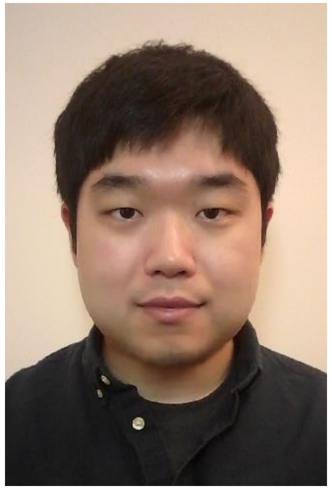
Preserving Linear Separability in Continual Learning by Backward Feature Projection



Qiao Gu
University of Toronto

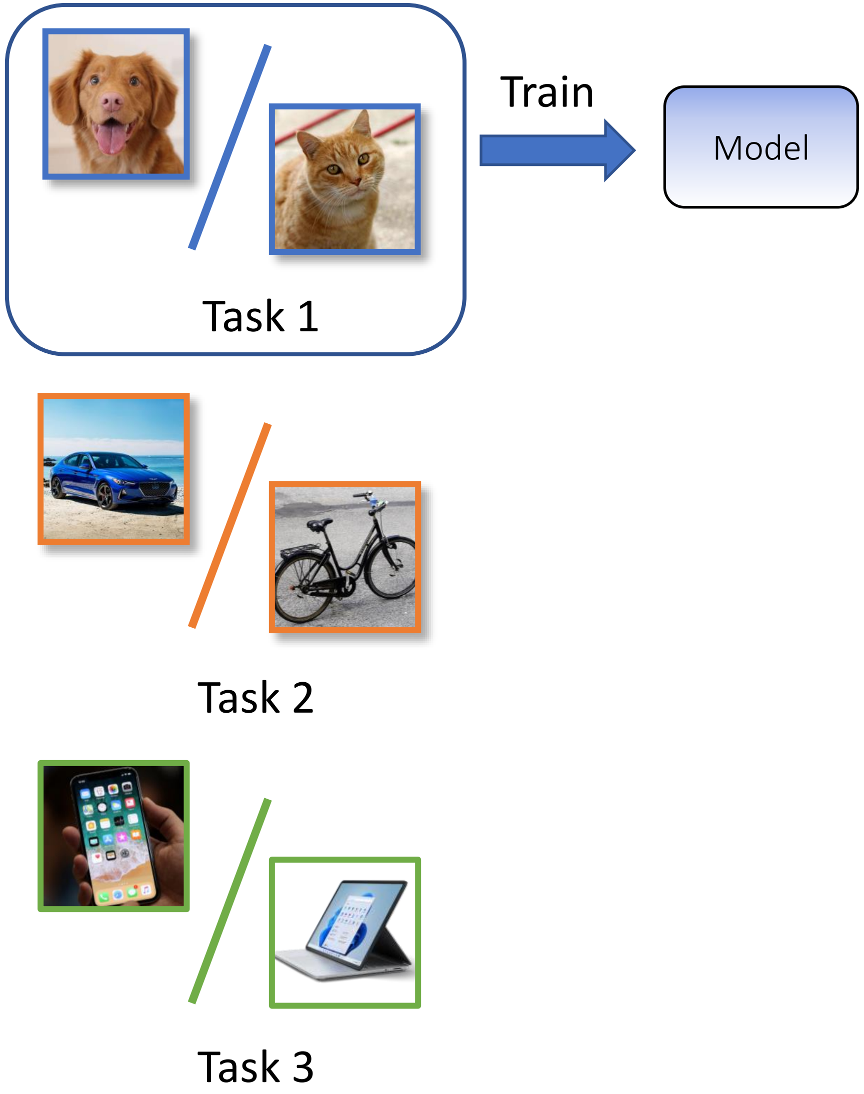
Dongsub Shim
LG AI Research

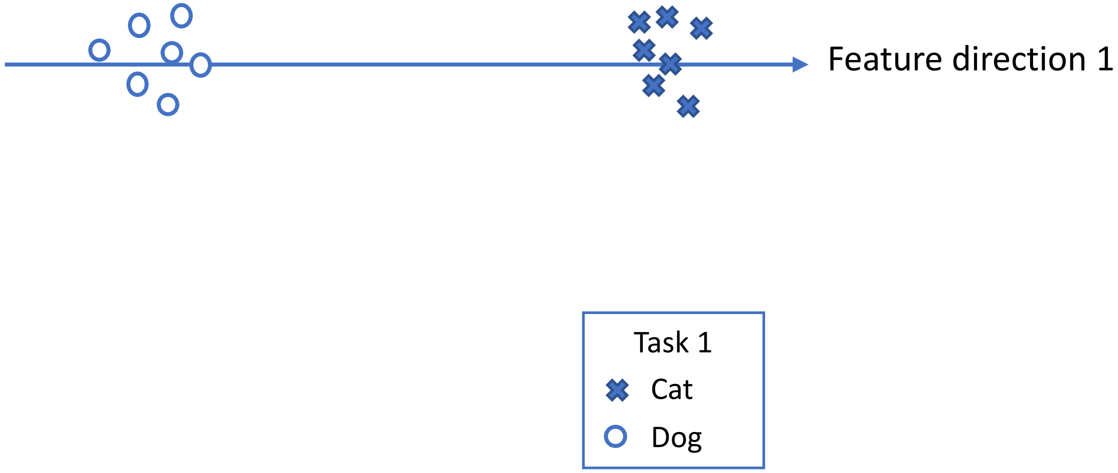
Florian Shkurti
University of Toronto

CVPR 2023
THU-PM-352

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

