DECOUPLING "WHAT" AND "WHERE": COGNITIVE CASCADES FOR CLASS-INCREMENTAL SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Class-incremental semantic segmentation (CISS) demands preserving previously learned classes while incorporating new ones, yet conventional Softmax-based heads suffer from background shift and incomparable per-task logits. Through a Hessian-based analysis we show that while Strict Parameter Isolation (SPI) provides a parameter-level guarantee against catastrophic forgetting, its naive instantiation still fails end-to-end: the underlying probabilistic factorization entangles "what classes exist in the image" with "where each class lies." In this study, we propose Cognitive Cascade Segmentation (CogCaS), which explicitly decouples these two questions. Phase I (the "what") is a text-anchored multi-label classifier that detects which learned classes are present; Phase II (the "where") activates only the corresponding class-specific binary segmenters. A frozen vision-language text anchor stabilizes the classifier as the vocabulary grows, allowing new classes to be added without retraining any historical module. Under SPI, this decomposition cleanly separates two phenomena commonly conflated in the CISS literature: parameter-level forgetting (which CogCaS provably eliminates) and routing error from the existence classifier (an open-vocabulary recognition problem, orthogonal to forgetting and modularly improvable). On PASCAL VOC 2012 and ADE20K, CogCaS matches or surpasses state-of-the-art CISS methods under both short- and long-sequence protocols, with particularly strong gains in extreme regimes such as VOC 1-1 (20 sequential tasks), while keeping storage at 1.6–2.0$\times$ the backbone.

1 INTRODUCTION

Deep learning has transformed semantic segmentation into a cornerstone of computer vision, enabling influential applications from autonomous navigation to medical diagnostics. Despite these advances, real-world deployment faces a fundamental limitation, i.e., the conventional paradigm requires all object categories to be predefined before training. When new classes emerge, as they inevitably do in dynamic environments, models must be retrained with training data of both old and new classes, incurring increased computational costs and potential privacy concerns.

This limitation has driven substantial research in continual learning (CL), where models incrementally acquire new knowledge while trying to preserve existing capabilities. The core challenge is catastrophic forgetting: the tendency of neural networks to abruptly forget previously learned information when updated. Current approaches span multiple paradigms, including regularization-based methods (Kirkpatrick et al., 2017; Li & Hoiem, 2017), parameter updates, replay-based strategies (Buzzega et al., 2020; Rebuffi et al., 2017), and optimization techniques (Farajtabar et al., 2020). Among these, SPI (Serra et al., 2018) stands out for providing theoretical guarantee of zero-forgetting through parameter compartmentalization, effectively addressing the stability-plasticity dilemma in classification tasks.

However, extending continual learning to semantic segmentation introduces unique complexities beyond classification. In particular in class-incremental semantic segmentation (CISS), models must adapt to new categories while maintaining pixel-precise understanding of previous classes, all without access to complete historical data. This setting introduces the challenge of background shift: pixels belonging to future classes are temporarily labeled as background, requiring the model to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

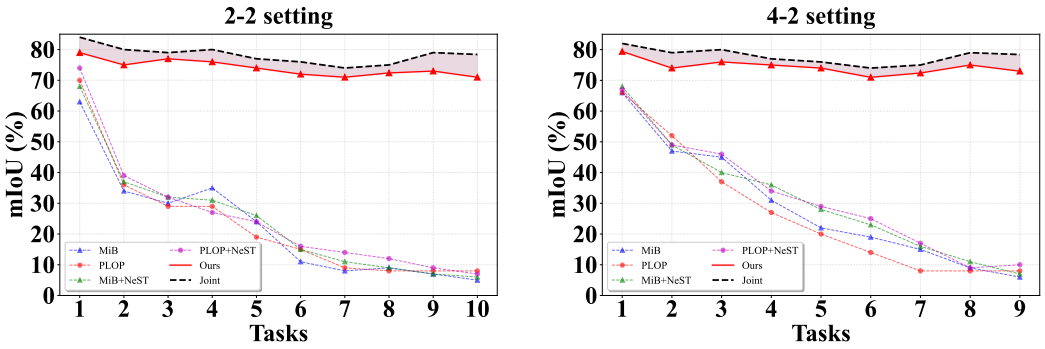


Figure 1: mIoU of classes learned at each task on PASCAL VOC under two CISS settings. While baselines exhibit declining learning capability as tasks progress, ours consistently achieves strong performance around 70% mIoU, demonstrating stable learning ability across all incremental stages.

