- What is the role of **hidden** layers in **Out-of-Distribution** learning?

- Where is the change of **representation** taking place in **OOD**?

“Catastrophic forgetting” in Neural Networks

Antoine Cornuéjols

AgroParisTech – INRAE   MIA Paris-Saclay

EKINOCS research group

Outline

1. How to measure the difficulty of a training example
2. What is catastrophic forgetting
3. Catastrophic forgetting and hidden representations
4. Catastrophic forgetting and the semantic similarity between tasks
5. Can forgetting be useful for transfer learning?
6. Is “forgetting less” useful for transfer learning?
7. Conclusions

How to measure the difficulty

of examples?

Measuring the **difficulty** of examples

- Previously
  - A **statistical** view
    - The probability of predicting the ground truth label for an **example omitted** from the training set
  - A **learning** view
    - The difficulty of learning an example, parameterized by **the earliest training iteration** after which the model (e.g. NN) predicts the ground truth class for that example in all subsequent iterations

Baldock, R., Maennel, H., & Neyshabur, B. (2021). **Deep learning through the lens of example difficulty**. *Advances in Neural Information Processing Systems, 34*.

Measuring the **difficulty** of examples

- A **new** proposition
  - The notion of "prediction depth"
  - And three distinct **difficulty types**:
    - Does this example **look mislabeled**?
    - Is classifying this example only easy if the label is given?
    - Is this **example ambiguous** both with and without its label?

Baldock, R., Maennel, H., & Neyshabur, B. (2021). **Deep learning through the lens of example difficulty**. *Advances in Neural Information Processing Systems, 34*.

Prediction depth

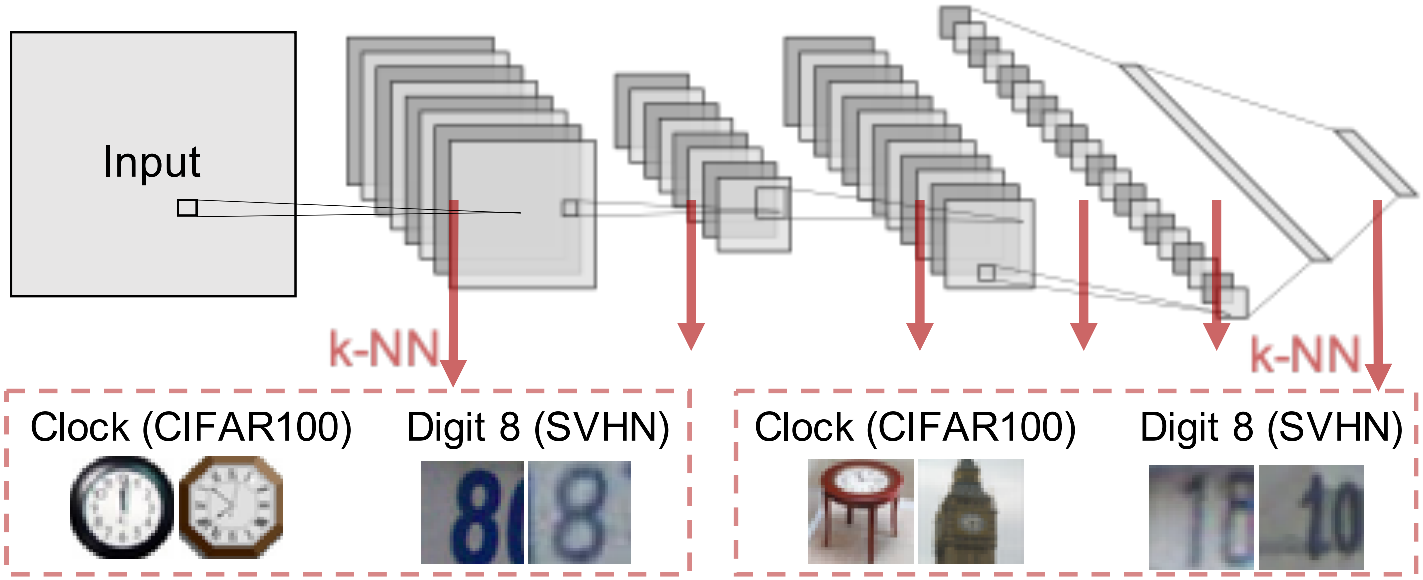
...

Prediction **depth**

- The number of hidden layers after which the network's final prediction **is** already **determined**

→ Prediction **depth**

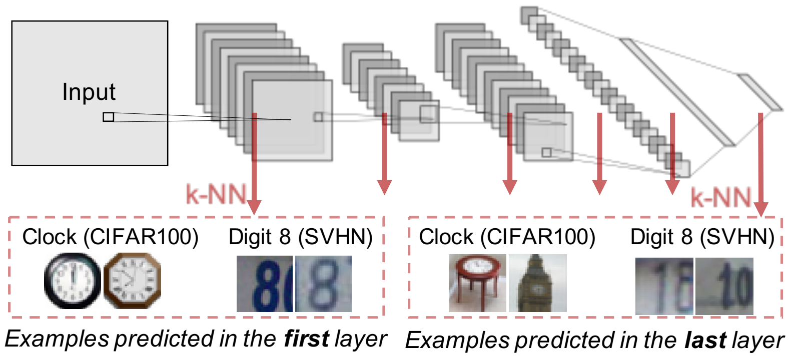
- The number of hidden layers after which the network's final prediction **is** already **determined**

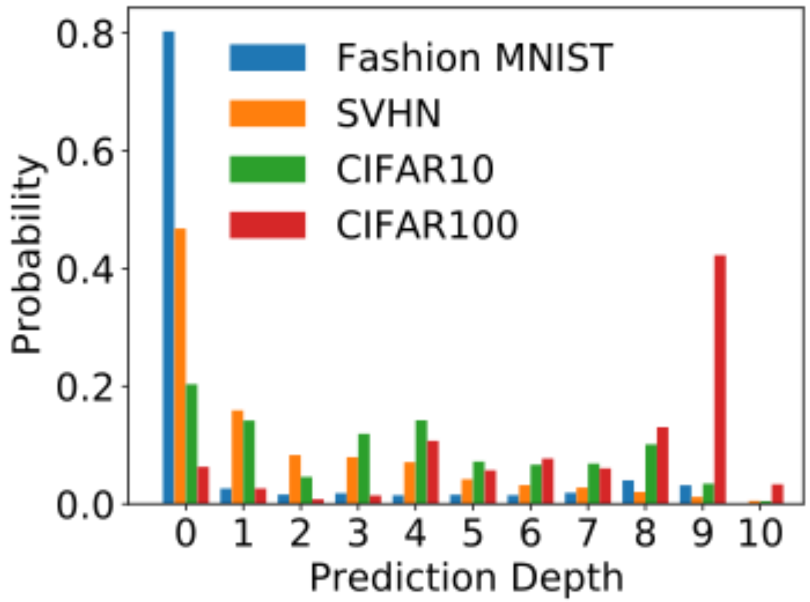


*Examples predicted in the **first** layer* *Examples predicted in the **last** layer*

Prediction depth

- The number of hidden layers after which the network's final prediction **is** already **determined**





Reflects the intuitive ranking from the **easier** to the more **difficult**

How to measure the prediction depth?

- k-NN classifier probes (with $k = 30$)
  - Compare the **hidden embedding** of an **input**
    to **those of the training set**

  (what is the **class** of the k nearest neighbors **in the embedding** considered)

- A prediction is defined to be made at a depth $L = l$ if
  - The k-NN classification **after** layer $L = l - 1$ is **different** from the network's final classification,
  - but the classification of k-NN probes **after** every layer $L \geq l$ are all **equal** to the final classification of the network

What they **claim** to show

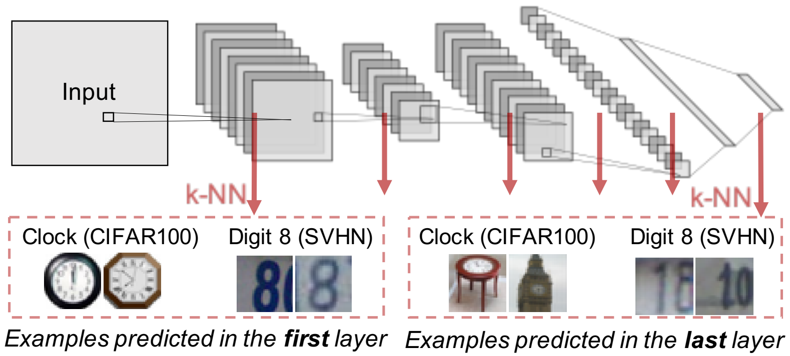
- The **prediction depth is larger** for examples that visually appear to be **more difficult**
  - And this is consistent between NN's architectures and random seeds
- Predictions are on average **more accurate** for validation points with **small prediction depths**
- Final predictions for data points that **converge earlier** during training are typically determined in **earlier layers**
- Both the adversarial input margin and output margin are **larger** for examples with **smaller prediction depths**
  - Intervention to reduce the output margin leads to predictions being made only in the **latest** hidden layers

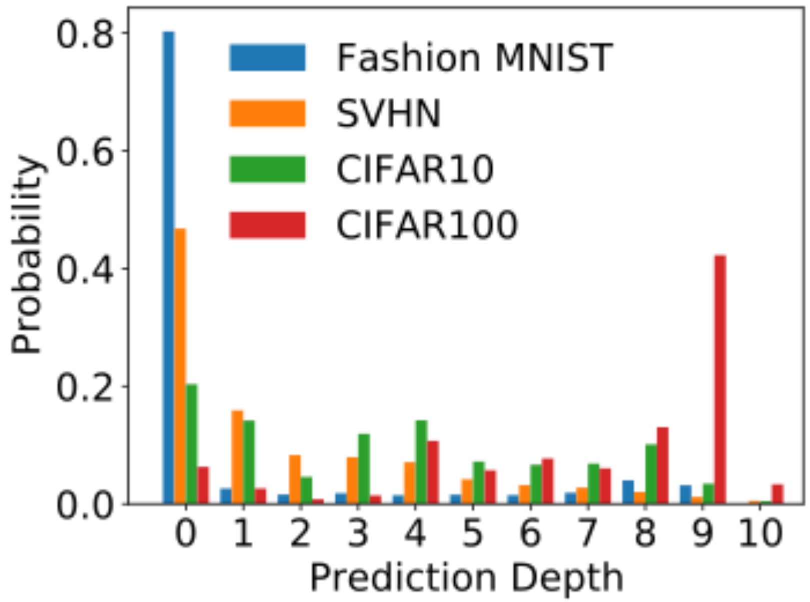
What they claim to show

1. Early layers **generalize** while later layers **memorize**
2. Networks converge **from** input layers **towards** output layers
3. **Easy** examples are learned **first**
4. Networks present **simpler functions earlier** in the training

What they claim to show

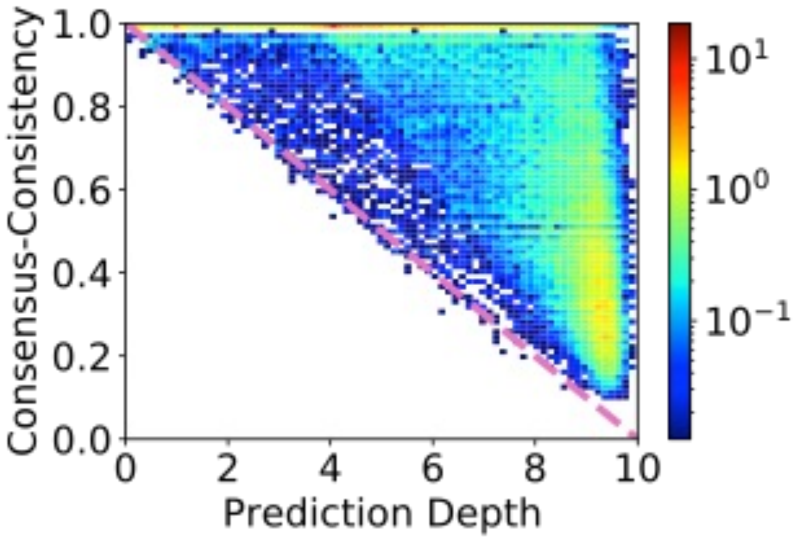
- The **prediction depth is larger** for examples that visually appear to be **more difficult**

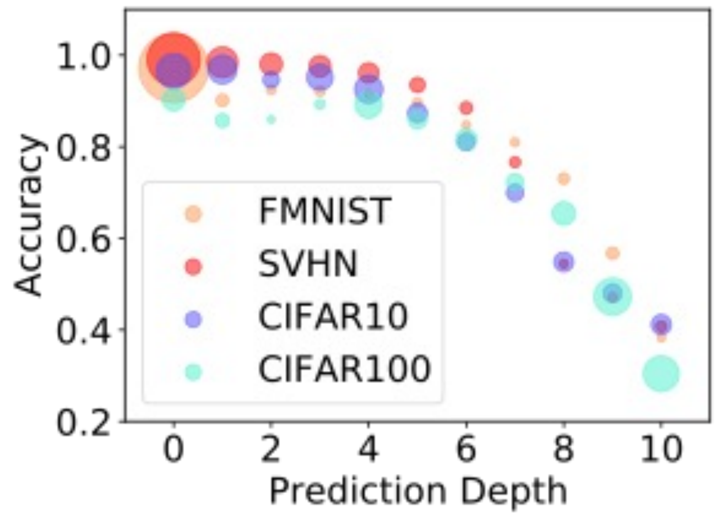




What they claim to show

...





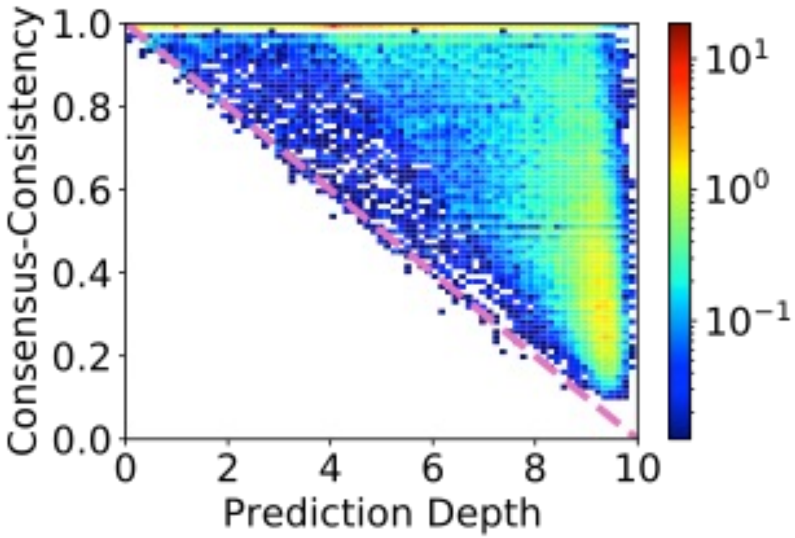
250 ResNet18 were trained on CIFAR100 (90:10% random train:validation splits). Comparison of the average **prediction depth** of a point to the **consensus-consistency** of the corresponding prediction.

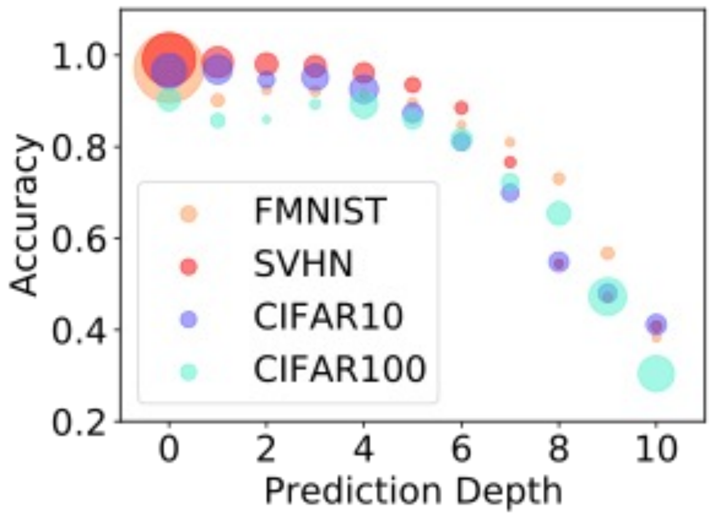
Consensus-consistency: the fraction of NNs that predict the ensemble's consensus class

For each dataset, **250** ResNet18 were trained on CIFAR100 (90:10% random train:validation splits). Each time a point appears in the validation split, its **prediction depth** and whether the **prediction was correct** was recorded.

What they claim to show

- Predictions are on average **more accurate** for validation points with **small prediction depths**





250 ResNet18 were trained on CIFAR100 (90:10% random train:validation splits). Comparison of the average **prediction depth** of a point to the **consensus-consistency** of the corresponding prediction.

