

# Variational Continual Learning in Deep Discriminative Models

Gwangbin Bae (gb585), Riccardo Barbano (rb876), Justin Bunker (jb2200)



## Continual Learning

- Data may arrive in non i.i.d. way
- Tasks may change and/or new tasks may emerge

**Plasticity** vs. **Stability**  
(catastrophic forgetting) (inability to adapt)

## Bayesian Inference

BI provides a framework for continual learning

$$p(\theta | \mathcal{D}_{1:T}) \propto p(\theta) \prod_{t=1}^T p(\mathcal{D}_t | \theta) \\ \propto p(\theta | \mathcal{D}_{1:T-1}) p(\mathcal{D}_T | \theta)$$

**Posterior Update:** normalize (current posterior  $\times$  likelihood of newly observed data)

## Projection Operation

PO finds a tractable normalized distribution that approximates the intractable un-normalized posterior

$$p(\theta | \mathcal{D}_{1:T}) \approx q_T(\theta) = \text{proj}(q_{T-1}(\theta) p(\mathcal{D}_T | \theta))$$

- Recursive relation between posteriors recovered
- $q_0(\theta) = p(\theta)$  (initialized with prior distribution)

## Variational Continual Learning<sup>[1][2]</sup>

VCL uses KL divergence minimization for projection

$$q_t(\theta) = \underset{q \in Q}{\text{argmin}} KL \left( q(\theta) \parallel \frac{1}{Z_t} q_{t-1}(\theta) p(\mathcal{D}_t | \theta) \right)$$

Q: set of available posterior functions

$q_t(\theta)$ : Gaussian mean-field approximate posterior

## Episodic Memory Enhancement

### Coreset

- Small subset of data from each task
- Excluded in training & used before prediction
- Avoids catastrophic forgetting