continuously revise its understanding of what constitutes "background" as new classes emerge (Park et al., 2024; Cermelli et al., 2020). This continuous re-evaluation of the "background" class due to background shift critically exacerbates the stability-plasticity dilemma, often compelling the model to make a detrimental trade-off between the forgetting of previously learned classes and the insufficient acquisition of new ones, a challenge visually depicted in Figure 1.

While recently CISS approaches show promising performance in knowledge preservation Su et al. (2025); Wang et al. (2024), we prove the localized optimization strategy from existing CISS approaches, even assuming perfect preservation of knowledge of past classes, structurally prevents convergence to the global optimum achievable through joint training on all classes. Our theoretical analysis also reveals that they face fundamental limitations. Architecturally, the commonly used task headers based on Softmax need to output complete probabilities for all classes at every stage. However, the probability output for each segmentation head is essentially local and is only optimized for a single task. This creates a stark trade-off: while freezing historical heads eliminates dynamic interference, it simultaneously cements distributional biases that prevent global optimization.

Motivated by these theoretical analyses, we propose a **Cog**nitive **Ca**scade **S**egmentation (CogCaS) architecture that fundamentally restructures the CISS paradigm. Our approach introduces two key innovations, i.e., **Existence-Driven Activation** by which segmentation heads are activated only for detected classes and thus background interference is eliminated, and **Parameter Modularity** by which independent detector and segmenter per class enable isolated evolution without cross-task contamination. Our implementation deliberately adopts elementary components: basic backbones, single-layer detectors and simple segmenters, and standard loss functions. This architectural transparency ensures that our empirical improvements stem purely from the cognitive cascade design rather than implementation sophistication, while maintaining the framework's extensibility for future enhancements.

In summary, our main contributions include (1) the first systematic characterization of the advantages and disadvantages of SPI strategy in CISS, (2) a cognitively inspired architecture that resolves the stability-plasticity dilemma through decoupling design, and (3) state-of-the-art performance with significantly increased margins particularly in challenging long-sequence CL scenarios.

2 Related Work

Class-Incremental Semantic Segmentation (CISS) as a relatively new research topic is often considered as an extension of image-level continual learning, and therefore those techniques developed originally for image-level continual learning have been adopted for CISS, including *regularization* (Cermelli et al., 2020), *pseudo-labeling/self-supervision* (Xie et al., 2024; Yang et al., 2023), and *experience replay* (Cha et al., 2021; Maracani et al., 2021; Zhang et al., 2022). Such adoption is still being actively explored, as shown by very recent attempts (Yu et al., 2025). All these methods model segmentation as *per-pixel multi-class Softmax classification*, where old and new logits directly compete, causing background drift that accumulate with the number of tasks (Farajtabar et al., 2020) which can be seen in Figure 1.

To mitigate Softmax competition, several studies recast CISS as a *pixel-level multi-label* problem combined with strict parameter freezing. SSUL (Cha et al., 2021) appends binary channels for each class and freezes them thereafter, but relies heavily on the initial task's class capacity to initialize its "unknown" class prediction; in dense incremental settings (e.g., 1-1), this causes performance collapse as incomparable logits from disjoint phases are forced into global competition. IPSeg (Yu et al., 2025) extends this paradigm with image posterior calibration but still requires exemplar replay for stability. In addition, built on Mask2Former (Cheng et al., 2021), methods such as CoMFormer (Cermelli et al., 2023) and CoMasTRe (Gong et al., 2024) treat CISS as a *mask-classification* problem. CoMasTRe decouples "where" and "what" via query-based segmenters and objectness transfer, yet relies on complex multi-stage distillation to mitigate forgetting. DecoupleCSS (Guo et al., 2026) more directly follows a detection-then-segmentation paradigm, using language-driven class-aware detection to prompt a class-agnostic SAM segmenter. ECLIPSE (Kim et al., 2024) employs visual prompt tuning within a shared decoder and logit manipulation to address drift in continual panoptic segmentation. Our *CogCaS* adopts *existence-driven activation*, which decouples "does class c exist?" from "segment class c", and freezes each class-specific segmentation head after learning, potentially achieving task-agnostic zero forgetting while maintaining high plasticity for new classes. Crucially, CogCaS is derived from Hessian-based theoretical analysis (Section 3.2): the explicit image-level router activates independent binary heads, structurally eliminating cross-task competition without distillation or replay.

