Logic-induced Diagnostic Reasoning for Semi-supervised Semantic Segmentation
Supplemental Material

Chen Liang, Wenguan Wang, Jiaxu Miao, Yi Yang*

ReLER, CCAI, Zhejiang University

https://github.com/leonnnop/LogicDiag

- §S1: Detailed introduction of First-order Logic.
- §S2: Detailed LogicDiag algorithm.
- §S3: Detailed label hierarchy.
- §S4: More experimental results.
- §S5: More qualitative visualization.
- §S6: Discussion of legal/ethical considerations.

S1. Detailed Introduction of First-order Logic

First-order logic (FOL), also known as predicate logic, is a formal language system used for representing and reasoning about statements involving objects and their properties. FOL extends propositional logic by introducing predicates, which are functions that take one or more arguments and return a truth value. The syntax of FOL is defined by a set of symbols, including variables, constants, function symbols, predicate symbols, and logical connectives. The basic components of FOL are (please also refer to §3.2.1):

• *Constants:* Objects that are always present and have a fixed interpretation. For example, a specific pixel x_i.
• *Variables:* Symbols that represent objects whose identity is not specified. For example, pixel data sample x.
• *Quantifiers:* Symbols that specify the extent of a statement's applicability, either 'for all' ($\forall$) or 'there exists' ($\exists$).
• *Connectives:* Symbols that combine statements to form more complex ones, *e.g.*, negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), implication ($\rightarrow$), and biconditional ($\leftrightarrow$).
• *Predicates:* Expressions that assert a relationship between objects. For example, $Cat(x)$: x is a Cat.

In FOL, a formula is formed by combining atomic formulas using logical connectives and quantifiers. An atomic formula is typically a predicate applied to a set of terms, and takes the form $P(t_1, \dots, t_n)$, where P is a predicate symbol of arity n and $t_1, \dots, t_n$ are terms, which may be variables, constants, or function symbols applied to terms. The semantics of FOL are determined by a truth function that assigns a truth value to each formula based on the values of its subformulas and the domain of discourse. As an example, in our main paper, we define a unary predicate $t(\cdot)$ that takes a single input o and evaluates whether o is logically consistent with the symbolic knowledge. The truth value of a formula of the form $\forall o, t(o)$ is true if and only if $t(o)$ is true for all objects in the domain of discourse.

Algorithm 1 LogicDiag Pseudocode, PyTorch-like style

```
# H: neural network predictor
# K: knowledge base of first-order logic rules
# L: loss function, e.g., binary cross entropy loss
# m: trade-off hyperparameter

# load minibatch from labeled and unlabeled dataset
# s_x, u_x: labeled and unlabeled datapoints
# o_s: ground-truth labels
for (s_x, o_s), u_x in loader:
  # Pose assumptions on observations
  p_s_o = H(s_x).sigmoid() # supervised prediction
  p_u_o = H(u_x).sigmoid() # unsupervised
      prediction

  loss_s = L(p_s_o, o_s) # supervised loss

  p_o = CONCAT([p_s_o, p_u_o])
  o = p_o.binarize() # original pseudo label

  # Compute diagnoses (Eq.4-5)
  Omega = DIAGNOSE(o, K)

  # Calculate diagnosis likelihood (Eq.7,8,11)
  p_Omega = FUZZY_LIKELIHOOD(Omega, p_o, K)
  omega = SAMPLING(p_Omega) # sample from
      likelihood

  # Resolve conflicts within pseudo labels (Eq.6)
  o_r = RESOLUTION(o, omega)

  # Calculate unsupervised loss on revised labels
  loss_u = L(p_o, o_r)

  # Calculate overall training objective (Eq.3)
  loss = loss_s + m*loss_u

  # Update the sub-symbolic network parameters
  loss.backward()
  update(H.params)
```

S2. Detailed LogicDiag Algorithm

Algorithm 1 provides the pseudocode for LogicDiag. By integrating symbolic reasoning into the neural learning process, LogicDiag enhances the model's ability to

*Corresponding author: Yi Yang.

