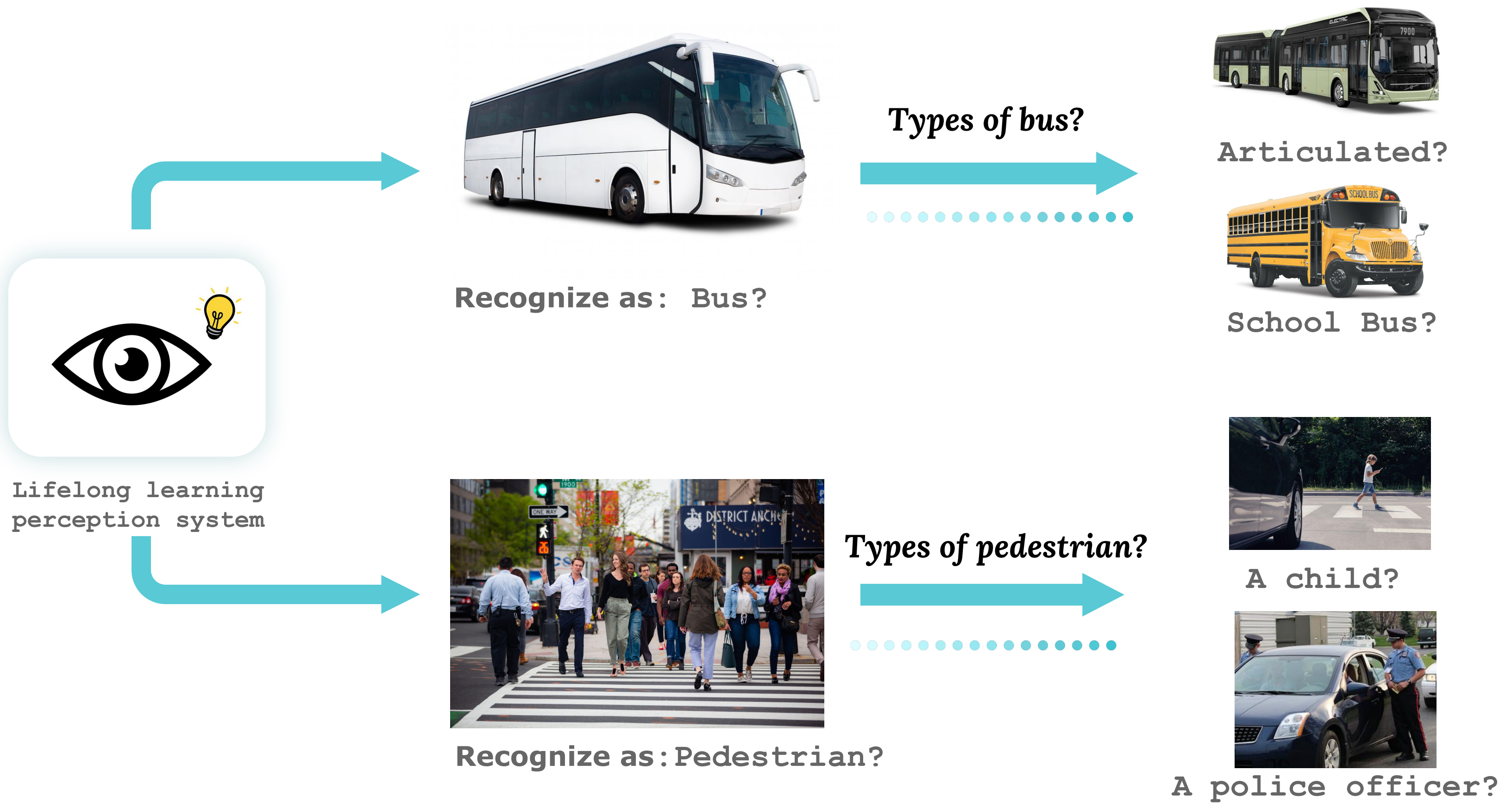
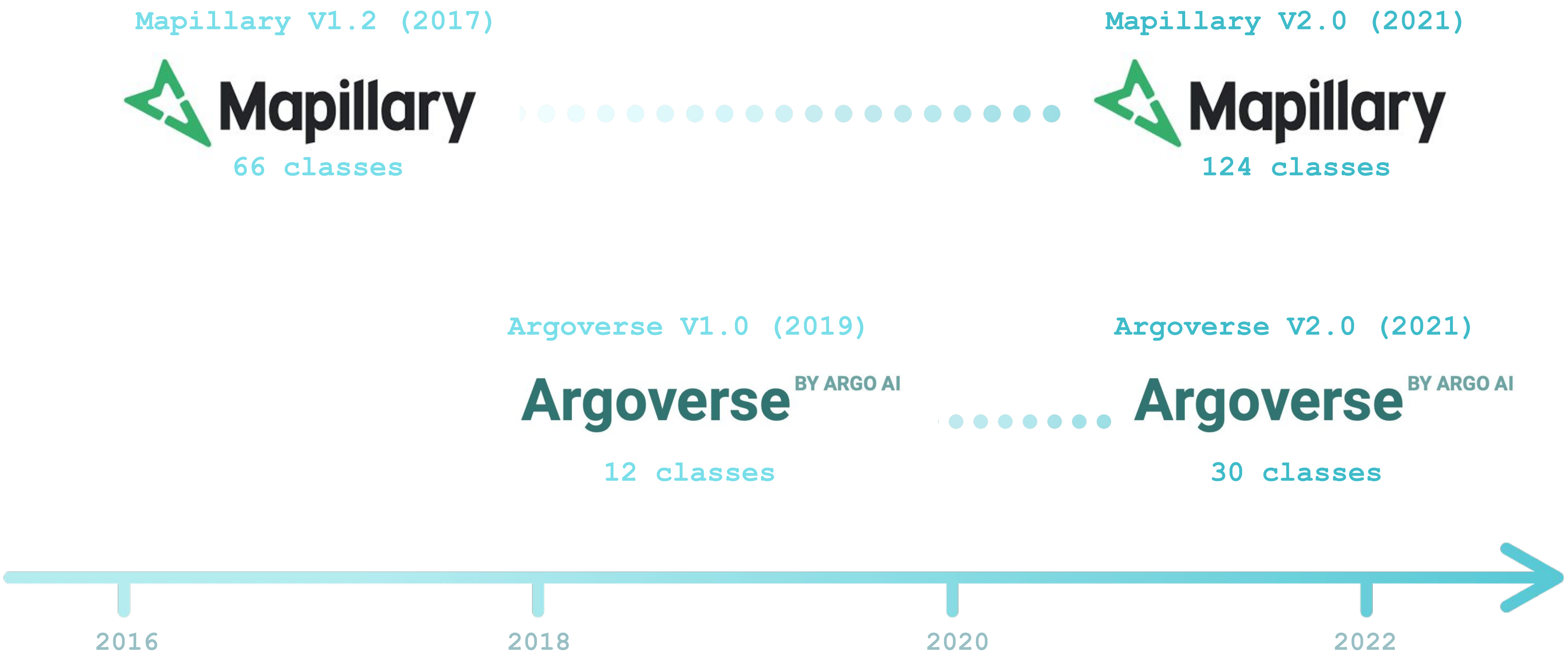
Learning with an Evolving Class Ontology