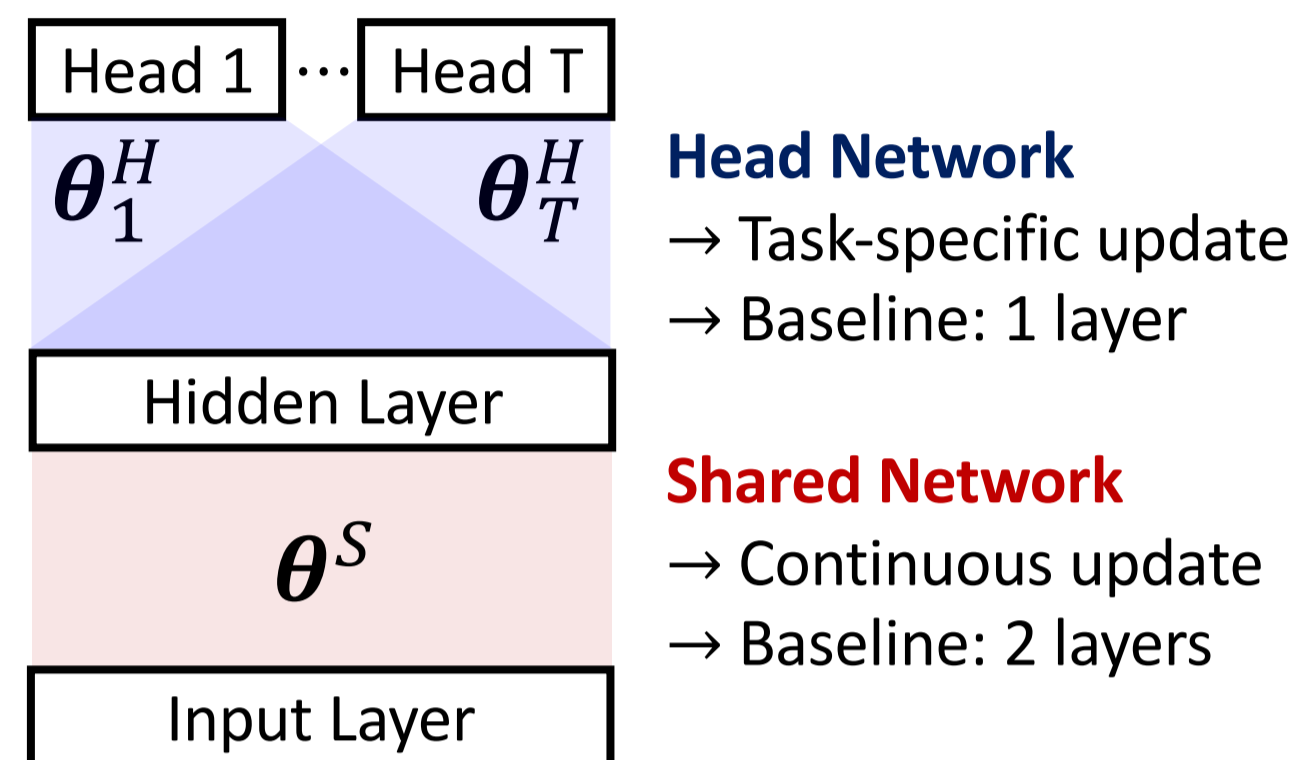
### Coreset Heuristics

- Random selection & K-center methods

## VCL in Deep Discriminative Models

### Multi-head Network

- Standard architecture for multi-task learning



**Posterior Update:** finds  $q_t(\theta)$  that maximizes the negative variational online free energy

$$\sum_{n=1}^{N_t} \mathbb{E}_{\theta \sim q_t(\theta)} [\log p(y_t^{(n)} | \theta, \mathbf{x}_t^{(n)})] - KL(q_t \parallel q_{t-1})$$

**Expected Likelihood** adapts to the new task

- Intractable & approximated using Monte Carlo sampling and local reparameterization trick<sup>[3]</sup>

**KL Divergence** avoids forgetting previous tasks

- Tractable &  $q_0(\theta)$  initialized with small variance