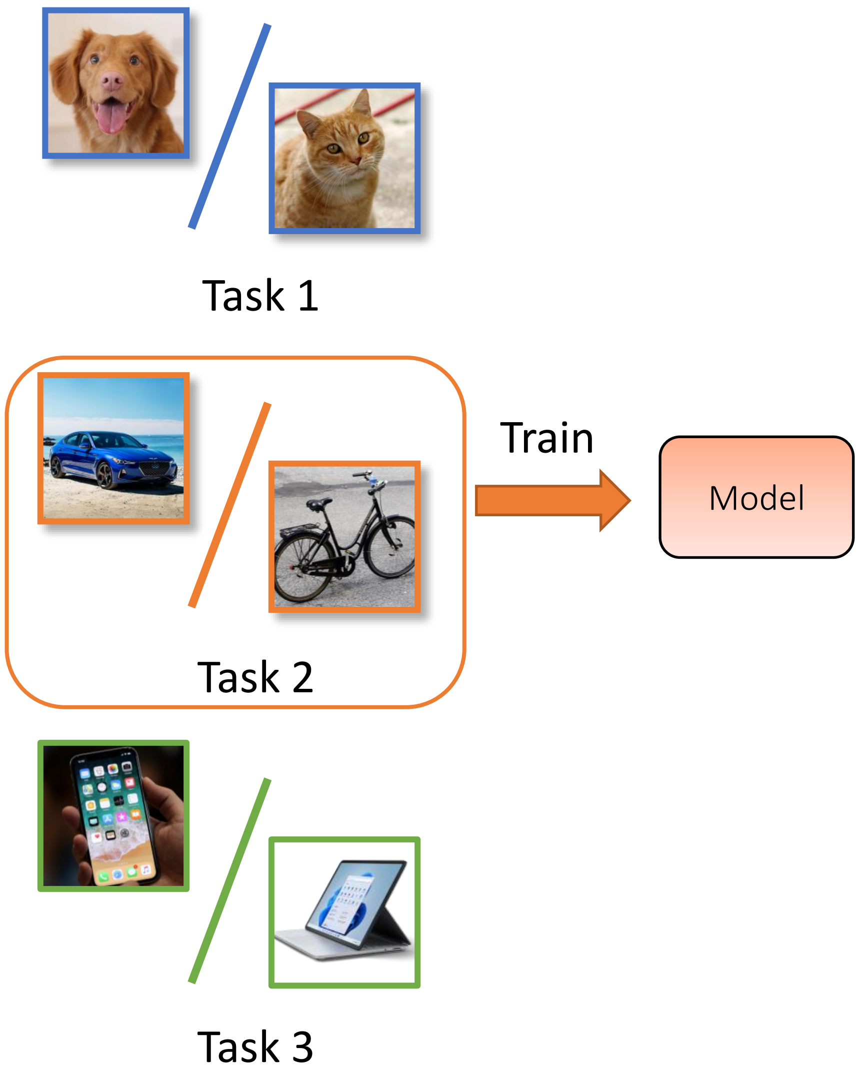
Class-Incremental Continual Learning

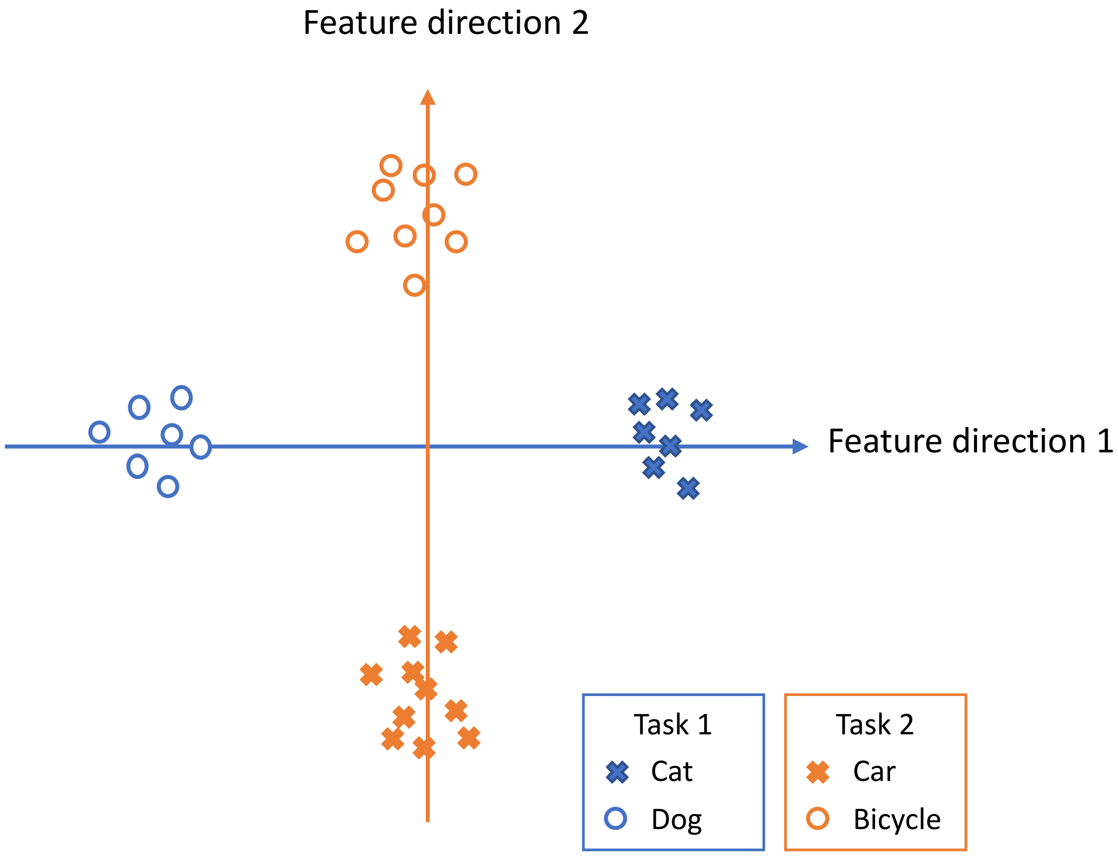




Learned feature space

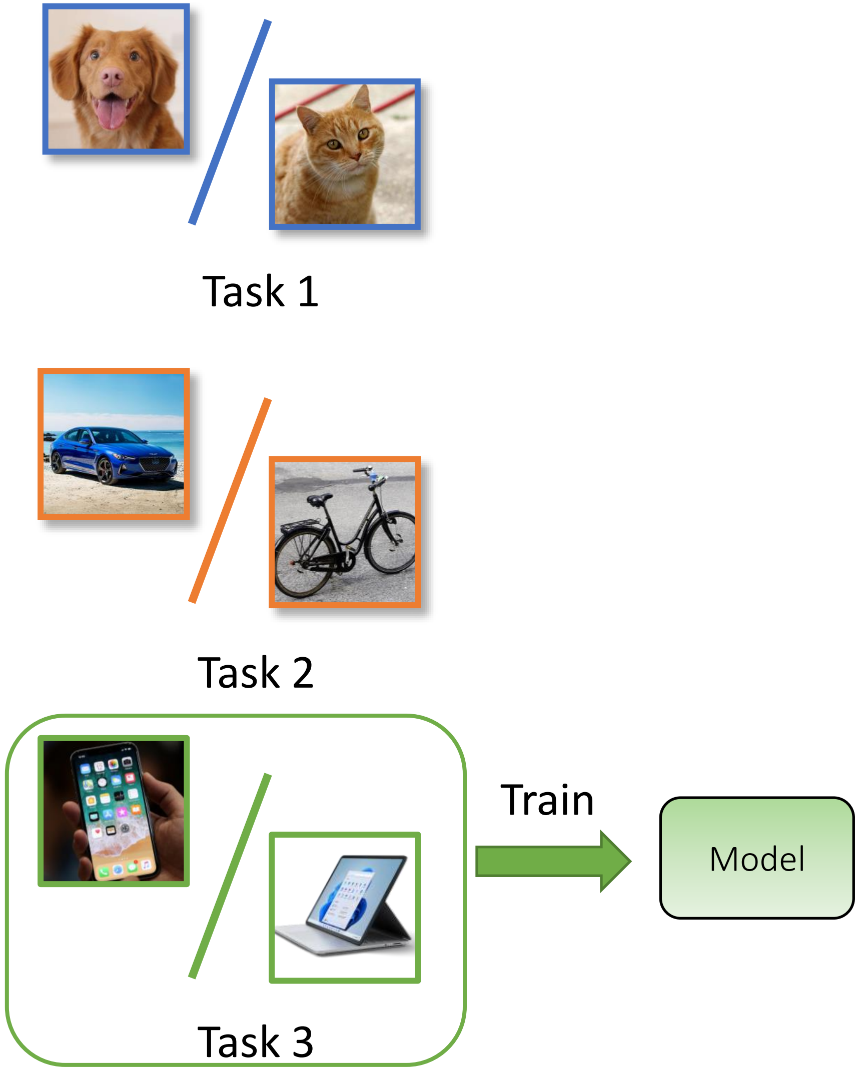
Class-Incremental Continual Learning

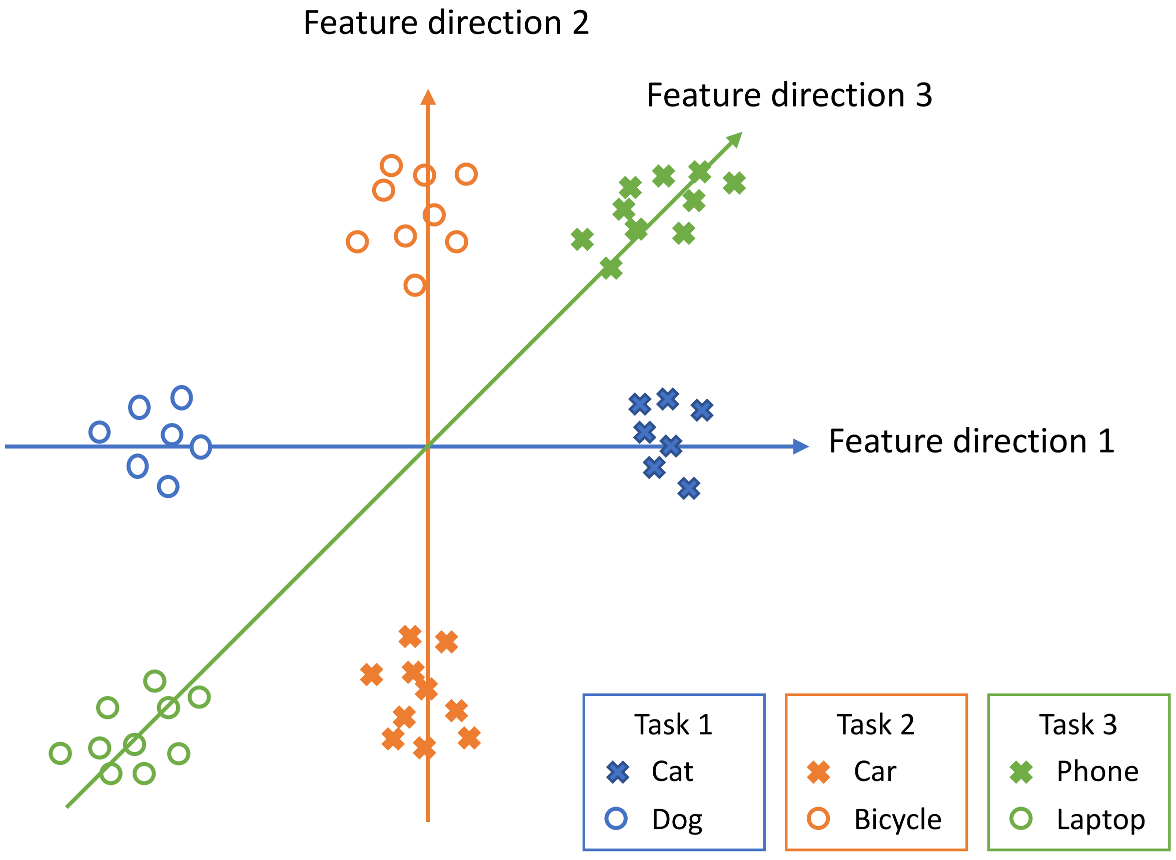




Learned feature space

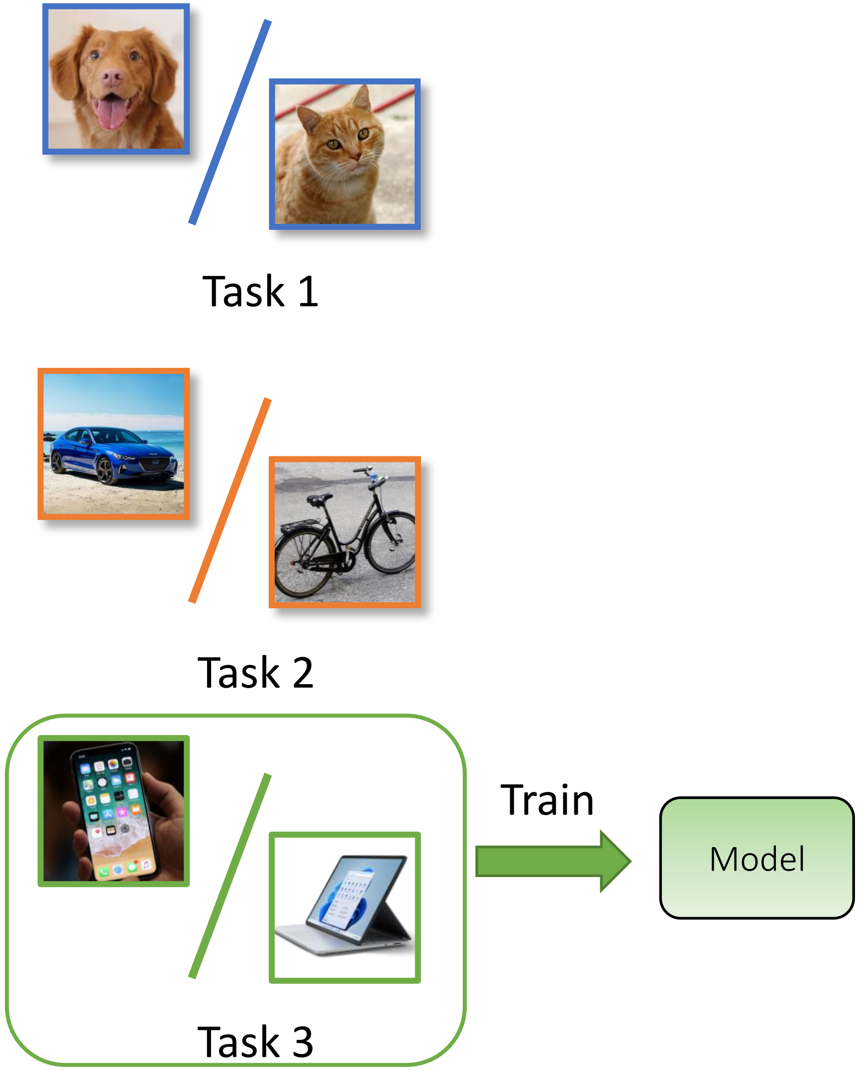
Class-Incremental Continual Learning