Zhiqiu Lin, Deepak Pathak, Yu-Xiong Wang, Deva Ramanan*, Shu Kong*

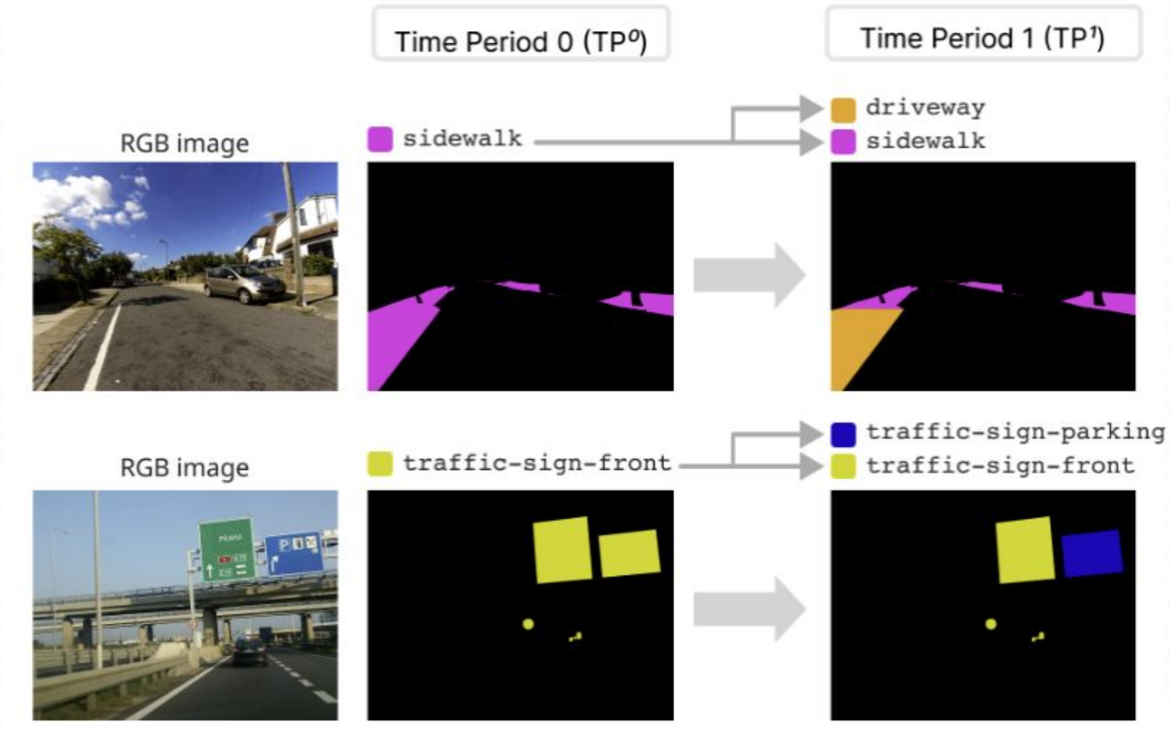
Visual perception systems need to cope with **evolving class ontology**..