## Algorithm<sup>[1][4]</sup>

```

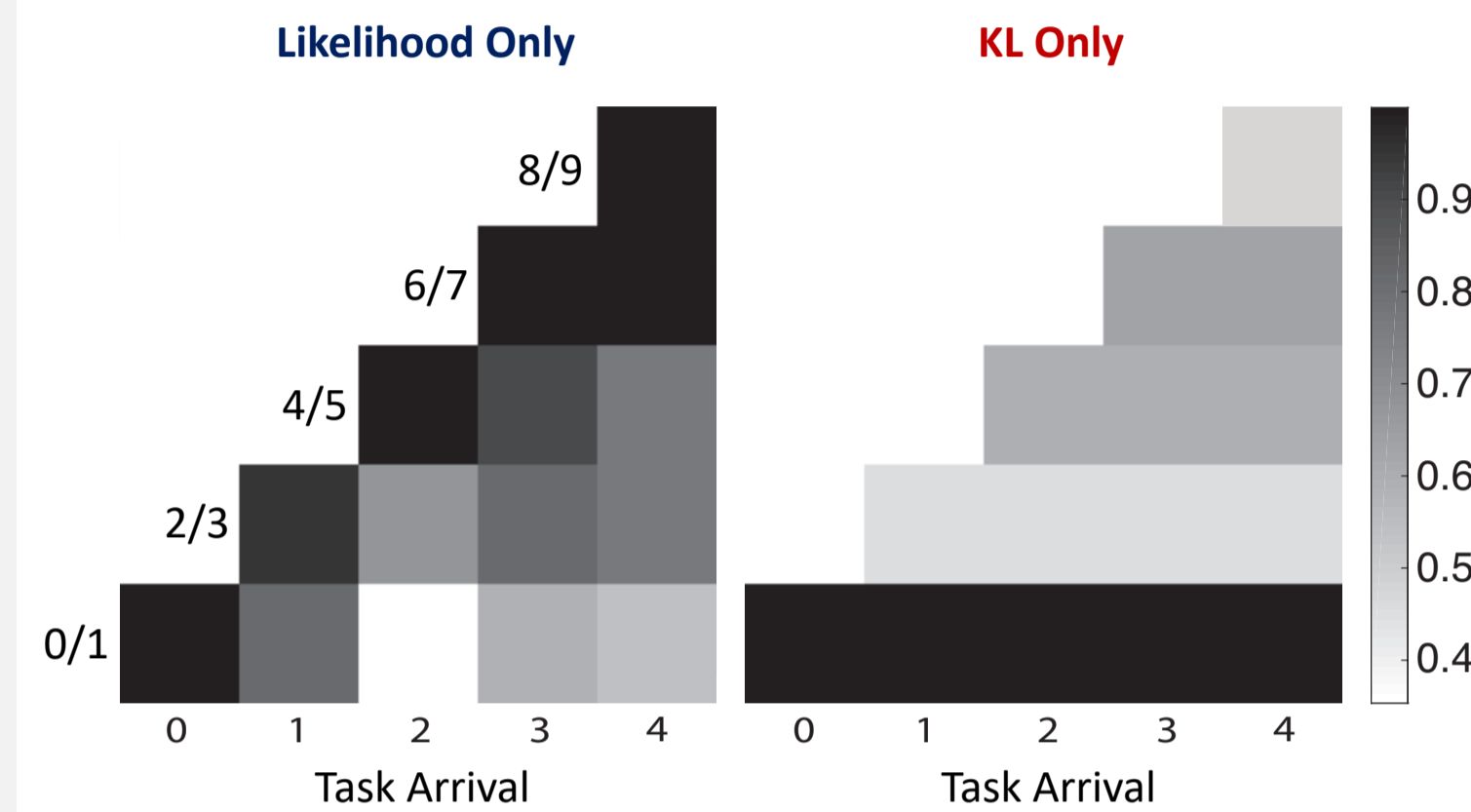
1  for t = 1, ..., T # T: total no. tasks
2
3  Observe new task  $\mathcal{D}_t$ 
4
5  if first task
6    # train feed-forward NN with  $\mathcal{D}_t$ 
7    Initialize  $q_0(\theta)$  with MLE
8
9  Update coreset  $C_t$  with  $C_{t-1}$  and  $\mathcal{D}_t$ 
10
11 # update posterior
12  $q_t(\theta) = \text{proj}(q_{t-1}(\theta), \mathcal{D}_t \cup C_{t-1} \setminus C_t)$ 
13
14 for task in previous tasks
15   # incorporate coreset
16    $q'_t(\theta) = \text{proj}(q_t(\theta), C_t)$ 
17   # make prediction
18   get_score(test_x, test_y,  $q'_t(\theta)$ )
    
```