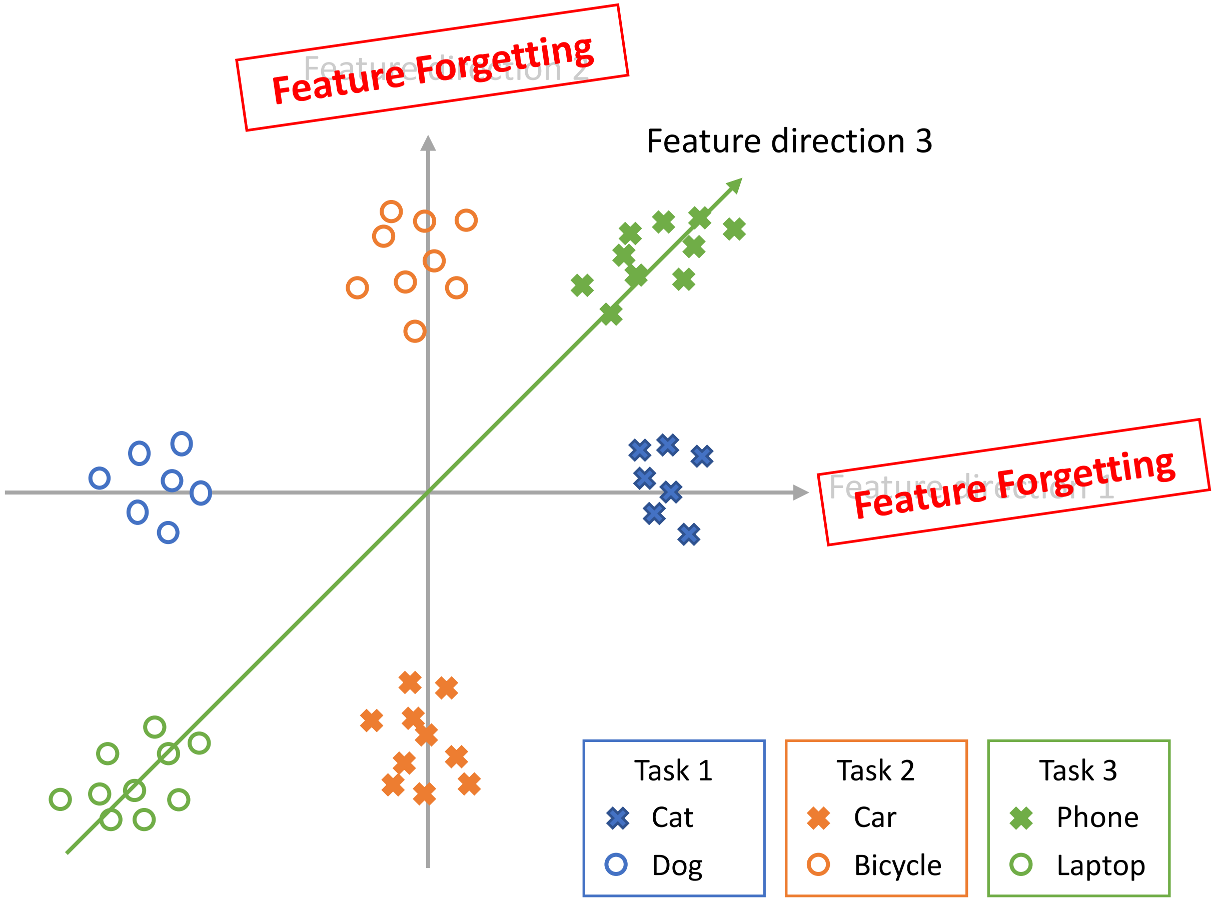




Learned feature space

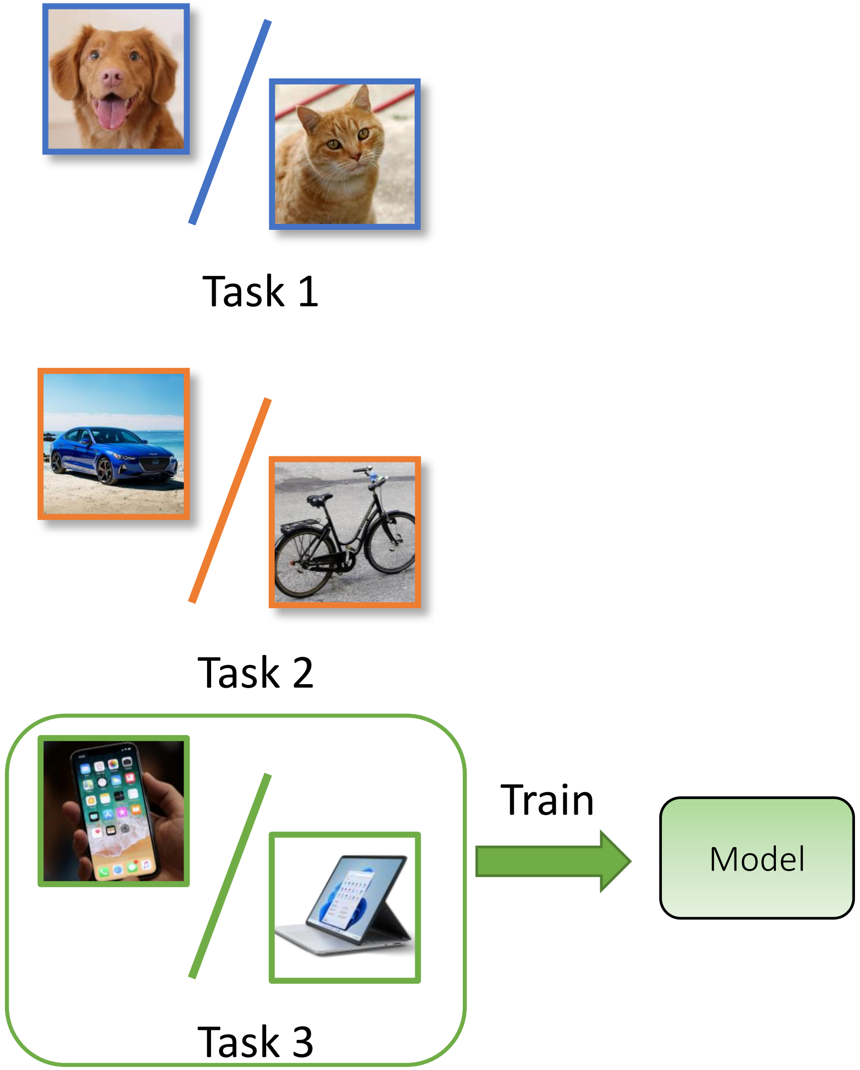
Class-Incremental Continual Learning

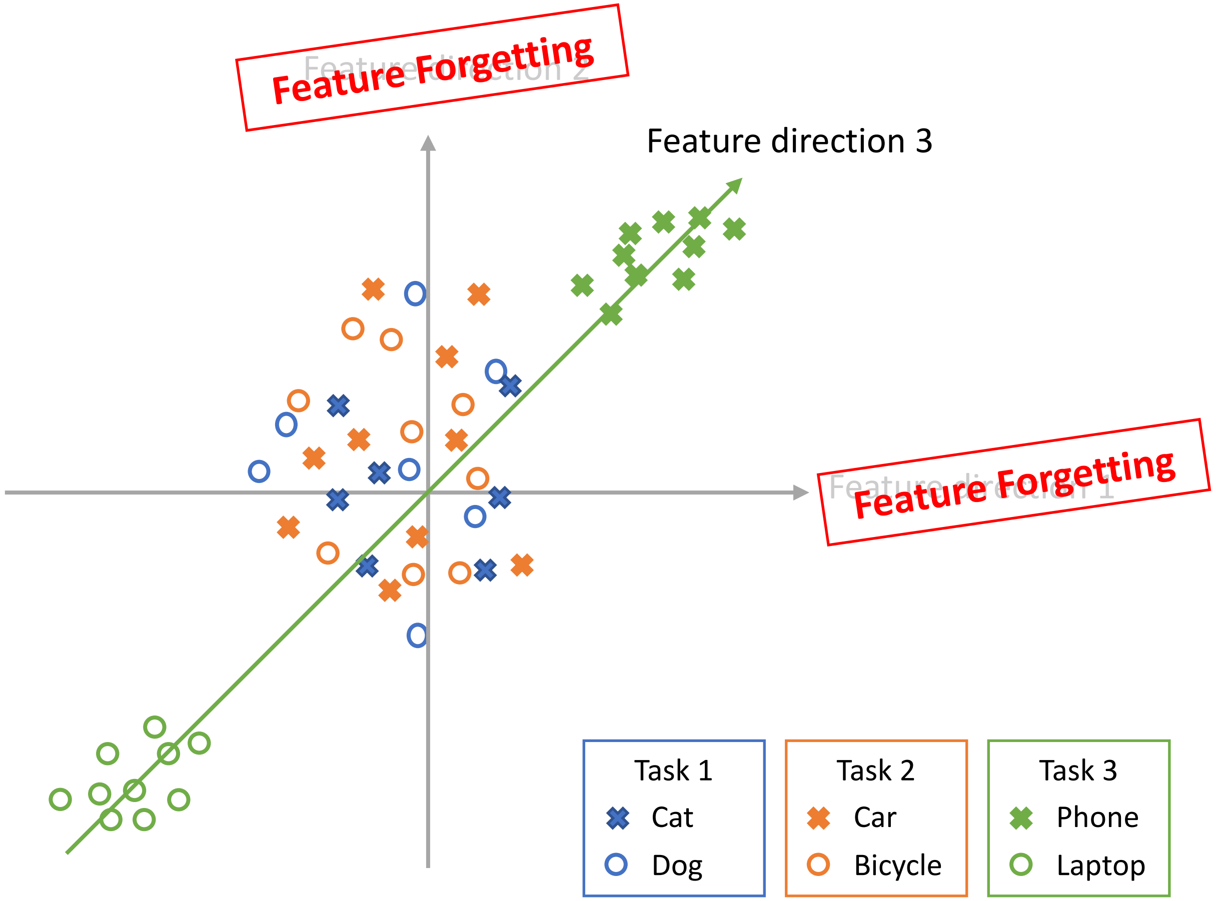




Learned feature space

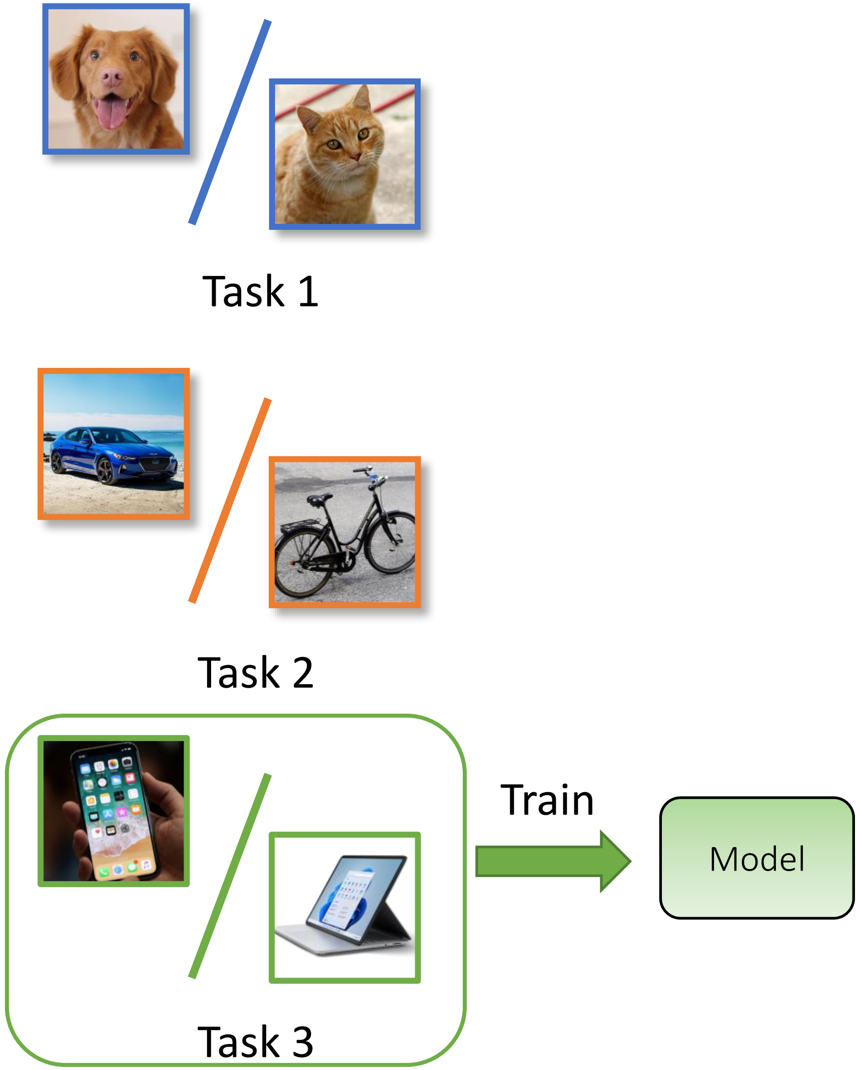
Class-Incremental Continual Learning

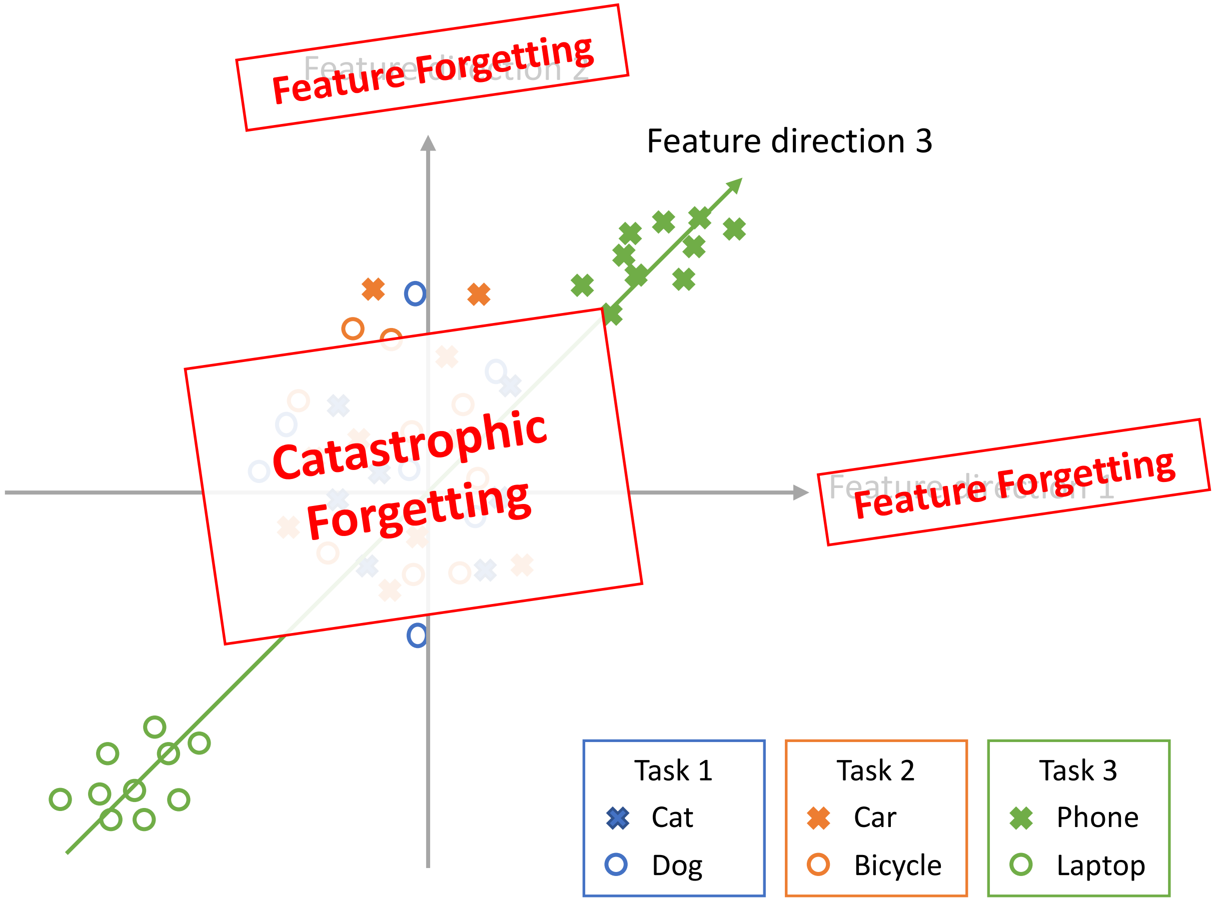




Learned feature space

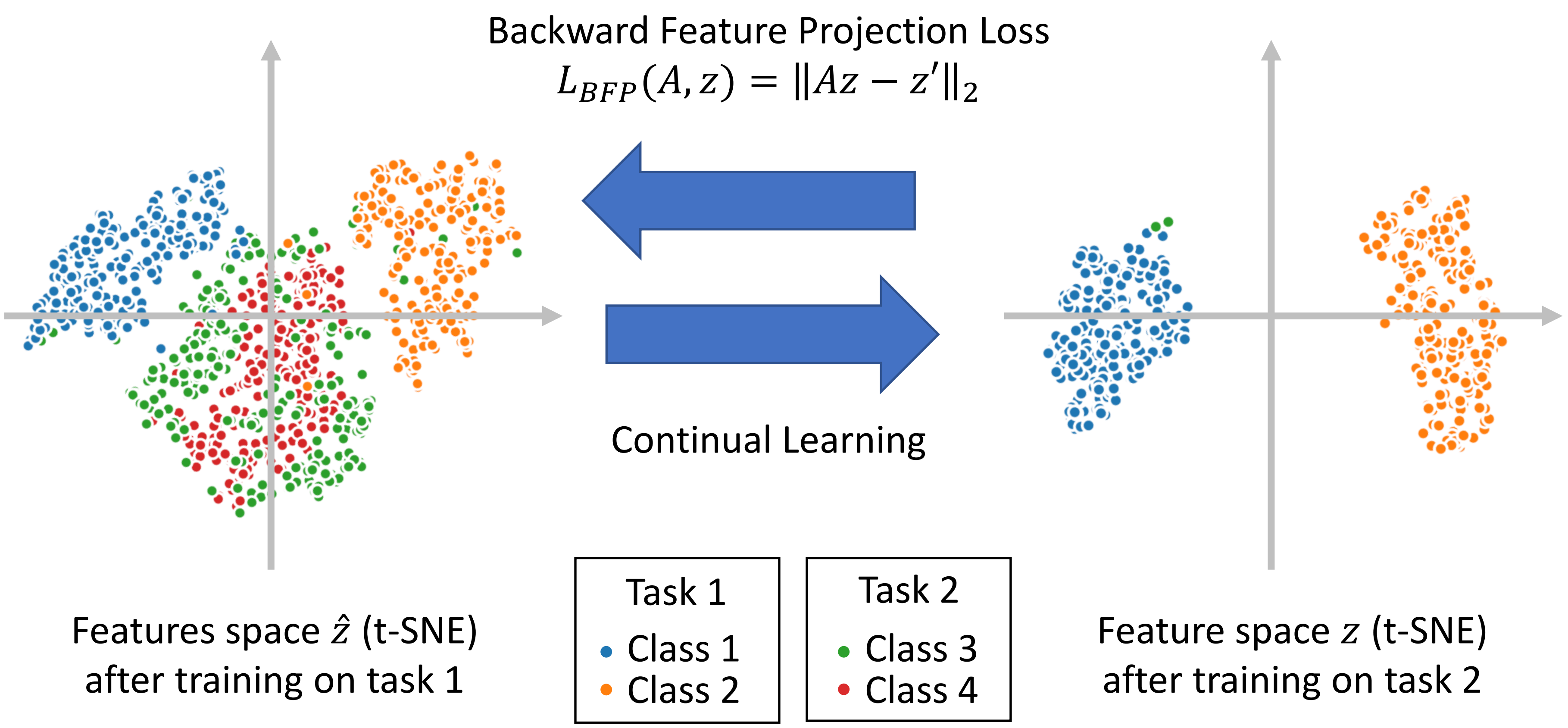
