

Masked Contrastive Learning for Coarse-Labelled Dataset

IEEE/CVF Conference on Computer Vision and Pattern Recognition
(CVPR), 2023

Feng Chen and Ioannis Patras

Course Instructor: Prof. C Krishna Mohan **TA:** Raj Popat **Presented By:** Aviraj Antala

Feng, Chen, and Ioannis Patras. "*MaskCon: Masked Contrastive Learning for Coarse-Labelled Dataset.*" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

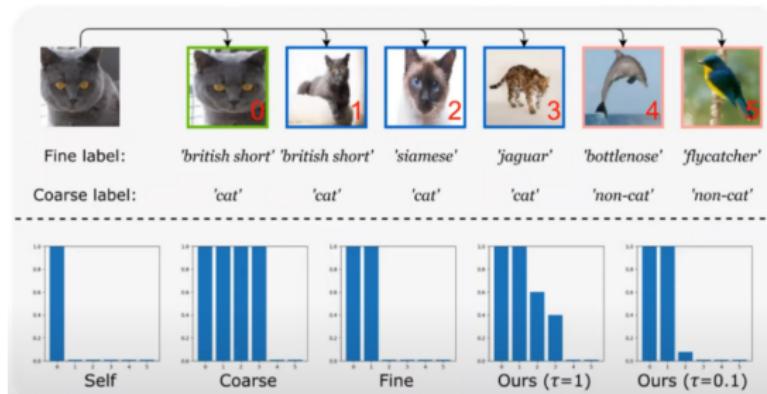
Contents

- ① Problem Statement
- ② MaskCon Methodology
- ③ MaskCon Framework
- ④ Experimental Setup
- ⑤ Paper Results
- ⑥ Reproduced Results
- ⑦ Novelty
- ⑧ Novelty Results
- ⑨ Implementation Challenges
- ⑩ Future Work
- ⑪ References

Problem Statement

Many real-world datasets use coarse labels because fine-grained annotation is expensive and time-consuming.

- Deep learning performs well with fine-grained labels (e.g., classifying dog breeds like “Golden Retriever”).
- In contrast, coarse labels (e.g., just “dog”) are cheap and easier to get.
- MaskCon is proposed to bridge the gap: learn fine-grained representations only from coarse labels.



MaskCon Methodology

MaskCon is a contrastive learning method that works even when only coarse labels are available.

- Traditional contrastive loss pulls all positive pairs together, but this fails when fine-grained labels are missing.
- MaskCon uses coarse labels to form soft groupings and masks out irrelevant pairs.
- For a given anchor sample i , only samples j with the same coarse label contribute to similarity.