Contemporary industry-made datasets, such as Mapillary[1] and Argoverse[2], continually refined the ontology from version 1.0 to 2.0.

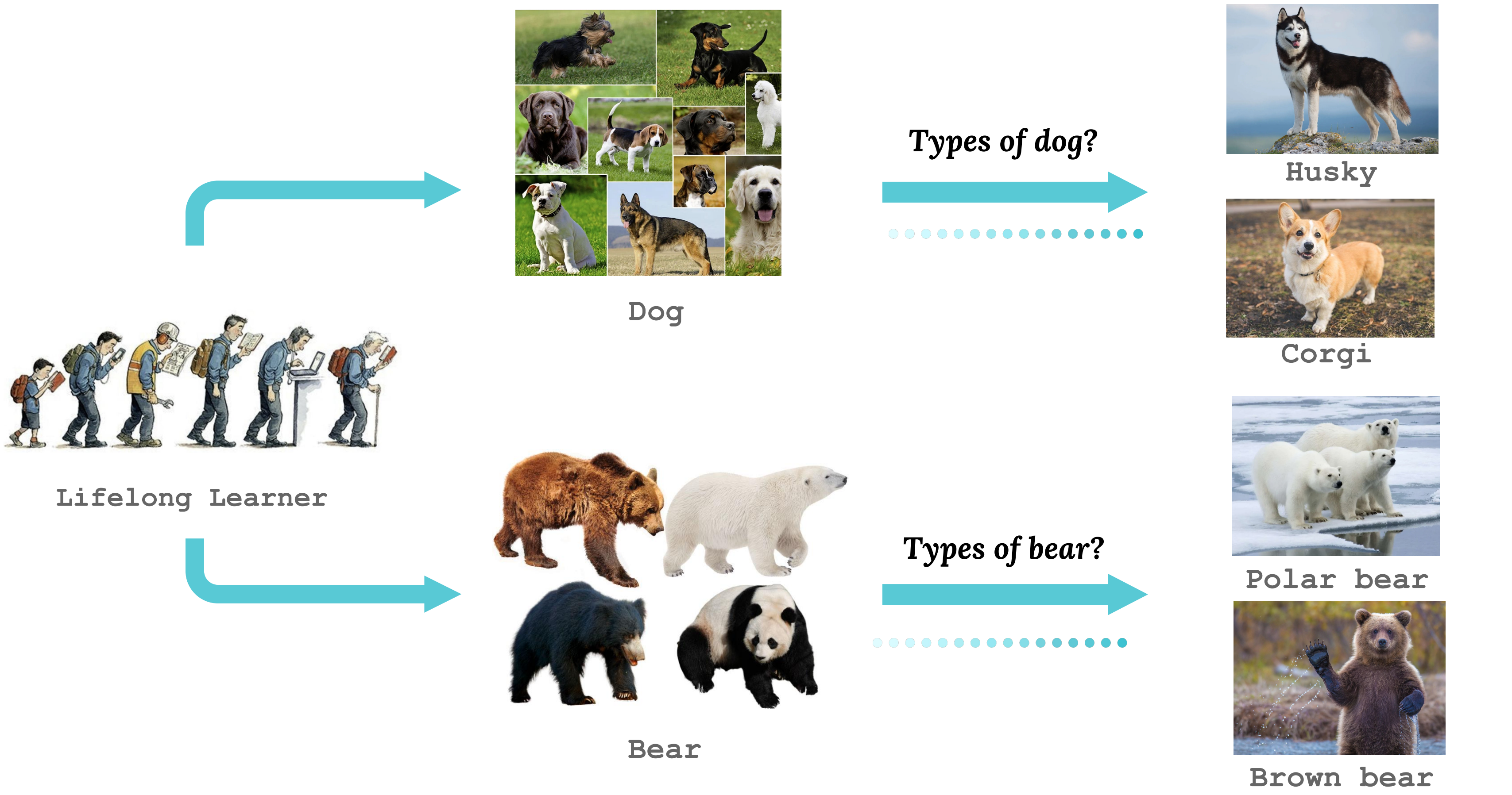


We study the problem of **LECO**: **L**earning with an **E**volving **C**lass **O**ntology.
Each time period (TP) of a LECO problem refines the class ontology:

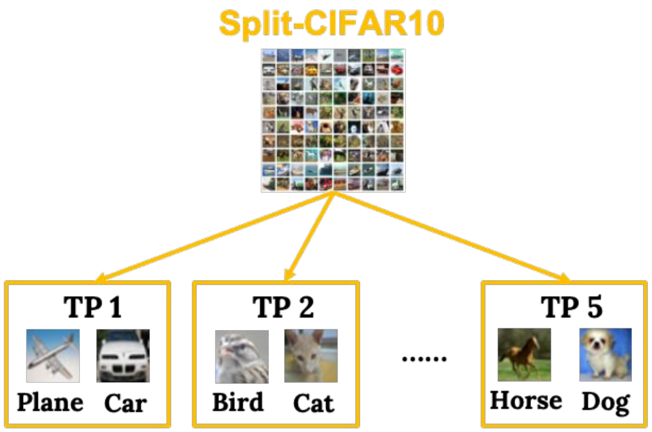


Mapillary-LECO* Benchmark**: Real-world ***dataset versioning scenario from V1.2(TP^0) to V2.0(TP^1).

Humans, as **lifelong learners**, are also good at solving LECO problems.



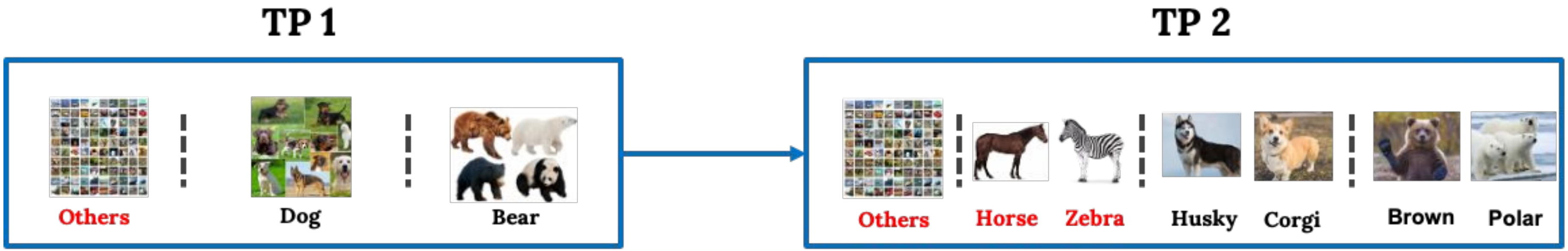
Class-Incremental Learning (CIL) v.s. LECO



In CIL, classes are disjoint across TPs, i.e., having no relationship with each other.

In LECO, the newly added classes are always refined ones from last TP.

Class-Incremental Learning (CIL) v.s. LECO



Note: This **Others** class is sometimes called "**Unlabeled**" or "**Void**" in many datasets [1, 2].

LECO is a more general form of CIL (class-incremental learning) problem by assuming a catch-all *Others* class.

[1] The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In ICCV 2017.
[2] The Cityscapes Dataset for Semantic Urban Scene Understanding. In CVPR 2016.

Class-Incremental Learning (CIL) v.s. LECO

In **CIL**:

- Training data from previous TPs will be **discarded.**
- Overall performance measured by **testsets of previous + current TPs**.

In **LECO**:

- **Keeping all history data** because storage is cheaper than annotation.
- Overall performance measured by the **testset of current TP only**.

LECO targets at practical applications by preserving all data (without setting an artificial small replay buffer).

Learning with an Evolving Class Ontology

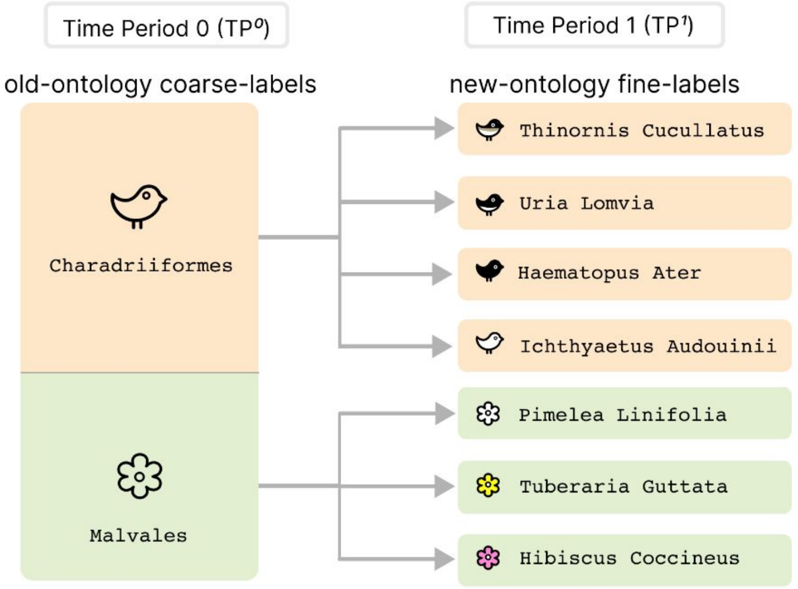
→ "LECO" benchmark for lifelong vision

Benchmark Construction

LECO-segmentation (Mapillary V1.2 -> V2.0)

Mapillary-LECO* Benchmark**: Real-world ***dataset versioning scenario from V1.2(TP^0) to V2.0(TP^1).