Class-Incremental Continual Learning



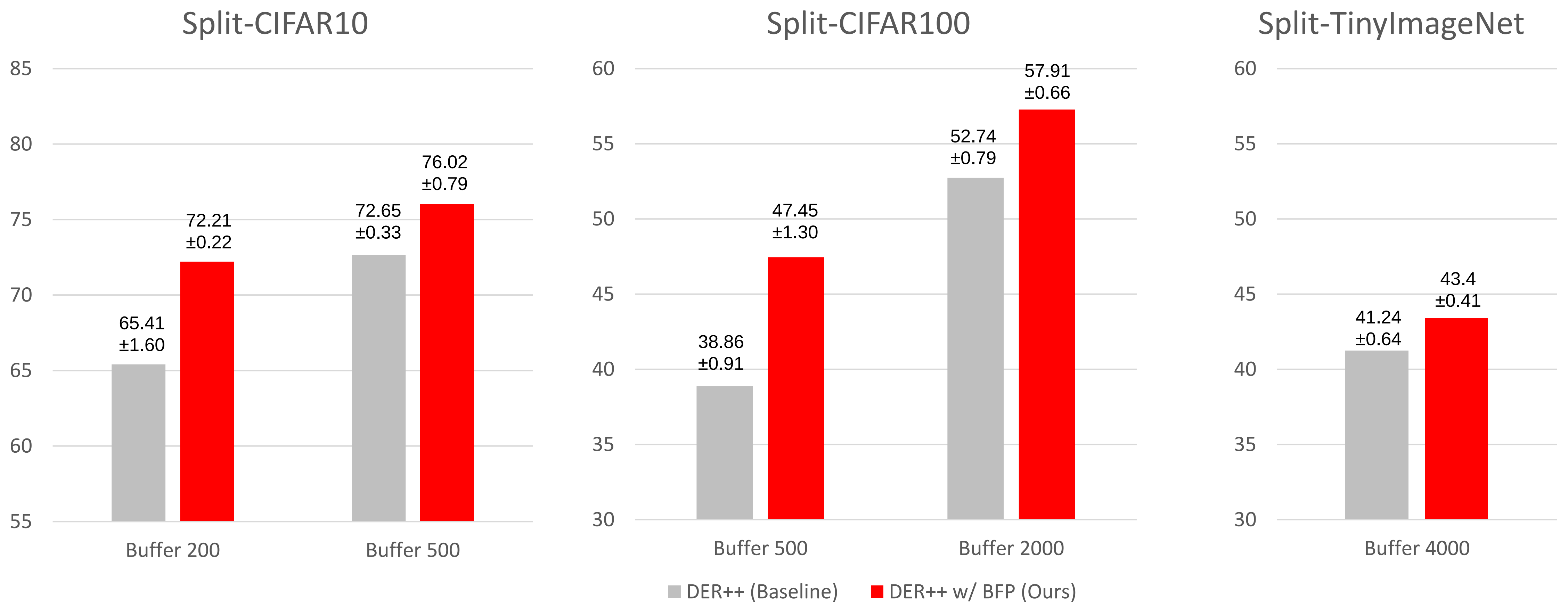


Learned feature space

BFP: Backward Feature Projection



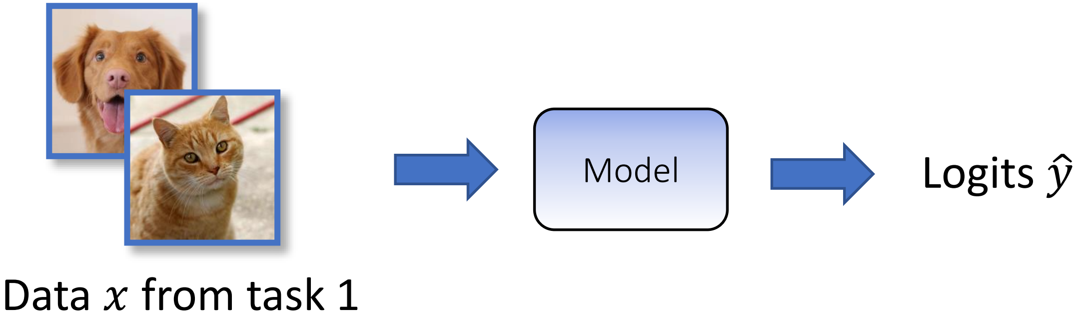
BFP brings significant performance boosts



[1] Buzzega, Pietro, et al. "Dark experience for general continual learning: a strong, simple baseline." NeurIPS 2020

How does feature space evolve in CL?