Consensus-consistency: the fraction of NNs that predict the ensemble's consensus class

For each dataset, **250** ResNet18 were trained on CIFAR100 (90:10% random train:validation splits). Each time a point appears in the validation split, its **prediction depth** and whether the **prediction was correct** was recorded.

What they claim to show

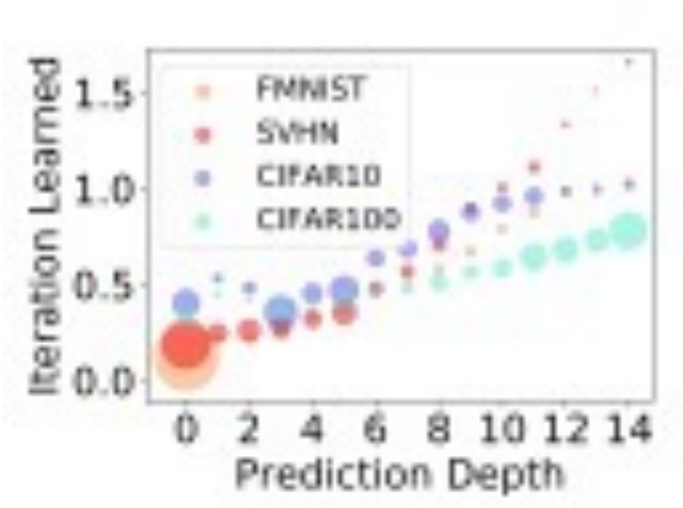
– Measure the difficulty of learning **an example** by the **speed at which the model's prediction converges** for that input during training

– **Iteration learned**. A data point is said to be learned by a classifier **at training iteration** $t = \tau$

if the predicted class at iteration $t = \tau - 1$ is different from the final prediction of the converged NN

and the predictions at all iterations $t \geq \tau$ are equal to the final prediction of the converged NN.

Renormalized



Each time an input appears in the validation split, the **prediction depth** and the **iteration learned** are recorded

Positive correlation between the **prediction depth** and the **iteration learned** appears for all datasets

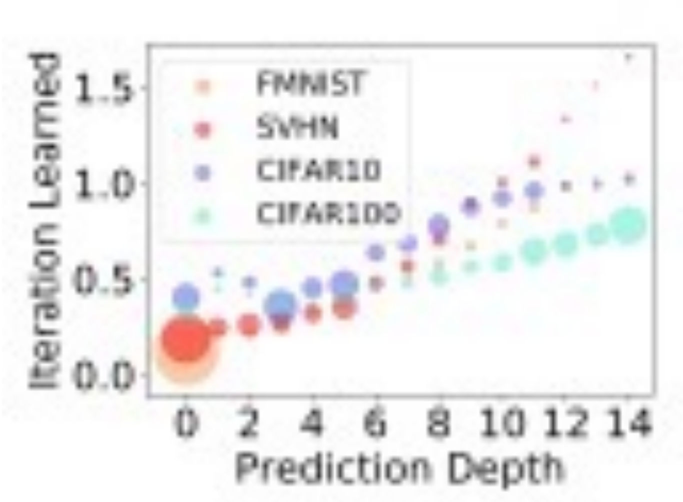
What they claim to show

- Final predictions for data points that **converge earlier** during training are typically determined in **earlier layers**
  - Measure the difficulty of learning **an example** by the **speed at which the model's prediction converges** for that input during training
  - **Iteration learned**. A data point is said to be learned by a classifier **at training iteration** $t = \tau$

    if the predicted class at iteration $t = \tau - 1$ is different from the final prediction of the converged NN

    and the predictions at all iterations $t \geq \tau$ are equal to the final prediction of the converged NN.

Renormalized



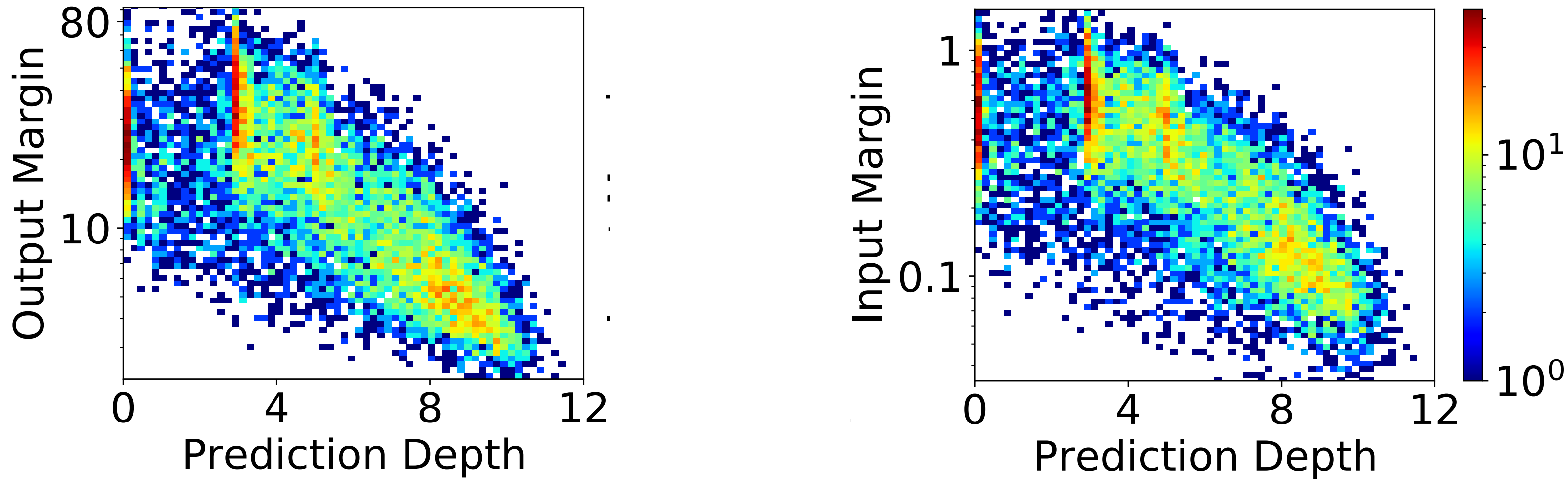
Each time an input appears in the validation split, the **prediction depth** and the **iteration learned** are recorded

Positive correlation between the **prediction depth** and the **iteration learned** appears for all datasets

What they claim to show

– **Output margin**: difference between the largest and second-largest output of the NN (logits)

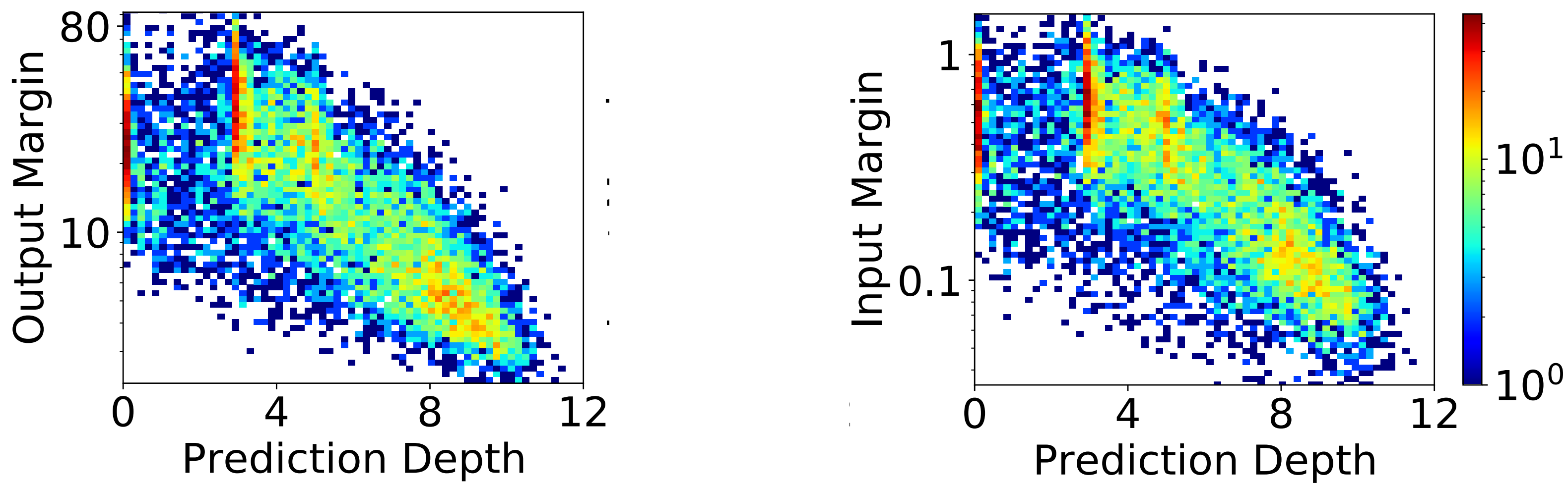
– **Adversarial input margin**: the smallest norm required for an adversarial perturbation in the input to change the NN's class prediction



Shows that data points with **smaller prediction depths** have both **larger** input and output margins on average, and that **variances** of the input and output margins **decrease** as the prediction depth increases

What they claim to show

- Both the adversarial input margin and output margin are **larger** for examples with **smaller prediction depths**
  - **Output margin**: difference between the largest and second-largest output of the NN (logits)
  - **Adversarial input margin**: the smallest norm required for an adversarial perturbation in the input to change the NN's class prediction



Shows that data points with **smaller prediction depths** have both **larger** input and output margins on average, and that **variances** of the input and output margins **decrease** as the prediction depth increases

What they claim to show

- **Different forms** of example difficulty
  - **Validation**: points with low prediction depth are "**clear**" and "**ambiguous**" otherwise
  - **Training** : idem

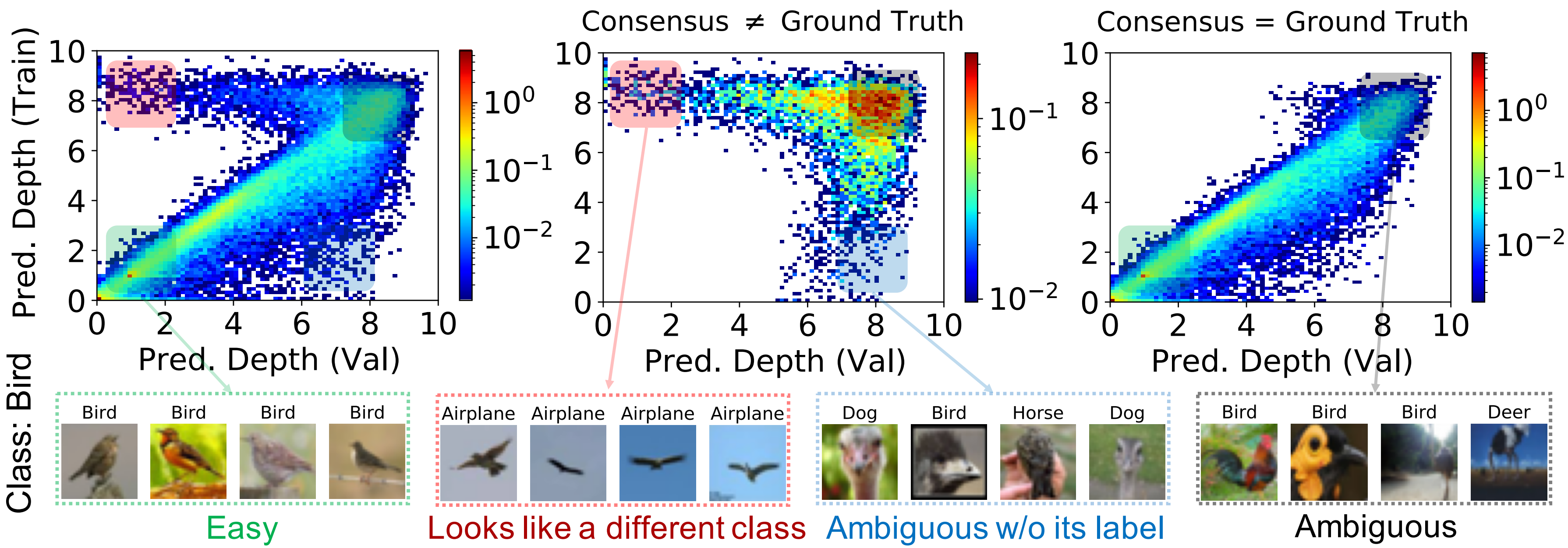
  – **Easy** examples (Low PD_{val} and low PD_{train})

  – **Look like a different class** (Low PD_{val} and high PD_{train}). (difficult to train, seemingly easy to classify)
    - E.g. **mislabeled** examples

  – **Ambiguous unless the label is given** (High PD_{val} and low PD_{train}).
    - E.g. resemble both their **own class** and **another** class
      Likely to be **misclassified**

  – **Ambiguous** (High PD_{val} and high PD_{train}).
    - Examples that may be **corrupted** or of a **rare** sub-class.

What they claim to show



Mislabeled Misclassified Corrupted

These examples are difficult to connect to their predicted class in the **validation** split but easy to connect to their ground truth class during **training**. These points may, for example, visually resemble both their own class and another class. They are likely to be misclassified.

Conclusion

Introduces a notion of example difficulty called the prediction depth

- which uses the **processing** of data **inside the network** to score the difficulty of **an example**

Conclusion

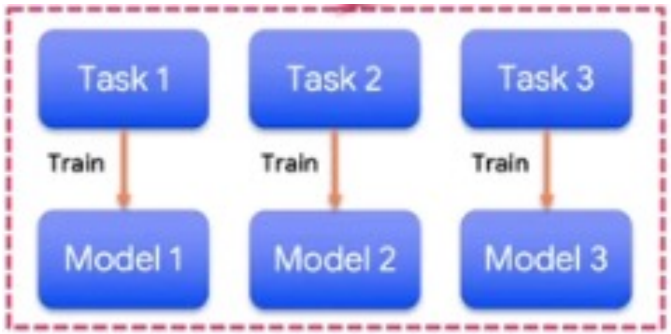
- **Easy** examples are learned and recognized **early** in the network

Outline

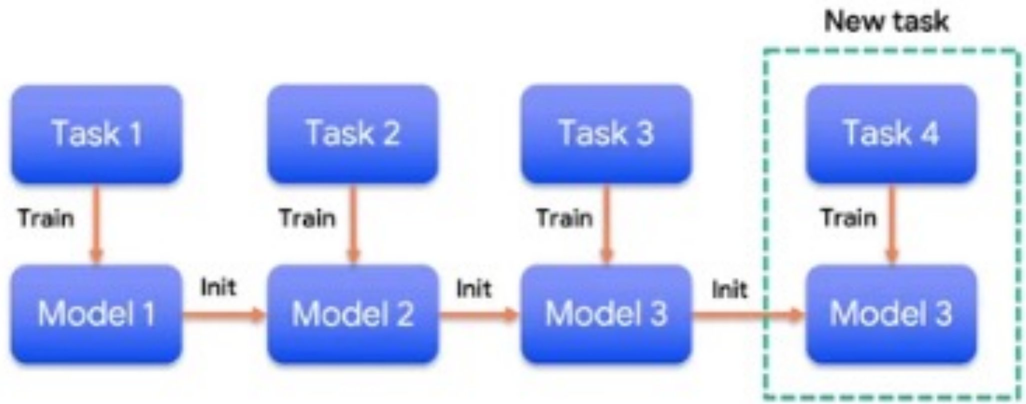
1. How to measure the difficulty of a training example
2. What is catastrophic forgetting
3. Catastrophic forgetting and hidden representations
4. Catastrophic forgetting and the semantic similarity between tasks
5. Can forgetting be useful for transfer learning?
6. Is “forgetting less” useful for transfer learning?
7. Conclusions

Continual learning of new tasks

Training new tasks **from scratch**



Continual learning



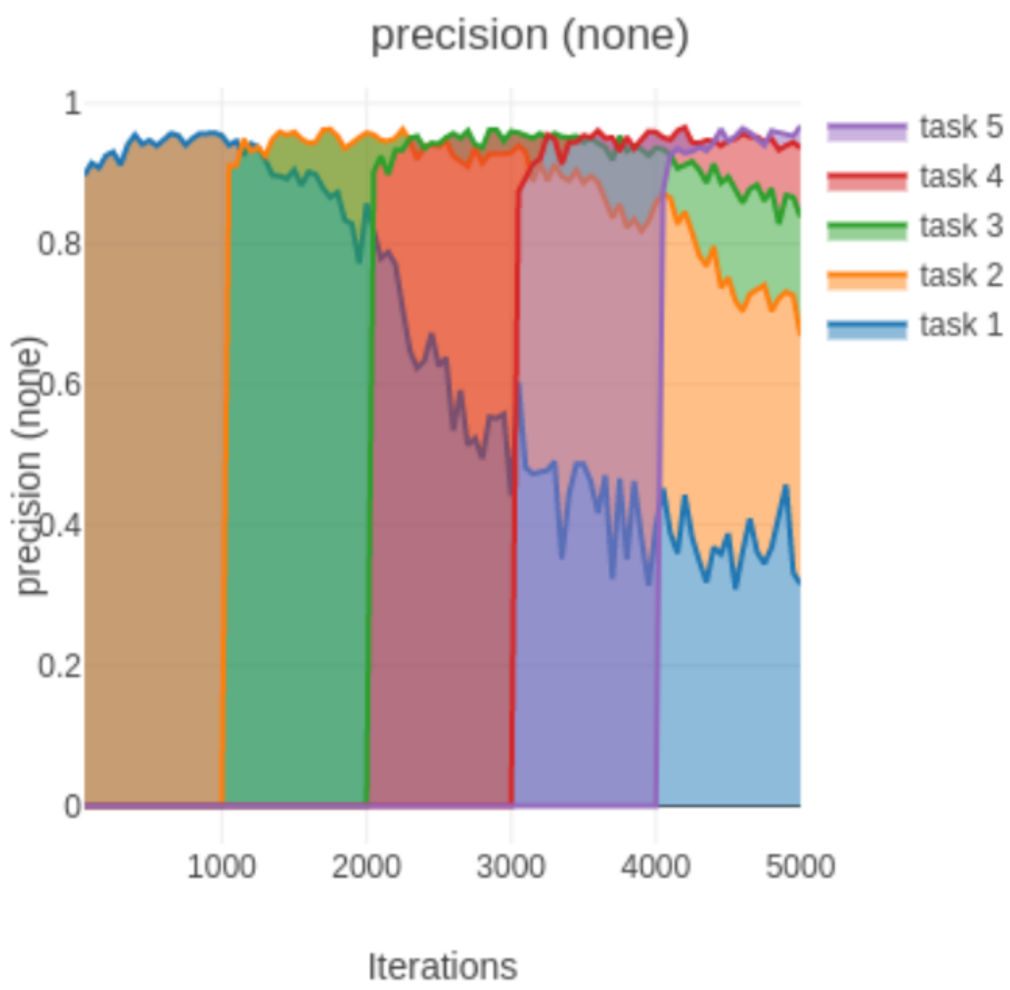
Chen, J., Nguyen, T., Gorur, D., & Chaudhry, A. (2023).
Is forgetting less a good inductive bias for forward transfer?
ICLR-2023.