3 METHODOLOGY

3.1 PRELIMINARIES

Notations. We consider the problem of Class-Incremental Semantic Segmentation (CISS), where a model sequentially learns a sequence of T tasks, denoted as $\{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$. Each task $\mathcal{T}_t$ is associated with a unique training set $\mathcal{D}_t$ which contains certain number of training images and corresponding pixel-wise annotations. The task introduces a set of task-specific foreground classes $\mathcal{C}_t$, which are mutually exclusive between tasks, i.e., $\mathcal{C}_\tau \cap \mathcal{C}_t = \emptyset$ for all $\tau \neq t$. For each training image in task $\mathcal{T}_t$, each pixel is annotated as one of the task-specific foreground classes in $\mathcal{C}_t$ or the special "background" class. Notably for the training set $\mathcal{D}_t$, all pixels not belonging to the foreground class set $\mathcal{C}_t$ are annotated as background. Thus, the annotated background regions in each training image of task $\mathcal{T}_t$ encompass the true background as well as image regions corresponding to foreground classes from past ($\mathcal{T}_{1:t-1}$) or future ($\mathcal{T}_{t+1:T}$) tasks. When updating the model with the training set $\mathcal{D}_t$ and validation set $\mathcal{D}_t^v$ of task $\mathcal{T}_t$, the model needs to be expanded to account for the set $\mathcal{C}_t$ of new foreground classes. Let $\boldsymbol{\theta}$ denote the collection of all the learnable model parameters throughout the continual learning process (from task $\mathcal{T}_1$ to $\mathcal{T}_T$). When the model learns task $\mathcal{T}_t$, the part of $\boldsymbol{\theta}$ which are uniquely associated with future tasks ($\mathcal{T}_{t+1:T}$) are frozen with certain default value (i.e., zero here). After updating the model based on certain task-specific loss function $\mathcal{L}_t$, the leranable model parameters are changed from $\boldsymbol{\theta}_{t-1}^*$ to $\boldsymbol{\theta}_t^*$, where $\boldsymbol{\theta}_t^*$ represents the locally optimal model parameters that minimizes $\mathcal{L}_t$. The change in parameter values from $\mathcal{T}_{t-1}$ to $\mathcal{T}_t$ is denoted by $\Delta_t := \boldsymbol{\theta}_t^* - \boldsymbol{\theta}_{t-1}^*$, and the Hessian matrix of the loss function $\mathcal{L}_t$ with respect to the learnable parameters $\boldsymbol{\theta}$ is denoted by $\mathbf{H}_t(\boldsymbol{\theta}) = \frac{\partial^2 \mathcal{L}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}$.

Convergence Assumption. To enable formal analysis of the CISS framework, we make the following assumption about the learning dynamics.

Assumption 3.1 (Convergence for each task)**.** For each task $\mathcal{T}_t$, the optimization process converges to a locally optimal model parameters $\boldsymbol{\theta}_t^*$, such that within its neighborhood $\mathcal{N}(\boldsymbol{\theta}_t^*)$, the magnitude of the loss gradient $\nabla \mathcal{L}_t(\boldsymbol{\theta})$ is bounded (smaller than ϵ) and the Hessian matrix $\mathbf{H}_t(\boldsymbol{\theta})$ remains positive semi-definite, satisfying the second-order conditions for local optimality, i.e.,

$$|\nabla \mathcal{L}_t(\boldsymbol{\theta})| \leq \epsilon \,, \mathbf{H}_t(\boldsymbol{\theta}) \succeq 0 \,, \quad \forall \boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{\theta}_t^*) \,. \tag{1}$$

