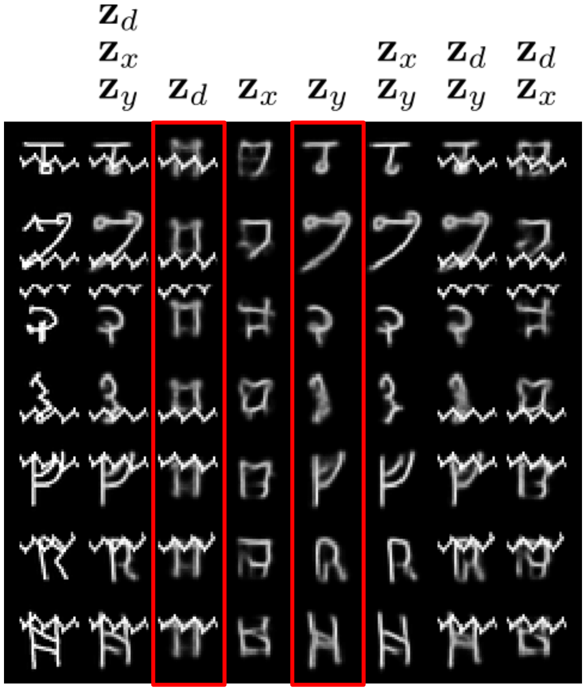
Pre-Trained Oracle DIVA Reconstructions



Very general character shows up when not considering z_y during reconstruction



Training Domain
Dotted Line



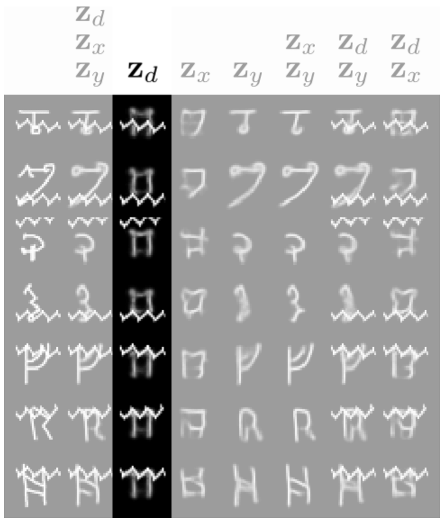
Test Domain
Zigzag

Meta-Learning Oracle Results

	Class Accuracy	
Embedding	Training Domains	Test Domains
Oracle DIVA	0.91	0.87
Oracle Classifier	0.94	0.90
Meta-Oracle	0.94	0.93

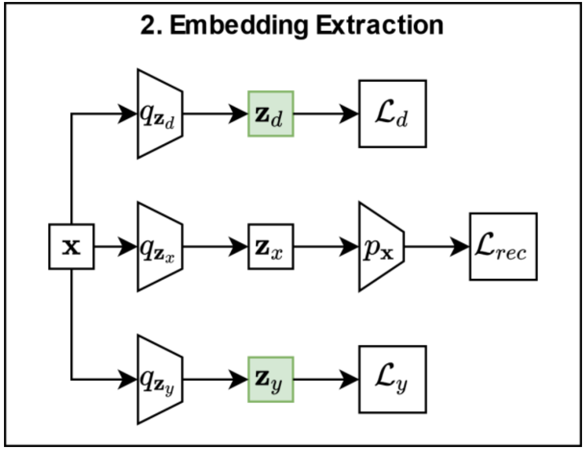
- Oracle Classifier **outperforms** Oracle DIVA!
  - Maybe disentanglement is not useful in our problem after all...
- Meta-Oracle is the best one, by far
  - Yet again hinting at importance of meta-training

Leveraging Oracle Domain Information



- We are not yet done with disentanglement
- Oracle DIVA provides us with high-quality domain information
- Is there any use for domain information in our problem?

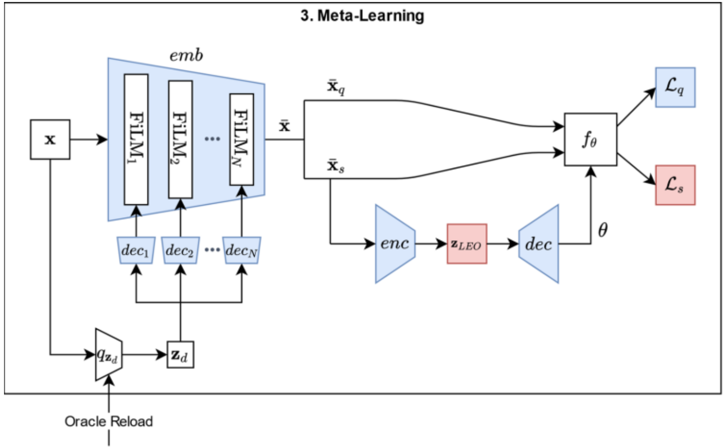
Including Oracle Domain in the Embedding



	Class Accuracy	
Embedding	Training Domains	Test Domains
Oracle DIVA zd+zy	0.90	0.64
Oracle DIVA zy	0.88	0.85

- We consider both class and domain information when reloading Oracle DIVA's embedding
    - LEO may improve the quality of adaptation based on domain information
- **The model overfits on training domains**

Oracle Domain-Based Modulations



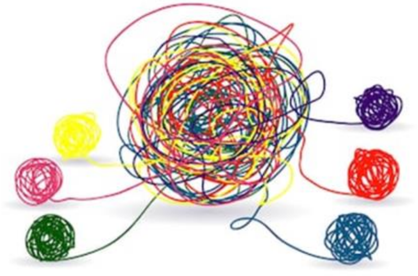
	Class Accuracy	
Embedding	Training Domains	Test Domains
FiLM	0.95	0.70
No FiLM	0.95	0.76

- We leverage modulations between the layers of the embedder
  - The parameters of the modulation are inferred based on domain information
  - Modulations may help in filtering domain information in the embedding, boosting performance
- **Again, overfitting on training domains**

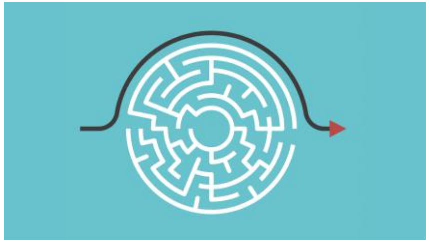
[Ilse et al. 2018]

Takeaways

- **Main takeaway: disentanglement does not seem to do much**
    - Domain information is hard to leverage
- Many works in the literature claim disentanglement is useful…
    - …**not enough experiments?**
    - …**biased literature?**
- Other observations:
    - Meta-training a simple embedding is beneficial as opposed to pre-training
    - Filtering domain information boosts performance



vs



Future Work

- Verify our results on disentanglement with **further experimentation**
- If our results are verified:
    - Another interesting direction is to pursue *domain agnostic* representations
    - The **adaptation process** can be refined to learn how to filter unseen domain information from few samples

Thank you!

Any questions?