

## Continual Learning Implementation

Paper: iCaRL - Incremental Classifier and Representation Learning

### Experimental Setup

Model Architecture

ResNet-32

- 기본 특징 추출기로 수정된 ResNet-32 를 사용합니다.
- layer1, layer2, layer3 의 세 개의 합성곱 층을 포함하고 있으며, 각 층은 배치 정규화(Batch Normalization)와 ReLU 활성화 함수를 갖춘 BasicBlock 으로 구성됩니다.

IncrementalNet

- 이 네트워크는 학습된 전체 클래스 수에 따라 동적으로 조정되는 완전 연결 계층을 포함합니다.
- 새로운 클래스가 추가될 때마다 출력 계층이 자동으로 업데이트됩니다.

### Continual Learning Framework

iCaRL (Incremental Classifier and Representation Learning)

- iCaRL 은 리허설 메모리(Exemplar Set)와 지식 증류(Knowledge Distillation)를 통해 새로운 클래스가 추가될 때 발생할 수 있는 망각 문제를 완화합니다.
- 이전에 학습한 클래스를 대표하는 샘플을 메모리에 저장하여 새 클래스 학습 중에도 이전 클래스를 반복 학습합니다.

Datasets

- CIFAR-10 데이터셋만을 사용하였으며, 코드 구조는 CIFAR-100, MNIST, ImageNet 등 다양한 데이터셋을 지원합니다.
- 각 데이터셋별 전처리는 모델의 입력 크기에 맞춰 조정되었습니다. 예를 들어, MNIST 는 32x32 로 리사이즈되고, CIFAR-10 과 CIFAR-100 은 RGB 표준화가 적용되었습니다.

### Training Setup

Hyperparameters

- 초기 학습 (init\_epoch): 20 에포크 동안 진행됩니다.
- 초기 학습률 (init\_lr): 0.1 로 시작하며, 이후 학습률(lr)도 0.1 로 설정되었습니다.
- 학습률 감소: 일정 에포크 이후 단계적으로 적용되며, init\_milestones 와 milestones 는 각각 [60, 120, 170], [80, 120]으로 설정되었습니다.
- Batch size: 실험마다 다양하게 설정되었습니다 (32, 64, 128, 256).
- num\_workers: 데이터 로딩 속도를 최적화하기 위해 num\_workers=4 로 설정되었습니다.

Knowledge Distillation

- 지식 증류 (Knowledge Distillation): 현재 모델과 이전 모델의 출력을 비교하여 이전 클래스에 대한 지식을 유지하도록 돕습니다.
- 온도 매개변수 (T): 지식 증류의 온도는 2, 5, 10 으로 실험되었습니다. 온도가 높을수록 이전 클래스에 대한 정보가 부드럽게 유지되지만, 너무 높으면 모델 성능이 떨어질 수 있습니다.

## Rehearsal Memory

- 메모리 예산 (Memory Budget): 클래스당 특정 수의 샘플을 저장하며, 메모리 예산은 500, 1000, 1500, 2000 으로 설정하여 실험하였습니다.
- 각 클래스에 대해 저장할 샘플 수는 메모리 예산과 총 클래스 수에 따라 계산됩니다. 이를 통해 새 클래스를 학습할 때도 이전 클래스 샘플을 사용할 수 있어 망각을 줄이는데 도움을 줍니다.

## PyCIL Settings (Original and Modified Settings)

### 원래 PyCIL 세팅값

- EPSILON: 1e-8
- 초기 학습 (init\_epoch): 200 에포크
- 초기 학습률 (init\_lr): 0.1
- 학습률 감소 주기 (init\_milestones): [60, 120, 170]
- 학습률 감소 비율 (init\_lr\_decay): 0.1
- 초기 weight decay (init\_weight\_decay): 0.0005
- 실험 설정 (epochs): 170 에포크
- 학습률 (lr): 0.1
- 학습률 감소 주기 (milestones): [80, 120]
- 학습률 감소 비율 (lr\_decay): 0.1
- 배치 크기 (batch\_size): 128
- weight decay: 2e-4
- 데이터 로딩 워커 수 (num\_workers): 8
- 온도 (T): 2