- After training on task 1

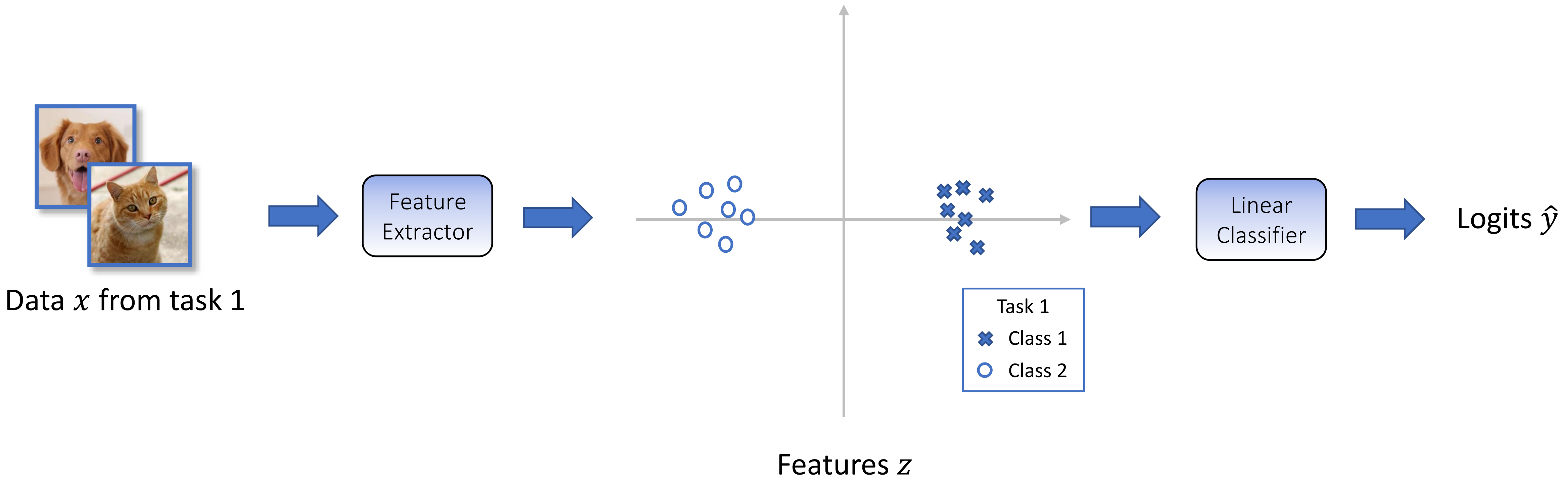


How does feature space evolve in CL?

- After training on task 1

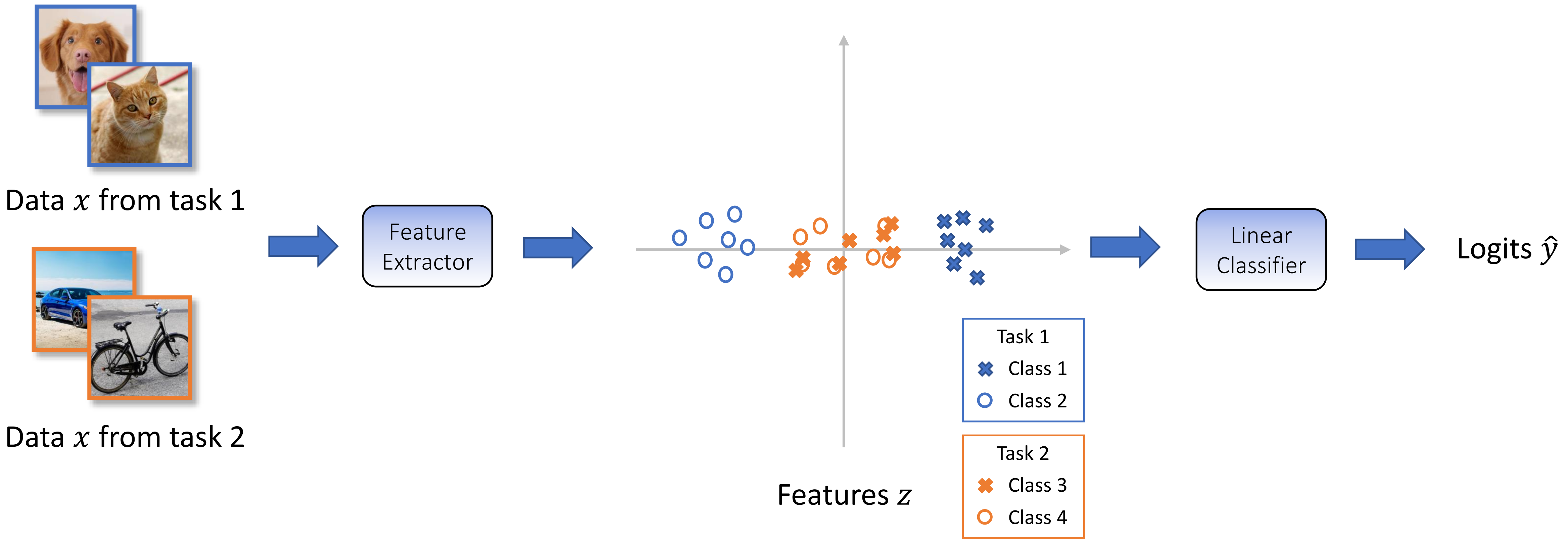
How does feature space evolve in CL?

- After training on task 1



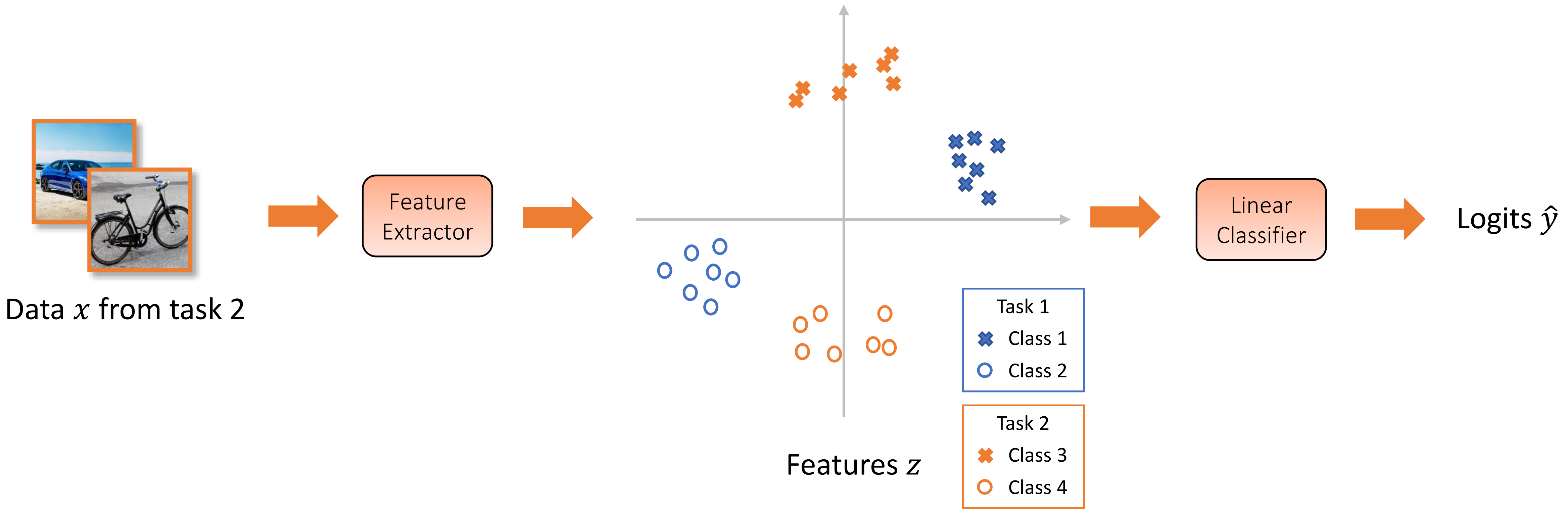
How does feature space evolve in CL?

- Before training on task 2

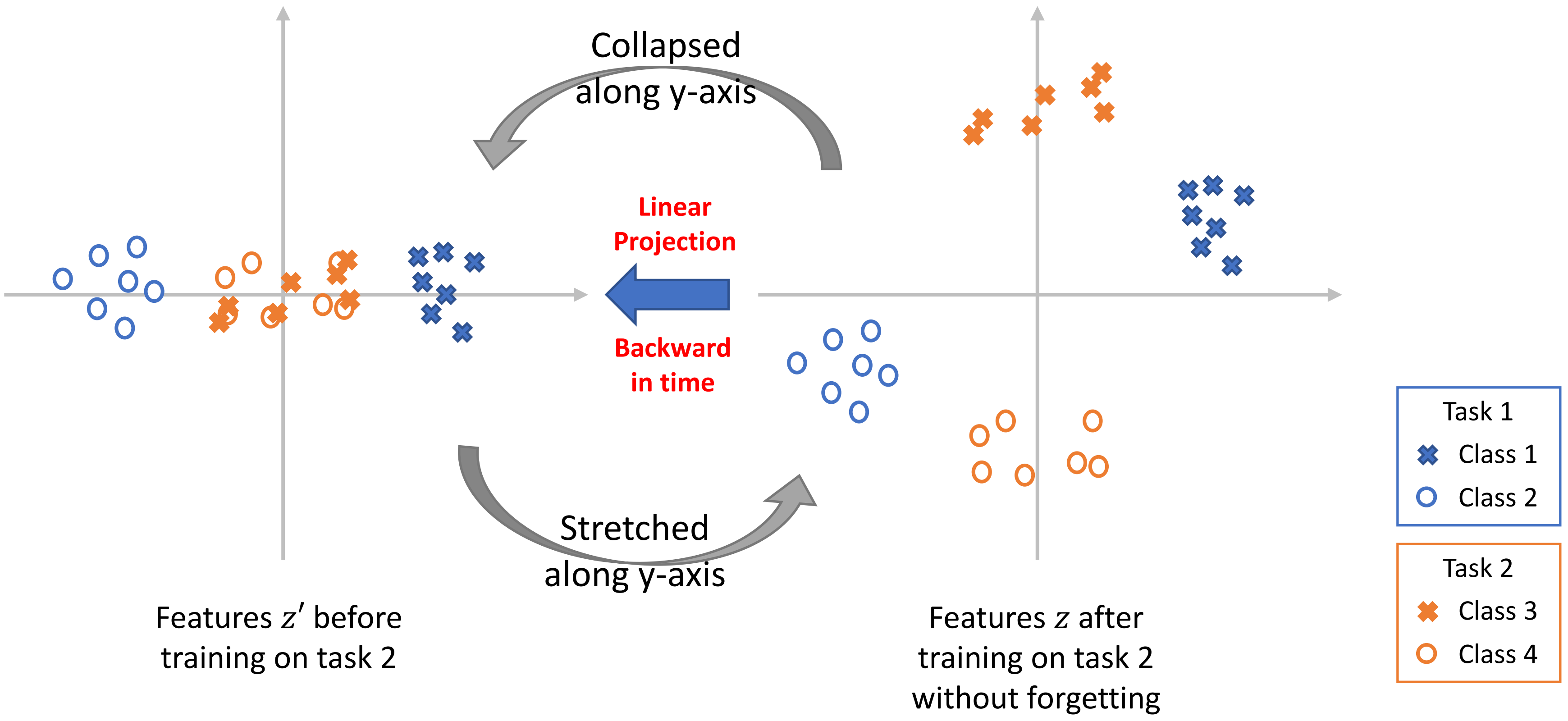


How does feature space evolve in CL?

