Rebalancing Batch Normalization for Exemplar-based Class-Incremental Learning

Sungmin Cha[1,2], Sungjun Cho[2], Dasol Hwang[2], Sunwon Hong[1], Moontae Lee[1,3], and Taesup Moon[1]

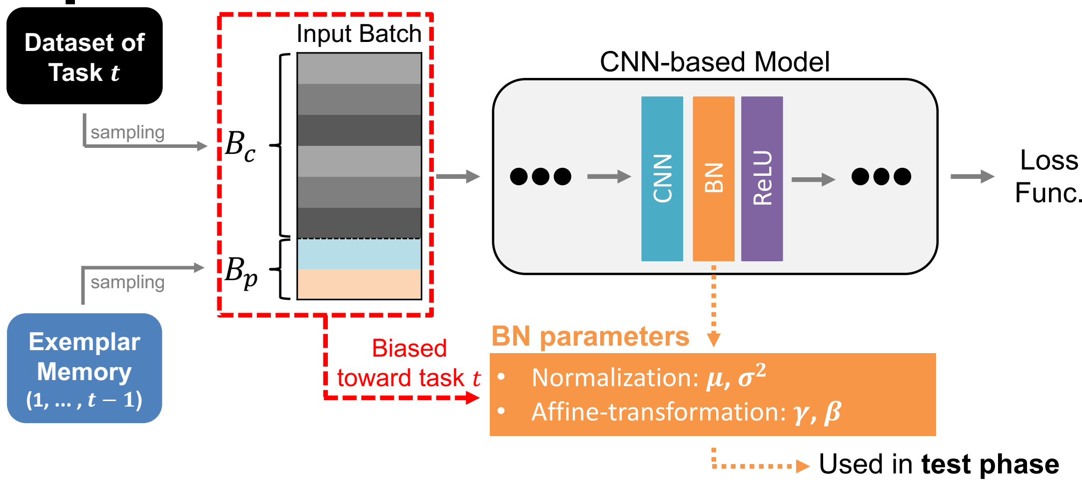
[1] *Seoul National University* [2] *LG AI Research* [3] *University of Illinois Chicago*