LECO-classification (iNaturalist/CIFAR)



iNat-LECO* Benchmark**: Long-tailed class ontology ***evolves from coarse to fine.

Question 1: Should one label new data, or relabel old data?

Mapillary V1.2 (2017)

Mapillary

Mapillary V2.0 (2021)

Mapillary

Same images, but relabeled!
(RelabelOld)

Argoverse V1.0 (2019)

Argoverse BY ARGO AI

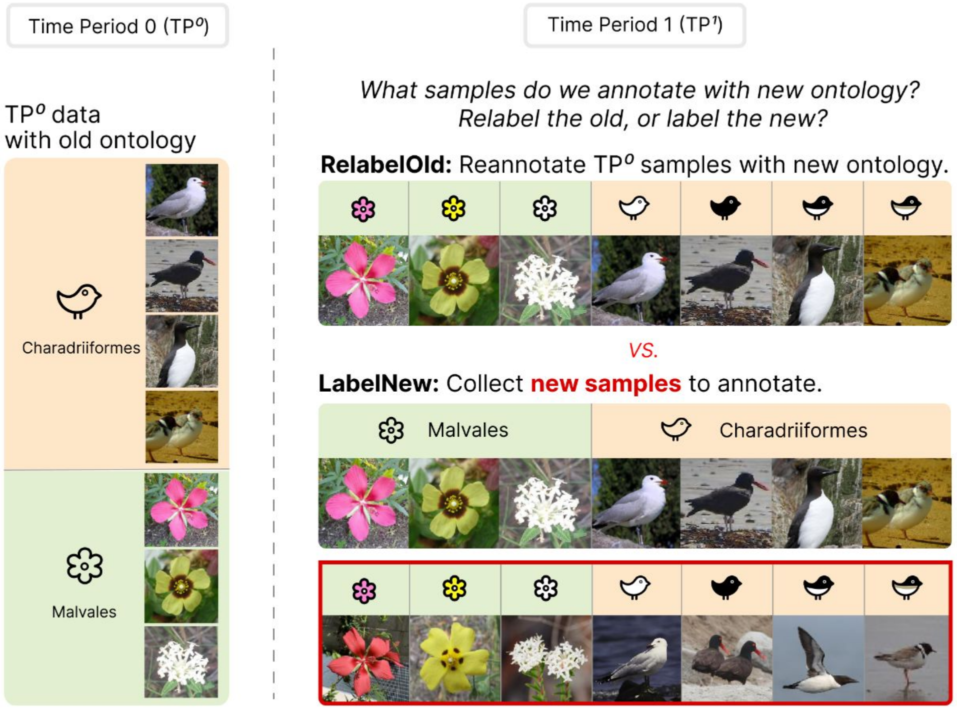
Argoverse V2.0 (2021)

Argoverse BY ARGO AI

Collect new data to label!
(LabelNew)

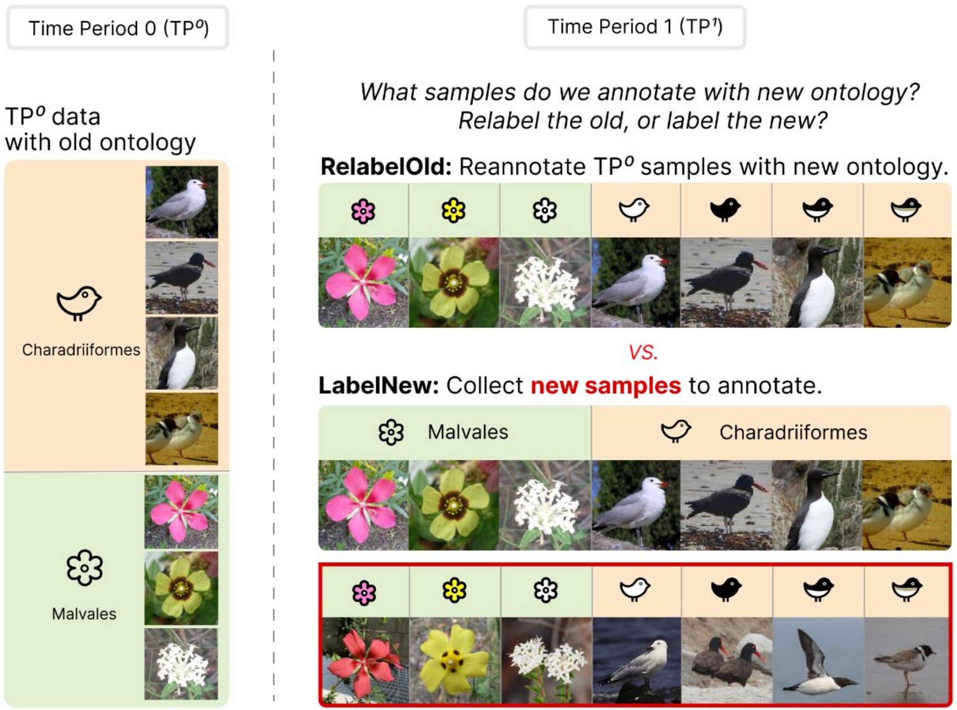
2016 2018 2020 2022

Question 1: Should one label new data, or relabel old data?