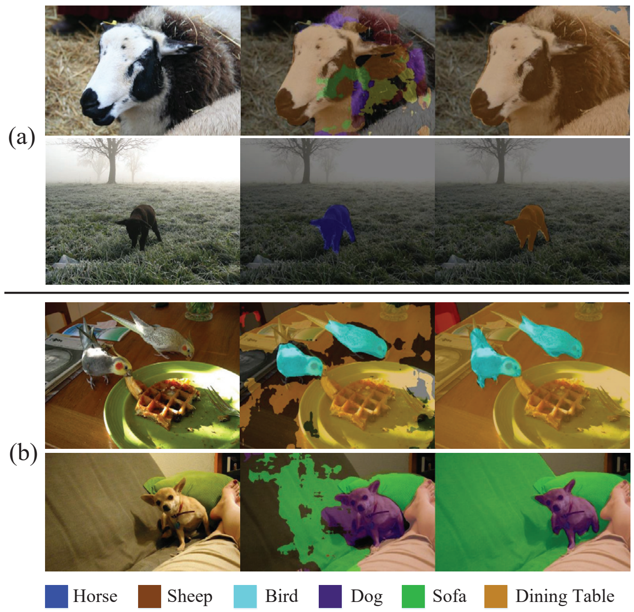

Figure S1: **Pseudo labels** obtained with confidence thresholding (middle) and LogicDiag (right). There are two key limitations associated with confidence thresholding. (a) Confidence thresholding is not able to rectify potentially incorrect predictions, leading to the generation of inaccurate pseudo-labels. For example, as shown in the second column, the error accumulation ultimately leads to the misclassification of sheep as a cow; (b) The typically high threshold used in confidence thresholding results in a significant reduction in the number of generated pseudo-labels.

align with background knowledge, correct conflicting predictions, and improves its predictive accuracy in SSL tasks.

S3. Detailed Label Hierarchy

In this paper, we leverage the label hierarchy present in each dataset to derive the logic rules. To this end, we leverage the official structured label hierarchies for the PASCAL VOC 2012[1] [1], Cityscapes[2] [2], and COCO[3] [3] datasets. To represent the most general concept, we introduce a virtual root node labeled `Root`. The detailed hierarchies for PASCAL VOC 2012, Cityscapes, and COCO are illustrated in Fig. S3, Fig. S4, and Fig. S5, respectively.

S4. More Experimental Results

Semantics of Label Hierarchy. We further examine the impact of hierarchical structure, which is used in deriving the Composition, Decomposition, and Exclusion rules of visual concepts. By default, we use the official label hierarchies defined in PASCAL VOC 2012, Cityscapes, and COCO (*c.f.* §S3). We additionally explore an alternative where a random hierarchy was constructed with the same number of superclasses and classes per superclass as the official hierarchy. The results are presented in the table below. For completeness, we also include the baseline without employing our framework. Our findings indicate that when using the randomly constructed hierarchy, the achieved results are only comparable or even slightly worse than the baseline, due to the presence of noisy supervision. This result again shows that the structured semantic concepts and derived set of logic rules are indeed helpful in the semi-supervised learning of semantic segmentation models.

[1] http://host.robots.ox.ac.uk/pascal/VOC/
[2] https://www.cityscapes-dataset.com/dataset-overview/
[3] https://github.com/nightrome/cocostuff

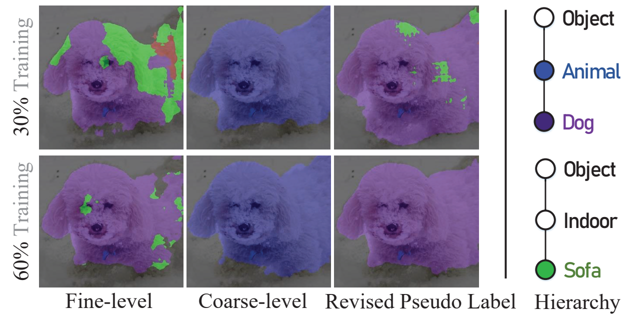

Figure S2: **Pseudo labels** *before* and *after* logic-induced diagnostic reasoning across training iterations.

Label Hierarchy	mIoU²	mIoU¹
None	-	68.02
Random	76.13	67.11
Official	87.91	73.25

Table S1: **Impact of semantics within label hierarchy**, evaluated on PASCAL VOC 2012 [1] `val` with 1/16 *augmented* set.