Machine Intelligence & Data science LAB

CVPR 2023

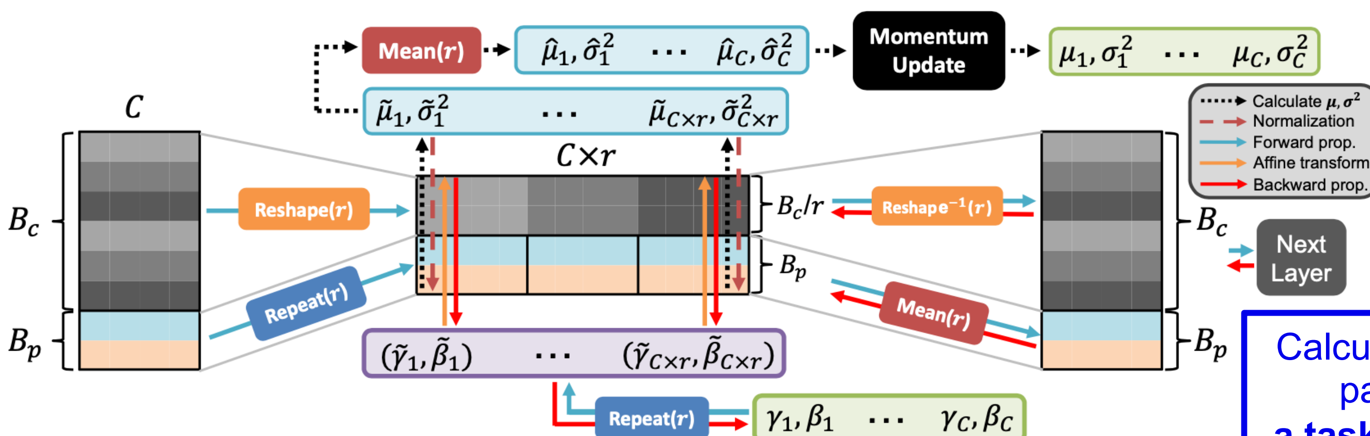
Quick Preview

- **The problem of batch normalization** in exemplar-based CIL



Biased BN parameters cause a problem in **the test phase**

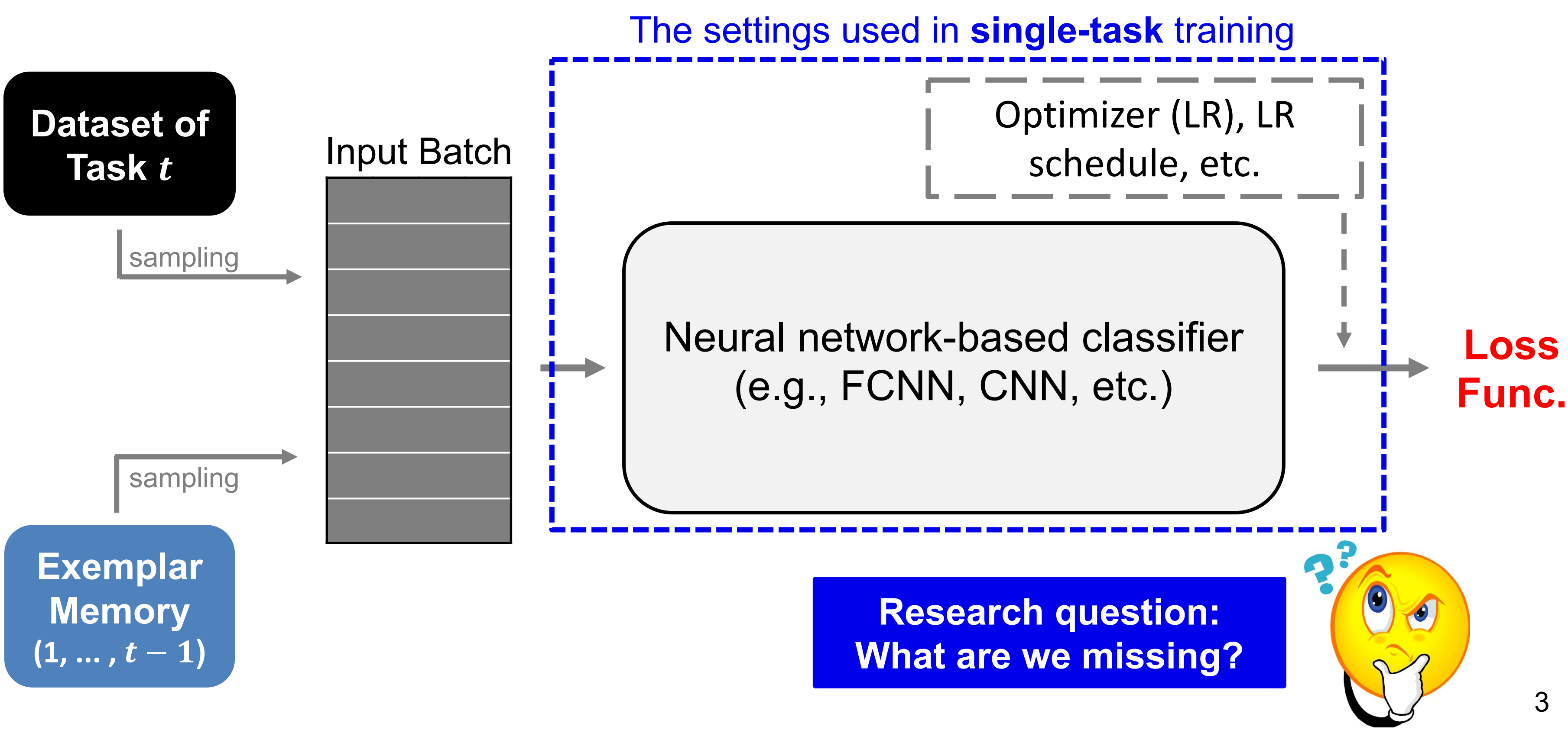
- **Task-Balanced Batch Normalization (TBBN)** for exemplar-based CIL