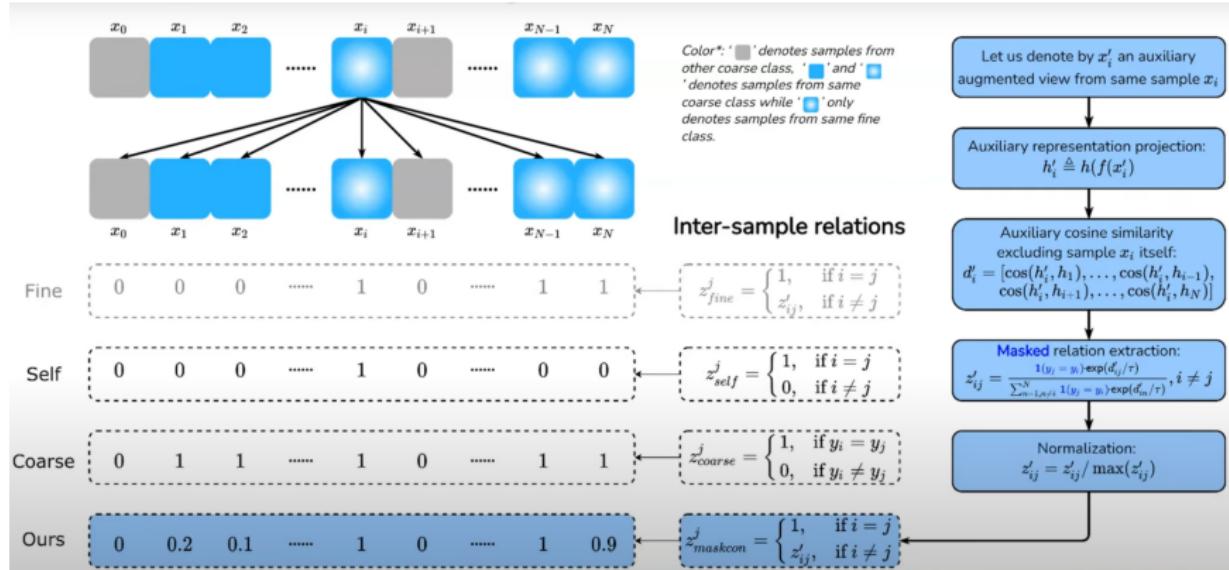
Masked Similarity and Final Loss

$$z'_{ij} = \frac{\mathbf{1}(y_i = y_j) \cdot \exp(\cos(h_i, h_j)/\tau)}{\sum_{n \neq i} \mathbf{1}(y_n = y_i) \cdot \exp(\cos(h_i, h_n)/\tau)}$$

where $\cos(h_i, h_j)$ is cosine similarity and τ is a temperature hyperparameter.

$$\mathcal{L} = w \cdot \mathcal{L}_{\text{MaskCon}} + (1 - w) \cdot \mathcal{L}_{\text{SelfCon}}$$

MaskCon Framework



The framework shows how MaskCon constructs soft similarity using coarse labels, cosine similarity, and masked contrastive loss.

Experimental Setup

Datasets Used

- **CIFAR-10 & CIFAR-100** (coarse labels)
- **ImageNet-1K** (coarse supervision)
- **Stanford Online Products (SOP)**
- **Stanford Cars 196**

Base Model

- **Backbone:** ResNet-18
- Modified: 3×3 conv, stride 1
- Max-pooling removed for CIFAR

Hyperparameters

- $w \in \{0, 0.2, 0.5, 0.8, 1.0\}$
- $\tau \in \{0, 0.01, 0.05, 0.1, 0.5, \infty\}$
- Best: $w = 1, \tau = 0.05$ or 0.1

Evaluation Metric

- **Recall@K:** retrieval accuracy
- 1 if any top-K result matches fine class
- Averaged over test samples

Paper Results: CIFARtoy Dataset

Method	CIFARtoy-goodsplit				CIFARtoy-badsplit			
	Recall@1	Recall@2	Recall@5	Recall@10	Recall@1	Recall@2	Recall@5	Recall@10
SelfCon	84.83	91.55	96.35	98.16	84.83	91.55	96.35	98.16
Grafit	86.61	92.33	97.01	98.38	89.96	94.36	97.61	98.10
SupCon	73.84	84.25	92.14	95.46	84.66	90.93	95.15	96.71
CoIns	86.15	92.76	97.21	98.46	90.55	94.94	97.73	98.71
SupCE	76.30	85.26	94.65	97.46	87.15	92.85	96.78	98.34
SupFINE	94.11	96.53	98.25	98.96	94.11	96.53	98.25	98.96
MaskCon (Ours)	90.28 (13.98↑)	94.04	97.33	98.53	91.56 (4.41↑)	95.23	97.70	98.70

Table 1. Results on CIFARtoy dataset.

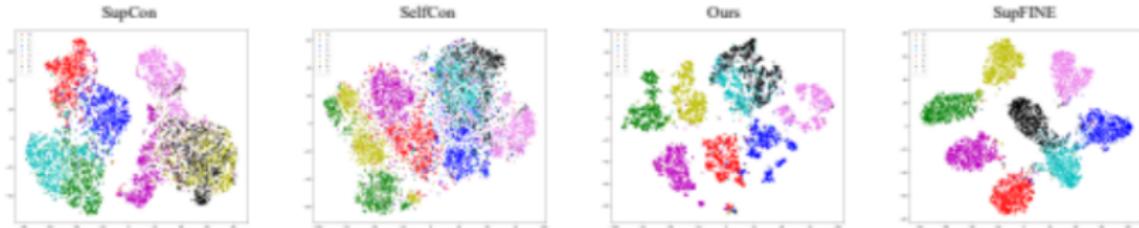


Figure 3. t-SNE visualization of learned representation on CIFARtoy dataset.