S5. More Qualitative Visualization

Pseudo-label Quality. In Fig. S1, we present the visualization that compares the pseudo labels generated by our LogicDiag with those obtained through the confidence thresholding method. Our visualization highlights two key advantages of LogicDiag. First, as demonstrated in Fig. S1 (a), the integration of conflict resolution mechanism enables LogicDiag to rectify possibly erroneous predictions, leading to more precise and cohesive pseudo-labels that align with the existing knowledge. Second, Fig. S1 (b) illustrates that the confidence thresholding method generates significantly fewer pseudo-labels compared to the LogicDiag framework. This limitation, in turn, hinders the effectiveness of semi-supervised learning process, which relies on large amounts of high-quality pseudo-labeled data.

Diagnostic Reasoning. To further depict the effect of logic-induced diagnostic reasoning, we illustrate the process w.r.t.

training iterations on one typical example in Fig. S2. Initially (30% training), a substantial portion of the pseudo labels undergo revisions based on label hierarchy, resulting in improved accuracy. As training progresses (60% training), the revised pseudo labels become more accurate, requiring fewer modifications. This demonstrates LogicDiag's efficacy in refining and enhancing pseudo labels iteratively.

Qualitative Results. We illustrate representative qualitative results of our method and confidence thresholding upon the DeepLabV3+ [4], on the PASCAL VOC [1] (Fig. S6), and Cityscapes [2] (Fig. S7). It is evident that LogicDiag produces more accurate predictions attributed to the successful incorporation of symbolic visual semantics, which helps to resolve ambiguous classes and subtle textures that often cause confusion for the baselines.

S6. Discussion

Asset License and Consent. In this work, we study the semi-supervised semantic segmentation problem with three famous semantic segmentation datasets, *i.e.*, PASCAL VOC 2012 [1], Cityscapes [2], and COCO [3] that are all publicly and freely available for academic purposes. All these assets release annotations obtained from human experts with agreements. PASCAL VOC 2012 (`https://groups.csail.mit.edu/vision/datasets/ADE20K/`) is released under the Flickr Terms of use for images and CC BY 4.0 for annotations; Cityscapes (`https://www.cityscapes-dataset.com/`) is released under this License; COCO (`https://cocodataset.org/`) is released under CC BY 4.0 license. We implement all the variants of LogicDiag with modifications on the MMSegmentation [5] codebase (`https://github.com/open-mmlab/mmsegmentation`), which is released under the Apache-2.0 license.

Limitation Analysis. As a very first attempt that demonstrates the power of neural-symbolic computation in large-scale semi-supervised semantic segmentation, our approach inevitably introduces some open problems that need to be acknowledged. In particular, the Monte Carlo approximation (§3.2.2) employed in our approach requires repeated sampling from the probability distribution of interest. It can be inefficient in terms of computational cost. However, we have found that optimizing with only one sample per datapoint is sufficient for achieving global convergence, and this incurs only a minor computational overhead, *i.e.*, $\sim$4.7% training speed delay. Moving forward, we are committed to designing more powerful algorithms that can further improve both the efficiency and efficacy of our approach.

Broader Impact. This work introduces a neural-symbolic framework that not only achieves promising results in terms of model performance, but also enhances the understandability of the pseudo-label generating process. One advantage of this approach is its ability to reduce human labor and energy consumption in the semi-supervised training of segmentation models. The potential real-world applications are broad and varied, including precision agriculture, robot navigation, *etc*. However, one potential drawback of the approach is that the generated results could be misused for malicious purposes, such as identifying minority groups. While this issue falls outside the scope of this paper, we plan to release our models in a gated manner to ensure that they are not used for anything beyond academic research.