### 수정된 PyCIL 세팅값

- num\_workers: 4
- init\_epoch: 20
- epochs: 20

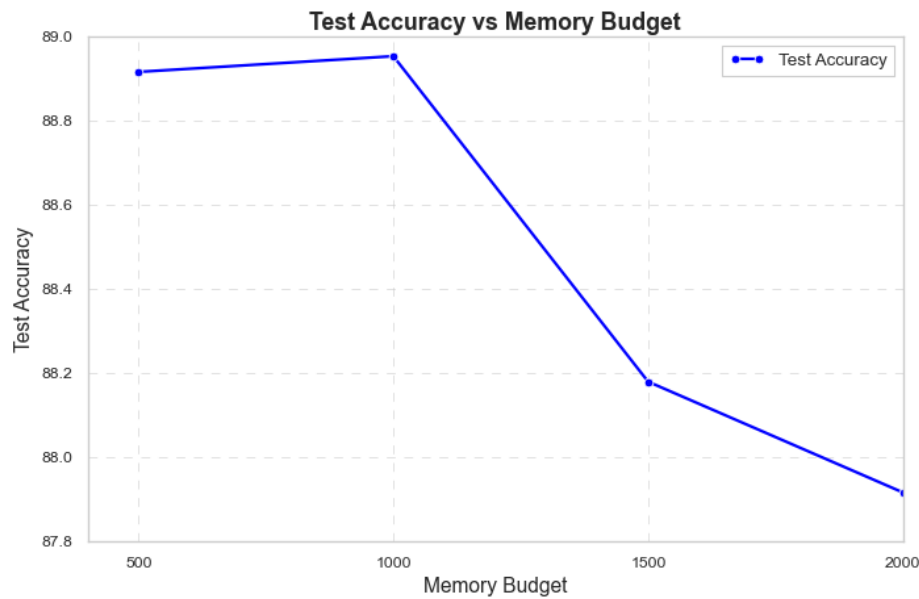
### 실험에서 사용한 하이퍼파라미터

- Memory Budgets: [500, 1000, 1500, 2000]
- Exemplar Sizes: [20, 50, 100, 200]
- Temperatures: [2, 5, 10]
- Batch Sizes: [32, 64, 128, 256]

## Evaluation

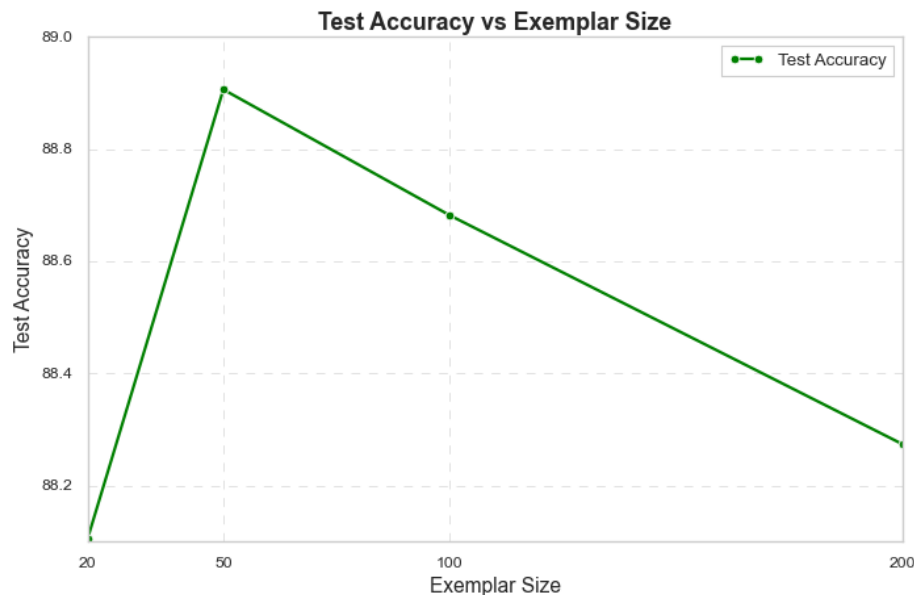
- 각 증분 학습 단계 후 테스트 데이터셋에서 평균 정확도 (average accuracy)를 측정하여 성능을 평가합니다.
- 모든 실험 결과는 메모리 예산, 예제 크기, 온도, 배치 크기 등 다양한 매개변수에 따른 테스트 정확도와 평균 정확도로 기록하여 성능 변화를 분석합니다.