Reproduced Results: CIFARtoy Dataset

CIFARtoy-goodsplit

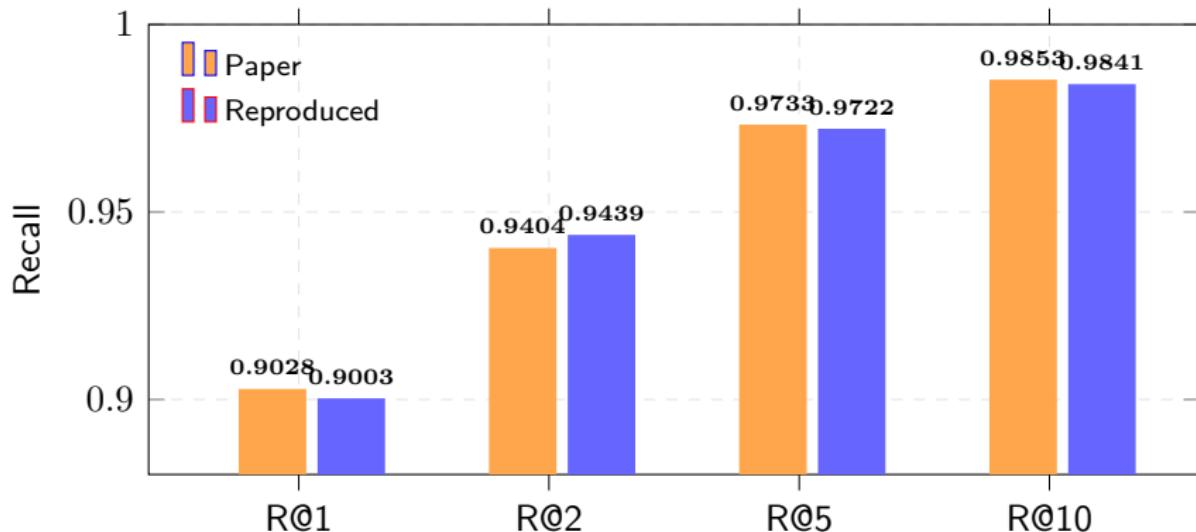
```
Train Epoch: [199/200], lr: 0.000000, Loss: 3.0741 100% | 312/312 [00:23<00:00, 13.21it/s]
...
wandb: Run history:
wandb: R@1
wandb: R@10
wandb: R@100
wandb: R@2
wandb: R@5
wandb: R@50
wandb: Run summary:
wandb: R@1 0.50025
wandb: R@10 0.98413
wandb: R@100 0.99888
wandb: R@2 0.98838
wandb: R@5 0.9725
wandb: R@50 0.99725
wandb: 
wandb: ★ View run train\_arch\_\[resnet18\]\_data\[cifartoy\_good\].epochs\[200\].memorysize\[8192\].mode\[maskcon\].contrastive\_temperature\[0.1\].temperature\_maskcon\[0.05\].weight\[1.0\]
wandb: Synced 5 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
wandb: Find logs at: ./wandb/run-20230815_100524-27jw0lkk/logs
(maskcon_env) cs24mtech12012@cse-node009i-/MaskCon_CVPR2023 screen -r+[{2-}
(maskcon_env) cs24mtech12012@cse-node009i-/MaskCon_CVPR2023 screen -r+[{2-}
(maskcon_env) 0:bash*
```

CIFARtoy-badsplit

```
Train Epoch: [199/200], lr: 0.000000, Loss: 5.9393 100% | 312/312 [00:23<00:00, 13.14it/s]
...
wandb: Run history:
wandb: R@1
wandb: R@10
wandb: R@100
wandb: R@2
wandb: R@5
wandb: R@50
wandb: Run summary:
wandb: R@1 0.91338
wandb: R@10 0.99533
wandb: R@100 0.99863
wandb: R@2 0.95938
wandb: R@5 0.97625
wandb: R@50 0.996
wandb: 
wandb: ★ View run train\_arch\_\[resnet18\]\_data\[cifartoy\_bad\].epochs\[200\].memorysize\[8192\].mode\[maskcon\].contrastive\_temperature\[0.1\].temperature\_maskcon\[0.05\].weight\[1.0\]
wandb: Synced 5 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
wandb: Find logs at: ./wandb/run-20230815_201032-5h30mlmt7/logs
(maskcon_env) cs24mtech12012@cse-node009i-/MaskCon_CVPR2023
(maskcon_env) cs24mtech12012@cse-node009i-/MaskCon_CVPR2023
(maskcon_env) 0:bash*
```