References

[1] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. S2, S3, S4, S6

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. S2, S3, S4, S7

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. S2, S3, S5

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. S3, S6, S7

[5] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. S3

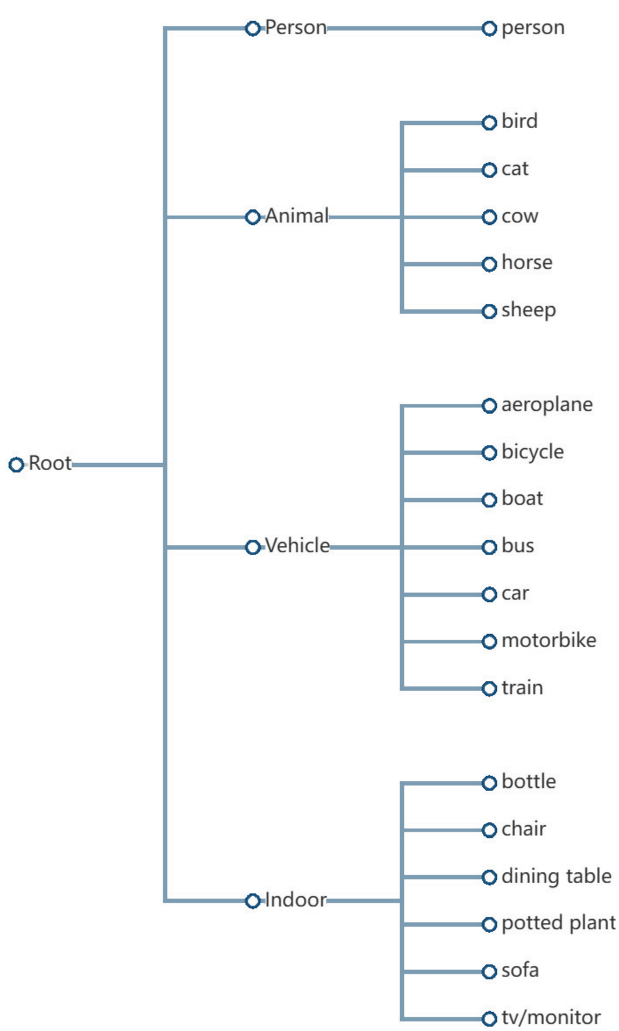

Figure S3: **Official label hierarchy** of PASCAL VOC 2012 [1]. Please check more details at `http://host.robots.ox.ac.uk/pascal/VOC/`.

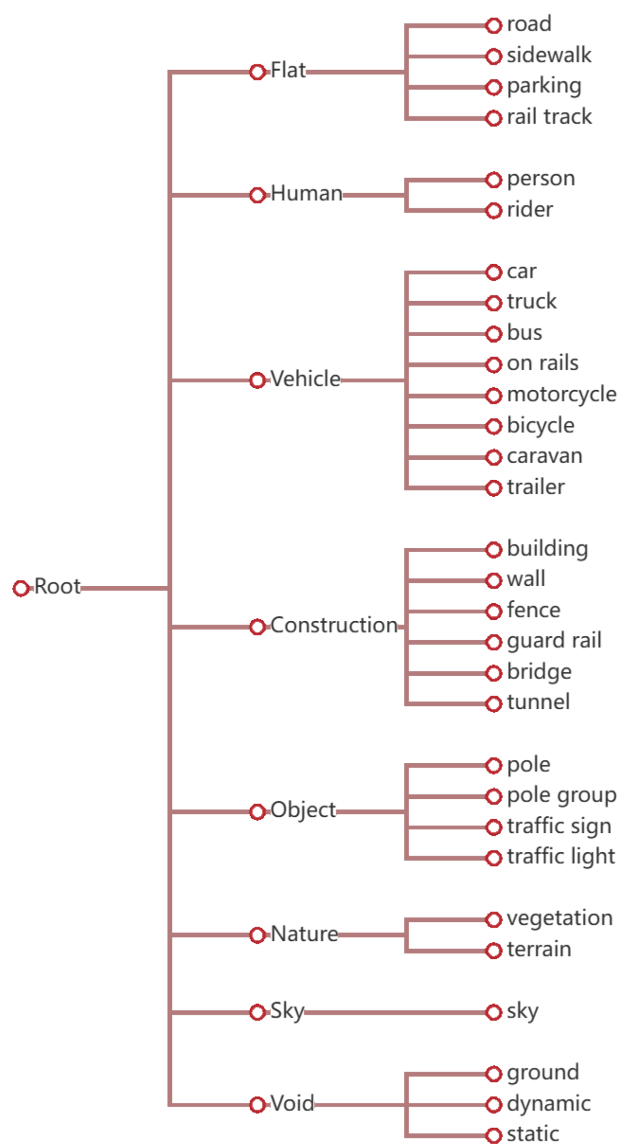

Figure S4: **Official label hierarchy** of Cityscapes [2]. Please find more details at `https://www.cityscapes-dataset.com/dataset-overview/`.