Continual learning of new tasks

Continuously updating the model on new tasks results in severely **degraded** performance on **old tasks**

Catastrophic forgetting

McCaffary, D. (2021).
Towards continual task learning in artificial neural networks: current approaches and insights from neuroscience.
arXiv preprint arXiv:2112.14146.



Catastrophic forgetting

- ANNs have the tendency to completely and **abruptly forget** previous learned information upon learning new information
  - Therefore **ANNs** are unable to learn multiple tasks sequentially
  - Lifelong or continual learning would not be possible for **ANNs**
  - In **humans**, catastrophic forgetting does not happen
    - Learning to **drive a car** does not result in not knowing anymore how to **ride a bike**

Catastrophic forgetting: how to avoid it

Classical approach

- Training for ImageNet typically involves
  - to **break** the training dataset into M **distinct batches**,
  - for ImageNet each batch typically has about **100,000 instances** from 100 classes that are **not seen in later batches**,
  - and then the algorithm sequentially **loops** over each batch many times.

- **Not** efficient
- **Not** biologically plausible

Reasons for catastrophic forgetting

- **Interferences** in the hidden layers
  - Training on **task B** modifies a lot the weights learnt for **task A**
    - No guarantee that the representation of deeper layers learned for **task A** will be sufficient to losslessly encode novel information, for **task B**

- The major issue is **balancing**
  - the **stability** of existing representations
  - with the **plasticity** required to efficiently learn new ones

Outline

1. How to measure the difficulty of a training example
2. What is catastrophic forgetting
3. Catastrophic forgetting and **hidden representations**
4. Catastrophic forgetting and the semantic similarity between tasks
5. Can forgetting be useful for transfer learning?
6. Is “forgetting less” useful for transfer learning?
7. Conclusions

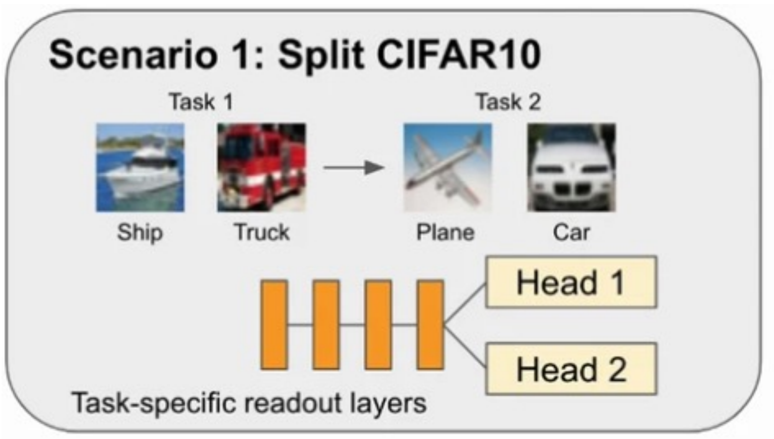
Catastrophic forgetting

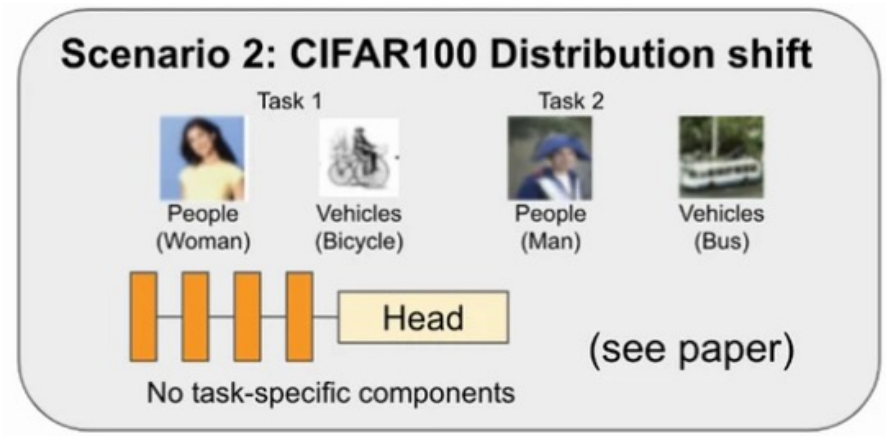
- Questions
  - What happens to the **internal representations** of neural networks as they undergo catastrophic forgetting?
  - Does the degree to which a network forgets depend on the ***semantic similarity*** between the successive tasks?

- What do we expect?

Catastrophic forgetting and **hidden representations**

- What role **hidden layers** play in forgetting?





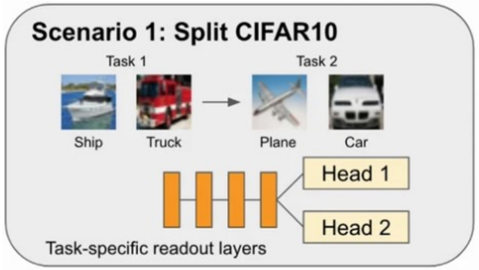
On CIFAR-10:
task 1 (5 classes) then **task 2** (5 ≠ classes)

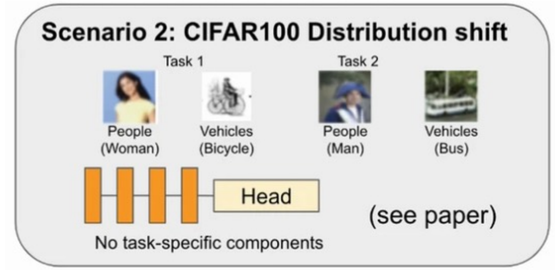
On CIFAR-100:
task 1 (examples of 5 subsets of 5 superclasses) then **task 2** (examples of 5 ≠ subsets of same 5 superclasses)

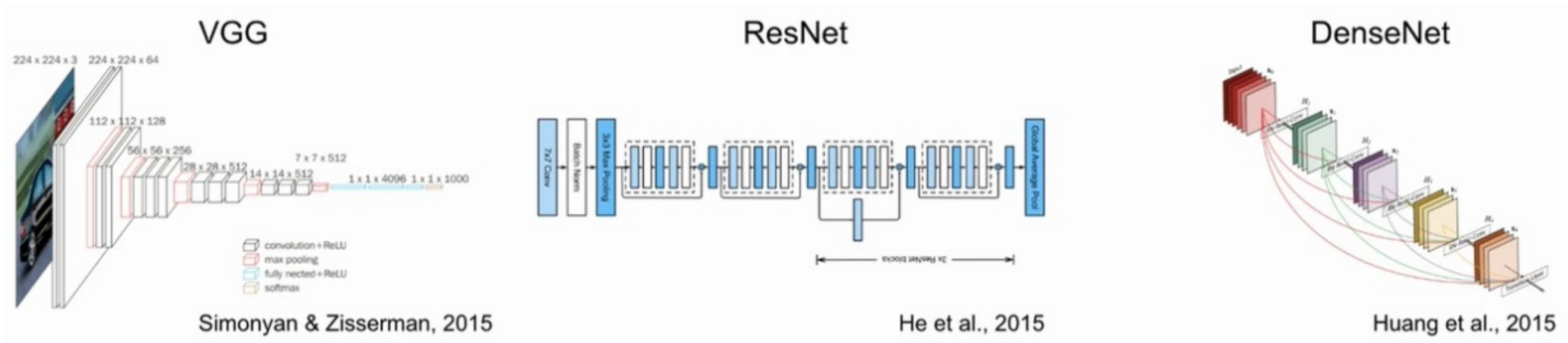
RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*

Catastrophic forgetting and **hidden representations**

- What role **hidden layers** play in forgetting?
  - Tested on 3 different Deep Neural Networks



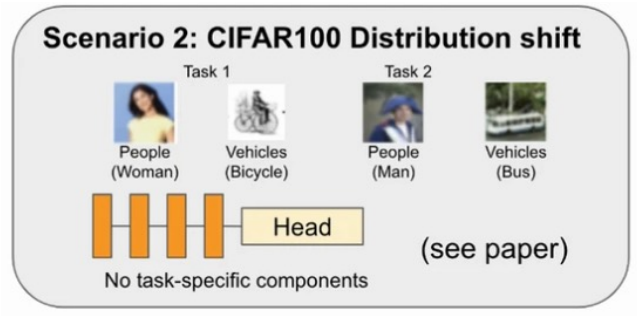




RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*

Catastrophic forgetting and **hidden representations**

- Manifestation of catastrophic forgetting?

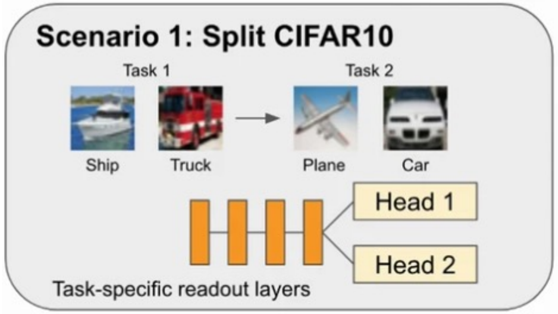


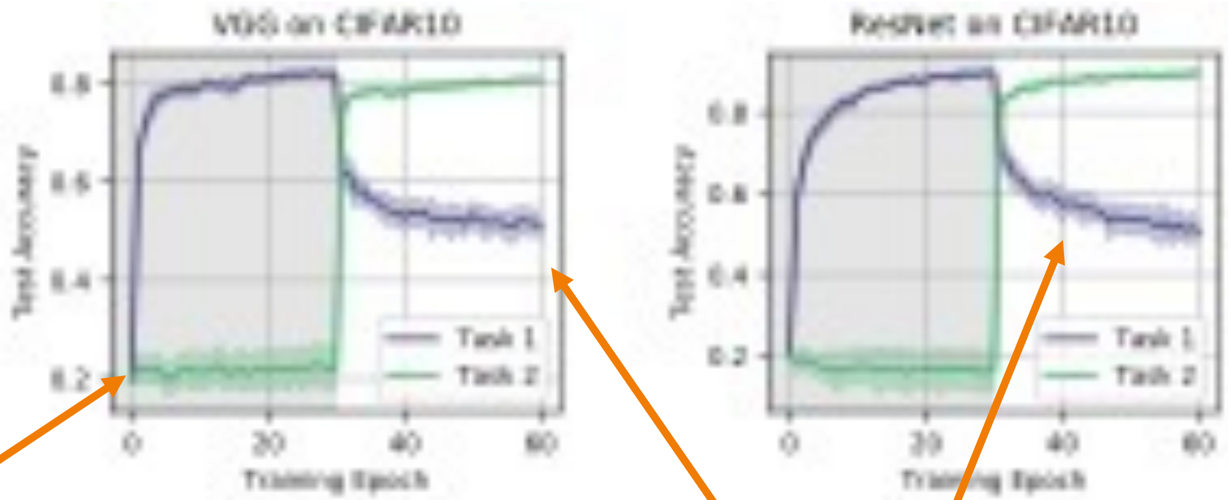
What do we expect?

RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*

Catastrophic forgetting and **hidden representations**

- Manifestation of catastrophic forgetting?





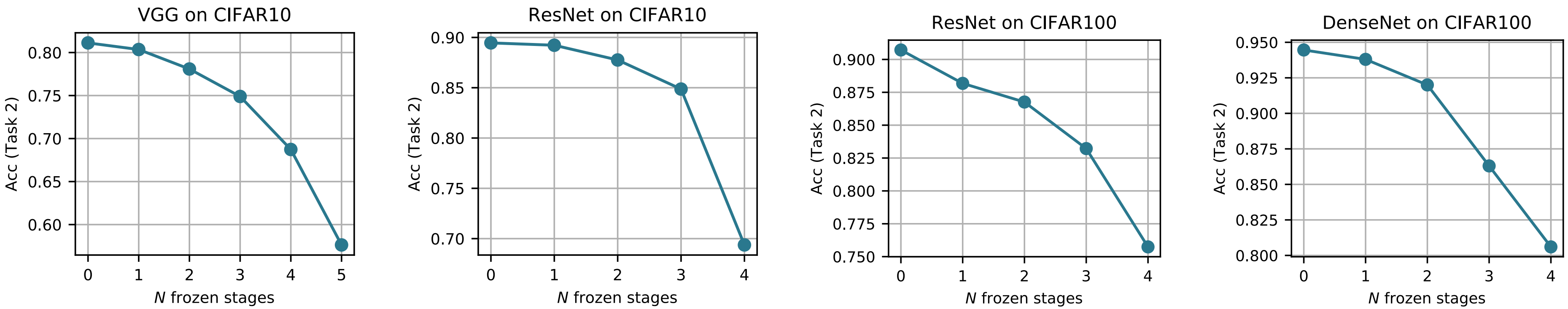
Starts at ~20% recognition rate: normal

Significant drop of performance on task 1 when learning task 2

RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*

Catastrophic forgetting and **hidden representations**

- What role **hidden layers** play in forgetting?