### Memory Budget에 따른 평균 Test Accuracy 변화



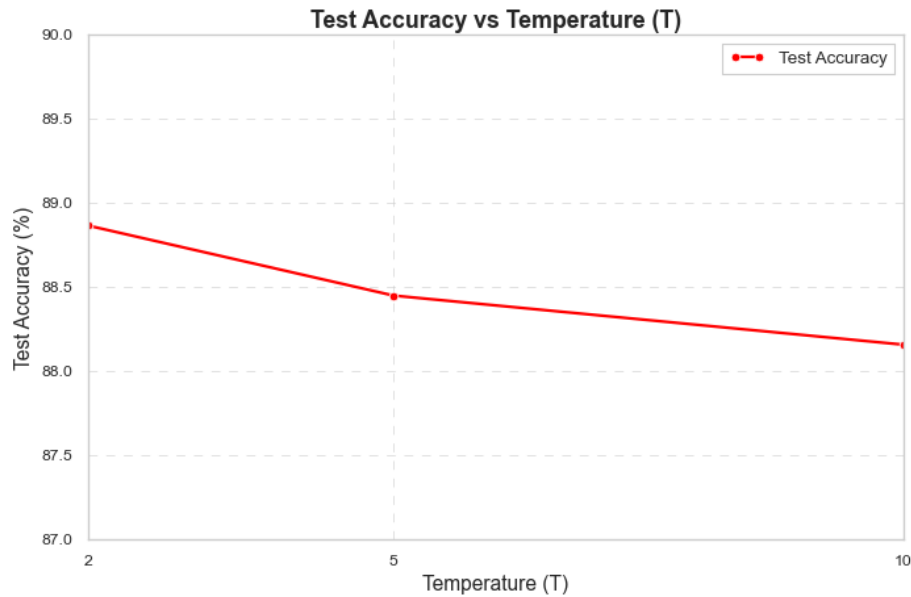
Memory Budget이 증가할수록 Test Accuracy가 감소하는 경향을 보입니다. 이는 리허설 메모리를 과도하게 확장할 경우 새로운 클래스에 적응하는 데 어려움이 있을 수 있다는 점을 의미합니다. iCaRL은 이전 클래스를 저장하는 방식에서 중요한 예제를 선택하여 메모리 예산을 최적화하는 것이 중요하다는 점을 보여줍니다.

### Exemplar Size에 따른 평균 Test Accuracy 변화



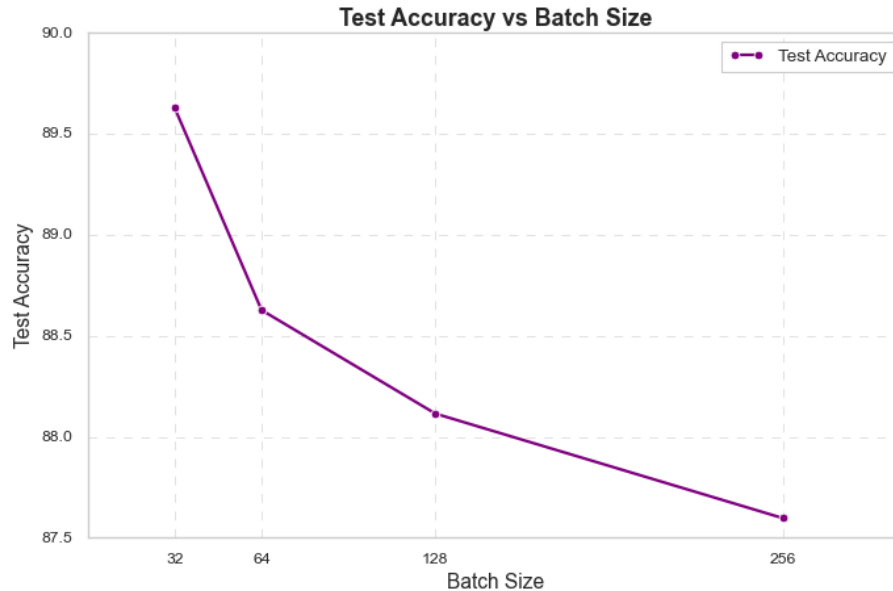
Exemplar Size가 커질수록 Test Accuracy가 향상되는 경향을 보입니다. 이는 더 많은 예제가 리허설 메모리에 저장될수록 모델이 더 많은 정보를 유지하고, 이전 클래스에 대한 망각을 줄일 수 있음을 나타냅니다. 그러나, Exemplar Size가 너무 커지면, 모델이 이전 클래스들에 대한 지식을 너무 많이 저장하려 하면서 새로운 클래스에 대한 적응을 방해할 수 있습니다. 따라서 적절한 크기의 Exemplar Size 설정이 중요합니다.