Comparison: MaskCon Paper vs Reproduced(Good Split)

CIFARtoy-Good split(Scale: 0 to 1)

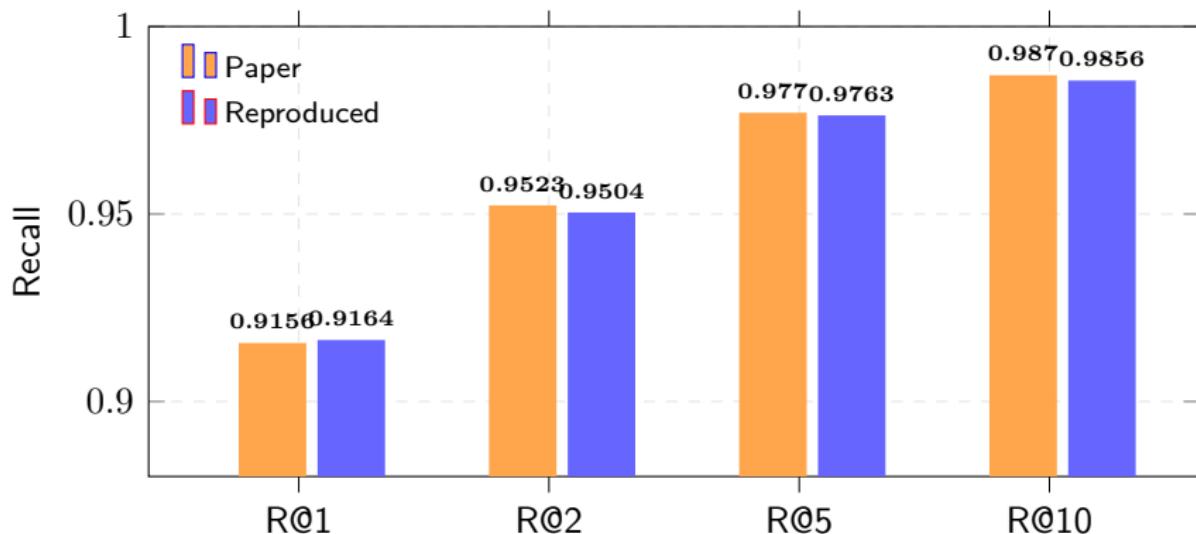


Setup

Epochs = 200, Temperature = 0.05, Weight $w_1 = 1.0$, Train size= 40k,
Test size= 8k

Comparison: MaskCon Paper vs Reproduced (Bad Split)

CIFARtoy-badsplit (Scale: 0 to 1)



Setup

Epochs = 200, Temperature = 0.05, Weight $w_1 = 1.0$, Train size= 40k,
Test size= 8k

CIFAR100: Paper Results & Reproduced Results

Paper Results

Method	Recall@1	Recall@2	Recall@5	Recall@10
SelfCon	40.50	51.83	66.23	76.66
Grafit	60.57	71.13	82.32	89.21
SupCon	58.65	70.04	82.18	89.09
CoIns	60.10	70.89	83.14	89.52
SupCE	47.25	61.24	77.78	87.01
SupFINE	71.13	80.03	87.61	91.59
MaskCon (Ours)	65.52 (18.17↑)	74.46	83.64	89.25