Stages = hidden layers starting from the earliest ones

- Freezing the **earliest hidden layers** after learning **task 1** has **little impact** on the performance of **task 2**
- **Higher layers** are disproportionately **responsible** for catastrophic forgetting

RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*

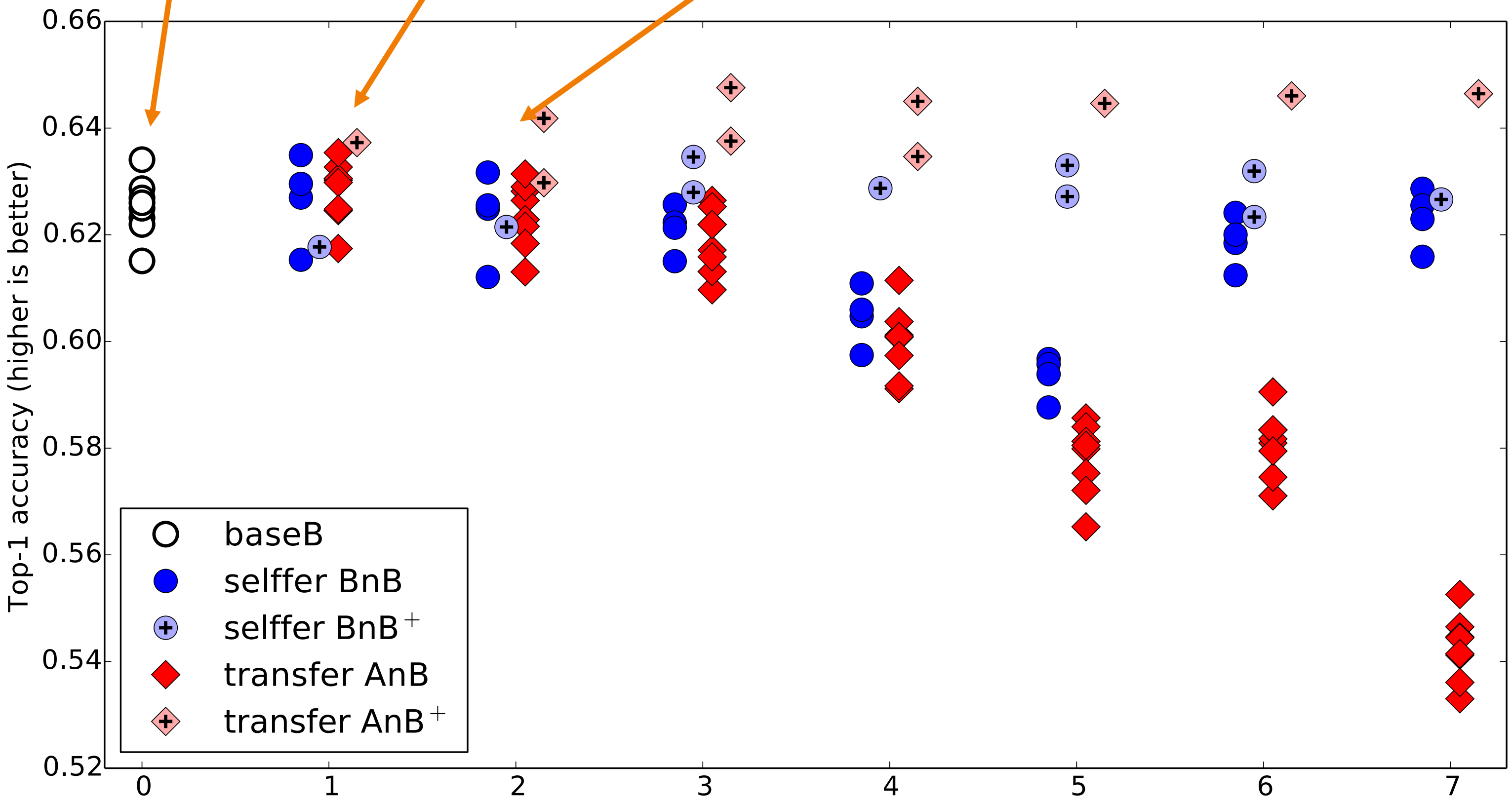
Results: what to think of them?

All layers **but the first** are retrained

NN (8 layers) trained from scratch

All layers **but the first two** are retrained

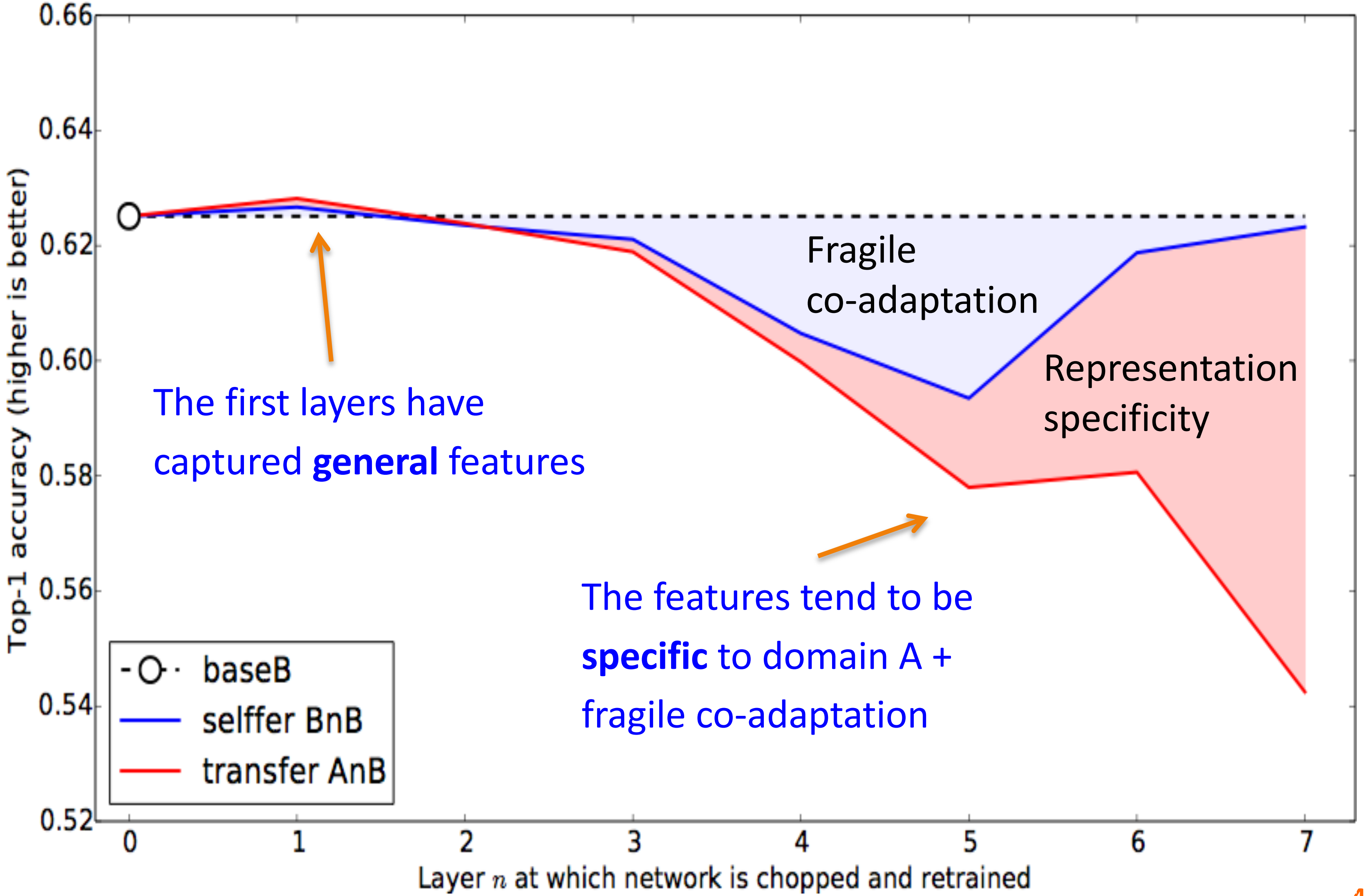
All layers but … are retrained



Yosinski J, Clune J, Bengio Y, and Lipson H. **How transferable are features in deep neural networks?** In *Advances in Neural Information Processing Systems* 27 (NIPS '14), NIPS Foundation, 2014.

Interpretation

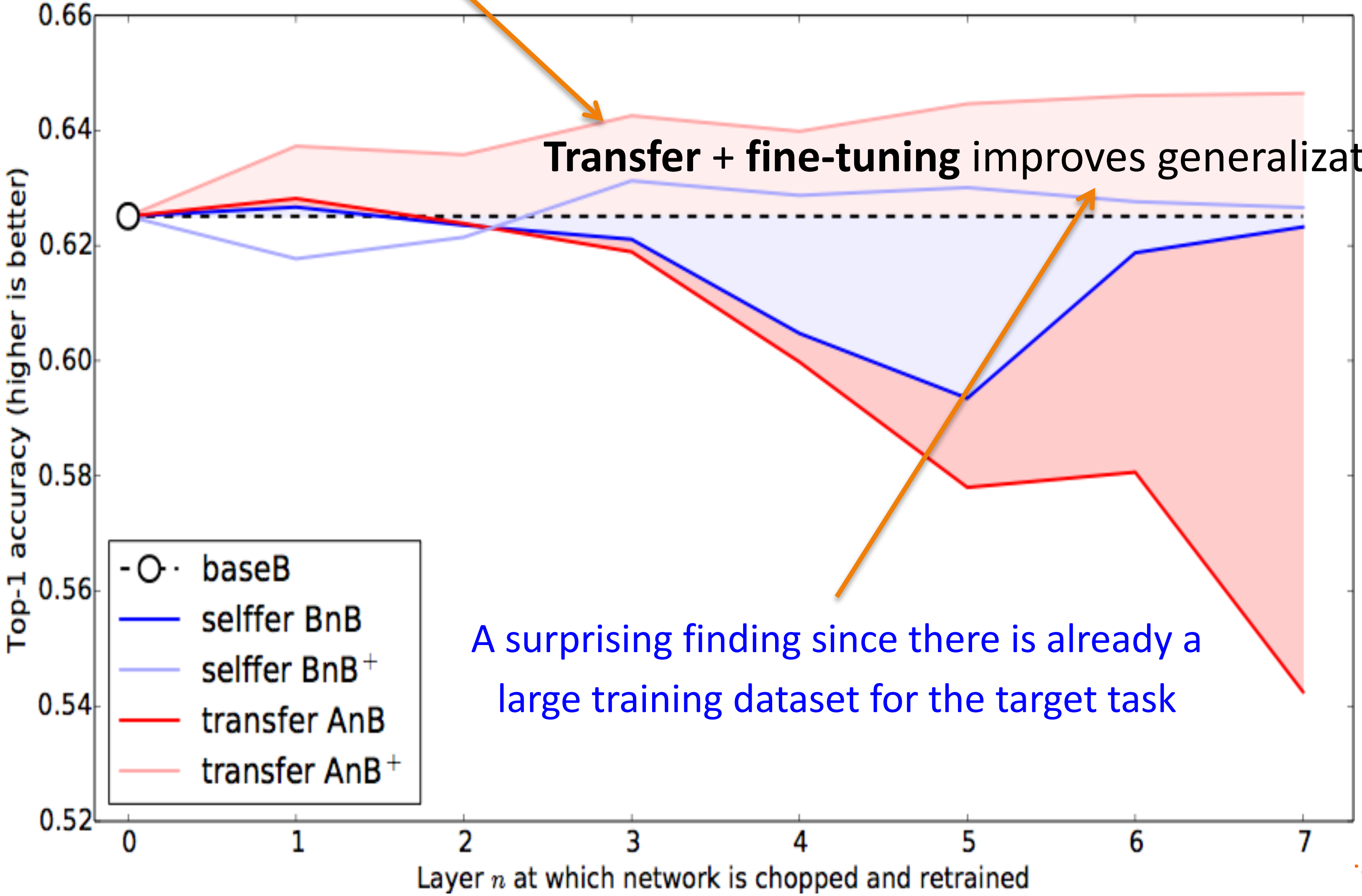
!!??



...

Interpretation

Retrain on all layers (fine-tuning) on domain B **after transfer** from domain A



Transfer + fine-tuning improves generalization

A surprising finding since there is already a large training dataset for the target task

Catastrophic forgetting and **hidden representations**

- What role **hidden layers** play in forgetting?
  - Measure *how similar* is each hidden layer **before** and **after** learning **task 2**
    - Use Centered Kernel Alignment (CKA)

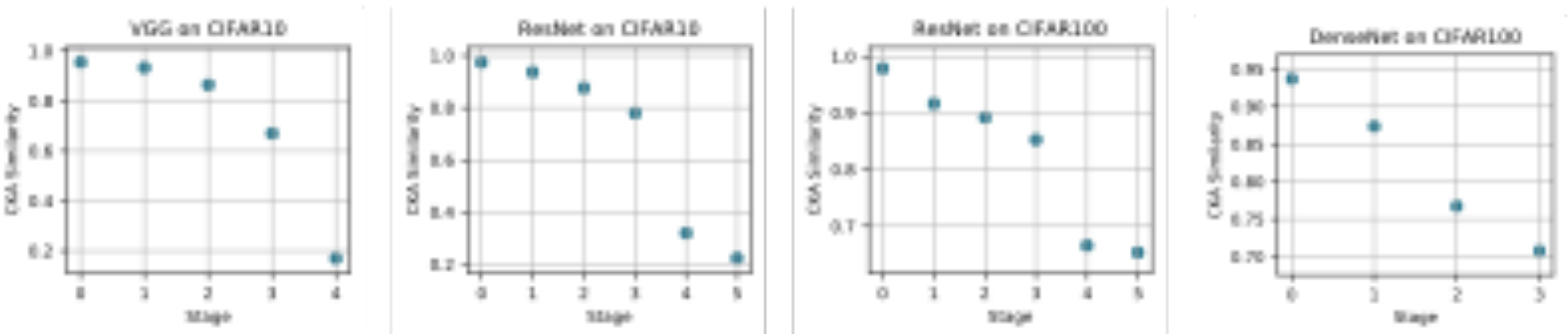
Specifically, letting $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times p}$ be (centered) layer activation matrices of (the same) n datapoints and p neurons, CKA computes

$$\mathrm{CKA}(\mathrm{X}, \mathrm{Y}) = \frac{\mathrm{HSIC}(\mathrm{XX}^{\mathrm{T}}, \mathrm{YY}^{\mathrm{T}})}{\sqrt{\mathrm{HSIC}(\mathrm{XX}^{\mathrm{T}}, \mathrm{XX}^{\mathrm{T}})}\sqrt{\mathrm{HSIC}(\mathrm{YY}^{\mathrm{T}}, \mathrm{YY}^{\mathrm{T}})}} \tag{1}$$

for HSIC Hilbert-Schmidt Independence Criterion (Gretton et al., 2005). We use linear-kernel CKA.

RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*

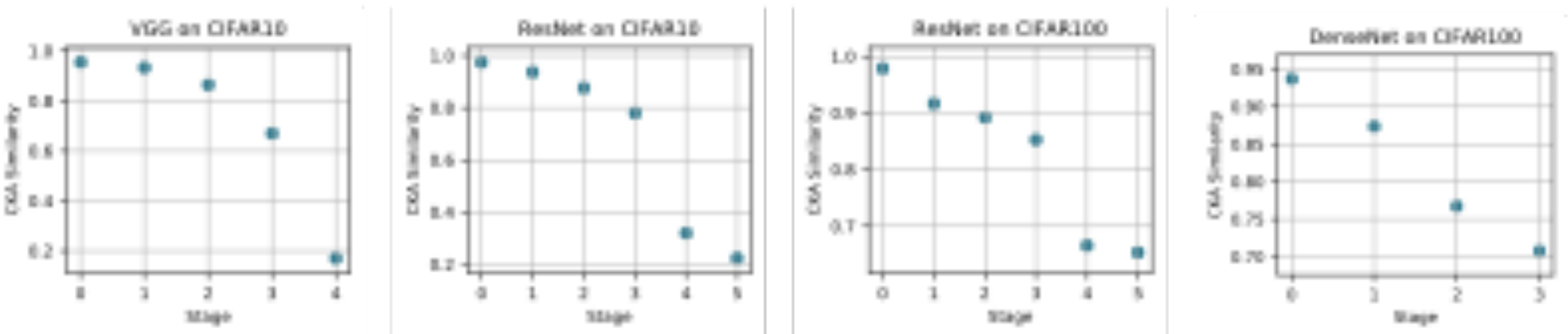
- What role **hidden layers** play in forgetting?
  - Measure how similar is each hidden layer **before** and **after** learning **task 2**
    - Use Centered Kernel Alignment (CKA)



RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*

Catastrophic forgetting and **hidden representations**

- What role hidden layers play in forgetting?
  - Measure how similar is each hidden layer **before** and **after** learning **task 2**
    - Use Centered Kernel Alignment (CKA)



- Again, the effect of learning task 2 is **biggest** on **higher** hidden layers

For all **tasks** and all **NNs**

RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*

Catastrophic forgetting and **hidden representations**

- What role **hidden layers** play in forgetting?
  - Measure how similar is each subspace (PCA of activations) of the hidden layers **before** and **after** learning **task 2**

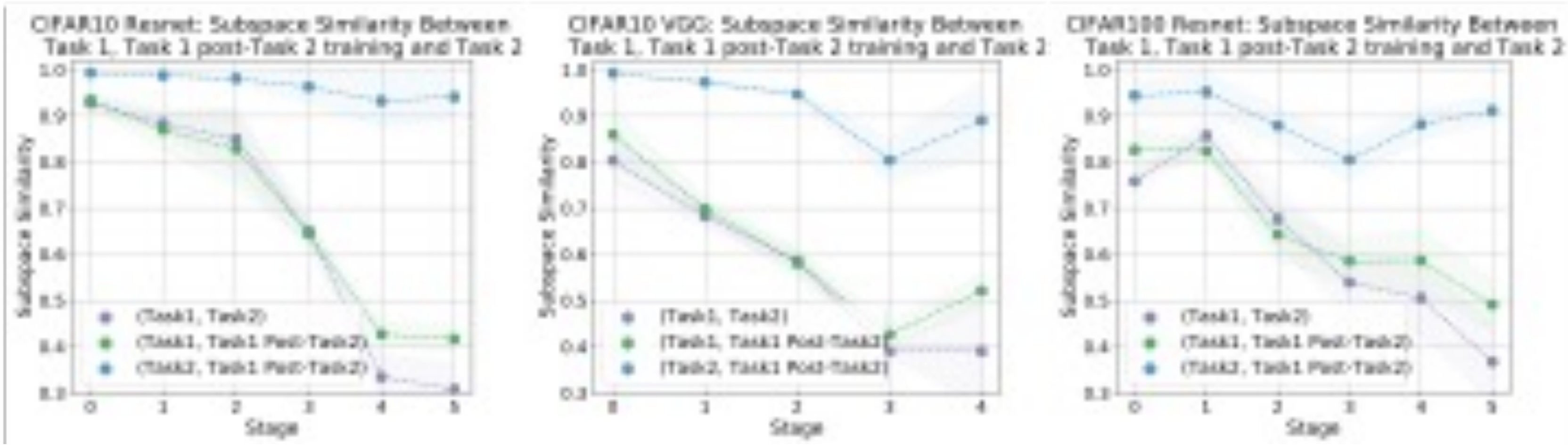
Letting $X \in \mathbb{R}^{n \times p}$ be the (centered) layer activation matrix of n examples by p neurons, we compute the PCA decomposition of X, i.e. the eigenvectors $(v_1, v_2, \ldots)$ and eigenvalues $(\lambda_1, \lambda_2, \ldots)$ of $X^\top X$. Letting V_k be the matrix formed from the top k principal directions, $v_1, \ldots, v_k$ as columns, and U_k the corresponding matrix for a different activation matric Y, we compute

$$\mathrm{SubspaceSim}_k(X, Y) \;=\; \frac{1}{k}\,||V_k^\top U_k||_F^2$$

This measures the **overlap** in the subspaces spanned by $(v_1, ..., v_k)$ and $(u_1, ..., u_k)$. Concretely, if X and Y correspond to layer activation matrices for two different tasks, $\mathrm{SubspaceSim}_k(X, Y)$ measures **how similarly the top k representations** for those tasks are stored in the network.