### Temperature에 따른 평균 Test Accuracy 변화



Temperature의 값이 높을수록 Test Accuracy가 감소하는 경향을 보입니다. 이는 Knowledge Distillation(KD)에서 너무 높은 온도는 이전 클래스에 대한 지식을 과도하게 부드럽게 만들어, 중요한 정보를 희석시킬 수 있음을 시사합니다. 적절한 Temperature 설정이 중요하며, 온도가 낮을수록 현재 클래스와 이전 클래스 간의 차이를 명확하게 유지하고, 그에 따라 성능이 향상됩니다.

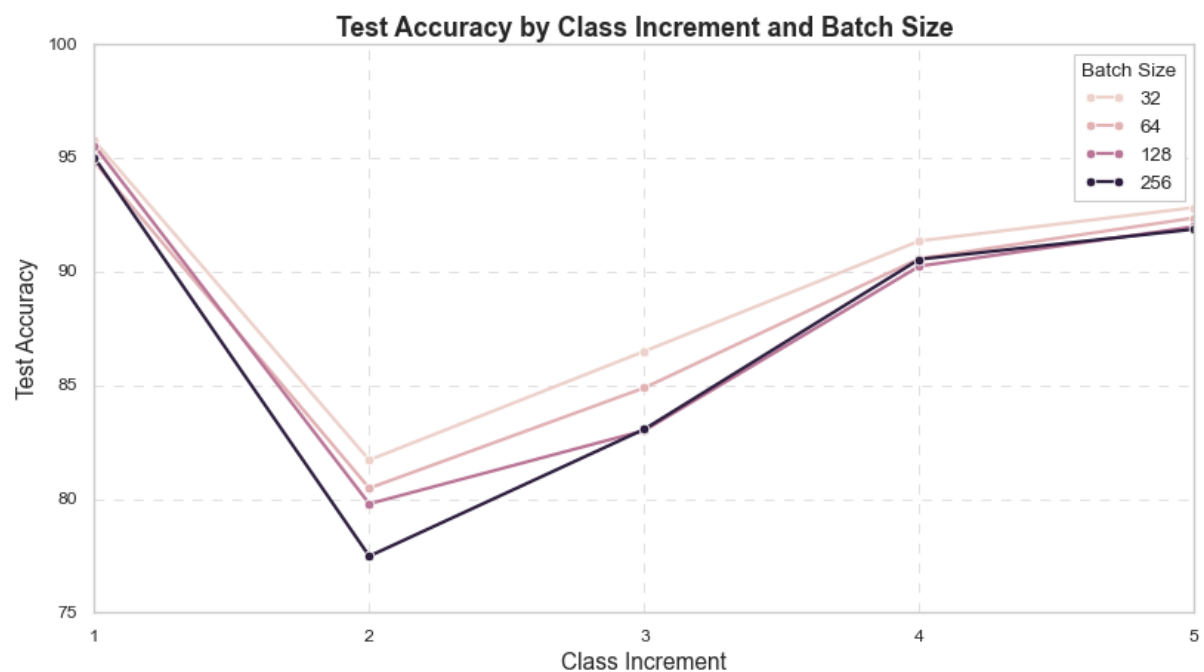
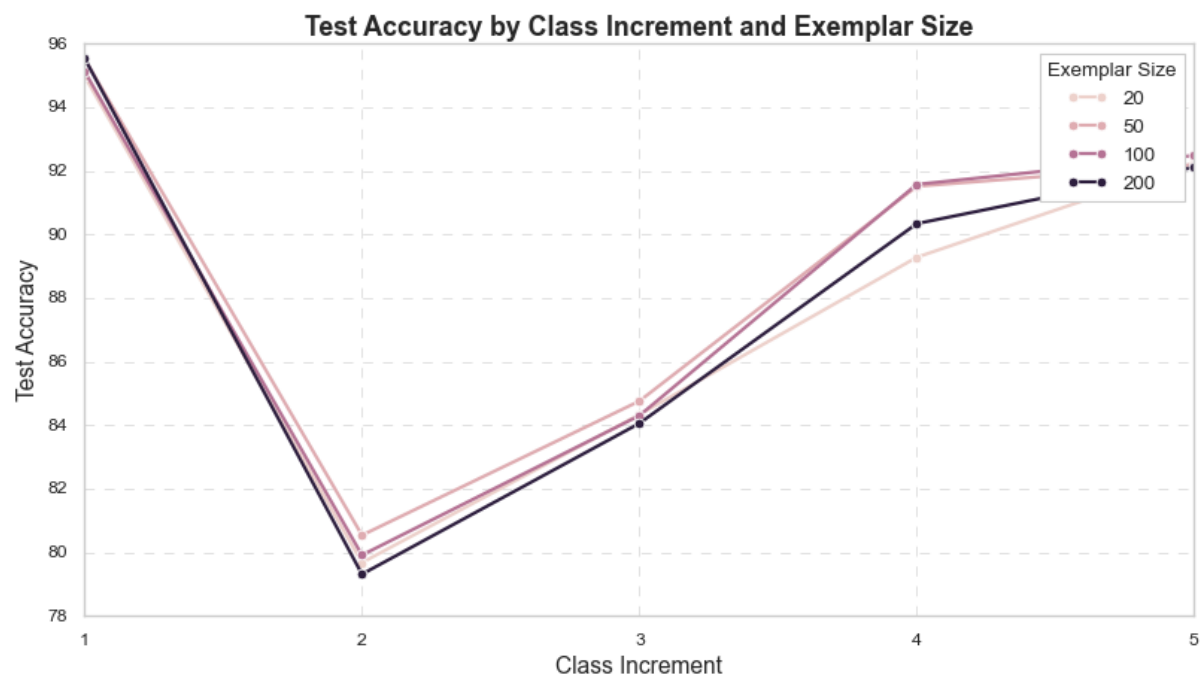
### Batch Size에 따른 평균 Test Accuracy 변화