Furthermore, it assumes that the magnitude of parameter updates satisfies $|\Delta_t| < \delta$, $\forall t \leq T$, for some small constant $\delta > 0$. When learning a new task (per Assumption 3.1), the model's performance on prior tasks can degrade, a phenomenon known as catastrophic forgetting. To quantify this, we will define the forgetting rate (see below) based on the change in the loss function, which serves as a continuous and differentiable measurement function for task performance and reflects generalization performance when evaluated on a validation set.

Definition 3.2 (Average Forgetting Rate)**.** The forgetting rate for a previously learned task $\mathcal{T}_\tau$ (where $\tau < t$) after the model parameters have been updated to θ_t for task $\mathcal{T}_t$, is defined as the change in the loss evaluated on the validation set $\mathcal{D}_\tau^v$. Formally, it is given by $\mathcal{E}_\tau(\theta_t) = \mathcal{L}_\tau^{val}(\theta_t) - \mathcal{L}_\tau^{val}(\theta_\tau^*)$, where $\mathcal{L}_\tau^{val}$ denotes the loss computed on the validation data for task $\mathcal{T}_\tau$. By definition, this rate is zero when evaluated at the task's own optimal parameters, i.e., $\mathcal{E}_\tau(\theta_\tau^*) = 0$. In the context of continual learning, the locally optimal parameters θ_t^* for task $\mathcal{T}_t$ may cause forgetting of knowledge from previous tasks $\mathcal{T}_{1:t-1}$. This effect is measured by the Average Forgetting Rate, defined as

$$\bar{\mathcal{E}}_t(\theta_t^*) = \frac{1}{t-1}\sum_{\tau=1}^{t-1}\mathcal{E}_\tau(\theta_t^*) \quad t \geq 2\,. \tag{2}$$

3.2 Strict Parameter Isolation in CISS

Based on Assumption 3.1 and Definition 3.2, the relationship between the average forgetting rate for task $\mathcal{T}_t$ and $\mathcal{T}_{t-1}$ can be obtained as (see Appendix A.1)

$$\bar{\mathcal{E}}_t(\theta_t^*) = \frac{1}{t-1}\left((t-2)\cdot\bar{\mathcal{E}}_{t-1}(\theta_{t-1}^*) + \frac{1}{2}\Delta_t^\intercal\Big(\sum_{\tau=1}^{t-1}\mathrm{H}_\tau(\theta_\tau^*)\Big)\Delta_t + v^\intercal\Delta_t\right) + \mathcal{O}(\delta\cdot\epsilon)\,, \tag{3}$$

where $v^\intercal = \sum_{\tau=1}^{t-1}(\theta_{t-1}^* - \theta_\tau^*)^\intercal\mathbf{H}_\tau(\theta_\tau^*)$. From Equation (3), it is clear that the average forgetting rate $\bar{\mathcal{E}}_t(\theta_t^*)$ of locally optimal parameters θ_t^* for task $\mathcal{T}_t$ is directly related to the average forgetting rate $\bar{\mathcal{E}}_{t-1}(\theta_{t-1}^*)$ of locally optimal parameters θ_{t-1}^* for task $\mathcal{T}_{t-1}$. With such relationship, we can obtain the following zero-forgetting condition (see Appendix A.1).

Theorem 3.3 (Zero-forgetting Condition)**.** *For any continuous learning algorithm that satisfies Assumption 3.1, (1) if* $\bar{\mathcal{E}}_\tau(\theta_\tau^*) = 0$, $\forall\tau < t$, *then* $\bar{\mathcal{E}}_t(\theta_t^*) = \frac{1}{2(t-1)}\Delta_t^\intercal\left(\sum_{i=1}^{t-1}\mathbf{H}_i(\theta_i^*)\right)\Delta_t$, *and (2)* $\mathcal{E}_\tau(\theta_t^*) = 0, \forall\tau < t$, *if and only if* $\Delta_t^\intercal(\sum_{\tau=1}^{t-1}\mathbf{H}_\tau(\theta_\tau^*))\Delta_t = 0$.