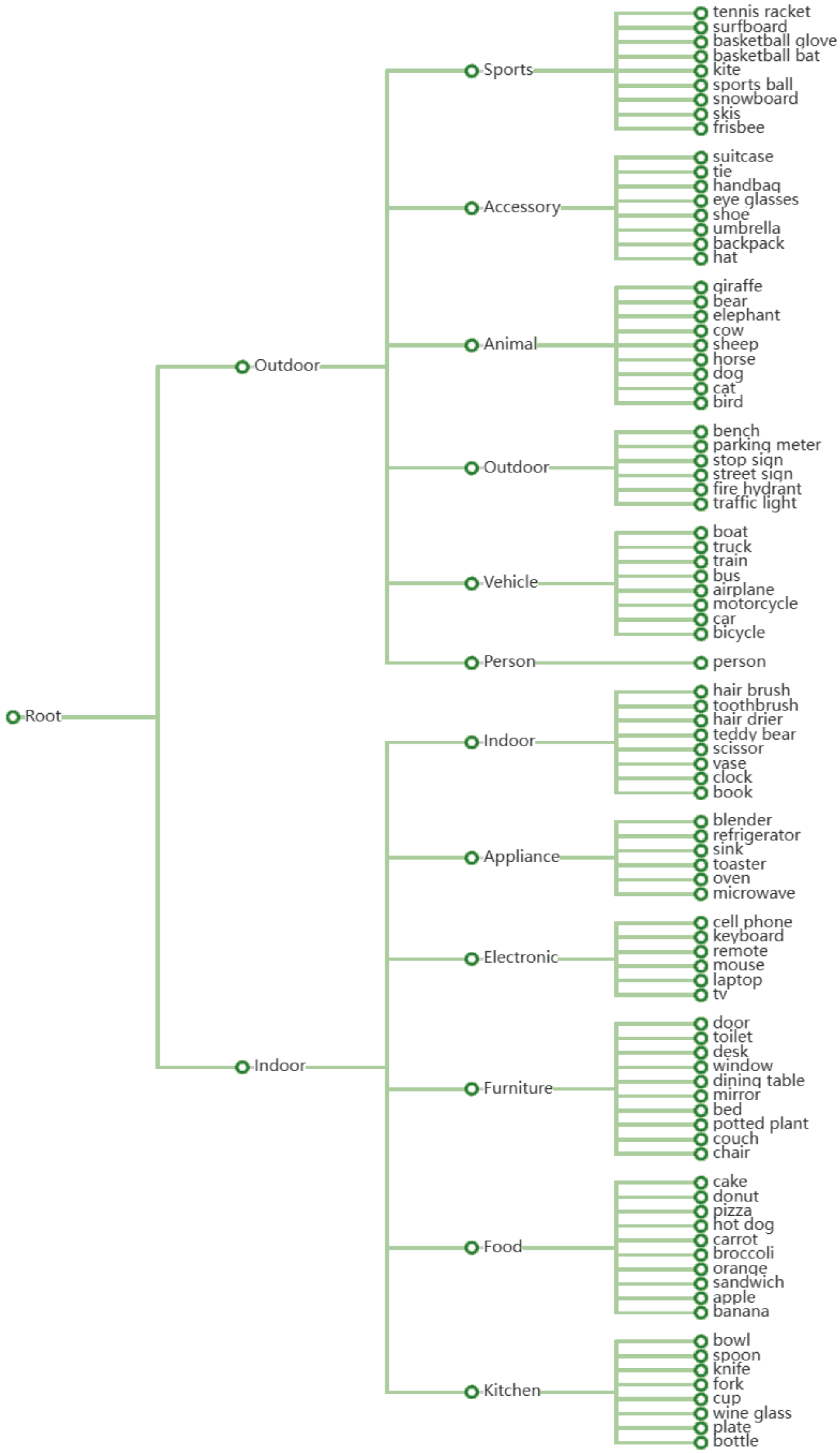

Figure S5: **Official label hierarchy** of COCO [3]. Please find more details at https://cocodataset.org/.