Calculate or learn BN parameters in **a task-balanced way**

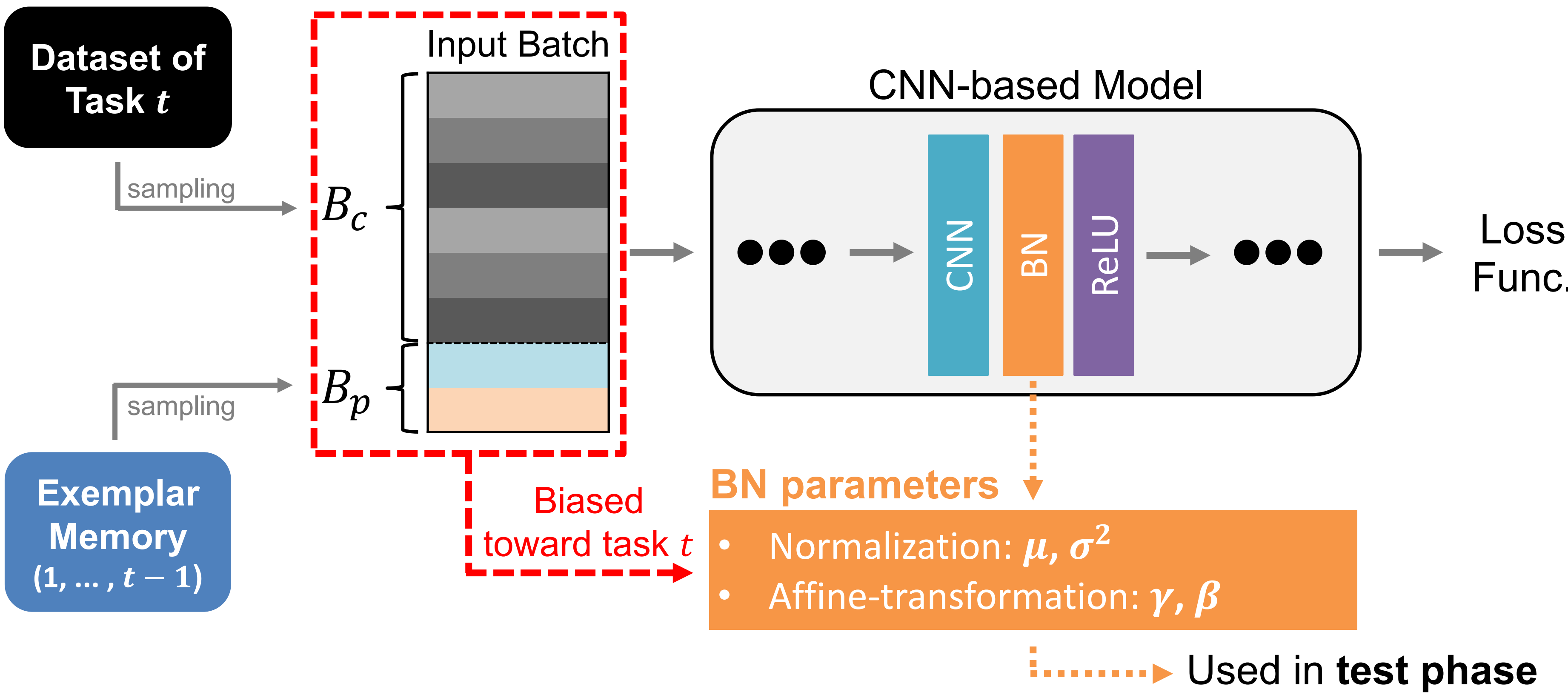
Motivation

- In exemplar-based Class-Incremental Learning (CIL)
  – Most methods only focus to propose **a novel loss func**.