Theorem 3.3 offers two significant implications for achieving zero forgetting. First, even if zero average forgetting ($\bar{\mathcal{E}}_\tau(\theta_\tau^*) = 0$) is achieved for all previous tasks $\tau < t$ when evaluated at their respective optimal parameters, learning a new task $\mathcal{T}_t$ and thereby updating parameters (resulting in $\Delta_t \neq 0$) can still cause considerable average forgetting. Second, Theorem 3.3 (second half) provides a direct mathematical condition for achieving true zero forgetting on all past tasks (i.e., $\mathcal{E}_\tau(\theta_t^*) = 0, \forall\tau < t$) after the model learns task $\mathcal{T}_t$, i.e., the quadratic term $\Delta_t^\intercal\left(\sum_{\tau=1}^{t-1}\mathbf{H}_\tau(\theta_\tau^*)\right)\Delta_t$ must be zero. Consequently, any continual learning algorithm aiming for zero forgetting must be designed to ensure this quadratic term vanishes. Indeed, existing zero-forgetting methods, such as orthogonal gradient method (Farajtabar et al., 2020) and projected gradient method (Saha et al., 2021), work by satisfying such a condition (see details in Appendix A.2).

Strict parameter isolation (SPI) which is used in existing work (Cha et al., 2021; Yu et al., 2025) is a strategy that can satisfy the zero forgetting condition in Theorem 3.3. By optimizing a unique set of parameters for each new task while freezing those for previous tasks, the parameter update Δ_t is guaranteed to be in a subspace orthogonal to the parameters of all previous tasks. Consequently, the quadratic term $\Delta_t^\intercal(\sum_{\tau=1}^{t-1}\mathbf{H}_\tau)\Delta_t$ is always zero, thus ensuring theoretical zero-forgetting.

Although the SPI strategy can theoretically prevent catastrophic forgetting by isolating task-specific parameters, its direct application in CISS introduces a critical challenge: the problem of incomparable outputs. In the SPI framework, each task-specific segmentation head is trained independently on a subset of classes. Consequently, the output logits from different heads are not mutually calibrated; a high score from one head is not directly comparable to a score from another trained on a different task. This renders the standard approach of applying a simple $argmax$ operation across all heads' outputs to determine the final class for each pixel fundamentally flawed (also can be seen in Figure 2(A)). This issue stems from the SPI model's inability to determine a pixel's task affiliation before classifying it. This problem is conceptually analogous to challenges in image-level continual learning. A prior study (Kim et al., 2022), for instance, factorizes the image-level prediction into an intra-task prediction and a task affiliation probability. Borrowing this formulation for our pixel-level problem, the prediction for a pixel y_p can be written as

$$P\left(y_p = c \mid \mathbf{x}\right) = P\left(y_p = c \mid \mathbf{x}, \mathcal{T}_t\right)\cdot P\left(\mathcal{T}_t \mid \mathbf{x}\right), \quad \forall c \in \mathcal{C}_t\,, \tag{4}$$