Catastrophic forgetting and **hidden representations**

- What role **hidden layers** play in forgetting?
  - Measure how similar is each subspace (PCA of activations) of the hidden layers **before** and **after** learning **task 2**



- (Task 1, task 2): low similarity for higher hidden layers
- (Task 1, and again on task 1 **after** training on task 2): much has been lost
- (Task 2, task 1 **after** training on task 1 then task 2): higher hidden layers are more similar to task 2 than to task 1!

Catastrophic forgetting and **hidden representations**

- During sequential training,
  - effective **feature reuse happens** in the **lower layers**,
  - but in **the higher layers**, after Task 2 training, **Task 1 representations are mapped into the same subspace as Task 2**.

Specifically, **Task 2** training **causes subspace erasure** of **Task 1** in the **higher layers**.

Catastrophic forgetting and **hidden representations**

- During sequential training,
  - effective **feature reuse happens** in the **lower layers**,
  - but in **the higher layers**, after Task 2 training, **Task 1 representations are mapped into the same subspace as Task 2**.

Specifically, **Task 2** training **causes subspace erasure of Task 1** in the **higher layers**.

→ Do popularly used mitigation methods act to **stabilize higher layers**?

Mitigation strategies and hidden representations

Types of mitigation **strategies**

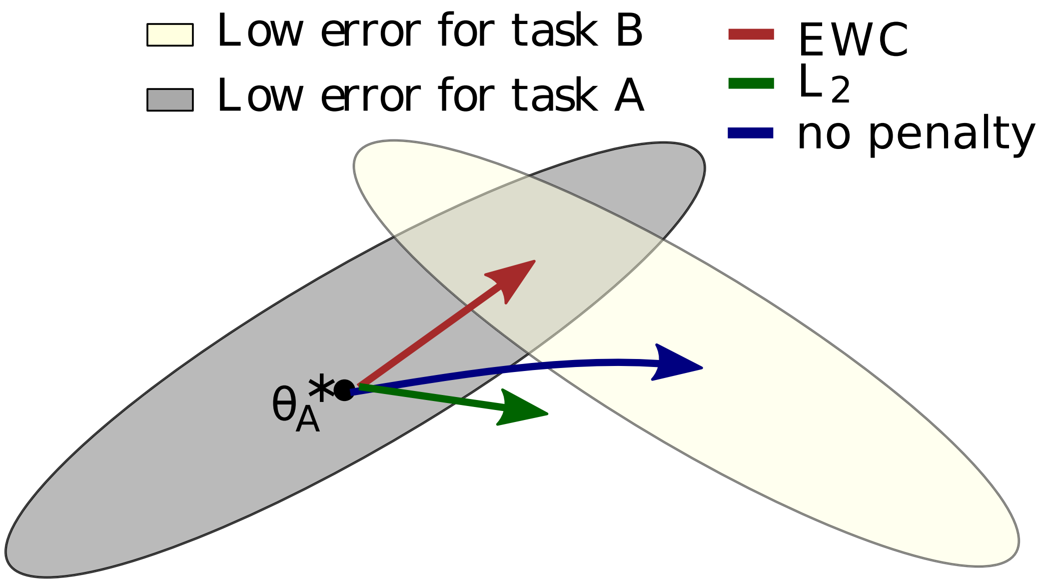
- **Regularization**-based approaches
- **Replay**-based approaches

- What are their **impact**? Are they successful?
- **How do they act** on the hidden layers?

Mitigation strategies - **Regularization** approaches

...

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). **Overcoming catastrophic forgetting in neural networks**. *Proceedings of the national academy of sciences*, *114*(13), 3521-3526.



While learning task B, **EWC** protects the performance in task A by constraining the parameters to **stay in a region of low error** for task A centered around $\theta_A{}^*$

Mitigation strategies - **Regularization** approaches

...

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). **Overcoming catastrophic forgetting in neural networks**. *Proceedings of the national academy of sciences*, *114*(13), 3521-3526.

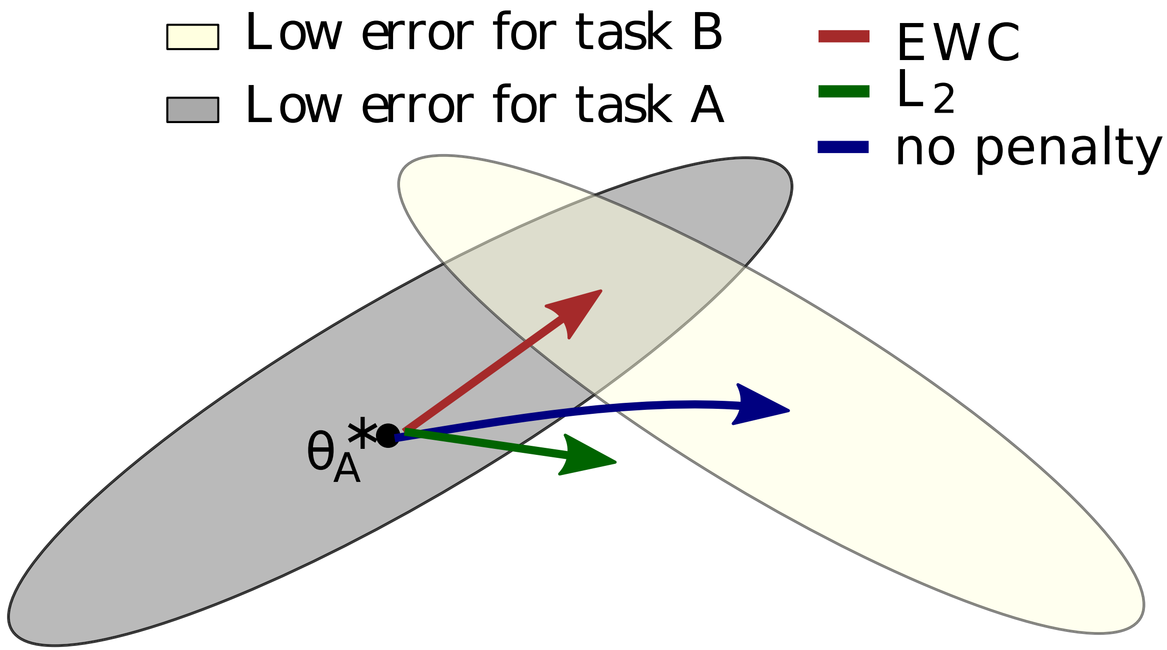


Fig. 1. EWC ensures task *A* is remembered while training on task *B*. Training trajectories are illustrated in a schematic parameter space, with parameter regions leading to good performance on task *A* (gray) and on task *B* (cream color). After learning the first task, the parameters are at θ_A^*. If we take gradient steps according to task *B* alone (blue arrow), we will minimize the loss of task *B* but destroy what we have learned for task *A*. On the other hand, if we constrain each weight with the same coefficient (green arrow), the restriction imposed is too severe and we can remember task *A* only at the expense of not learning task *B*. EWC, conversely, finds a solution for task *B* without incurring a significant loss on task *A* (red arrow) by explicitly computing how important weights are for task *A*.

Mitigation strategies - **Regularization** approaches

- "Elastic Weight consolidation" (EWC)
  - **EWC** works by slowing learning of the network weights which are most relevant for solving **previously encountered tasks**

Hyperparameter indicating the relative importance of previously encountered tasks compared to new ones

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i \left(\theta_i - \theta^*_{A,i}\right)^2$$

Parameters of the network

Loss for **task B**

Parameters important for solving **task A**

Fisher information matrix

Mitigation strategies - **Regularization** approaches

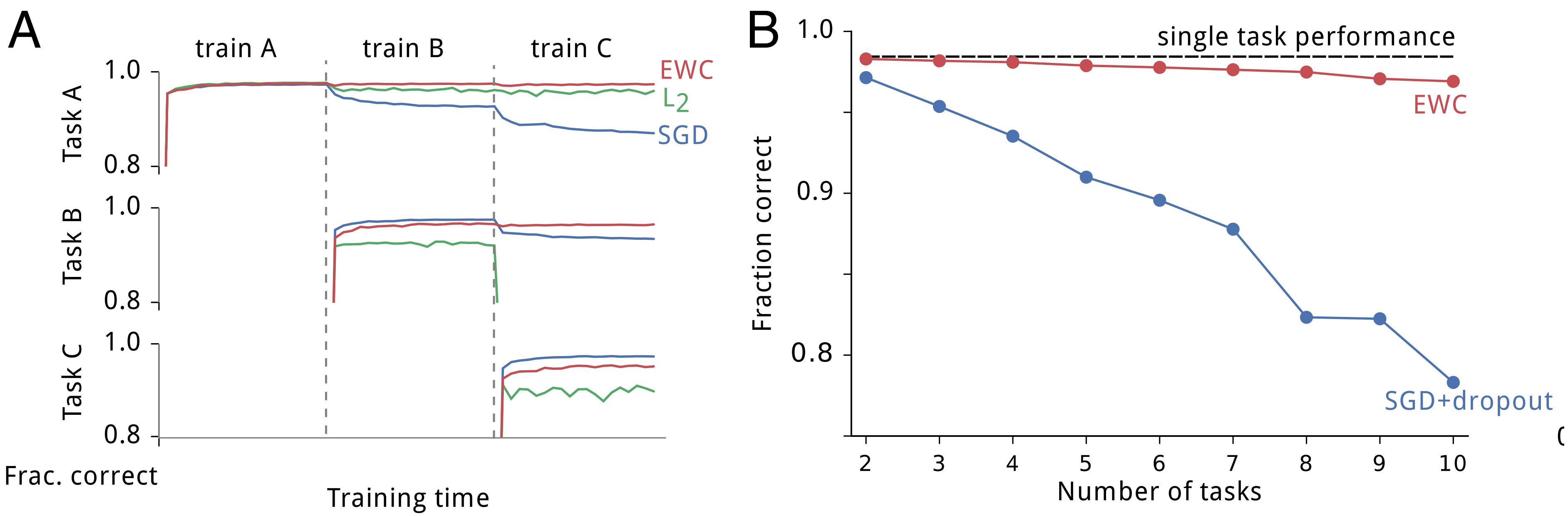
- Elastic Weight consolidation (**EWC**)
  - the Fisher information matrix is used to give an estimation of the **importance of weights** for solving tasks
    - The **importance weighting** is proportional to the diagonal of the Fisher information metric over the old parameters for the previous task

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i \left(\theta_i - \theta^*_{A,i}\right)^2$$

Parameters important for solving task A

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). **Overcoming catastrophic forgetting in neural networks**. *Proceedings of the national academy of sciences*, *114*(13), 3521-3526.

Mitigation strategies - **Regularization** approaches



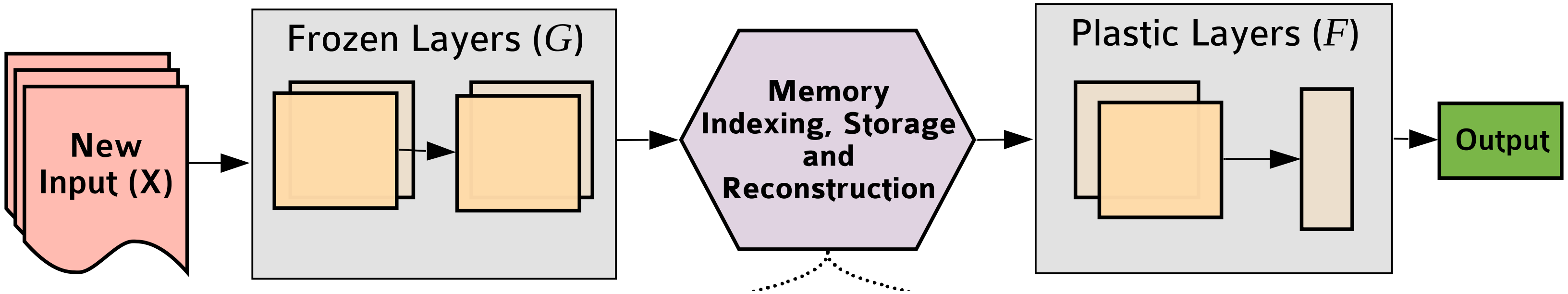
- (**A**) **Training curves** for three random permutations A, B, and C, using **EWC** (red), L_2 **regularization** (green), and **plain SGD** (blue).
  Note that only EWC is capable of maintaining a high performance on old tasks, while retaining the ability to learn new tasks.
- (**B**) **Average performance across all tasks**, using **EWC** (red) or **SGD** with dropout regularization (blue). The dashed line shows the performance on a single task only.

Mitigation strategies – **Replay-based** approaches

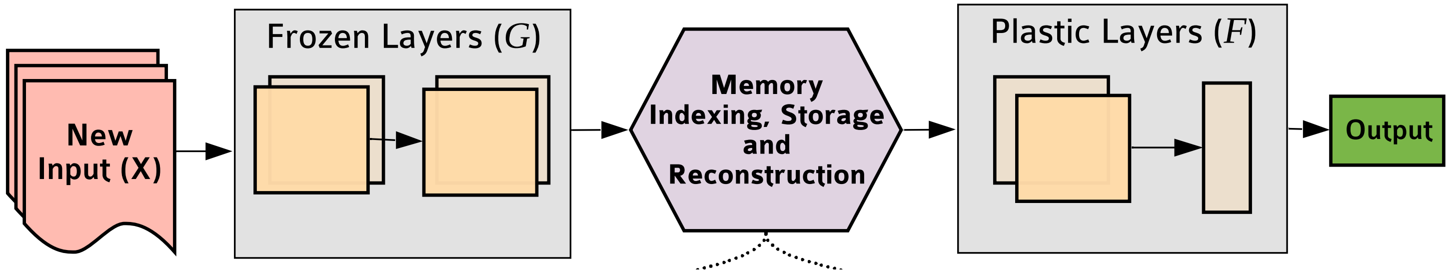
- Deep generative replay
  - A generative model is used to **generate representative data** from **previous** tasks
  - From which a sample is **selected** and **interspersed** with the dataset of the new task
  - Example: REMIND (Replay using Memory Indexing)
    - Replays a compressed representation of previously encountered training data
    - Using hidden layers (e.g. a feature map)

Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., & Kanan, C. (2020, August). **Remind your neural network to prevent catastrophic forgetting**. In *European Conference on Computer Vision* (pp. 466-483). Springer, Cham.