Motivation

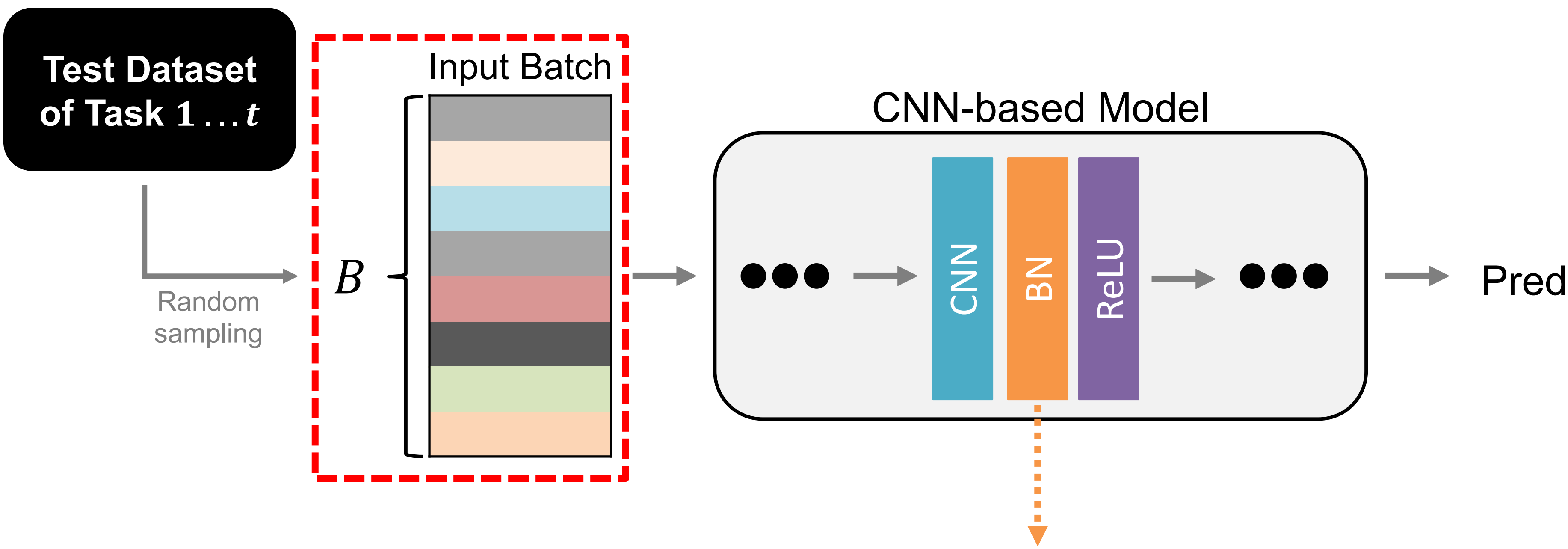
- The case of Batch Normalization (BN) [10]
  - In **training phase** of exemplar-based CIL, BN statistics gets **biased**



[10] S. Loffe, et al., "Batch normalization: Accelerating deep network training by reducing internal covariate shift". ICML 2015.

Motivation

- The case of Batch Normalization (BN)
  - In **test phase**, BN layer is **mismatched** with test data distribution



Biased BN parameters

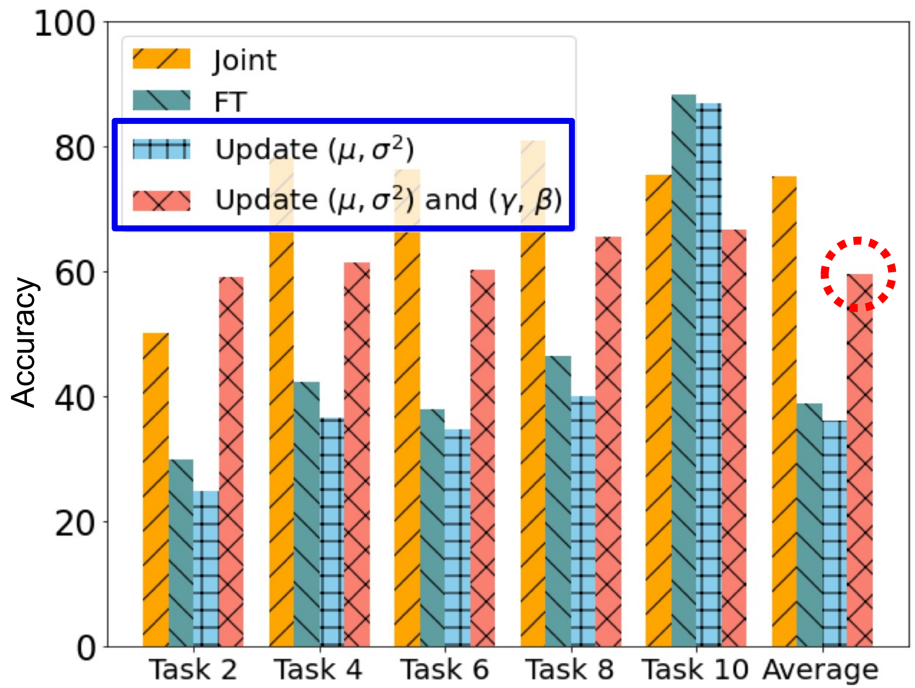
- Normalization: μ, σ^2
- Affine-transformation: γ, β