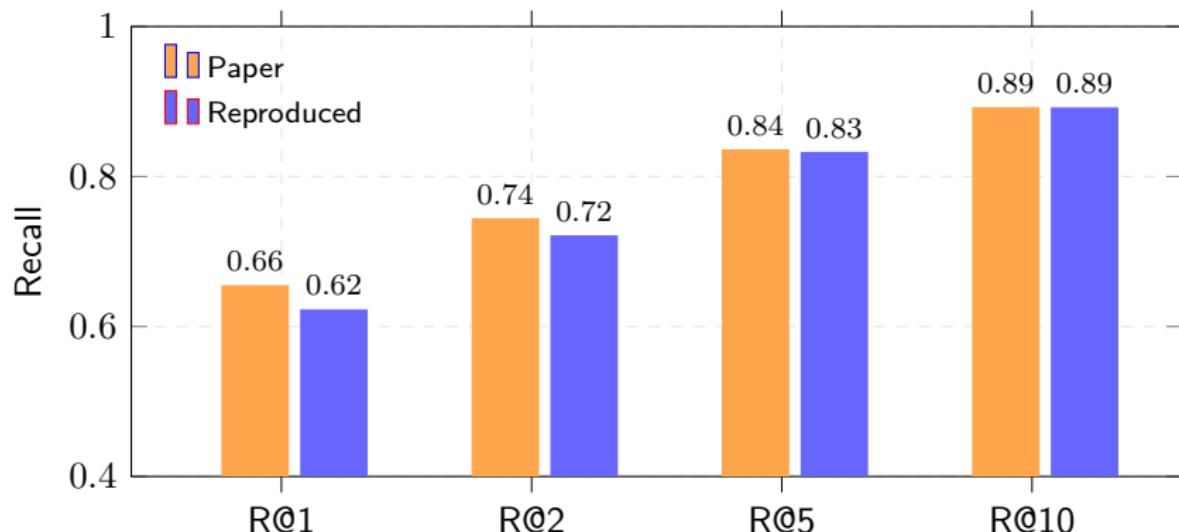
Table 2. Results on CIFAR100 dataset.

Reproduced Results

```
Train Epoch: [199/200], lr: 0.000000, Loss: 6.1147; 100%|██████████| 0/0 [00:00, 0.000000]
wandb: 
wandb: Run history:
wandb: R01
wandb: R02
wandb: R03
wandb: R04
wandb: R05
wandb: R06
wandb: 
wandb: Run summary:
wandb: R01 0.6228
wandb: R02 0.8922
wandb: R03 0.8919
wandb: R04 0.7216
wandb: R05 0.8328
wandb: R06 0.9649
wandb: 
wandb: View run train.arch[resnet18].data[cifar100].epochs[200].memorysize[8192].model[maskcon].contrastive_temperature[0.1].temperature_maskcon[0.1].weight[1.0]
wandb: ★ View project at: https://wandb.ai/cz2hnten14011/lit-hyperparametermaskcon
wandb: Synced 5 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
wandb: Find logs at: https://wandb.ai/cz2hnten14011/lit-hyperparametermaskcon/logs
A   M   A   I   N   C   O   D   E   S   ↵   m   v   o   n   ↵   ↵   ↵
```

Comparison: MaskCon Paper vs Reproduced

CIFAR100 (Scale: 0 to 1)



Setup

Epochs = 200, Temperature = 0.05, Weight $w_1 = 1.0$, Train size= 50k,
Test size= 10k

Novelty: Proposed Methodology

Key Idea

Can we make the model learn better by leveraging the information associated with the binary vectors corresponding to every data point?

Caltech-UCSD Birds-200-2011

```
12 012.Yellow_headed_Blackbird  
13 013.Bobolink  
14 014.Indigo_Bunting  
15 015.Lazuli_Bunting  
16 016.Painted_Bunting  
17 017.Cardinal  
18 018.Spotted_Catbird  
19 019.Gray_Catbird  
20 020.Yellow_breasted_Chat  
21 021.Eastern_Towhee  
22 022.Chuck_will_Widow  
23 023.Brandt_Cormorant  
24 024.Red_faced_Cormorant  
25 025.Pelagic_Cormorant  
26 026.Bronzed_Cowbird  
27 027.Shiny_Cowbird  
28 028.Brown_Creeper  
29 029.American_Crow  
30 030.Fish_Crow  
31 031.Black_billed_Cuckoo
```

```
48 048.European_Goldfinch  
49 049.Boat_tailed_Grackle  
50 050.Eared_Grebe  
51 051.Horned_Grebe  
52 052.Pied_billed_Grebe  
53 053.Western_Grebe  
54 054.Blue_Grosbeak  
55 055.Evening_Grosbeak  
56 056.Pine_Grosbeak  
57 057.Rose_breasted_Grosbeak  
58 058.Pigeon_Guillemot  
59 059.California_Gull  
60 060.Glaucous_winged_Gull  
61 061.Heermann_Gull  
62 062.Herring_Gull  
63 063.Ivory_Gull  
64 064.Ring_billed_Gull  
65 065.Slaty_backed_Gull  
66 066.Western_Gull  
67 067.Anna_Hummingbird
```