REMIND



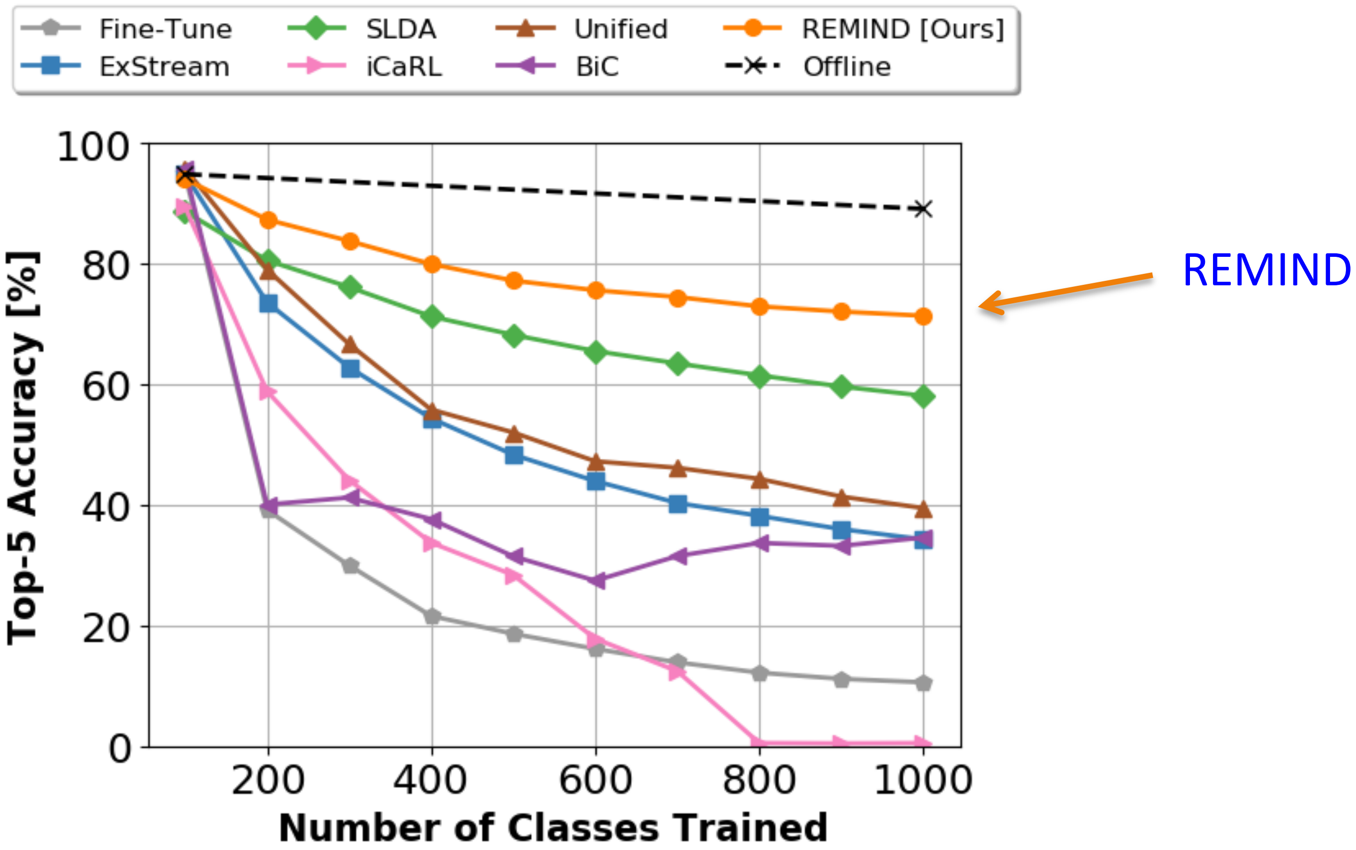
New
Input (X)
Frozen Layers (G)
Memory
Indexing, Storage
and
Reconstruction
Plastic Layers (F)
Output



- First, **train the complete network**: G + F layers on the training set
- Froze (G) and **store** sort of **prototype features** of the training examples
- Later, during training of **new tasks**, use the stored prototype features to **generate training instances** before (F) related to the previous tasks together with new training examples and train only (F)

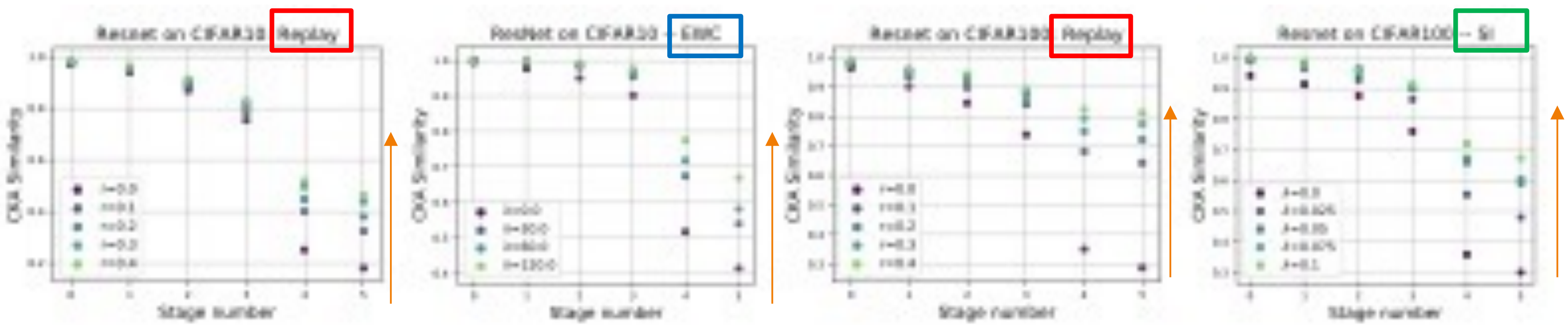
REMIND

- Performances when learning additional classes of ImageNet



Now the **analysis**

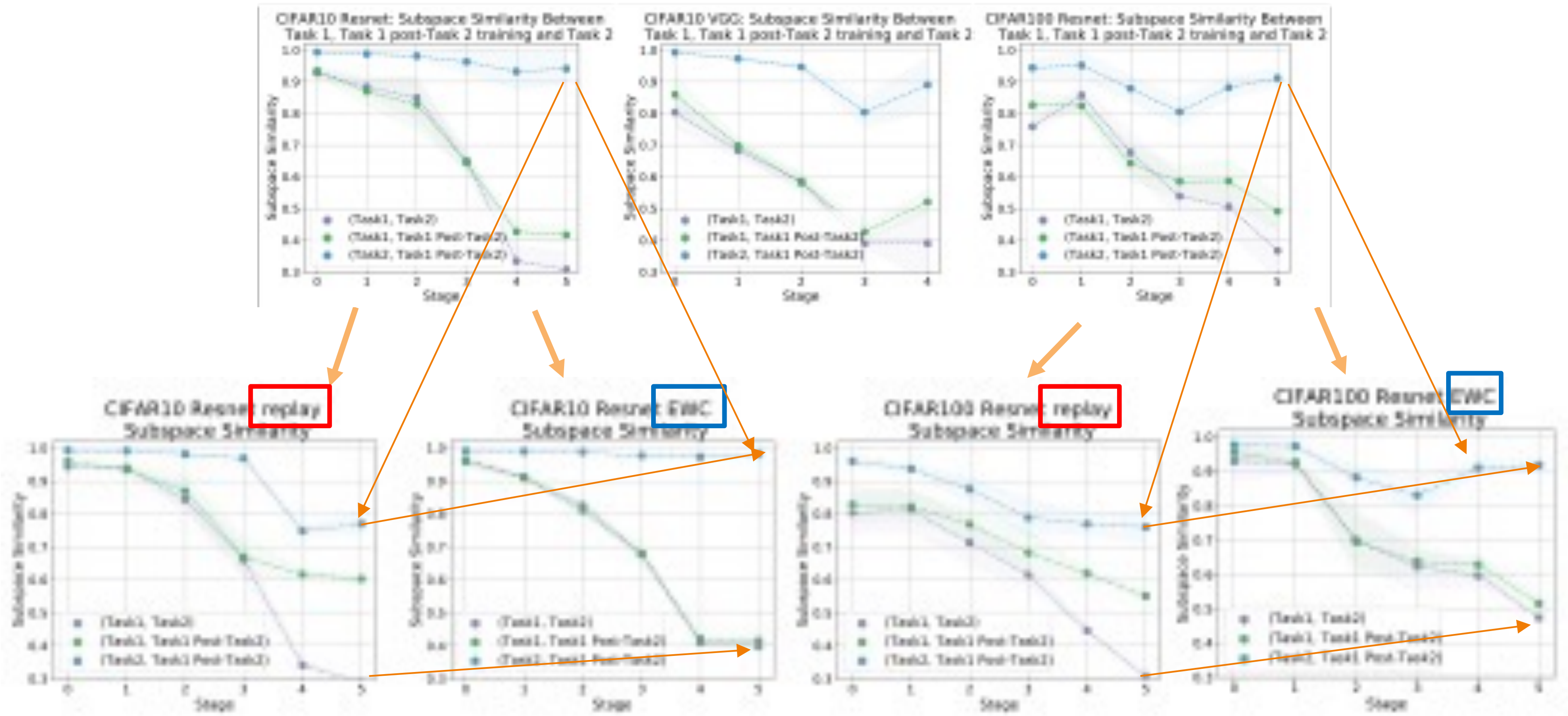
- CKA analysis
  - Measures how similar a pair of hidden layer representations are



- Compute CKA between layer representations of Task 1 **before** and **after** Task 2 training
- With varying amounts and types of mitigation.

→ Mitigation methods **stabilize** the **higher layer** representations

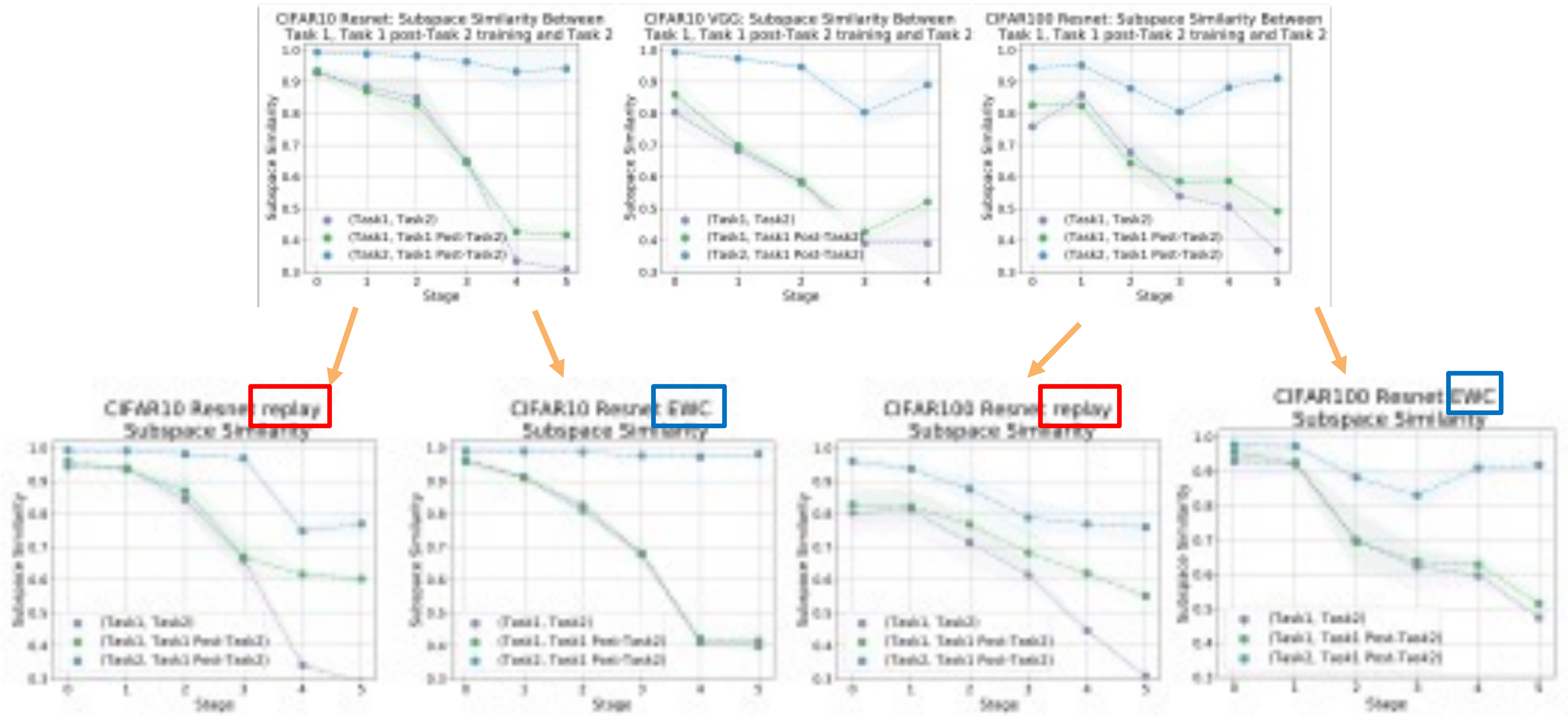
- But what about **subspace** similarity?



- (**Task 2, Task 1** post-Task 2 training) similarity is **lower** in **replay** compared to EWC and SI **regularization-based methods**
- As is (**Task1, task 2**)

Mitigation strategies and hidden representations

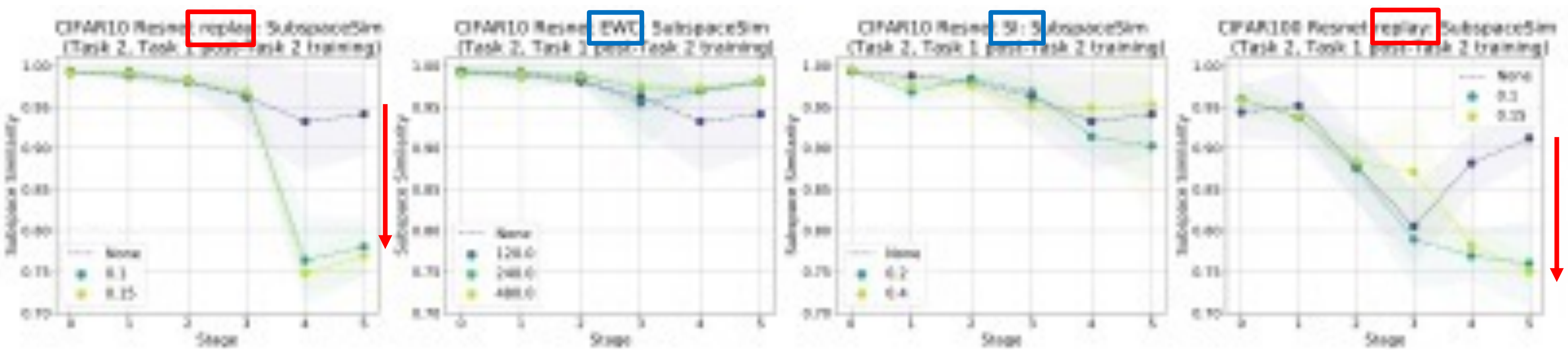
- But what about **subspace** similarity?



- Replay stores Task 1 and Task 2 representations in **orthogonal subspaces**
- EWC and SI promote feature reuse in the **higher layers**

Mitigation strategies and hidden representations

- But what about **subspace** similarity?
  - With varying degree of mitigation



- Again (Task 2, Task 1 post-Task 2 training) is **much lower** in **replay** compared to no mitigation
- When **EWC** and **SI** maintain **similar** subspaces for (Task2, Task 1 post Task 2 training)

Outline

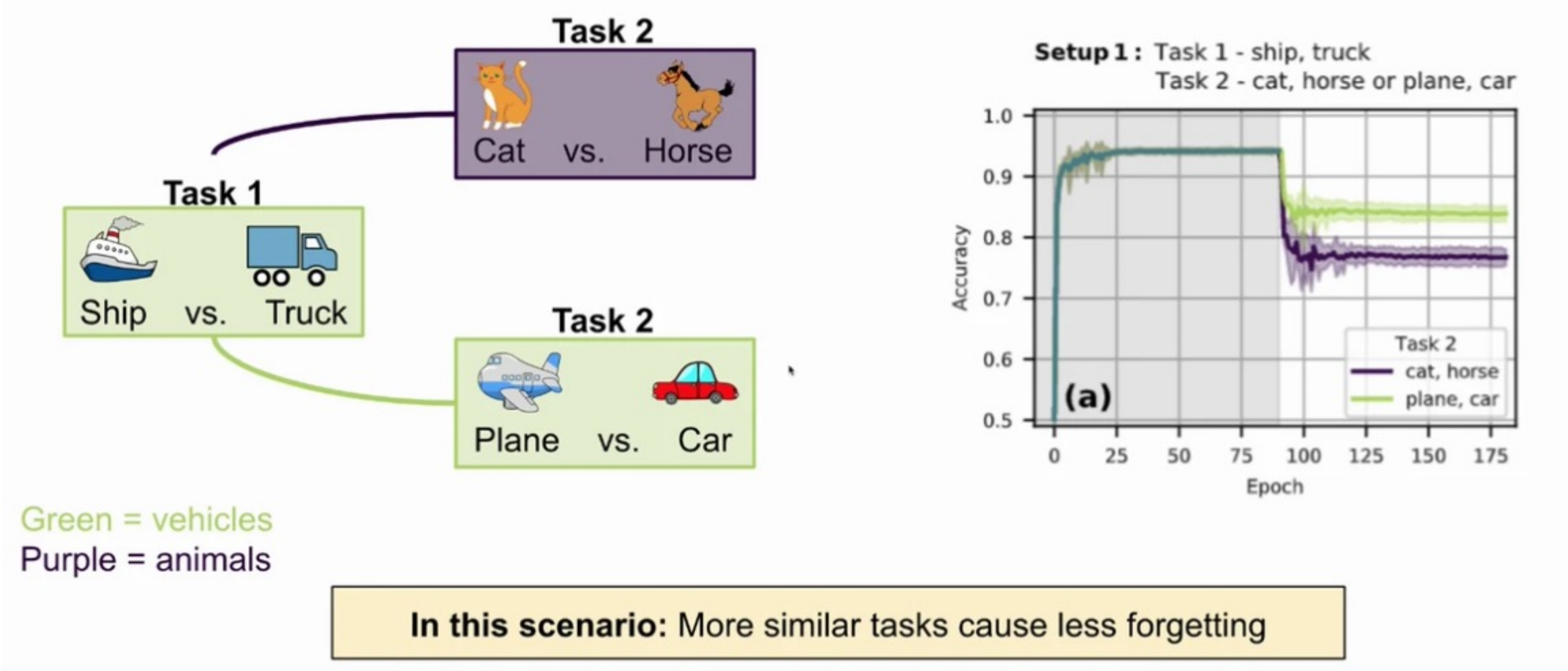
1. How to measure the difficulty of a training example
2. What is catastrophic forgetting
3. Catastrophic forgetting and hidden representations
4. Catastrophic forgetting and the semantic similarity between tasks
5. Can forgetting be useful for transfer learning?
6. Is “forgetting less” useful for transfer learning?
7. Conclusions

Catastrophic forgetting

- Questions
  - What happens to the **internal representations** of neural networks as they undergo catastrophic forgetting?
  - Does the degree to which a network forgets depend on the ***semantic similarity*** between the successive tasks?