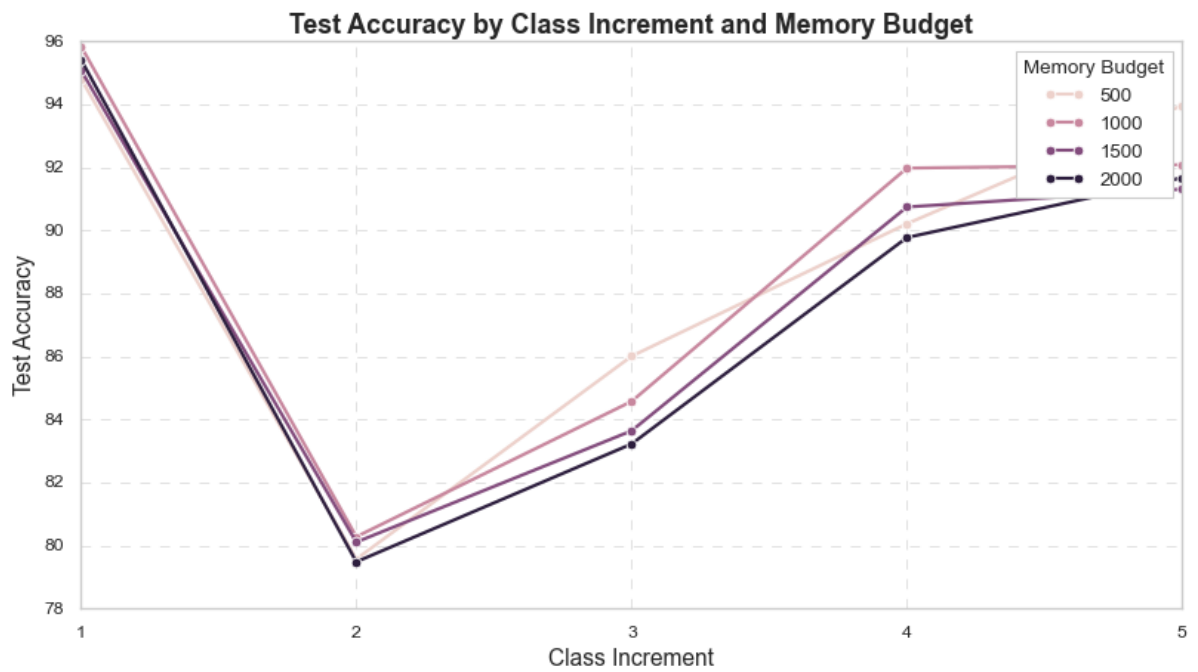
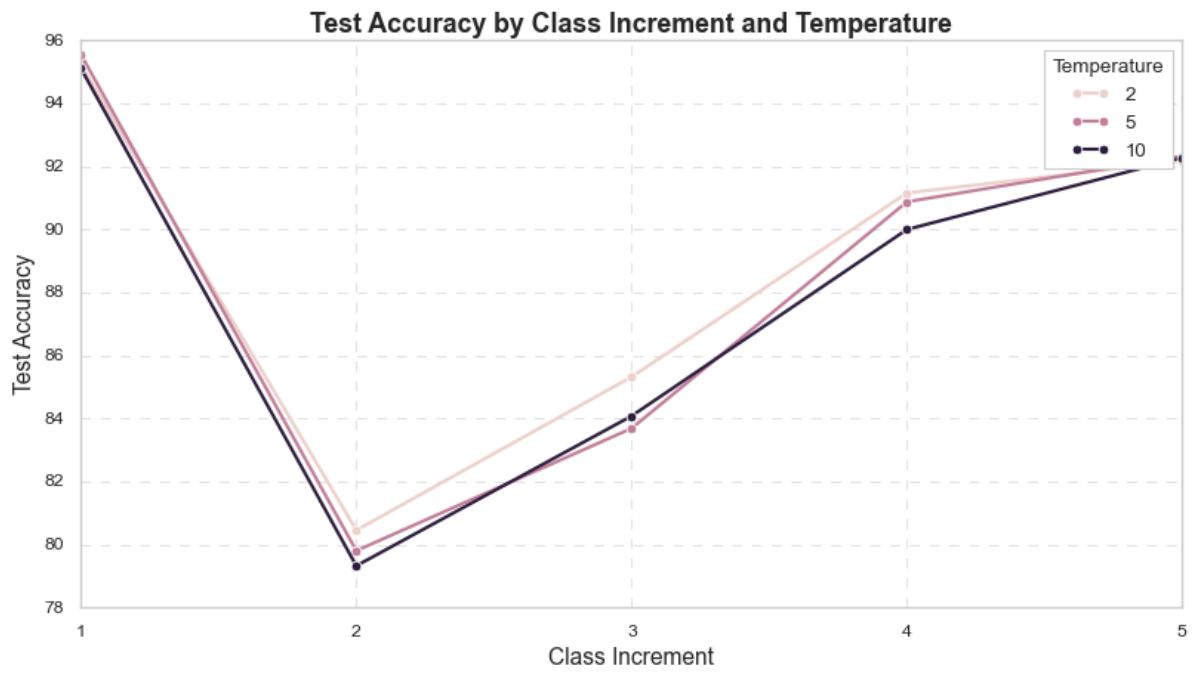