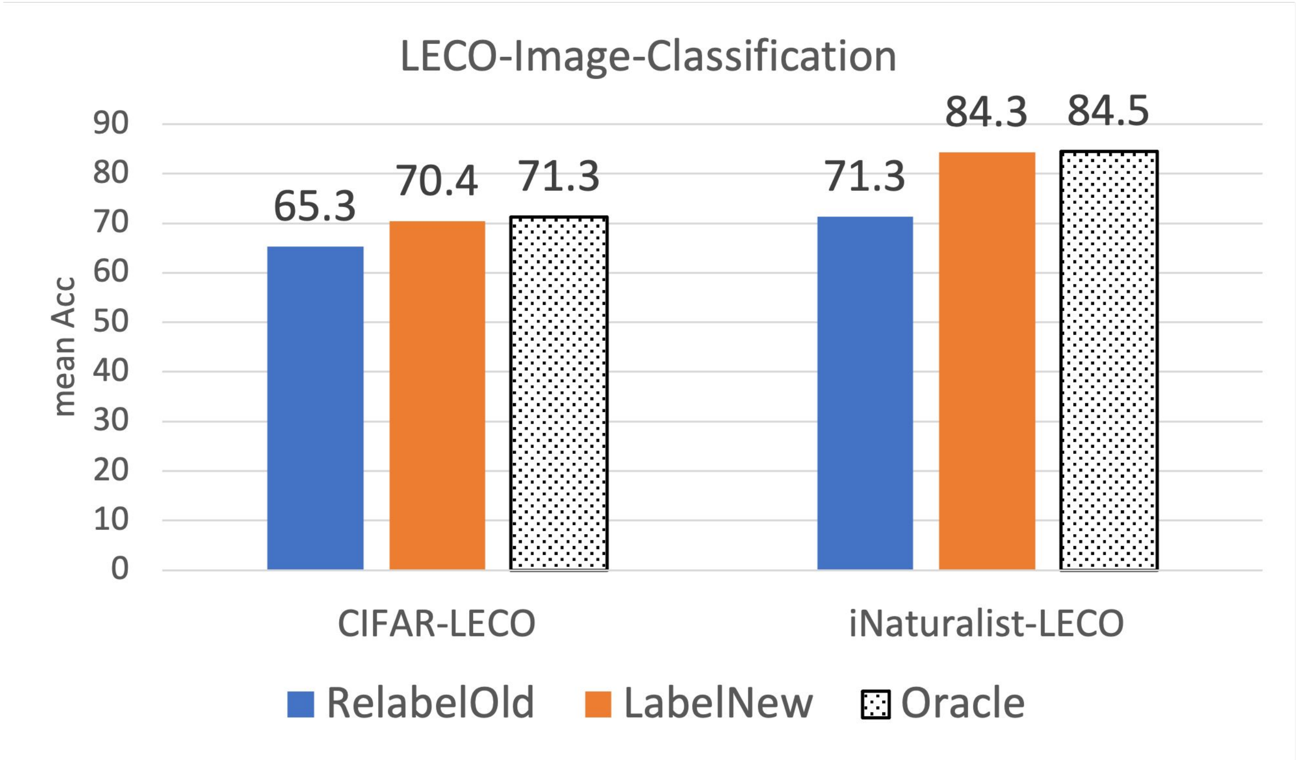
Key insight:
LabelNew will produce more data for training (though with inconsistent labels)

Question 1: Should one label new data, or relabel old data?



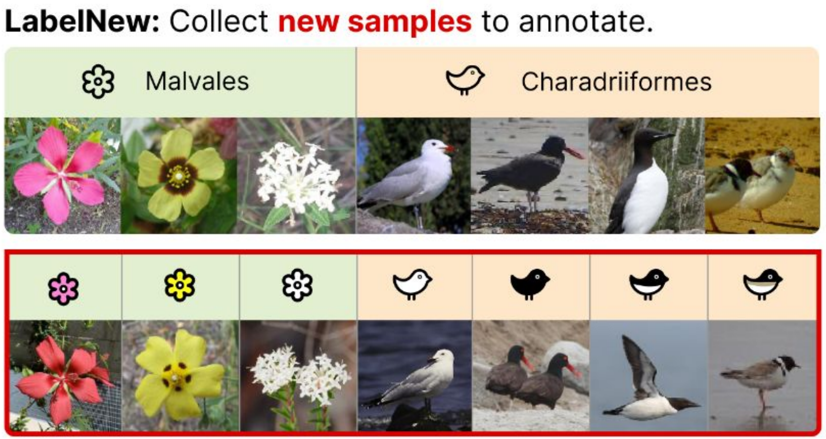
Key insight:
LabelNew produces more data for training!