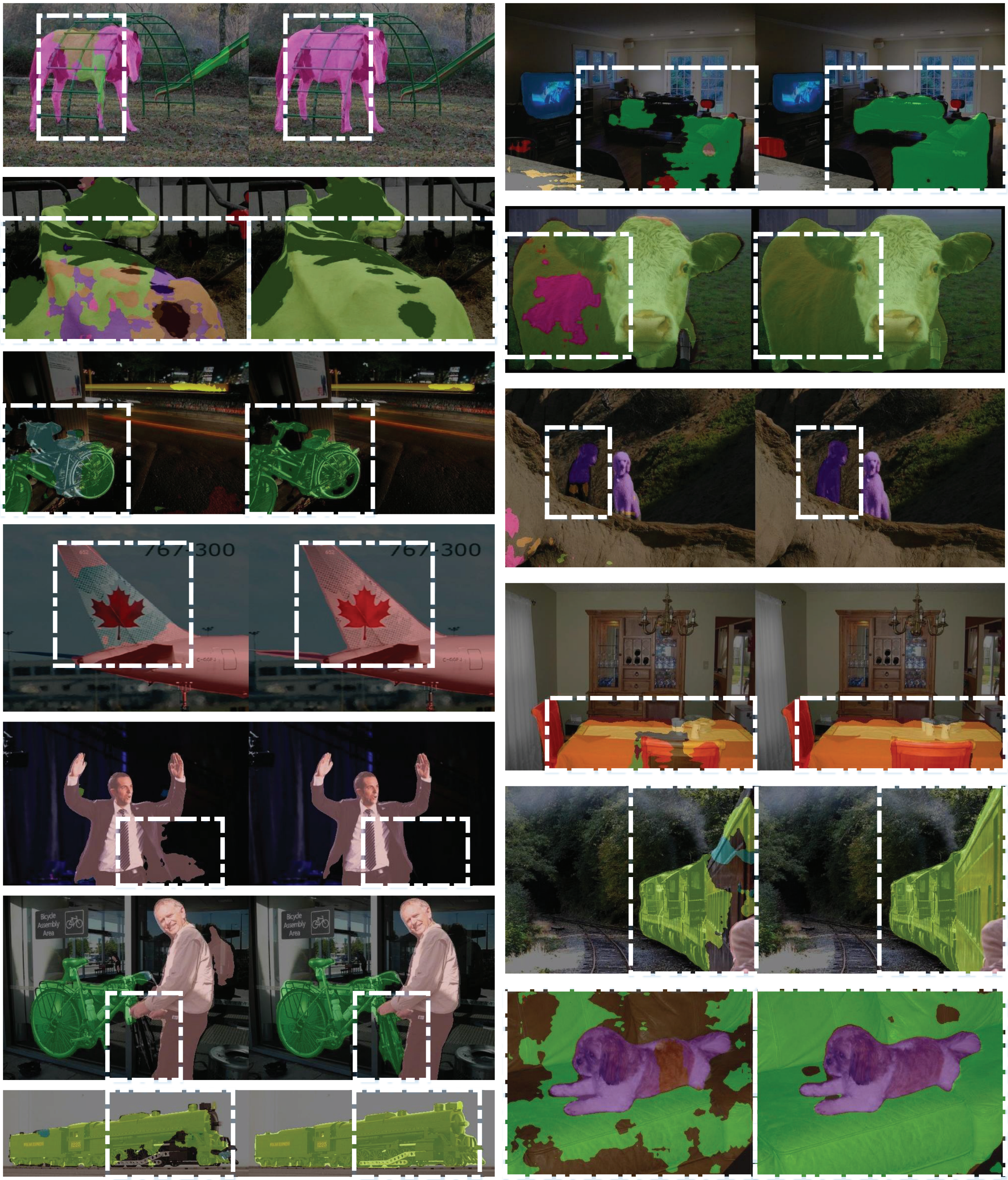

Figure S6: **Qualitative results** (§S5) obtained from confidence thresholding (left) and LogicDiag (right) methods with DeepLabV3+ [4] as the basic segmentation architecture on PASCAL VOC 2012 [1].