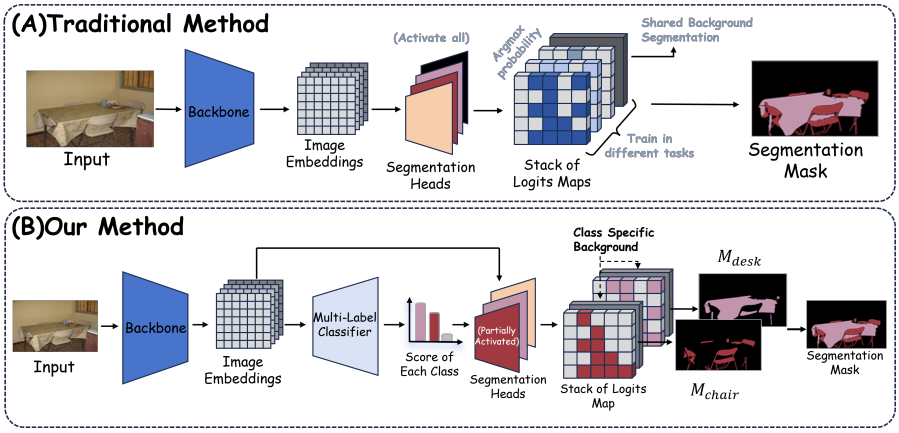


Figure 2: Demonstration of existing typical CISS framework and the proposed CogCaS framework. (A) Traditional CISS framework consists of a set of segmentation heads which are trained sequentially and activated for inference simultaneously. (B) Our proposed CogCaS restructures the traditional CISS formulation into a dual-phase cascade using multi-label classifier and class-specific segmentation heads. The multi-label classifier determine whether each learned class exists in the image, and only segmentation heads corresponding to existing classes are activated to produce a foreground-background mask. These masks are then fused to obtain the final segmentation mask using a mask fusion strategy.

where p is the index of a pixel in the input image $\mathbf{x}$, and c is a class learned from task $\mathcal{T}_t$. While SPI perfectly preserves the intra-task prediction term $P(y_p = c|x, \mathcal{T}_t)$ due to its zero-forgetting nature. There is no mechanism to estimate the crucial task affiliation probability $P(\mathcal{T}_t|x)$, thus hindering effective final prediction.

3.3 Class-Incremental Segmentation via Cognitive Cascades

To resolve the issue of incomparable outputs in Equation (4) and fully leverage the zero-forgetting property of SPI, we introduce **Cog**nitive **Cas**cade Segmentation (CogCaS), a novel framework that fundamentally restructures the CISS paradigm. Our approach is motivated by the coarse-to-fine strategy employed by human annotators who first identify the object categories present in an image before delineating their precise boundaries. CogCaS mimics this cognitive process by decoupling the problem into two sequential phases. First, an image-level classifier which acts as a Task Router determines the existence of all learned classes within the input image. Second, only the parameter-isolated segmentation heads corresponding to the detected classes are activated to perform binary foreground-background segmentation. This restructuring can be formalized (in Appendix A.4) by reframing the probabilistic decomposition from Equation (4) into a more intuitive, class-centric model as detailed below.

Notation Clarification. Let $Y_p \in \mathcal{C}_{[1:t]} \cup \{0\}$ denote the random variable representing the class label of pixel p, and let $Z_c \in \{0, 1\}$ be a binary random variable indicating whether class c is present in image $\mathbf{x}$ (i.e., $Z_c = 1$ if $\exists\, p : Y_p = c$). The cascade decomposition is then:

$$P(Y_p = c \mid \mathbf{x}) \propto \underbrace{P\big(Y_p = c \mid \mathbf{x},\, Z_c = 1\big)}_{\text{Binary segmentation}} \cdot \underbrace{P\big(Z_c = 1 \mid \mathbf{x}\big)}_{\text{Class existence (router)}}, \quad \forall c \in \mathcal{C}_{[1:t]}, \tag{5}$$

where the first term is the conditional probability that pixel p belongs to class c given that class c exists in the image, and the second term is the existence probability estimated by the multi-label classifier. In practice, we threshold the existence probability $P(Z_c = 1|\mathbf{x}) \geq \alpha$ to obtain the predicted class set $\mathcal{C}_{pred} = \{c : P(Z_c = 1|\mathbf{x}) \geq \alpha\}$.

The overall pipeline of this framework is depicted in Figure 2(B). An input image $\mathbf{x}$ is first passed through a task-shared, frozen pretrained feature extractor Φ to generate feature maps $F = \Phi(\mathbf{x})$. Subsequently, F is processed by the following two cascade phases.

Phase I: Image-Level Category Recognition. This phase is responsible for estimating the class-existence probability $P(Z_c{=}1 \mid \mathbf{x})$ for every class $c \in \mathcal{C}_{1:t}$. We instantiate it as a *text-anchored* ex-

istence detector: a frozen vision-language backbone provides a shared semantic space, and for every class we learn only a small set of continuous prompt vectors that produce a class-specific text query. Class existence is then read off as the cosine similarity between this query and the pooled image embedding. This design keeps each new task's parameter footprint at the kilobyte scale, places all classes in a single semantic space so cross-class relations are inherited from pretraining rather than learned head-by-head, and—because only the newly introduced prompts are ever trained—remains fully compatible with the SPI guarantee.