Batch Size가 커질수록 모델의 성능이 초기에는 향상되지만, 어느 시점 이후에는 성능 향상이 정체되거나 감소하는 경향을 보입니다. 이는 Batch Size가 커지면 한 번에 많은 데이터를 처리하게 되어 새로운 클래스 학습 시 이전 클래스에 대한 망각이 발생할 수 있기 때문일 수 있습니다. 작은 배치 크기는 더 자주 모델을 업데이트하고, 각 예제가 모델에 미치는 영향을 더 명확하게 만들어 이전 클래스 정보 보존에 유리한 결과를 보일 수 있습니다.

## Class Increment 에 따른 Test Accuracy 변화

Class Increment가 증가함에 따라 Test Accuracy는 급격히 감소한 후 점진적으로 회복되는 양상을 보입니다. 이는 iCaRL이 새로운 클래스가 추가될 때마다 일시적인 망각이 발생하지만, 리허설 메모리와 지식 증류를 통해 이를 완화하고 성능을 회복하는 방식으로 망각을 줄이려는 노력이 반영된 결과로 보입니다.





논문의 결과는 Memory Budget, Exemplar Size, Batch Size 와 같은 주요 하이퍼파라미터들이 iCaRL 성능에 미치는 긍정적인 영향을 강조하고 있습니다. 그러나 사용자의 실험 결과에서는 이러한 하이퍼파라미터의 영향이 논문의 결과와 상반되거나 약간 다른 경향을 보였습니다. 이는 사용자가 CIFAR-10 데이터셋을 사용한 반면, 논문에서는 CIFAR-100 데이터셋을 사용했기 때문에 발생한 차이일 가능성이 큼니다.

CIFAR-10 과 CIFAR-100 은 클래스 수와 데이터 분포에 큰 차이가 있으며, 이는 iCaRL 의 성능에 큰 영향을 미칠 수 있습니다. CIFAR-100 은 더 많은 클래스(100 개)를 포함하고 있어, 메모리 관리와 클래스 간 구별에서 CIFAR-10 보다 더 큰 도전 과제를 제공합니다. 이로 인해 Memory Budget 과 Exemplar Size 의 설정이 CIFAR-100 에서는 더 중요하게 작용하여, 성능에 민감하게 반응할 수 있습니다. 반면, CIFAR-10 에서는 클래스 수가 적어 메모리 예산이나 예제 크기에 상대적으로 덜 민감하게 반응할 수 있습니다.

따라서, 데이터셋의 차이가 주요 하이퍼파라미터가 성능에 미치는 영향을 다르게 만든 것으로 보이며, CIFAR-10 과 같은 소규모 클래스 데이터셋에 맞는 최적의 하이퍼파라미터를 찾는 것이 중요할 것입니다.

Table 1: Average multi-class accuracy on iCIFAR-100 for different modifications of iCaRL.

(a) Switching off different components of iCaRL (*hybrid1*, *hybrid2*, *hybrid3*, see text for details) leads to results mostly inbetween iCaRL and LwF.MC, showing that all of iCaRL’s new components contribute to its performance.

batch size	iCaRL	<i>hybrid1</i>	<i>hybrid2</i>	<i>hybrid3</i>	LwF.MC
2 classes	57.0	36.6	57.6	57.0	11.7
5 classes	61.2	50.9	57.9	56.7	32.6
10 classes	64.1	59.3	59.9	58.1	44.4
20 classes	67.2	65.6	63.2	60.5	54.4
50 classes	68.6	68.2	65.3	61.5	64.5

(b) Replacing iCaRL’s mean-of-exemplars by a nearest-class-mean classifier (NCM) has only a small positive effect on the classification accuracy, showing that iCaRL’s strategy for selecting exemplars is effective.

batch size	iCaRL	NCM
2 classes	57.0	59.3
5 classes	61.2	62.1
10 classes	64.1	64.5
20 classes	67.2	67.5
50 classes	68.6	68.7

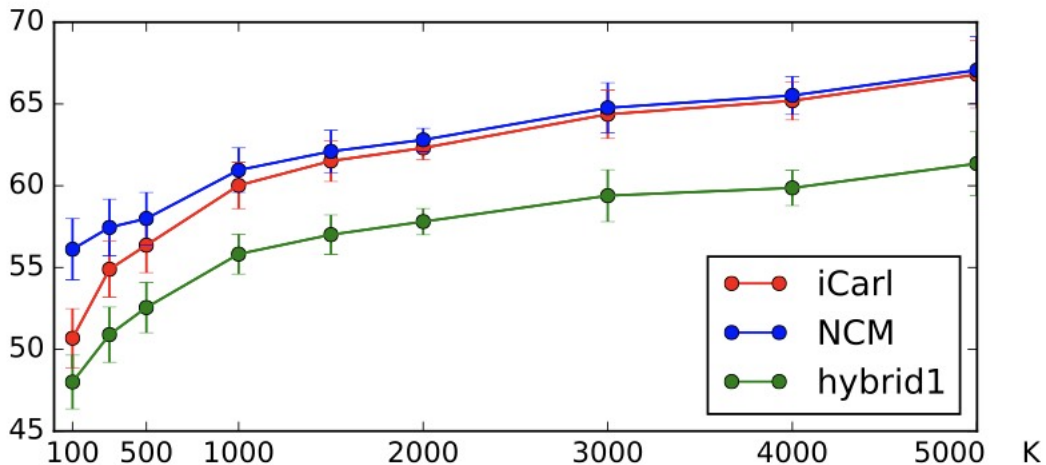


Figure 4: Average incremental accuracy on iCIFAR-100 with 10 classes per batch for different memory budgets  $K$ .