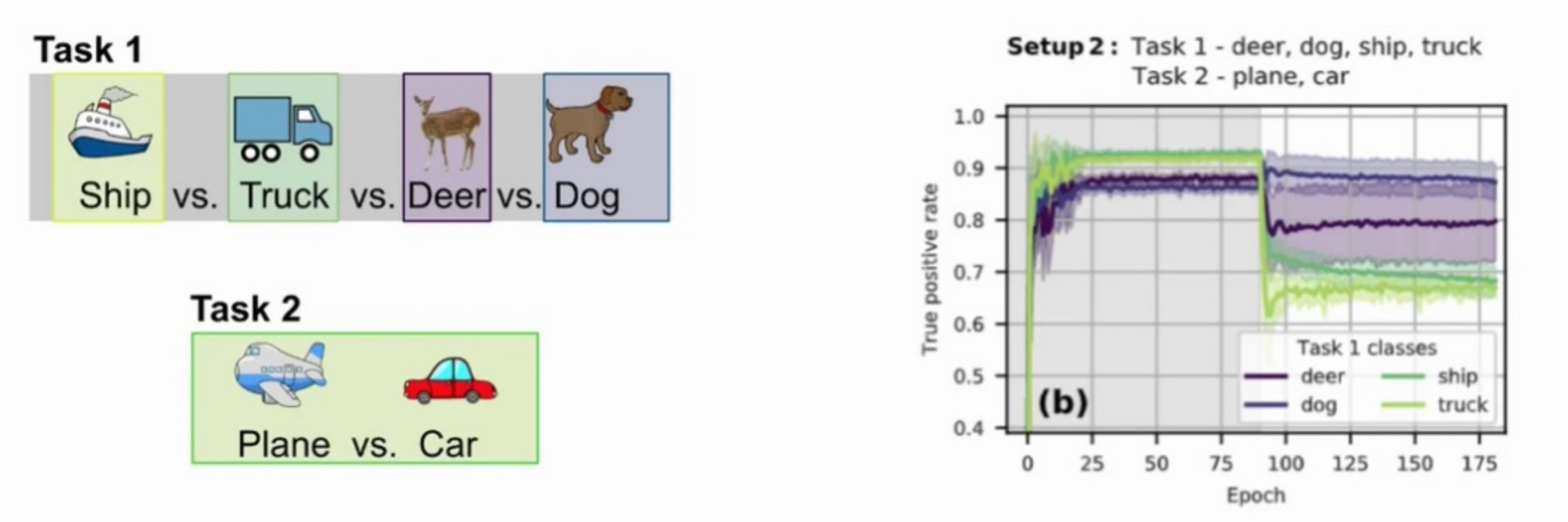
Catastrophic forgetting and **semantic similarity** between tasks

...



Catastrophic forgetting and **semantic similarity** between tasks

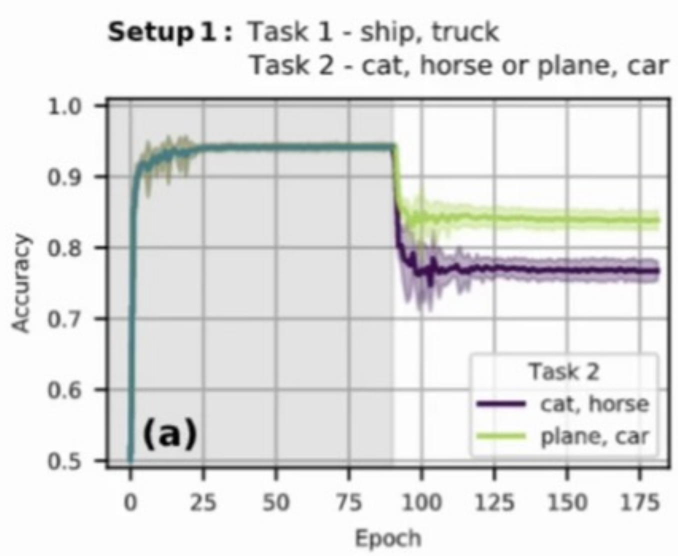
- But …



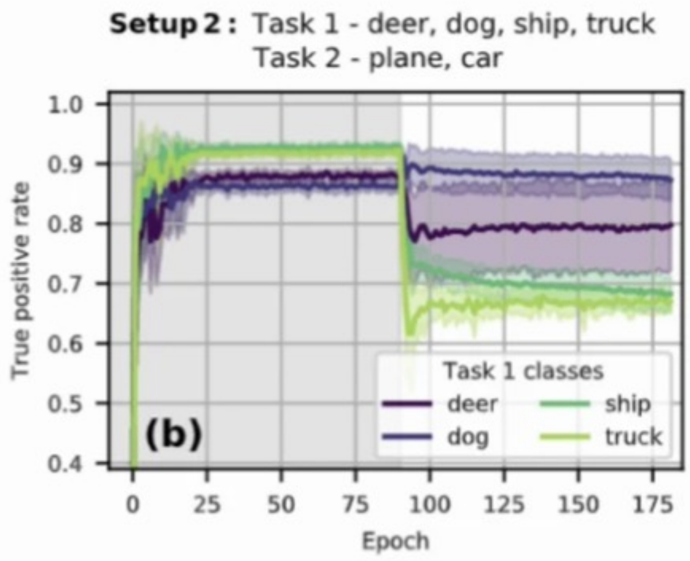
In this scenario: similar categories are forgotten more

Catastrophic forgetting and **semantic similarity** between tasks

- A **contradiction**?



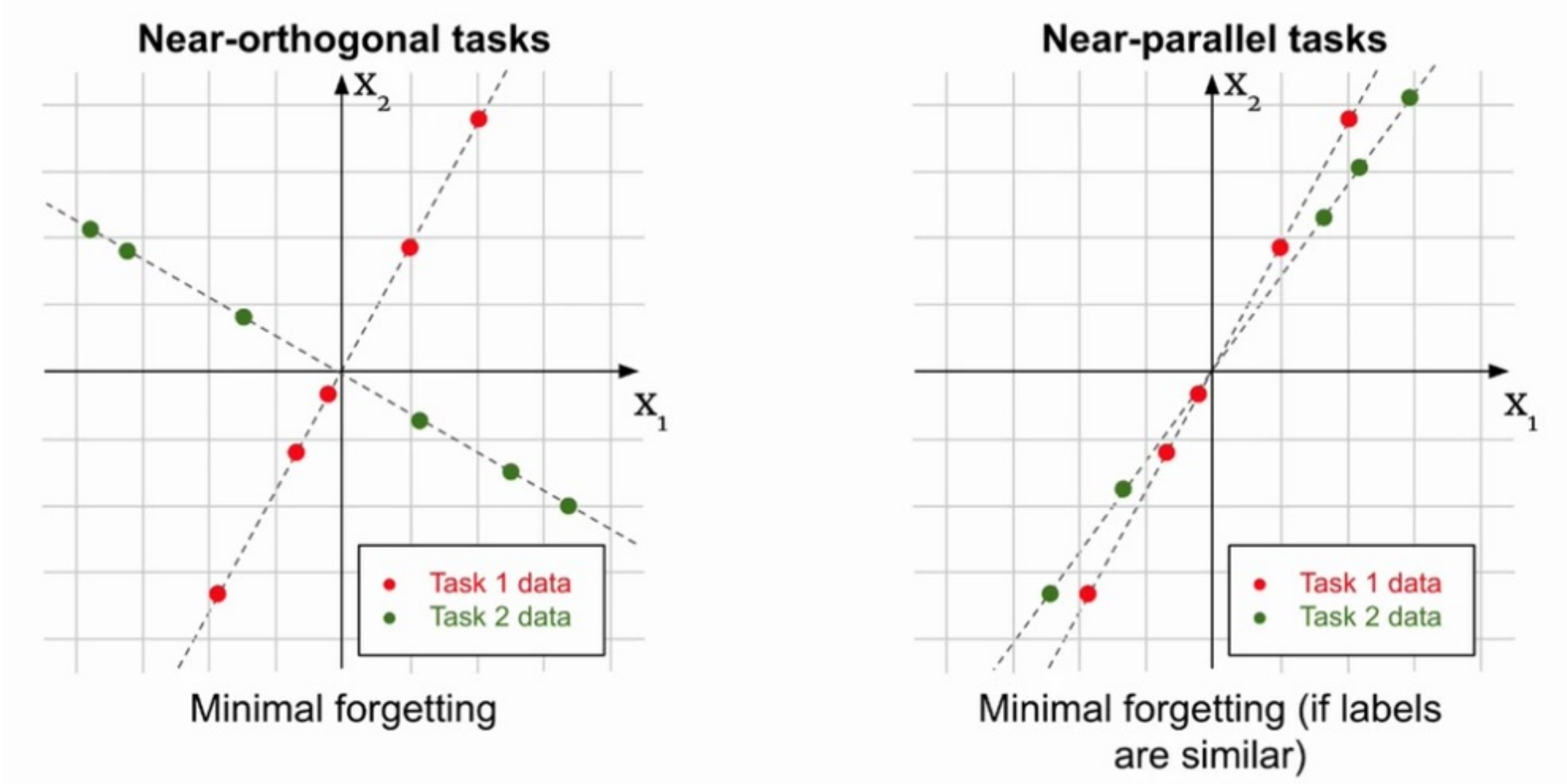
Similar tasks cause less forgetting



Similar categories are forgotten more

Catastrophic forgetting and **semantic similarity** between tasks

- Alignment of subspaces



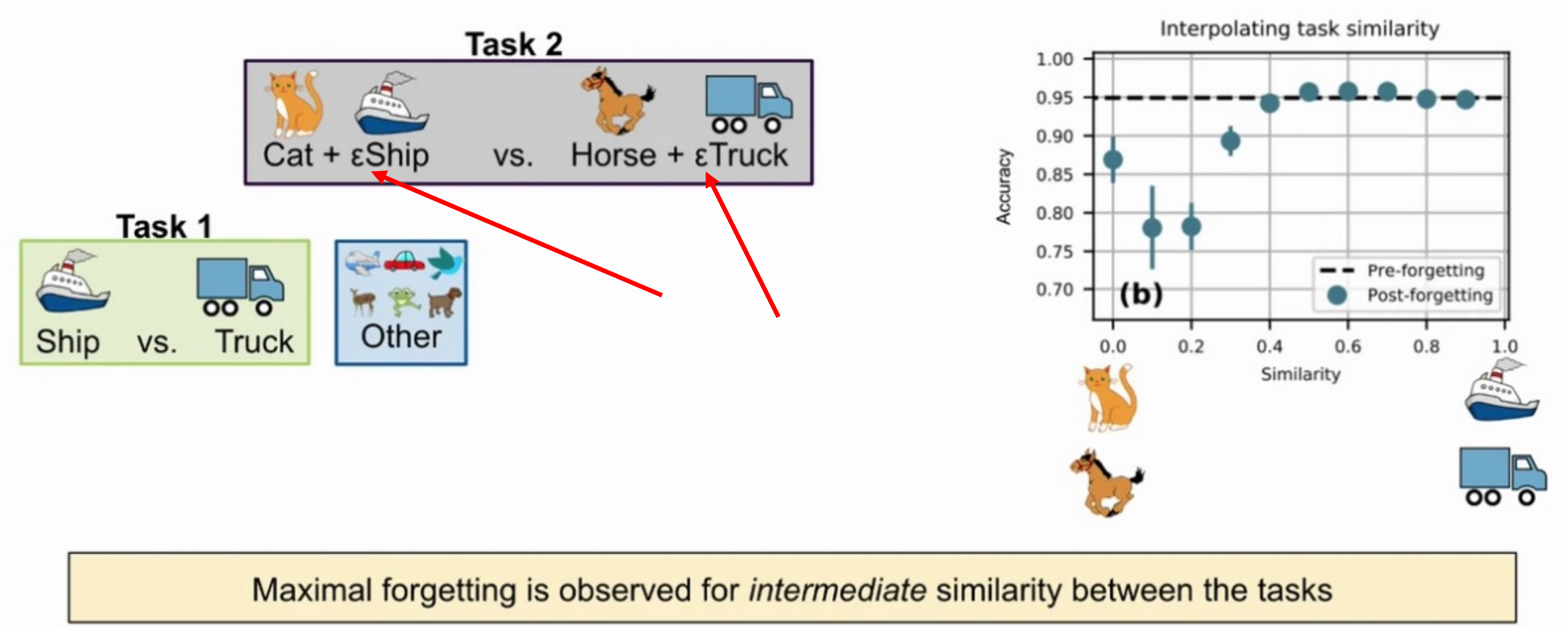
Near **orthogonal** model representations

Near **equal** model representations

Little forgetting

Catastrophic forgetting and **semantic similarity** between tasks

- …



Conclusions

- **Higher layers** are disproportionately **responsible** for catastrophic forgetting

- Different methods for **mitigating** forgetting exist
  - all **stabilize higher layer** representations,
    - But some methods encourage **greater feature reuse** in higher layers, (e.g. EWC and SI)
    - Others **store task** representations **as orthogonal subspaces**, preventing interference (e.g. REPLAY)

- **Semantic similarity** between subsequent tasks consistently **controls** the degree of forgetting
  - forgetting is **most severe** for tasks with **intermediate similarity**

Outline

1. How to measure the difficulty of a training example
2. What is catastrophic forgetting
3. Catastrophic forgetting and hidden representations
4. Catastrophic forgetting and the semantic similarity between tasks
5. Can **forgetting** be **useful** for transfer learning?
6. Is “forgetting less” useful for transfer learning?
7. Conclusions

Can **forgetting** be **useful** in transfer learning?

...

Zhou, H., Vani, A., Larochelle, H., & Courville, A. (2022).
Fortuitous forgetting in connectionist networks.
ICLR-2022.

Can **forgetting** be **useful** in transfer learning?

- "Forgetting"

Noise

$$P\big[\mathrm{Acc}\big(f(N_t, U)\big) \; < \; \mathrm{Acc}(N_t)\,|\,\mathrm{Acc}(N_t) > C\big] = 1$$

Adding noise **decreases the accuracy**, given that the accuracy was better than random

$$I\big(f(N_t, U), \mathcal{D}\big) > 0$$

Adding noise equates to a **partial removal of information** (still "aligned")

Zhou, H., Vani, A., Larochelle, H., & Courville, A. (2022). **Fortuitous forgetting in connectionist networks**. *ICLR-2022*.

Can **forgetting** be **useful** in transfer learning?