Text-Anchored Existence Detection. Following the prompt-learning paradigm of CoOp (Zhou et al., 2022) and PromptSRC (Khattak et al., 2023), we instantiate Phase I on top of a frozen vision-language backbone, namely the SigLIP2 image encoder Φ_v and text encoder Φ_t (Tschannen et al., 2025). For each class $c \in \mathcal{C}_t$, we attach a learnable prompt $\boldsymbol{p}_c \in \mathbb{R}^{N_p \times D}$ in front of the embedding $\mathbf{e}_{\mathrm{CLS}_c}$ of a unified, LLM-generated textual template instantiated with the class name (e.g., "*a photo of a* CLS_c"), and feed the concatenation through Φ_t to obtain the class-specific text query: $\boldsymbol{t}_c = \Phi_t\big([\,\boldsymbol{p}_c,\, \mathbf{e}_{\mathrm{CLS}_c}\,]\big)$, where $\boldsymbol{p}_c$ is the only trainable component.

Given an input image $\mathbf{x}$, the class-existence probability is computed by cosine similarity in the shared embedding space:

$$P(Z_c{=}1 \mid \mathbf{x}) \;=\; \sigma\big(\langle \tilde{\boldsymbol{v}},\, \tilde{\boldsymbol{t}}_c \rangle \,/\, \tau\big), \qquad \boldsymbol{v} = \mathrm{Pool}\big(\Phi_v(\mathbf{x})\big), \tag{6}$$

where $\tilde{\cdot}$ denotes ℓ_2 normalization, σ is the sigmoid, and τ is the SigLIP2-pretrained logit scale, kept fixed to preserve a globally calibrated similarity scale. The predicted class set is $\mathcal{C}_{\mathrm{pred}} = \{c : P(Z_c{=}1 \mid \mathbf{x}) \geq \alpha\}$.

Geometric Regularization. Optimizing only an alignment loss on a few thousand prompt parameters risks (i) drifting from the well-calibrated SigLIP2 text geometry (Liang et al., 2022), and (ii) collapsing visually similar but semantically distinct categories (e.g., *stairs* vs. *stairway*). We therefore minimize

$$\mathcal{L}_{\mathrm{Phase\,I}} \;=\; \mathcal{L}_{\mathrm{align}} \;+\; \lambda_s\, \mathcal{L}_{\mathrm{geo}} \;+\; \lambda_p\, \mathcal{L}_{\mathrm{plast}}, \tag{7}$$

where $\mathcal{L}_{\mathrm{align}}$ is the asymmetric multi-label loss (Ridnik et al., 2021) over positives and the near-OOD negatives shared with Phase II (Appendix B.1). To keep the trainable queries $\boldsymbol{t}_c = \Phi_t([\boldsymbol{p}_c, \mathbf{e}_{\mathrm{CLS}_c}])$ faithful to the original SigLIP2 topology, we precompute a frozen prompt-free anchor $\boldsymbol{t}_c^*$ for each class as the mean embedding of hand-crafted templates (e.g., "*a photo of* $\{c\}$.") and regress the pairwise distances and triangle angles of $\boldsymbol{t}_c$ toward those of $\boldsymbol{t}_c^*$ with a Huber loss ℓ_δ:

$$\mathcal{L}_{\mathrm{geo}} \;=\; \sum_{i \in \mathcal{C}_t} \sum_{\substack{j \in \mathcal{C}_{1:t} \\ j \neq i}} \ell_\delta\big(d_{ij},\, d_{ij}^*\big) \;+\; \sum_{i \in \mathcal{C}_t} \sum_{\substack{j,k \in \mathcal{C}_{1:t} \\ j,k \neq i}} \ell_\delta\big(a_{ijk},\, a_{ijk}^*\big), \tag{8}$$