## Experiments & Results

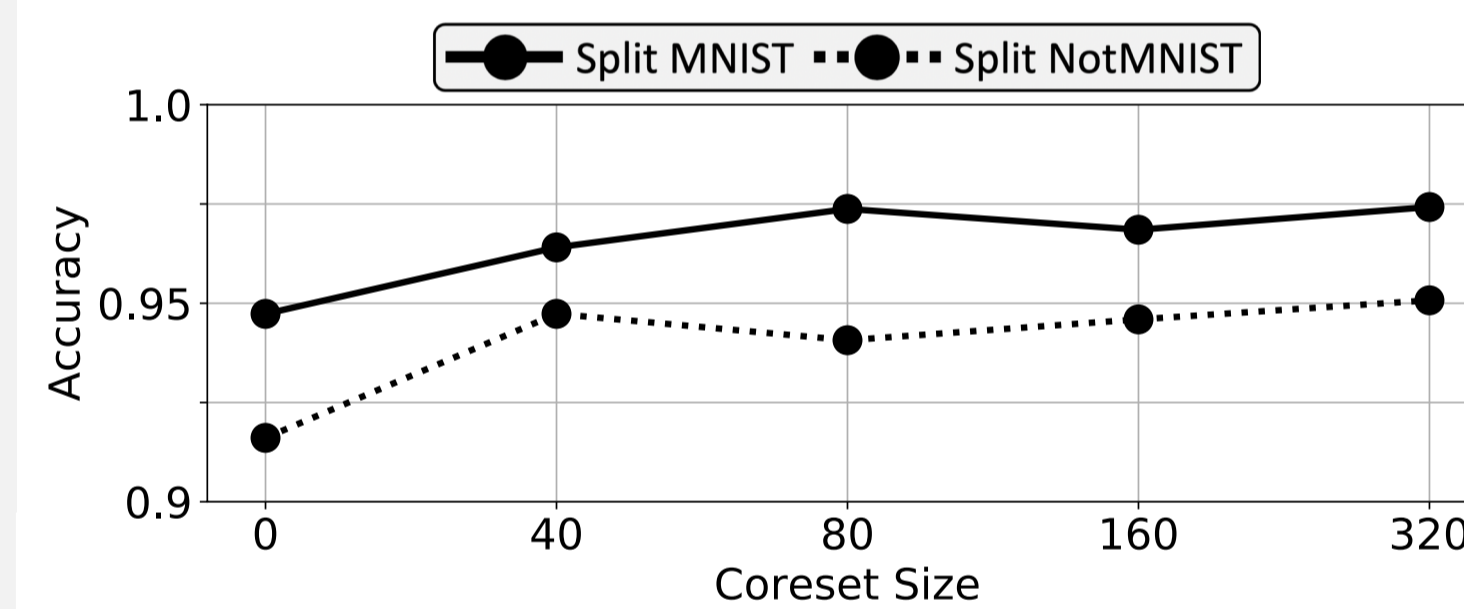
### 1. Plasticity vs. Stability (dataset: Split MNIST)