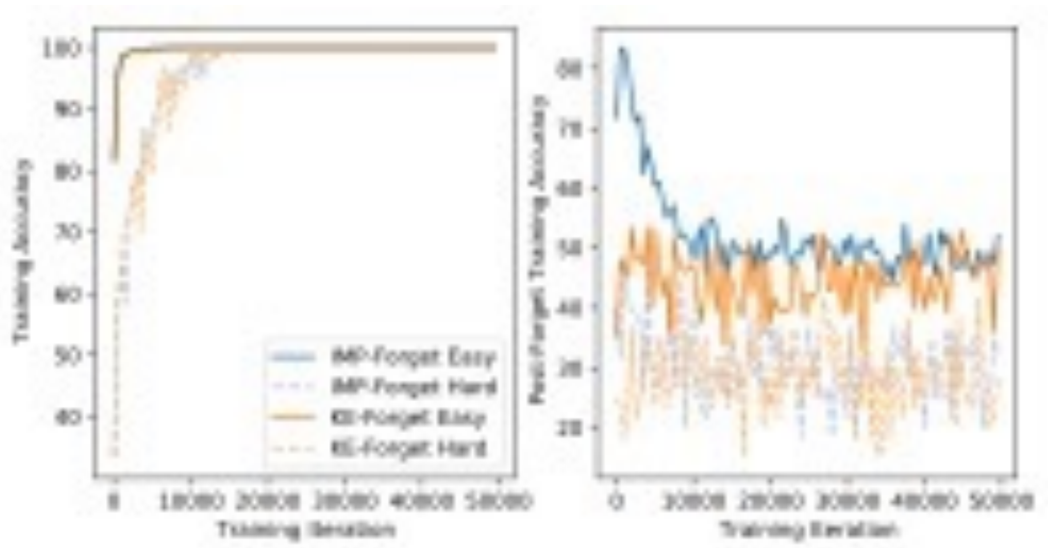
- The forget-and-relearn hypothesis
  - Given an appropriate forgetting operation, iterative **re-training AFTER forgetting** will **amplify unforgotten features** that are **consistently useful** under different learning conditions induced by the forgetting step.
  - A forgetting operation that favors the **preservation** of **desirable features** can thus be used to steer the model towards those desirable characteristics.

Many existing **algorithms** which have successfully demonstrated improved generalization **have a forgetting step** that disproportionately **affects undesirable information** for the given task.

Can **forgetting** be **useful** in transfer learning?

- Easy vs. Hard examples
  - Use the **output margin** between the **largest** and **second-largest** logits (outputs) for each example

Hard examples are **more adversely affected** than easy ones by weight perturbations



Normal training

Training with weight perturbations at each iteration

Targeted forgetting

- Later-Layer Forgetting (LLF)
  - **Reinitialization** of **later layers** at each learning iteration

Method	Flower	CUB	Aircraft	MIT	Dog
Smth (N1)	51.02 $\pm$0.09	58.92 $\pm$0.24	57.16 $\pm$0.91	56.04 $\pm$0.39	63.64 $\pm$0.16
Smth long (N3)	59.51 $\pm$0.17	66.03 $\pm$0.13	62.55 $\pm$0.25	59.53 $\pm$0.60	65.39 $\pm$0.55
Smth + KE (N3)	57.95 $\pm$0.65	63.49 $\pm$0.39	60.56 $\pm$0.36	58.78 $\pm$0.54	64.23 $\pm$0.05
Smth + LLF (N3) **(Ours)**	**63.52** $\pm$0.13	**70.76** $\pm$0.24	**68.88** $\pm$0.11	**63.28** $\pm$0.69	**67.54** $\pm$0.12

- Importance of having **variable conditions** for **refining first** layers
- Keeps and **amplifies** the **useful** features of the **first layers**

Lesson

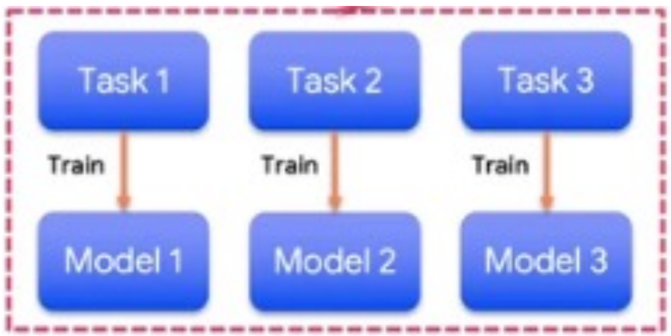
- Forgetting is **useful**
  - If it promotes the **amplification** of **useful** features in the **first layers**

Outline

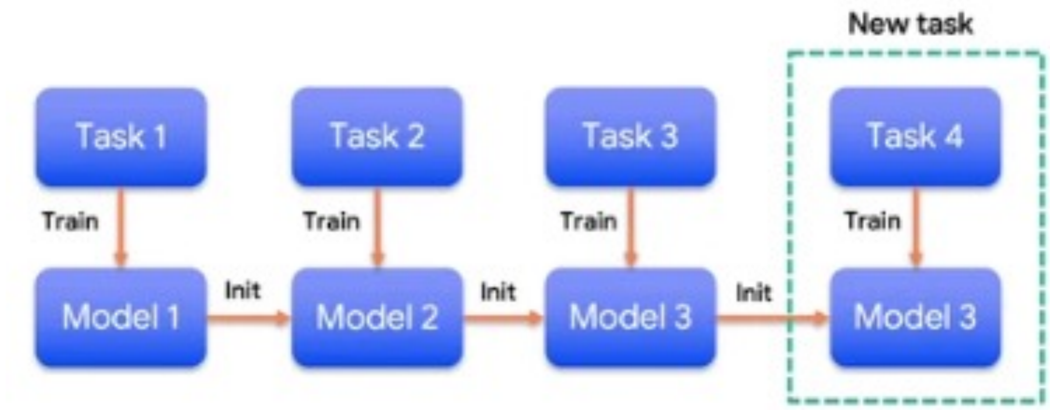
1. How to measure the difficulty of a training example
2. What is catastrophic forgetting
3. Catastrophic forgetting and hidden representations
4. Catastrophic forgetting and the semantic similarity between tasks
5. Can forgetting be useful for transfer learning?
6. Is “forgetting **less**” useful for transfer learning?
7. Conclusions

Is “**forgetting less**” good for **forward transfer**?

Training new tasks **from scratch**



Forward **transfer** learning



Chen, J., Nguyen, T., Gorur, D., & Chaudhry, A. (2023). **Is forgetting less a good inductive bias for forward transfer?** *ICLR-2023.*

Is “**forgetting less**” good for **forward transfer**?

- Claim that
  - many continual learning approaches **alleviate catastrophic forgetting** at the expense of **forward transfer**

Chen, J., Nguyen, T., Gorur, D., & Chaudhry, A. (2023). **Is forgetting less a good inductive bias for forward transfer?** *ICLR-2023*.

Is “**forgetting less**” good for **forward transfer**?

- Claim that
  - many continual learning approaches **alleviate catastrophic forgetting** at the expense of **forward transfer**

In which situation is it necessary to forget?

Is "**forgetting less**" good for **forward transfer**?

- They measure forward transfer in terms of **how *easy* it is to learn a new task** given continually trained **representations**
- The **easiness** is measured by learning a linear classifier on top of the *fixed* representations using a small subset of the data of the new task

Is "**forgetting less**" good for **forward transfer**?

- They measure forward transfer in terms of **how *easy* it is to learn a new task** given continually trained **representations**
- The **easiness** is measured by learning a linear classifier on top of the *fixed* representations using a small subset of the data of the new task

Remark: they say that this appropriate when considering **foundation models**

Is “**forgetting less**” good for **forward transfer**?

- They measure forward transfer in terms of **how *easy* it is to learn a new task** given continually trained **representations**
- The **easiness** is measured by learning a linear classifier on top of the *fixed* representations using a small subset of the data of the new task

Remark: they say that this appropriate when considering **foundation models**

Because we **finetune** them in order to address new tasks

Is “**forgetting less**” good for **forward transfer**?

- They measure forward transfer in terms of **how *easy* it is to learn a new task** given continually trained representations
- The **easiness** is measured by learning a **linear classifier** on top of the *fixed* representations using a small subset of the data of the new task

Linear classifier

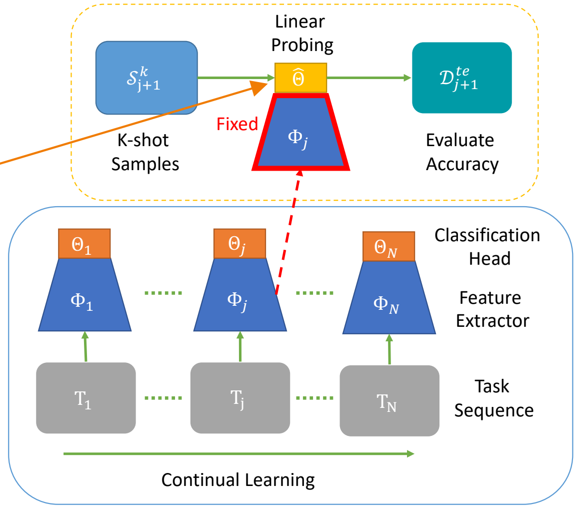
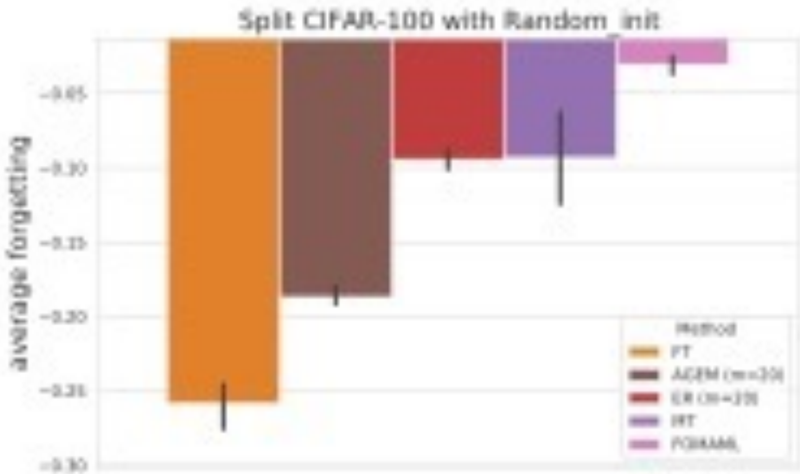


Figure 2: Illustration of continual learning and k-shot evaluation process. We continuously train the feature extractor and the classification head on a task sequence $T_1, \ldots, T_N$. $\Theta_j \circ \Phi_j$ is the model obtained after training on T_j. To evaluate the forward transfer of Φ_j, we use linear probing on k-shot samples from the next task T_{j+1} to learn a classifier $\hat{\Theta}$ and then evaluate the accuracy of $\hat{\Theta} \circ \Phi_j$ on the test set $\mathcal{D}_{j+1}^{te}$ from the task T_{j+1}.

Is “**forgetting less**” good for **forward transfer**?

YES!



(a) Average Forgetting (↑ better)

(b) Average Forward Transfer (↑ better)

- **Less** forgetting leads to **better** transfer learning
- Less forgetful models result in **more diverse** and **easily separable** representations

Is "**forgetting less**" good for **forward transfer**?

- Measure how **diverse** and **easily separable** are the features learned in Φ_j

$$\texttt{FDiv}_j = \log|\alpha \Psi_j^\top \Psi_j + \mathbf{I}| - \sum_{c=1}^{C_j} \log|\alpha_j \Psi_j^{c\top} \Psi_j^c + \mathbf{I}|$$

where $|\cdot|$ is a matrix determinant operator, $\alpha = D/(m\varepsilon^2)$, $\alpha_j = D/(m_j\varepsilon^2)$, $\varepsilon = 0.5$, and C_j denotes the number of classes for task 'j'.

Is "**forgetting less**" good for **forward transfer**?

- Measure how **diverse** and **easily separable** are the features learned in Φ_j

$$\texttt{FDiv}_j = \log\left|\alpha \Psi_j^\top \Psi_j + \mathbf{I}\right| - \sum_{c=1}^{C_j} \log\left|\alpha_j {\Psi_j^c}^\top \Psi_j^c + \mathbf{I}\right|$$

where $|\cdot|$ is a matrix determinant operator, $\alpha = D/(m\varepsilon^2)$, $\alpha_j = D/(m_j\varepsilon^2)$, $\varepsilon = 0.5$, and C_j denotes the number of classes for task 'j'.

Hypothesis: **less forgetful representations** maintain **more diversity** and **discrimination** in the features making it easy to learn a classifier head on top leading to better forward transfer

Is "**forgetting less**" good for **forward transfer**?

Average **forgetting** Average **diversity**

Dataset	Method	Random Init		Pre-trained	
		AvgFgt ↑	AvgFDiv ↑	AvgFgt ↑	AvgFDiv ↑
Split CIFAR-10	FT	-28.18 ± 2.97	35.59 ± 10.52	-29.01 ± 7.97	60.18 ± 36.35
	LP-FT	-	-	-3.39 ± 1.06	**171.41** ± 13.41
	ER (m=50)	-9.18 ± 1.50	37.33 ± 14.66	-7.15 ± 1.97	66.18 ± 35.74
	AGEM (m=50)	-13.77 ± 2.38	35.79 ± 16.34	-19.26 ± 5.01	60.77 ± 41.80
	MT	-3.88 ± 5.86	36.88 ± 13.21	-4.83 ± 5.56	86.88 ± 21.82
	FOMAML	**-0.75** ± 1.39	**45.52** ± 7.82	-1.40 ± 0.61	65.26 ± 10.36
Split CIFAR-100	FT	-25.83 ± 2.43	224.27 ± 3.63	-24.33 ± 4.19	263.31 ± 27.46
	LP-FT	-	-	-4.46 ± 0.46	**332.10** ± 2.97
	ER (m=20)	-9.44 ± 1.11	225.95 ± 2.38	-9.19 ± 0.28	281.31 ± 3.59
	AGEM (m=20)	-18.70 ± 1.00	224.46 ± 2.93	-20.05 ± 3.12	260.01 ± 20.32
	MT	-9.35 ± 4.96	225.33 ± 4.62	-7.93 ± 4.04	277.14 ± 8.31
	FOMAML	**-3.05** ± 0.98	225.87 ± 5.31	-4.40 ± 0.20	271.56 ± 7.45
CIFAR-100 Superclasses	FT	-14.45 ± 1.02	458.73 ± 12.99	-13.51 ± 0.56	599.29 ± 13.65
	LP-FT	-	-	-2.66 ± 0.53	**702.43** ± 4.10
	ER (m=5)	-11.33 ± 1.79	463.78 ± 7.86	-11.36 ± 1.44	600.23 ± 23.86
	AGEM (m=5)	-12.28 ± 0.84	459.65 ± 14.52	-12.11 ± 0.76	594.70 ± 27.51
	MT	1.30 ± 4.02	465.47 ± 7.84	-5.50 ± 3.65	601.38 ± 16.92
	FOMAML	**1.99** ± 0.76	**470.27** ± 5.17	-1.24 ± 0.44	620.66 ± 10.36

Less forgetting generally leads to **representations** that have **higher** AvgFDiv score, both for *randomly initialized* and for *pre-trained* models

Is “**forgetting less**” good for **forward transfer**?

- Here, no difference is made between **layers**
- But it emphasizes the **beneficial** role of **diversity** in the **features** learned in each learning task

Outline

1. How to measure the difficulty of a training example
2. What is catastrophic forgetting
3. Catastrophic forgetting and hidden representations
4. Catastrophic forgetting and the semantic similarity between tasks
5. Can forgetting be useful for transfer learning?
6. Is “forgetting less” useful for transfer learning?
7. Conclusions

Conclusions

- **Better** transfer
  - **If** the tasks are **orthogonal** or **similar** (as measured by PCA on the subspaces)
  - **If** the learnt **features** (in the first layers) are **diverse** and useful in general (for different tasks)

→ Devise **algorithms** that promote that

Bibliography

- Baldock, R., Maennel, H., & Neyshabur, B. (2021). **Deep learning through the lens of example difficulty**. *Advances in Neural Information Processing Systems*, *34*.
- Bauer, M., Klassen, E., Preston, S. C., & Su, Z. (2018). **A diffeomorphism-invariant metric on the space of vector-valued one-forms**. arXiv preprint arXiv:1812.10867.
- Chen, J., Nguyen, T., Gorur, D., & Chaudhry, A. (2023). Is forgetting less a good inductive bias for forward transfer?. *ICLR-2023*.
- Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., & Kanan, C. (2020, August). **Remind your neural network to prevent catastrophic forgetting**. In *European Conference on Computer Vision* (pp. 466-483). Springer, Cham.
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). **Overcoming catastrophic forgetting in neural networks**. *Proceedings of the national academy of sciences*, *114*(13), 3521-3526.
- RAMASESH, Vinay V., DYER, Ethan, et RAGHU, Maithra (2021). **Anatomy of catastrophic forgetting: Hidden representations and task semantics**. *ICLR-2021.*
- H. Venkateswara, S. Chakraborty, and S. Panchanathan, "Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 117–129, 2017.
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. In *Advances in neural information processing systems* (pp. 3320-3328).
- Zhou, H., Vani, A., Larochelle, H., & Courville, A. (2022). Fortuitous forgetting in connectionist networks. *ICLR-2022*.
- Zhang, C., Zhang, L., & Ye, J. (2012). Generalization bounds for domain adaptation. In *Advances in neural information processing systems* (pp. 3320-3328).