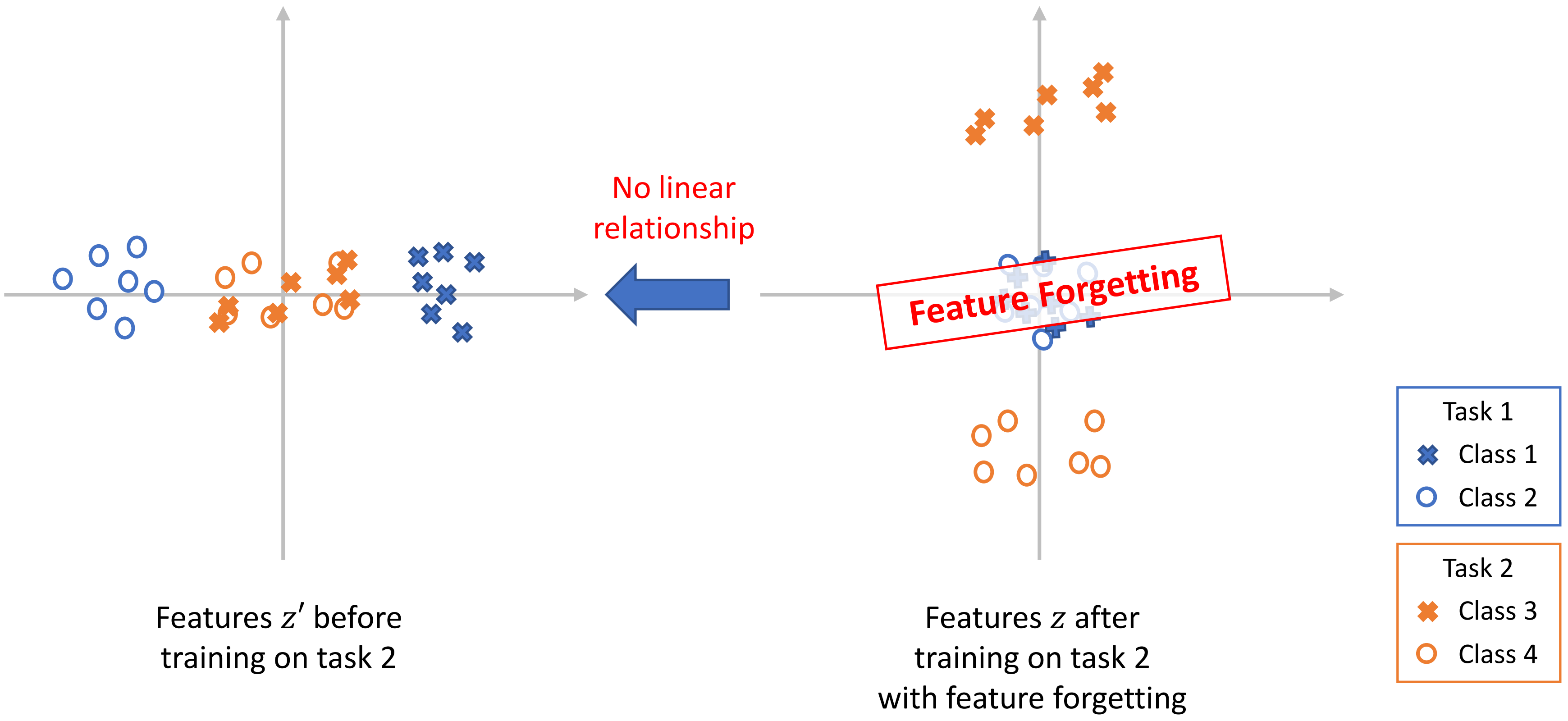
- After training on task 2, ideally



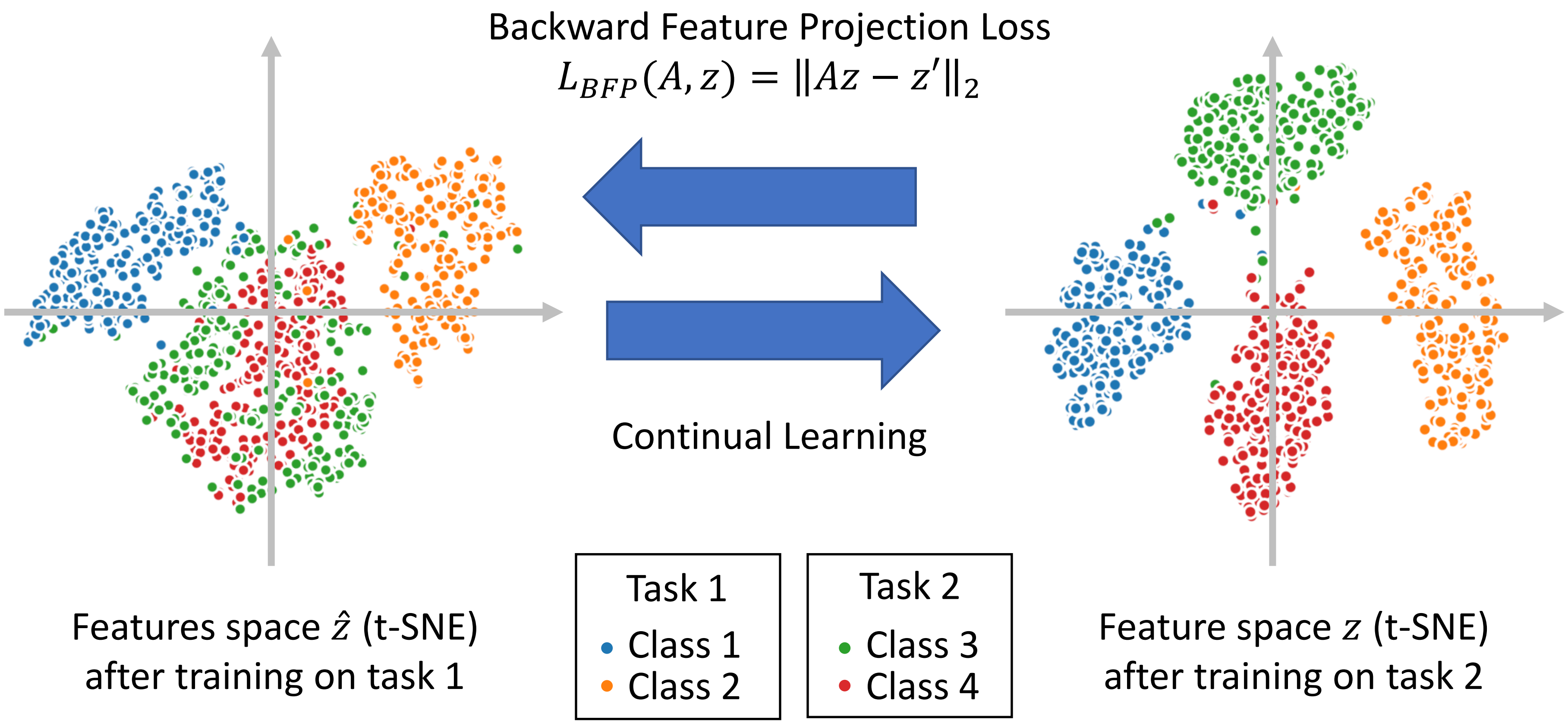
Learning new features results in a linear feature projection backward in time



In case of feature forgetting…



Backward Feature Projection



Backward Feature Projection

- Old feature extractor $f': x \to z' \in \mathbb{R}^d$
- New feature extractor: $f: x \to z \in \mathbb{R}^d$
- Learnable linear projection matrix $A \in \mathbb{R}^{d \times d}$

$$L_{BFP} = \sum_x \|f'(x) - Af(x)\|_2$$

- Baseline: Feature Distillation

$$L_{FD} = \sum_x \|f'(x) - f(x)\|_2$$

BFP Combined with Experience Replay

Algorithm 1 - Continual Learning with BFP

Input: dataset $\{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$, parameters $\theta = \{\phi, \psi\}$, scalars α, β and γ, optimizer $sgd, sgdm$,

$\boxed{M \leftarrow \{\}}$ Memory buffer

for t **from** 1 **to** T **do**

 $A \leftarrow \textit{random-init}()$

 $sgdm \leftarrow \textit{reinit}(sgdm)$

 for (x_o, y_o) **in** $\mathcal{D}_t$ **do**

  $x, y \leftarrow \textit{augment}(x_o, y_o)$

  $L \leftarrow \textit{cross-entropy}(y, f_\theta(x))$

  if $t > 1$ **then**

   $x, y \leftarrow \textit{augment}(\boxed{\textit{sample}(M)})$

   $L_{\text{rep-ce}} \leftarrow \textit{cross-entropy}(y, f_\theta(x))$

   $x, y \leftarrow \textit{augment}(\boxed{\textit{sample}(M)})$

   $L_{\text{rep-logits}} \leftarrow \|f_\theta(x) - f_{\text{old}}(x)\|_2$

   $x, y \leftarrow \textit{augment}(\boxed{\textit{sample}(M)})$

   $L_{\text{BFP}} \leftarrow \|Ah_\psi(x) - h_{\text{old}}(x)\|_2$

   $L = L + L_{\text{rep-ce}} + L_{\text{rep-logits}} + L_{\text{BFP}}$

  end if

  $\theta \leftarrow sgd(\theta, \nabla_\theta L)$

  $A \leftarrow sgdm(A, \nabla_A L)$

  $\boxed{M \leftarrow \textit{balanced-reservoir}(M, (x_o, y_o))}$ Reservoir sampling

 end for

 $f_{\text{old}} = \textit{freeze}(f_\theta)$

end for

Experience replay

[1] Buzzega, Pietro, et al. "Dark experience for general continual learning: a strong, simple baseline." NeurIPS 2020

BFP Combined with Experience Replay

Algorithm 1 - Continual Learning with BFP

Input: dataset $\{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$, parameters $\theta = \{\phi, \psi\}$, scalars α, β and γ, optimizer *sgd*, *sgdm*,

$M \leftarrow \{\}$

for t **from** 1 **to** T **do**

  $A \leftarrow \textit{random-init}()$

  $\textit{sgdm} \leftarrow \textit{reinit}(\textit{sgdm})$

Random init A

  for (x_o, y_o) **in** $\mathcal{D}_t$ **do**

    $x, y \leftarrow \textit{augment}(x_o, y_o)$

    $L \leftarrow \textit{cross-entropy}(y, f_\theta(x))$

    if $t > 1$ **then**

      $x, y \leftarrow \textit{augment}(\textit{sample}(M))$

      $L_{\text{rep-ce}} \leftarrow \textit{cross-entropy}(y, f_\theta(x))$

      $x, y \leftarrow \textit{augment}(\textit{sample}(M))$

      $L_{\text{rep-logits}} \leftarrow \|f_\theta(x) - f_{\text{old}}(x)\|_2$

Cross-entropy and logits regularization [1]

      $x, y \leftarrow \textit{augment}(\textit{sample}(M))$

      $L_{\text{BFP}} \leftarrow \|Ah_\psi(x) - h_{\text{old}}(x)\|_2$

BFP

      $L = L + L_{\text{rep-ce}} + L_{\text{rep-logits}} + L_{\text{BFP}}$

    end if

    $\theta \leftarrow \textit{sgd}(\theta, \nabla_\theta L)$

    $A \leftarrow \textit{sgdm}(A, \nabla_A L)$

Gradient Descent

    $M \leftarrow \textit{balanced-reservoir}(M, (x_o, y_o))$

  end for

  $f_{\text{old}} = \textit{freeze}(f_\theta)$

end for

[1] Buzzega, Pietro, et al. "Dark experience for general continual learning: a strong, simple baseline." NeurIPS 2020