**Likelihood only** → catastrophic forgetting

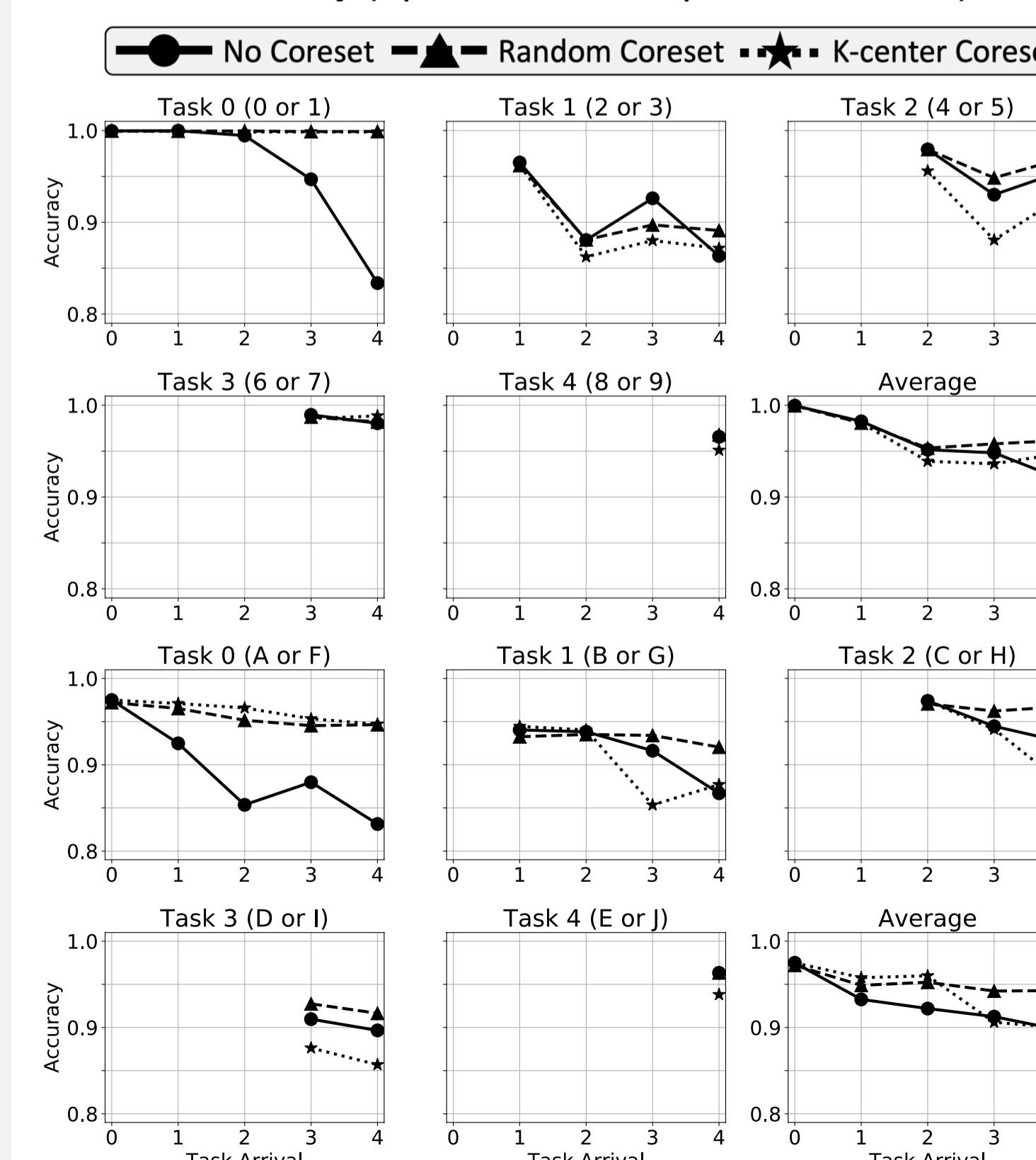
**KL only** → inability to adapt



### 2. Performance vs. Coreset Size



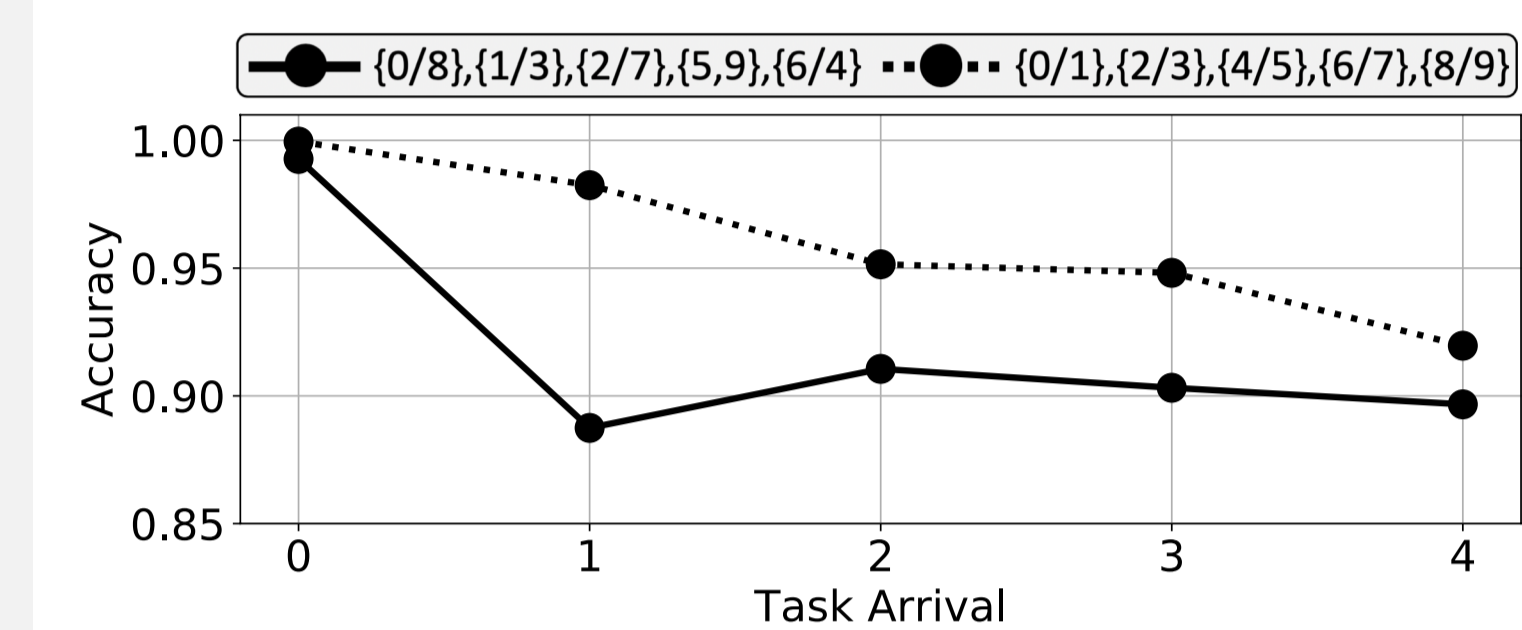
### 3. Task Accuracy (Split MNIST & Split NotMNIST)