Novelty: Caltech-UCSD Birds-200-2011 (CUB dataset)

- A collection of 11,788 images spanning 200 bird species.
- Each image is annotated with detailed attributes, including part locations and bounding boxes.
- Binary attributes for more precise identification and analysis.

```
143 has_eye_color::green  
144 has_eye_color::pink  
145 has_eye_color::orange  
146 has_eye_color::black  
147 has_eye_color::white
```

```
10 has_wing_color::blue  
11 has_wing_color::brown  
12 has_wing_color::iridescent  
13 has_wing_color::purple
```

```
11788 294 0  
11788 295 0  
11788 296 0  
11788 297 0  
11788 298 0  
11788 299 1  
11788 300 0  
11788 301 0  
11788 302 0  
11788 303 0  
11788 304 0  
11788 305 0  
11788 306 1  
11788 307 0  
11788 308 1  
11788 309 0  
11788 310 0  
11788 311 0  
11788 312 1
```

312-dimensional binary vector
for every data point.

Total Class = 200 bird species

Total images= 11,788

Fine labels Available

Novelty: Method

Our proposed method:

- Create vector representation for each image, given binary valued attributes.
- Find a similarity metric between each image attribute vector within the same coarse label.
- Obtain a probability distribution using softmax function.
- Take convex combination of the obtained probabilities with the vector Z'_{ij} .
- Use refined probabilities in the original MaskCon loss function to obtain fine labels.

Metrics to capture similarity:

- **Cosine similarity:** Dot product between similar species should be higher.
- **Hamming distance:** The difference between dissimilar data points is higher.
- **Euclidean distance:** Similar points have less distance between them.

Novelty: Adding the newly obtained information to the MaskCon loss

Augmenting Similarity into Loss

A convex combination of similarity measures Z'_{ij} and Z'_{ij_new} is used to augment the loss function:

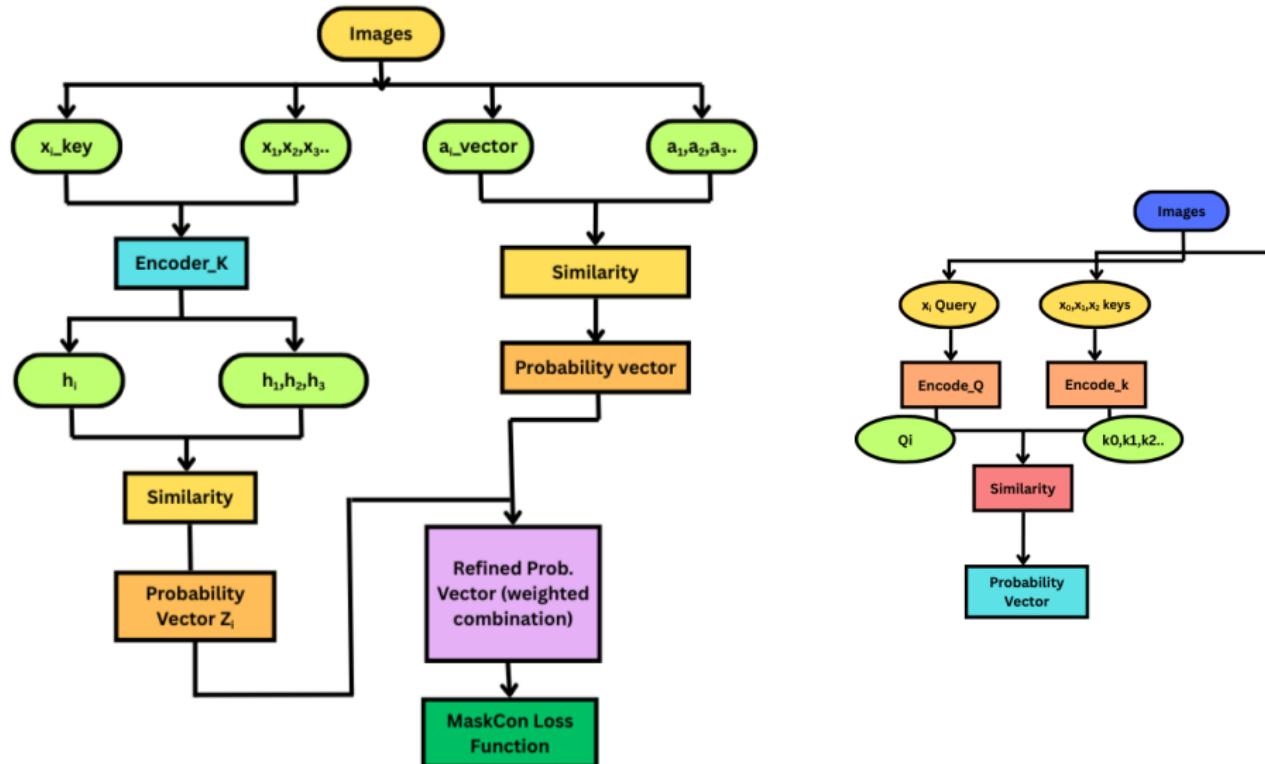
$$Z'_{ij_final} = w \cdot Z'_{ij} + (1 - w) \cdot Z'_{ij_new}, \quad 0 < w < 1$$