Question 1: Should one label new data, or relabel old data?



Takeaway: LabelNew produces a better classifier for training on more overall data.

Question 2: How to train on data with <u>both coarse- and fine-grained labels</u>?

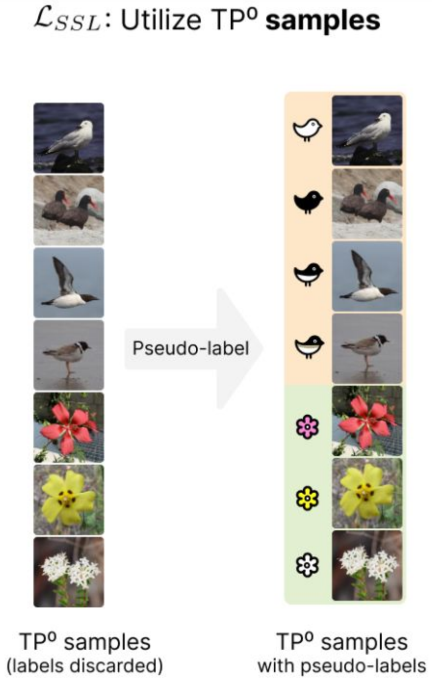


Our proposals:

1. Discard old-ontology labels and only use data.
2. Train on both coarse- and fine-grained labeled data.
3. Exploit the coarse-to-fine label hierarchy.

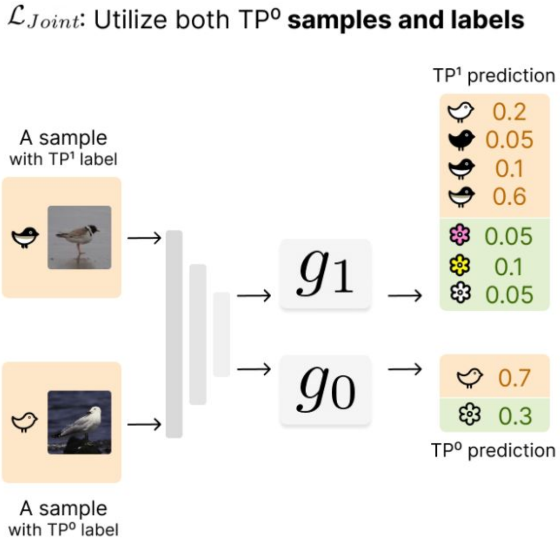
Question 2: How to train on data with both coarse- and fine-grained labels?

Proposal 1: Discard old-ontology labels and only use data.
⇒ Semi-supervised learning (SSL)



Question 2: How to train on data with both coarse- and fine-grained labels?

Proposal 2: Train on both coarse- and fine-grained labeled data.
⇒ Joint Training

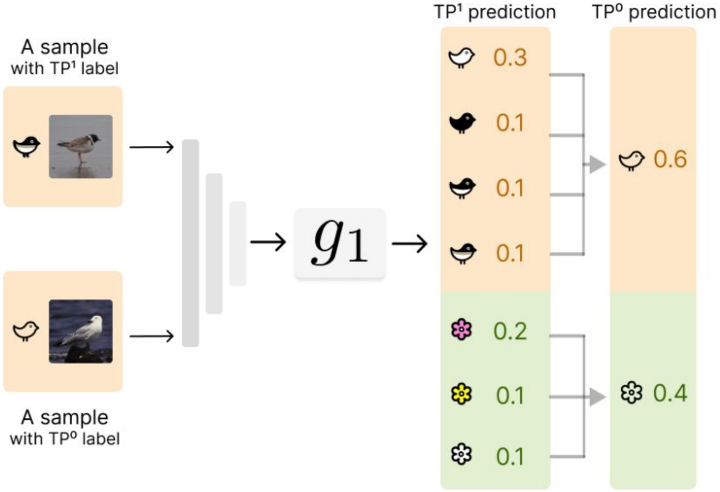


JointTraining with a two headed model

Question 2: How to train on data with both coarse- and fine-grained labels?