Biased BN parameters can cause a problem

Motivation

- Experimental verification of the biased BN
  - Dataset: ImageNet-100 (10 classes × 10 tasks)
  - Algorithm: finetuning (FT) with exemplar memory saving 2000 images
  - Model: ResNet-32

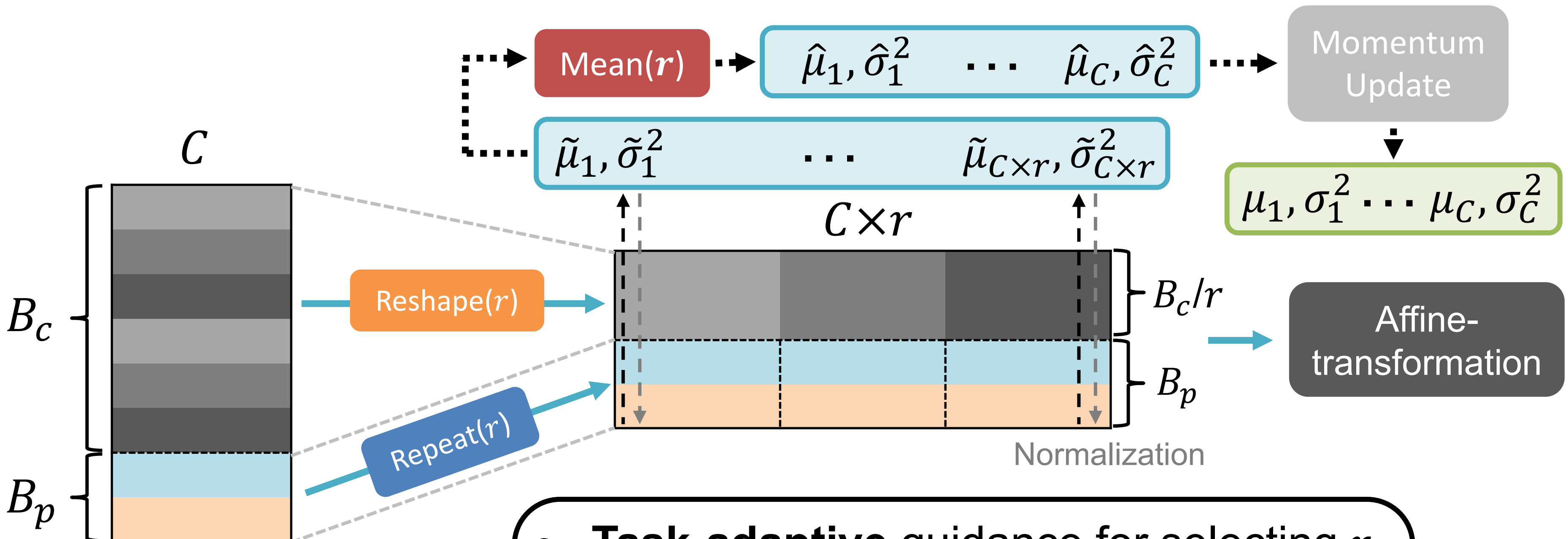
Re-tuning only BN parameters with the joint dataset



Debiasing BN parameters increases CIL accuracy

Proposed Method

- Task-Balanced Batch Normalization (TBBN)
  - Calculate **task-balanced** (mu, sigma^2) with a **balanced batch**