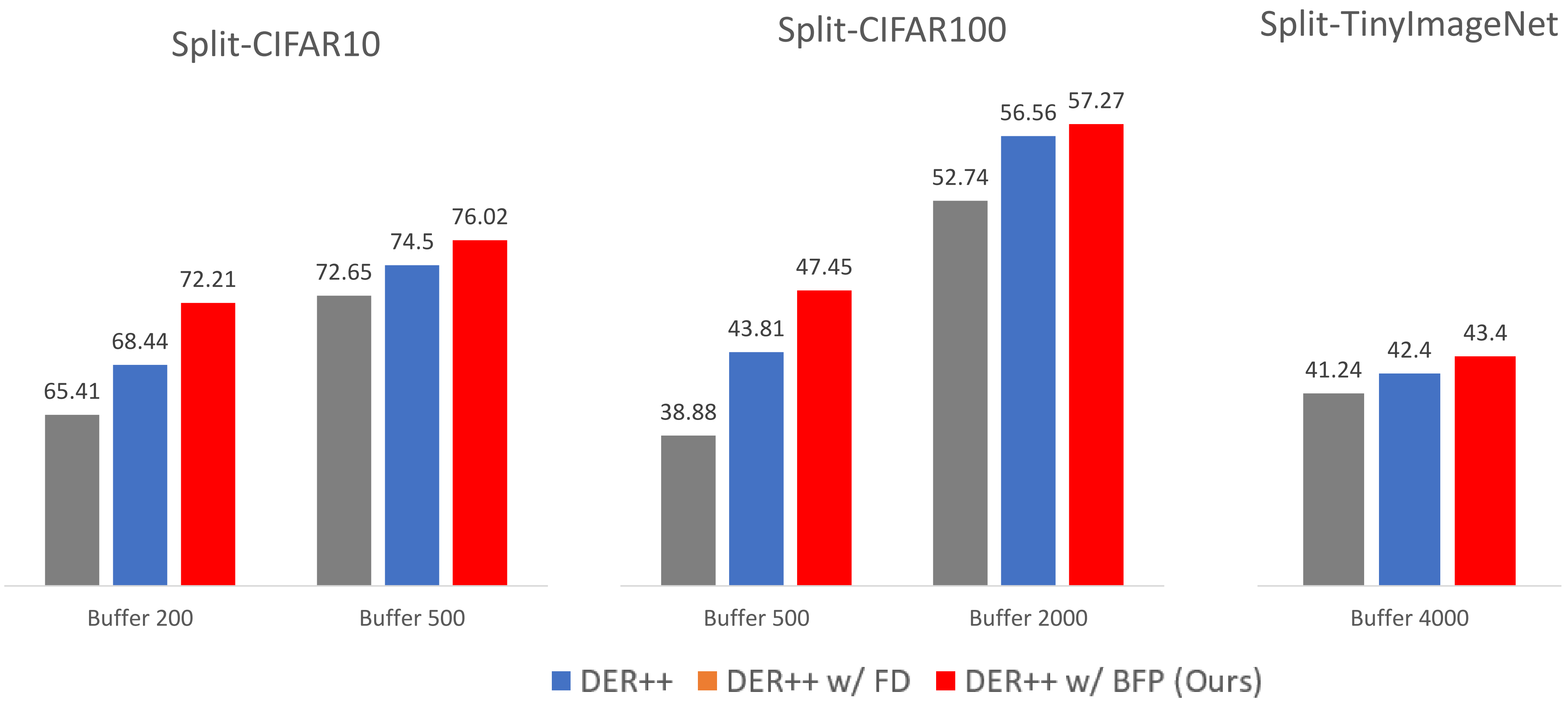
Datasets

- Class-incremental learning datasets
  - Split-CIFAR10
    - 5 tasks, 2 classes per task
  - Split-CIFAR100
    - 10 tasks, 10 classes per task
  - Split-TinyImageNet
    - 10 tasks, 20 classes per task
- Metrics
  - Final class-IL accuracy
  - Final forgetting (refer to the paper)

BFP improves performance by a large margin

Method	S-CIFAR10		S-CIFAR100		S-TinyImageNet
Buffer Size	200	500	500	2000	4000
Joint Training (JT)	91.27±0.57		70.68±0.57		59.61±0.25
Finetuning (FT)	36.20±2.02		9.36±0.07		8.11±0.08
iCaRL [41]	63.58±1.22	62.62±2.07	46.66±0.23	52.60±0.38	31.47±0.46
FDR [5]	31.24±2.61	28.72±2.86	22.64±0.56	34.84±1.03	26.52±0.41
LUCIR [23]	58.53±3.03	70.37±0.97	35.14±0.57	48.95±1.21	29.79±0.70
BiC [50]	59.53±1.77	75.41±1.14	35.96±1.38	45.44±0.96	15.98±1.01
ER-ACE [10]	63.54±0.42	71.17±1.38	38.86±0.72	50.20±0.39	37.72±0.16
ER [42]	58.07±2.92	68.04±2.18	20.34±0.96	37.44±1.48	23.29±0.54
ER w/ BFP (Ours)	63.27±1.09 (+5.21)	71.51±1.58 (+3.47)	22.54±1.10 (+2.20)	38.92±1.94 (+1.48)	26.33±0.68 (+3.04)
DER++ [8]	65.41±1.60	72.65±0.33	38.88±0.91	52.74±0.79	41.24±0.64
DER++ w/ BFP (Ours)	**72.21±0.22** (+6.80)	**76.02±0.79** (+3.37)	**47.45±1.30** (+8.56)	**57.91±0.66** (+5.17)	**43.40±0.41** (+2.16)

Ablation study with Feature Distillation

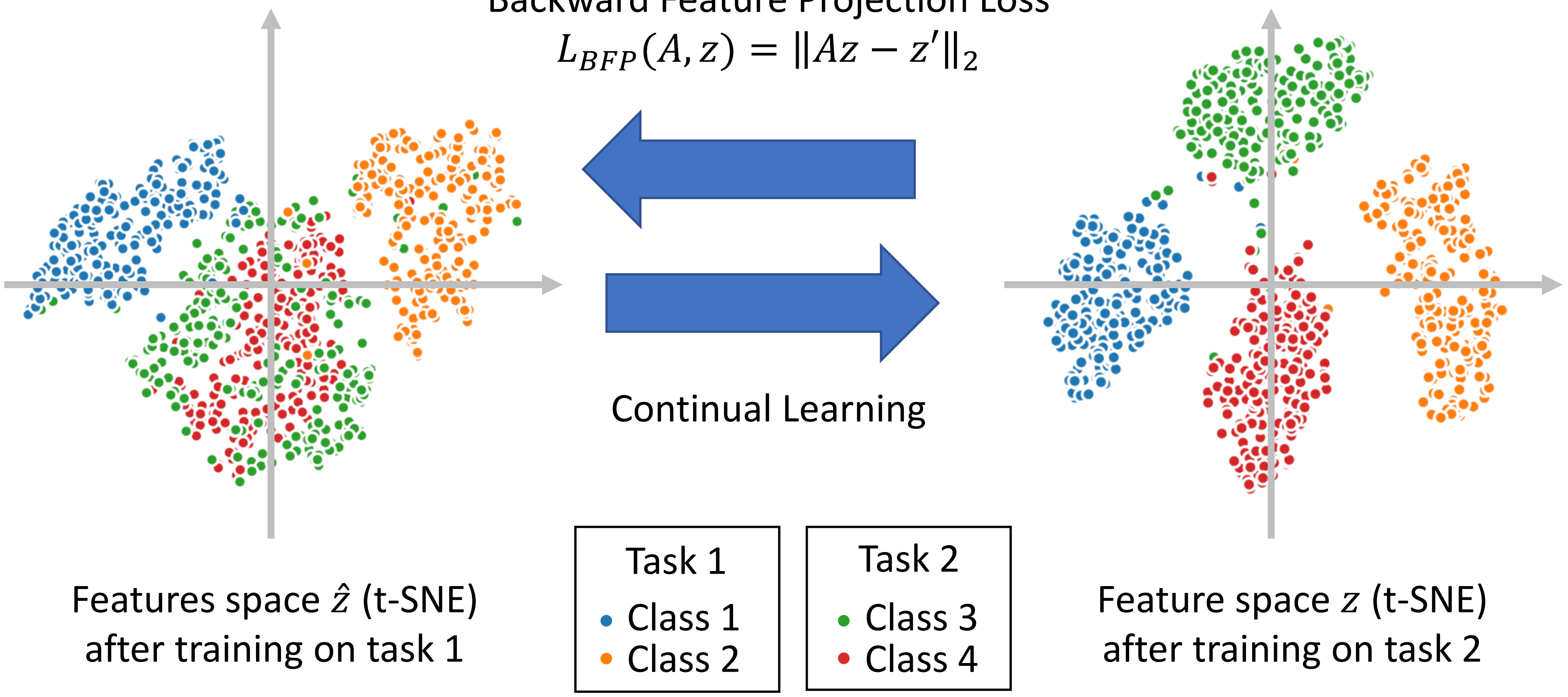


Results are averaged over 5 runs, with standard deviation in parentheses.