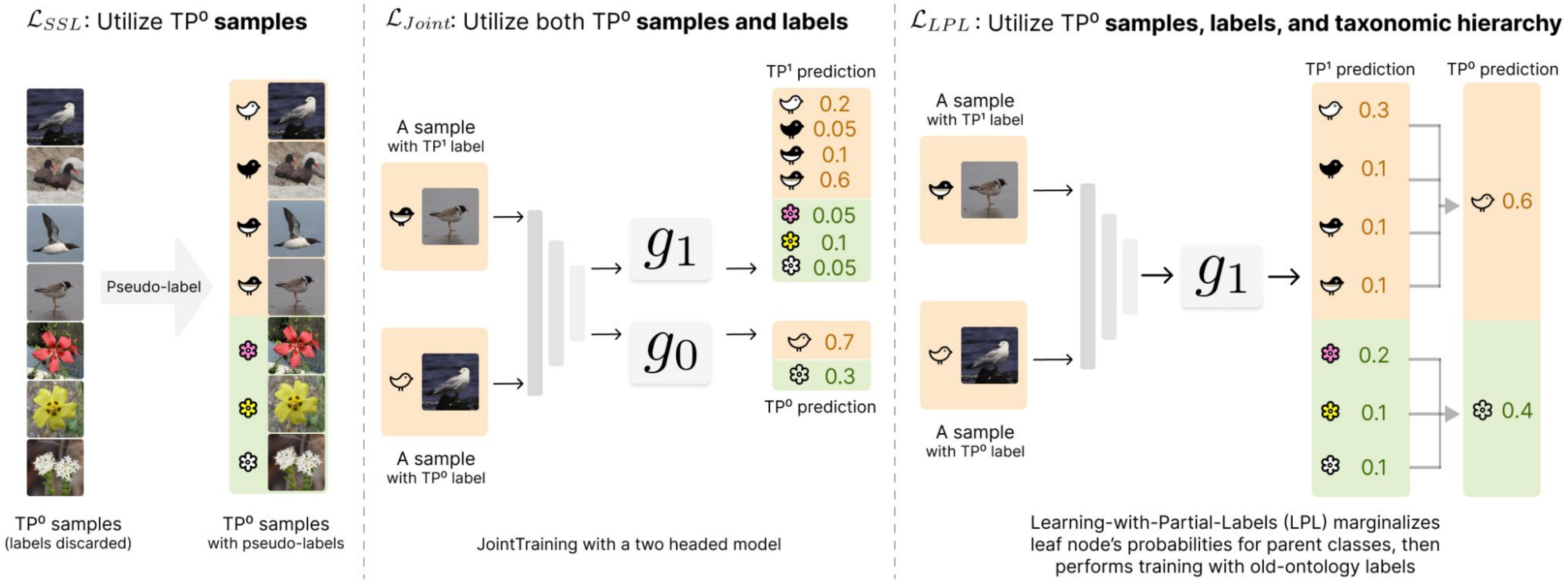
Proposal 3: Exploiting coarse-to-fine label hierarchy.
⇒ Learning-with-Partial-Labels (LPL)

$\mathcal{L}_{LPL}$: Utilize TP⁰ **samples, labels, and taxonomic hierarchy**



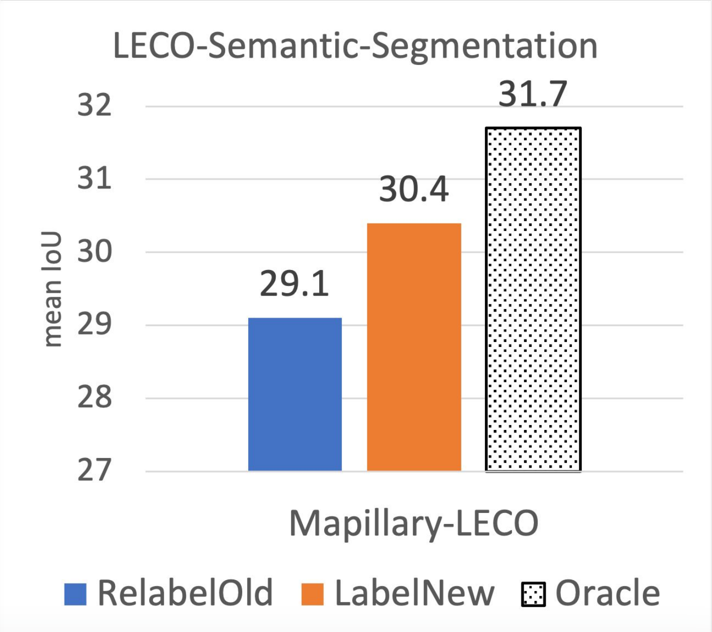
Learning-with-Partial-Labels (LPL) marginalizes leaf node's probabilities for parent classes, then performs training with old-ontology labels

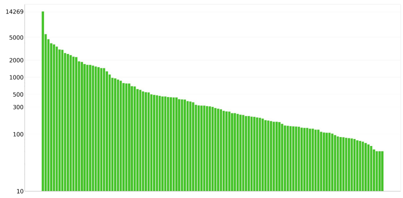
Question 2: How to train on data with both coarse- and fine-grained labels?

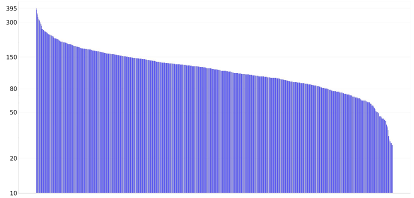


Check out the paper for comprehensive ablation results!

Question 3: Do our proposals generalize to real-world scenarios?







Our solutions generalize to real-world LECO scenario (Mapillary) without given the label hierarchy.

We show consistent improvements under:

- **Long-tailed distribution (Mapillary/iNaturalist)**
- **More than 2 TPs (iNaturalist)**

Thank You!

Scan for our site!