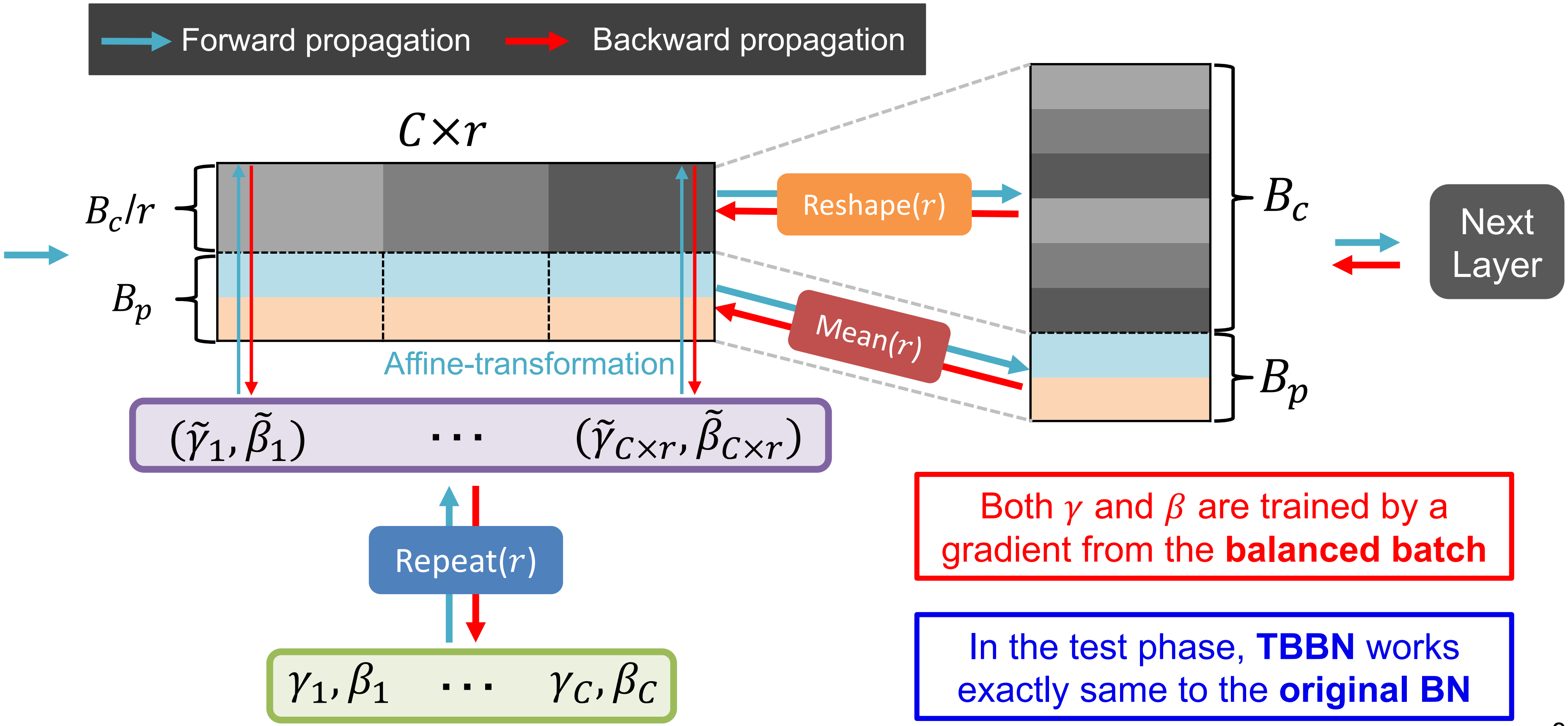
- **Task-adaptive** guidance for selecting r

$$\frac{B_c}{r} : \frac{B_p}{t-1} = 1:1 \rightarrow r = \frac{B_c}{B_p} \cdot (t-1)$$

- $t = 2$ ... T and r is set to 1 when $t = 1$

Proposed Method

- Task-Balanced Batch Normalization (TBBN)
  - Learn a **task-balanced** (γ, β) from a **balanced batch**



Both γ and β are trained by a gradient from the **balanced batch**

In the test phase, **TBBN** works exactly same to the **original BN**

Experimental Result

- Quantitative Result
  - Comparing TBBN with **other normalization layers**
    - Class-IL scenario: 10 classes × 10 tasks
    - Algorithm: finetuning (FT) with exemplar memory saving 2000 images