New loss: Replace Z'_{ij} with Z'_{ij_final} in the contrastive loss function.

Why adding information in the loss function is more intuitive?

To guide initial probabilities of Z_{ij} with Z_{ij}^{new} , which in turn helps the model better learn the similarities between data points that belong to the same coarse labels.

Novelty: End to End MaskCon with our Addition



Novelty: Results CUB Dataset (20 Classes)

MaskCon



$w_1 = 1.0, w_2 = 0.0$

Our Novelty



$w_1 = 0.5, w_2 = 0.5$

SupFINE



Supervised learning using
the true fine labels

Common Setup

ResNet-50 , Temperature = 0.5, Epochs = 100, Weight=1.0 Train=600 ,
Test=515

Novelty: Results CUB Dataset (50 Classes)

MaskCon

```
wandb: Run history:  
wandb: R@1 [REDACTED]  
wandb: R@10 [REDACTED]  
wandb: R@100 [REDACTED]  
wandb: R@2 [REDACTED]  
wandb: R@5 [REDACTED]  
wandb: R@50 [REDACTED]  
wandb:  
wandb: Run summary:  
wandb: R@1 0.69258  
wandb: R@10 0.9568  
wandb: R@100 0.9964  
wandb: R@2 0.79266  
wandb: R@5 0.90065  
wandb: R@50 0.99352
```

Our Novelty

```
wandb: Run history:  
wandb: R@1 [REDACTED]  
wandb: R@10 [REDACTED]  
wandb: R@100 [REDACTED]  
wandb: R@2 [REDACTED]  
wandb: R@5 [REDACTED]  
wandb: R@50 [REDACTED]  
wandb:  
wandb: Run summary:  
wandb: R@1 0.71898  
wandb: R@10 0.95896  
wandb: R@100 0.99784  
wandb: R@2 0.82126  
wandb: R@5 0.91997  
wandb: R@50 0.9892
```

SupFINE

```
wandb: Run history:  
wandb: R@1 [REDACTED]  
wandb: R@10 [REDACTED]  
wandb: R@100 [REDACTED]  
wandb: R@2 [REDACTED]  
wandb: R@5 [REDACTED]  
wandb: R@50 [REDACTED]  
wandb:  
wandb: Run summary:  
wandb: R@1 0.80922  
wandb: R@10 0.96616  
wandb: R@100 0.99928  
wandb: R@2 0.87113  
wandb: R@5 0.93161  
wandb: R@50 0.9964
```

$$w_1 = 1.0, w_2 = 0.0$$

$$w_1 = 0.5, w_2 = 0.5$$

Supervised learning using
the true fine labels

Common Setup

ResNet-50 , Temperature = 0.5, Epochs = 100, Weight=1.0 Train=1500 ,
Test=1389

Novelty: Results CUB Dataset (200 Classes)