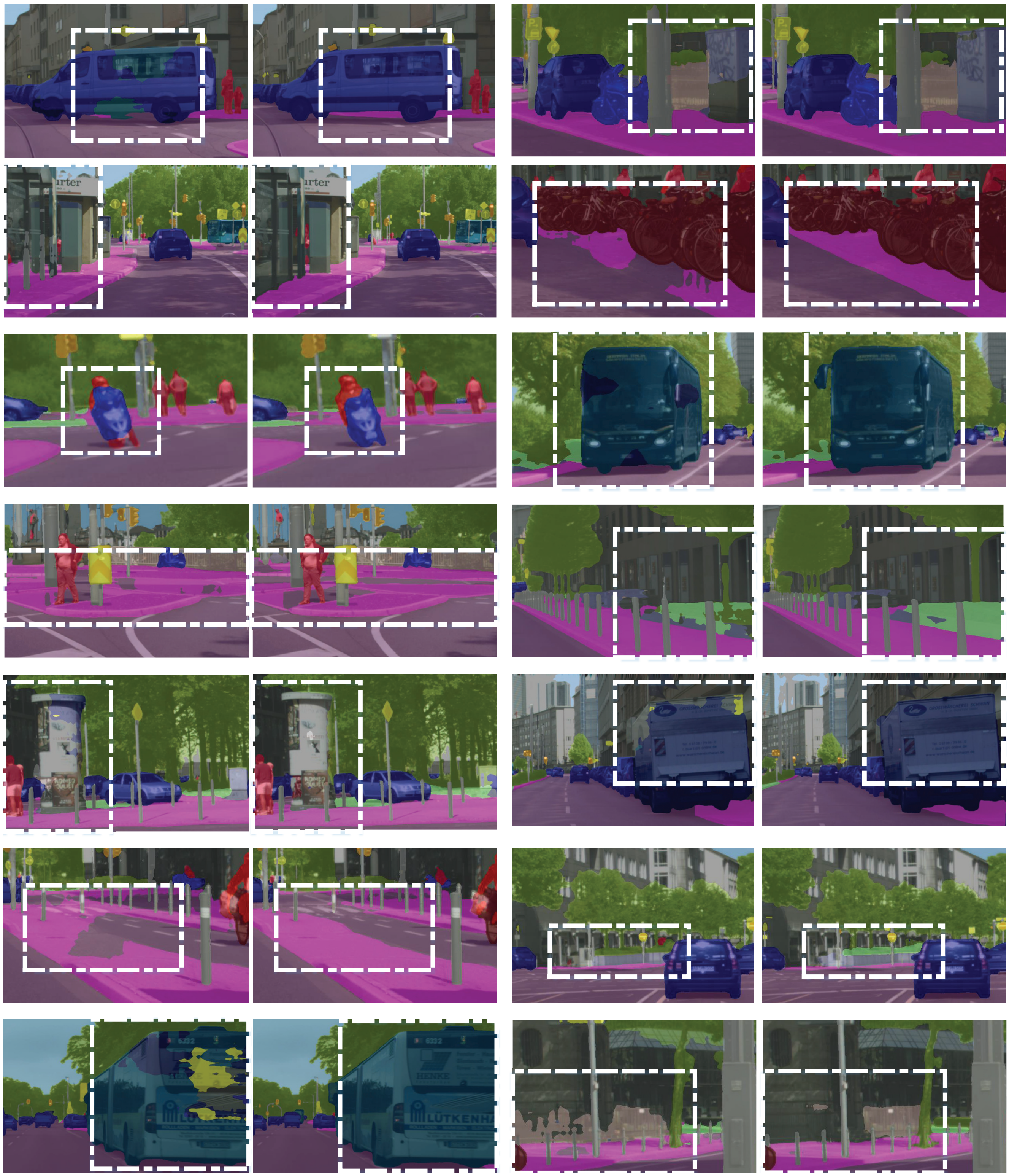

Figure S7: **Qualitative results** (§S5) obtained from confidence thresholding (left) and LogicDiag (right) methods with DeepLabV3+ [4] as the basic segmentation architecture on Cityscapes [2].