BFP results in linearly separable features

Backward Feature Projection Loss

$$L_{BFP}(A, z) = \|Az - z'\|_2$$

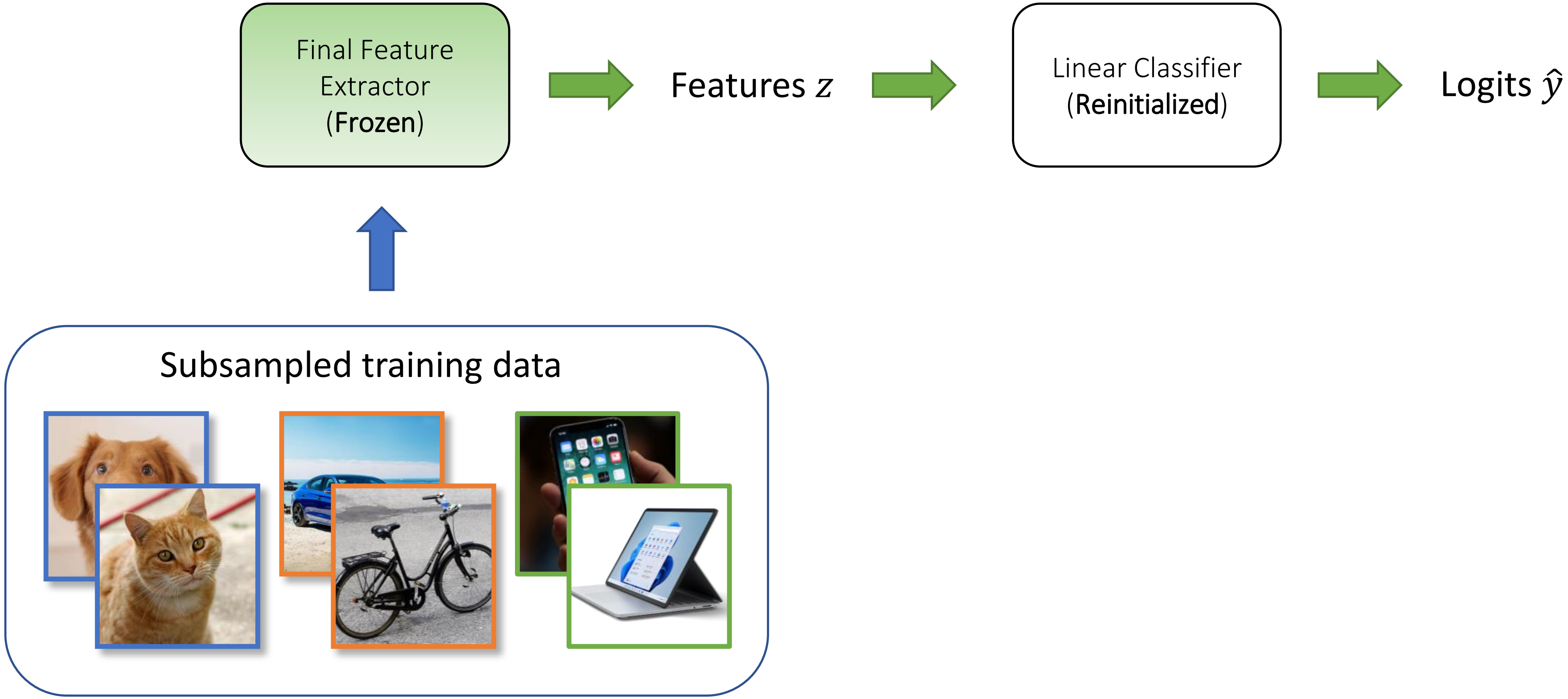


Features space $\hat{z}$ (t-SNE) after training on task 1

Feature space z (t-SNE) after training on task 2

Linear Probing

- After continual learning on all tasks …



Linear Probing

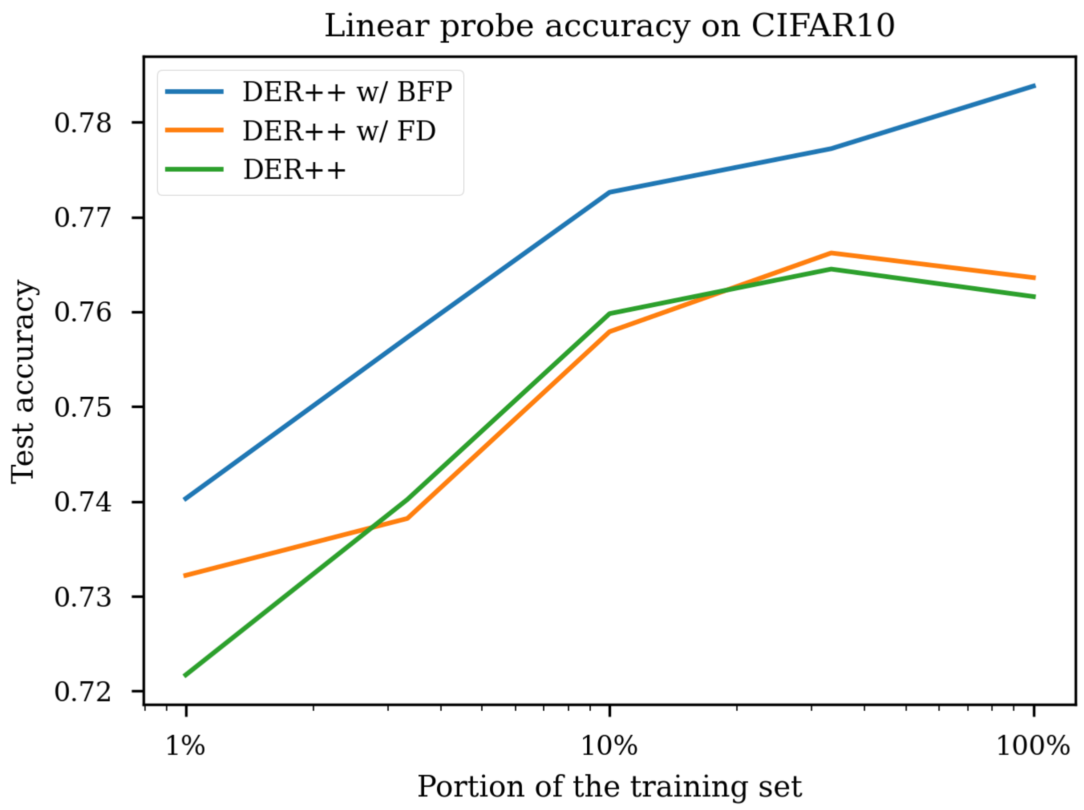
- After continual learning on all tasks …

Final Feature Extractor (**Frozen**) → Features z → Linear Classifier (**Reinitialized & retrained**) → Logits $\hat{y}$

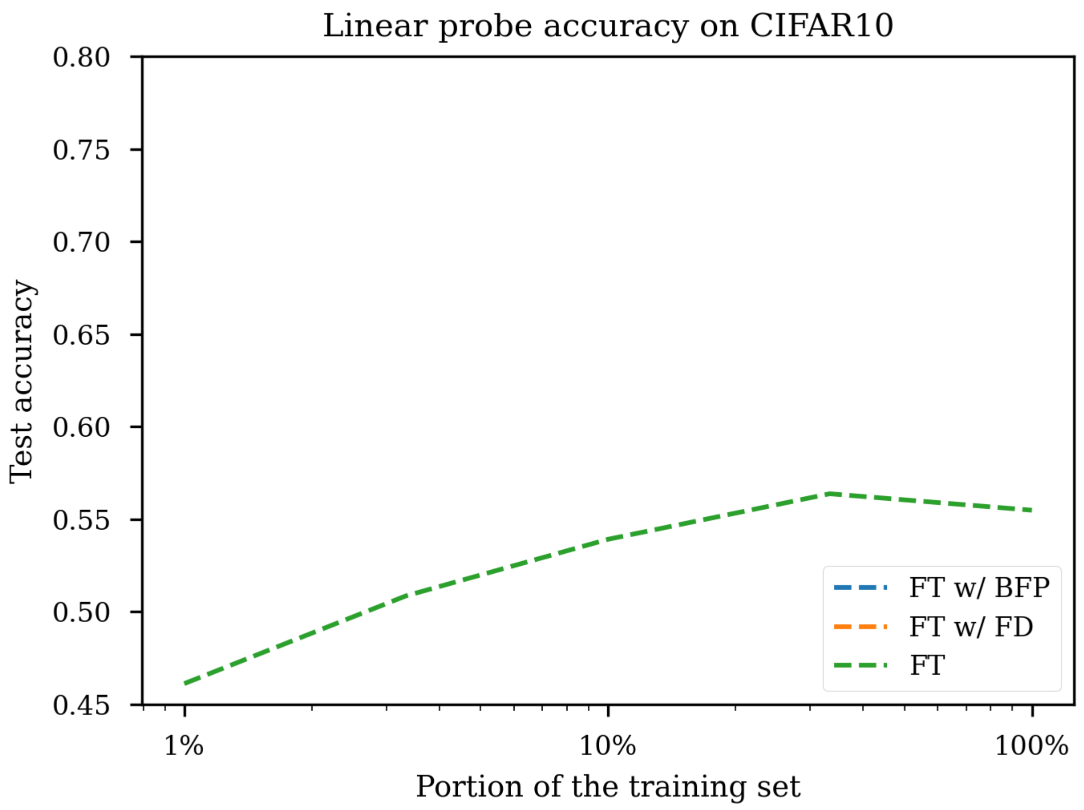
Linearly Separable Feature Space ← Higher accuracy

Subsampled training data

BFP results in a linearly separable feature space and higher linear probing accuracies

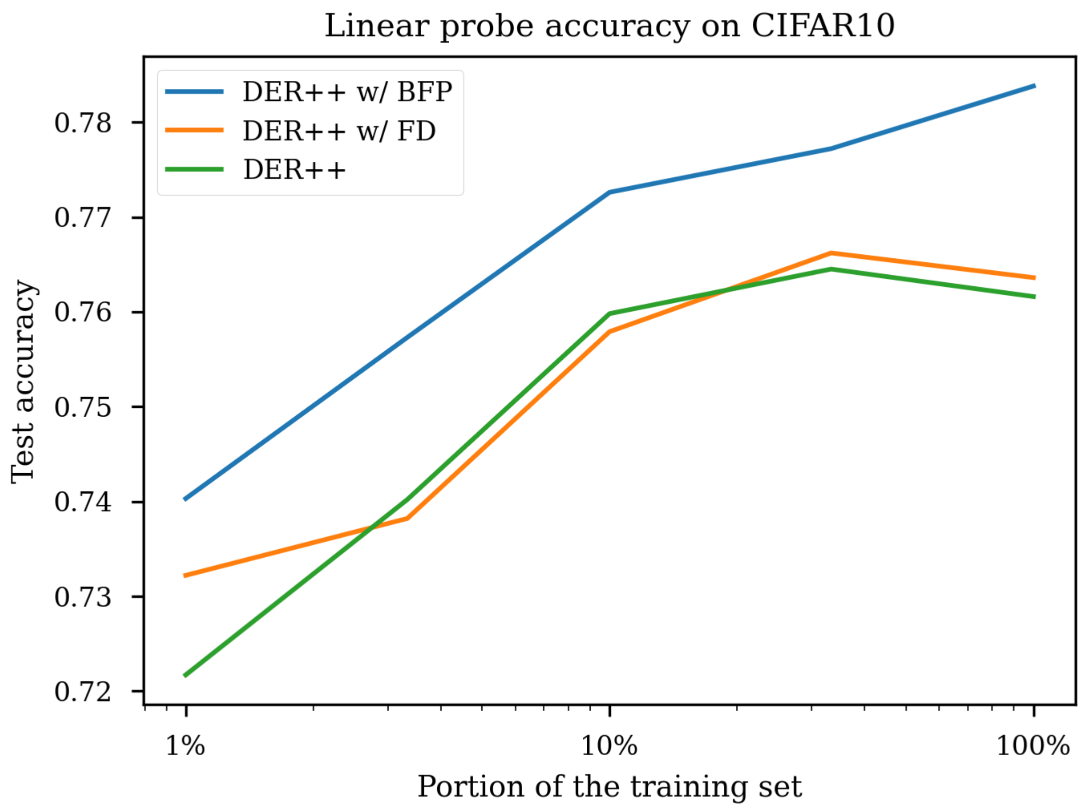


Based on DER++, with memory buffer and replay

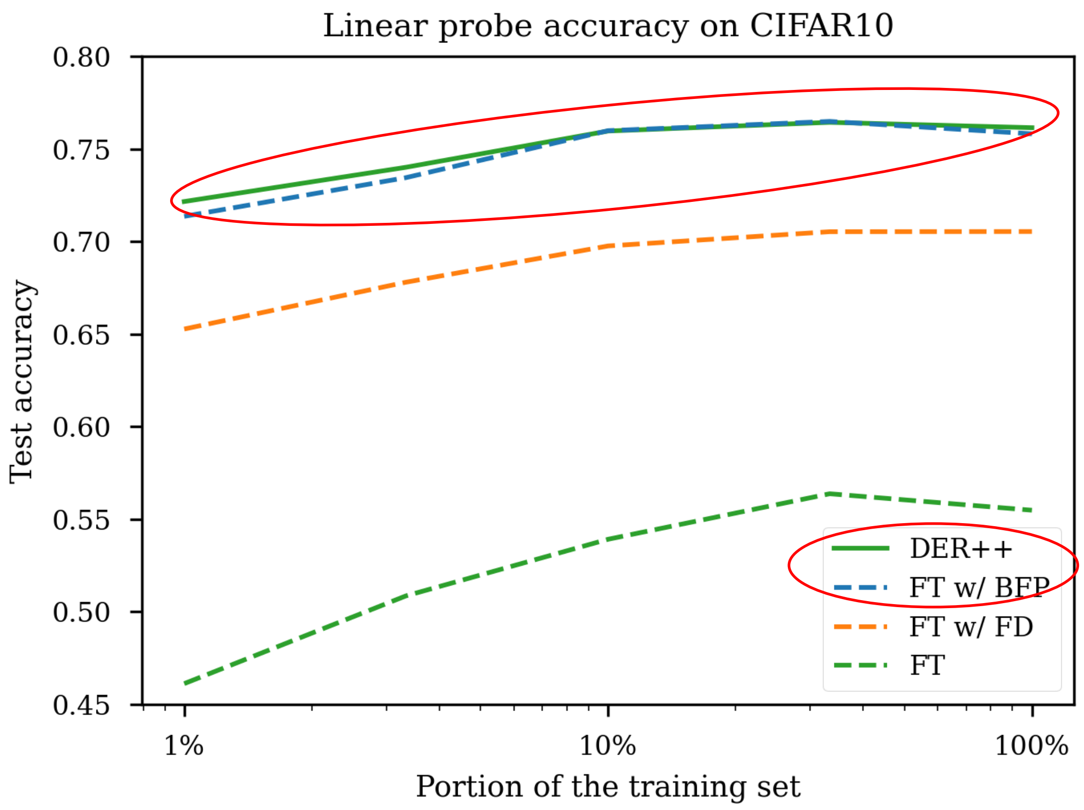


Based on Finetune (FT) baseline, where **no replay** is applied.

BFP results in a linearly separable feature space and higher linear probing accuracies



Based on DER++, with memory buffer and replay



Based on Finetune (FT) baseline, where **no replay** is applied.

Conclusion

- We proposed Backward Feature Projection, a simple yet strong method to reduce forgetting in continual learning.
- We showed that BFP can reduce feature forgetting by learning a more linearly separable feature space.
- Experiments showed that BFP can boost CL performance by a significant margin, achieving state-of-the-art results.