MaskCon

```
wandb: Run history:  
wandb: R@1 [REDACTED]  
wandb: R@10 [REDACTED]  
wandb: R@100 [REDACTED]  
wandb: R@2 [REDACTED]  
wandb: R@5 [REDACTED]  
wandb: R@50 [REDACTED]  
wandb:  
wandb: Run summary:  
wandb: R@1 0.59821  
wandb: R@10 0.90352  
wandb: R@100 0.98257  
wandb: R@2 0.7154  
wandb: R@5 0.84329  
wandb: R@50 0.97204
```

$$w_1 = 1.0, w_2 = 0.0$$

Our Novelty

```
wandb: Run history:  
wandb: R@1 [REDACTED]  
wandb: R@10 [REDACTED]  
wandb: R@100 [REDACTED]  
wandb: R@2 [REDACTED]  
wandb: R@5 [REDACTED]  
wandb: R@50 [REDACTED]  
wandb:  
wandb: Run summary:  
wandb: R@1 0.62997  
wandb: R@10 0.91538  
wandb: R@100 0.98205  
wandb: R@2 0.73136  
wandb: R@5 0.85518  
wandb: R@50 0.97273
```

$$w_1 = 0.5, w_2 = 0.5$$

SupFINE

```
wandb: Run history:  
wandb: R@1 [REDACTED]  
wandb: R@10 [REDACTED]  
wandb: R@100 [REDACTED]  
wandb: R@2 [REDACTED]  
wandb: R@5 [REDACTED]  
wandb: R@50 [REDACTED]  
wandb:  
wandb: Run summary:  
wandb: R@1 0.79013  
wandb: R@10 0.92837  
wandb: R@100 0.98999  
wandb: R@2 0.84777  
wandb: R@5 0.89972  
wandb: R@50 0.9805
```

Supervised learning using
the true fine labels

Common Setup

ResNet-50 , Temperature = 0.5, Epochs = 100, Weight=1.0 Train=5994 ,
Test=5794

Novelty Results Across Class Settings

Classes	Method	Recall@1	Recall@2	Recall@5
20	MaskCon	0.8019	0.8835	0.9611
	Novelty	0.8267	0.8935	0.9123
	SupFINE	0.9068	0.9320	0.9553
50	MaskCon	0.6925	0.7926	0.9006
	Novelty	0.7189	0.8212	0.9199
	SupFINE	0.8092	0.8711	0.9316
200	MaskCon	0.5982	0.7154	0.8433
	Novelty	0.6300	0.7314	0.8555
	SupFINE	0.7901	0.8477	0.8997

Implementation Challenges

Challenges Faced During Implementation

- **Colab Limitations:**

Training on 200-class datasets (e.g., CUB) was slow and often interrupted due to limited GPU runtime and memory.

- **Missing Coarse Labels:**

Datasets like CUB and Cars196 lacked coarse labels. Manual grouping based on domain knowledge was required.

- **Long Training Time:**

MaskCon's contrastive learning setup significantly increased training time for fine-grained classes.

- **Hyperparameter Sensitivity:**

Model performance was sensitive to the choice of w and temperature τ , requiring repeated tuning.

Future Work

Planned Extensions and Improvements

- **Generalization to Other Datasets:**

Extend MaskCon to other coarse-labeled or weakly supervised datasets like ImageNet-Coarse.

- **Automated Coarse Labeling:**

Use clustering or external metadata to generate coarse labels automatically.

- **Improved Similarity Functions:**

Explore learned or adaptive metrics beyond cosine, Hamming, and Euclidean.

- **Scalability Enhancements:**

Support larger models and distributed training for higher-resolution, high-class datasets.

References

- Feng, Chen, and Ioannis Patras.
MaskCon: Masked Contrastive Learning for Coarse-Labelled Dataset.
CVPR 2023.
[\[Link\]](#)
- David Berthelot et al.
MixMatch: A Holistic Approach to Semi-Supervised Learning.
arXiv:1905.02249, 2019.
[\[PDF\]](#)
- Guy Bukchin et al.
Fine-Grained Angular Contrastive Learning with Coarse Labels.
CVPR 2021, pp. 8730–8740.
[\[Link\]](#)
- Mathilde Caron et al.
Deep Clustering for Unsupervised Learning of Visual Features.
ECCV 2018, pp. 132–149.
[\[PDF\]](#)

Thank You

THANK YOU!