## Extensions

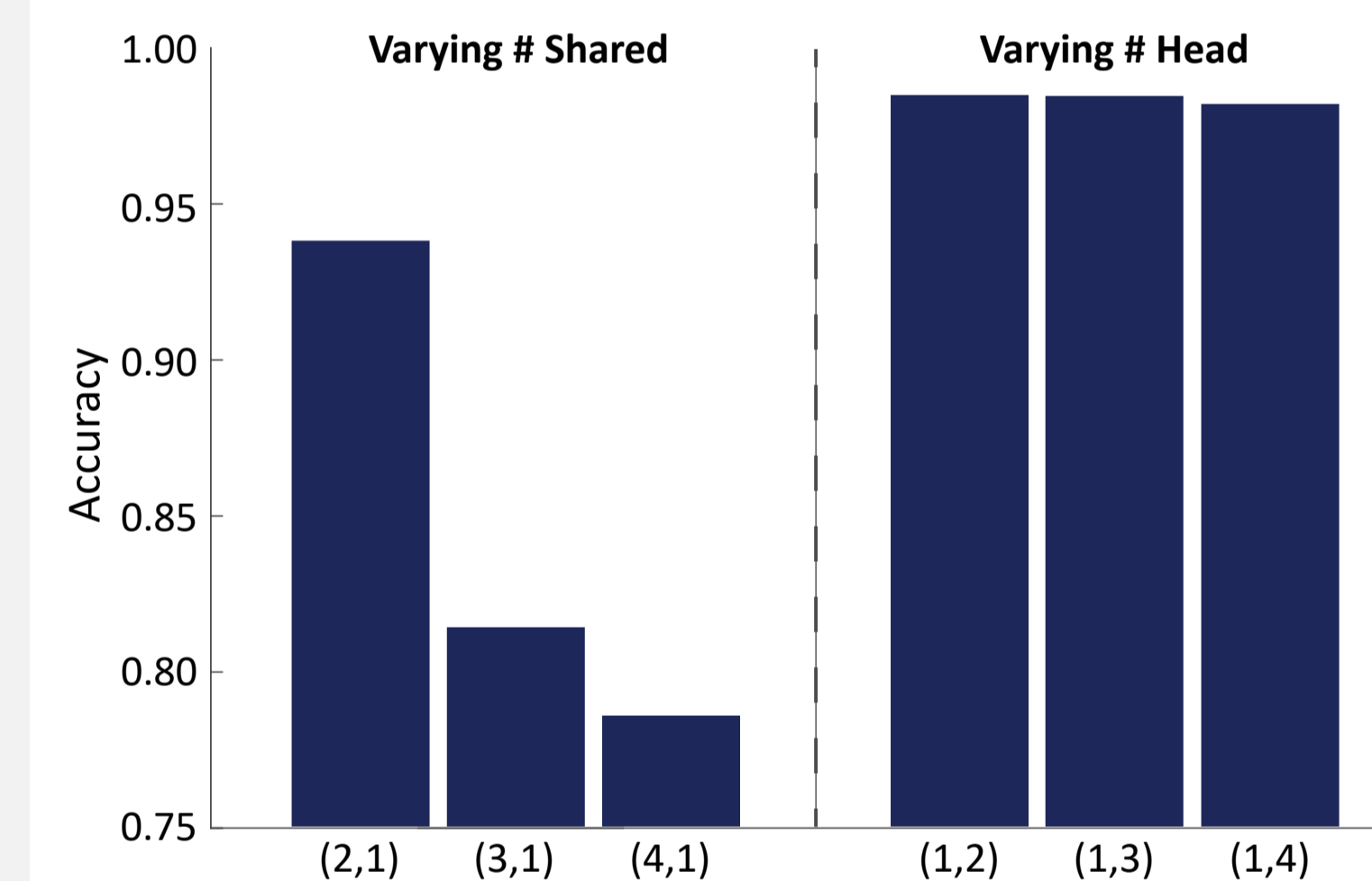
### 1. Adversarial Ordering

- Changing the incoming task order



### 2. Different Network Architectures

- Changing the number of shared/head layers



## Conclusion

Main contribution of this project includes:

- Customizable implementation of VCL pipeline in PyTorch (network architecture, task ordering, etc.)
- Demonstration of various consequences from changing model characteristics
- Performance increase with episodic memory

## References

- Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, Richard E. Turner. *Variational Continual Learning*. ICLR, 2018.
- Siddharth Swaroop, Cuong V. Nguyen, Thang D. Bui, Richard E. Turner. *Improving and Understanding Variational Continual Learning*. NIPS workshop on Continual Learning, 2018
- Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, Daan Wierstra. *Weight Uncertainty in Neural Networks*. ICML, 2015
- Alex Graves. *Practical Variational Inference for Neural Networks*. NIPS, 2011.