Method	CIFAR-100 w/ ResNet-32				ImageNet-100 w/ ResNet-18			
	$A_f(\uparrow)$	$A_a(\uparrow)$	$F(\downarrow)$	$A_l(\uparrow)$	$A_f(\uparrow)$	$A_a(\uparrow)$	$F(\downarrow)$	$A_l(\uparrow)$
BN	35.41	53.88	43.48	78.79	39.40	59.60	48.02	87.42
IN	31.72	46.84	47.72	79.44	33.45	53.59	50.69	84.66
GN	31.26	44.53	44.01	75.27	28.83	47.79	49.79	79.19
SN	36.29	53.64	42.91	79.20	39.45	59.55	48.04	87.79
BRN	36.08	52.58	44.34	80.42	37.49	57.77	48.13	86.57
CN	35.06	54.43	43.82	80.64	41.96	60.02	45.32	87.28
CN*	36.05	54.18	44.81	**80.85**	40.46	59.03	46.74	87.20
TBBN	**38.46**	**56.17**	**41.90**	80.36	**43.20**	**61.69**	**43.62**	**87.82**

CN[11] requires to tune a **hyperparameter**

TBBN outperforms others **without hyperparameter tuning**

[11] Q. Pham et al., "Continual normalization: Rethinking batch normalization for online continual learning". *ICLR 2022*.

Experimental Result

- Quantitative Result
  - Applying TBBN to **various CIL baselines**
    - Class-IL scenario: 10 classes × 10 tasks, Exemplar memory: 2000 images

Method		CIFAR-100 w/ ResNet-32				ImageNet-100 w/ ResNet-18			
		$A_f(\uparrow)$	$A_a(\uparrow)$	$F(\downarrow)$	$A_l(\uparrow)$	$A_f(\uparrow)$	$A_a(\uparrow)$	$F(\downarrow)$	$A_l(\uparrow)$
FT	+BN	35.41	53.88	43.38	78.79	39.40	59.60	48.02	87.42
	+CN	35.06	54.43	43.82	**80.64**	41.96	60.02	45.32	87.28
	+TBBN	**38.46**	**56.17**	**41.90**	80.36	**43.20**	**61.69**	**43.62**	**87.82**
EEIL	+BN	39.82	55.25	39.40	79.22	40.06	61.15	47.78	**87.84**
	+CN	39.98	55.09	39.31	79.29	42.48	61.43	44.44	86.92
	+TBBN	**41.93**	**57.53**	**37.80**	**79.93**	**45.18**	**63.48**	**42.66**	**87.84**
LUCIR	+BN	38.06	54.20	32.35	70.41	42.26	63.82	41.68	**83.94**
	+CN	38.07	55.60	33.78	**71.85**	40.44	61.44	42.04	83.48
	+TBBN	**41.45**	**56.13**	**29.23**	70.68	**43.72**	**64.36**	**40.18**	83.90
PODNet	+BN	**38.10**	52.95	14.70	**52.58**	49.05	65.41	22.40	**69.99**
	+CN	34.80	50.26	15.69	50.52	46.20	62.91	23.66	68.50
	+TBBN	37.90	**52.98**	**13.90**	51.78	**49.30**	**65.70**	**21.85**	69.76
SSIL	+BN	41.34	53.00	15.64	56.02	49.56	65.79	21.20	69.94
	+CN	40.74	52.38	**14.60**	54.44	50.58	64.81	**18.56**	65.04
	+TBBN	**43.80**	**54.28**	15.12	**59.37**	**51.30**	**66.51**	19.58	**70.64**
AFC	+BN	39.90	53.93	33.17	73.10	52.50	67.53	19.70	72.22
	+CN	37.50	51.16	33.40	70.94	48.00	65.21	20.10	70.68
	+TBBN	**41.30**	**57.31**	**32.89**	**73.57**	**54.00**	**68.68**	**19.00**	**73.22**

TBBN can be applied to **various existing CIL baselines**

Concluding Remarks

- We tackle **the biased problem of BN** in exemplar-based CIL.
- To solve the above problem, we propose **Task-Balanced Batch Normalization** for exemplar-based class-IL.
- Applying TBBN to existing CIL methods makes performance improvement of them, without **any hyperparameter tuning**.

Thank you!