with $d_{ij} = \|\tilde{\boldsymbol{t}}_i - \tilde{\boldsymbol{t}}_j\|_2$, $a_{ijk} = \cos \angle(\tilde{\boldsymbol{t}}_i, \tilde{\boldsymbol{t}}_j, \tilde{\boldsymbol{t}}_k)$, and $*$-quantities computed on anchors. Restricting the anchor index i to the new-class set $\mathcal{C}_t$ enumerates exactly the pairs and triplets through which gradients can flow—queries of all old classes $\mathcal{C}_{1:t-1}$ are frozen, so SPI is preserved without explicit masking. Finally, $\mathcal{L}_{\mathrm{plast}}$ resolves residual near-synonym collisions by repelling each new-class query from its top-k most similar current neighbors:

$$\mathcal{L}_{\mathrm{plast}} \;=\; \sum_{i \in \mathcal{C}_t} \sum_{j \in \mathcal{N}_k(i)} \big(1 - \cos(\boldsymbol{t}_i, \boldsymbol{t}_j)\big), \qquad \mathcal{N}_k(i) = \text{Top-}k_{\,j \in \mathcal{C}_{1:t},\, j \neq i}\, \cos(\boldsymbol{t}_i, \boldsymbol{t}_j). \tag{9}$$

Phase II: Class-Specific Binary Segmentation. This phase is responsible for estimating the conditional segmentation term, $P(Y_p = c|\mathbf{x}, Z_c = 1)$. For each class $c \in \mathcal{C}_{pred}$ identified in Phase I, its dedicated and parameter-isolated segmentation head $H_c(\cdot)$ is activated. This head takes the feature maps F as input and produces a two-channel probability map, $M_c = (m_c^{bg}, m_c^{fg})$, representing the probability of each pixel belonging to the "relative background for class c" and the "foreground for class c," respectively. The binary segmentation schema offers key advantages: it simplifies the complex multi-class decision boundary into a *single binary decision boundary*, leading to faster convergence and higher fidelity.

Mask Fusion. Since Phase II yields independent binary masks, a fusion step $\mathcal{U}$ is required to integrate them into a coherent multi-class segmentation map by resolving overlapping predictions. We evaluate several fusion strategies and analyze their impact in Table 3b.

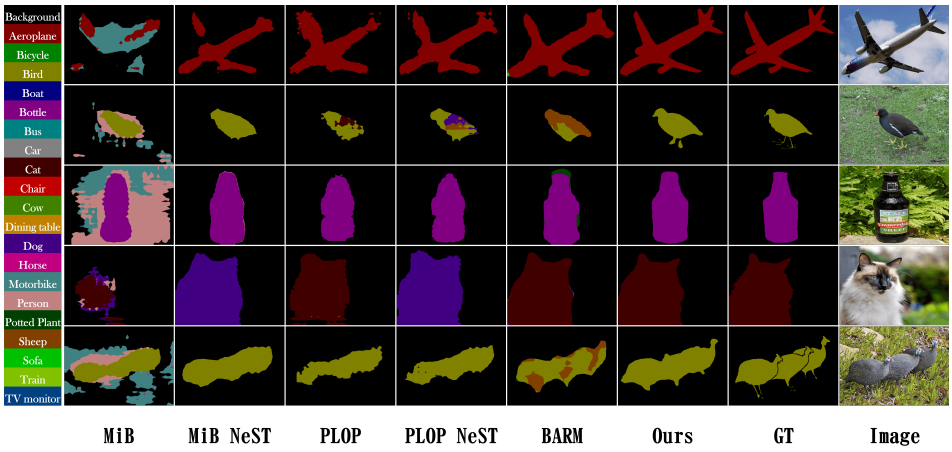


Figure 3: Representative segmentation results from different methods after the model learns all tasks in the Pascal VOC 2012 15-1 setting.

Precise Model Lifecycle. To ensure reproducibility and clarity, we provide a rigorous description of the complete model lifecycle as follows.

Instantiation. For Phase I, the visual feature $\boldsymbol{v} = \mathrm{Pool}(\Phi_v(\mathbf{x}))$ is produced by the frozen SigLIP2 image encoder, while Phase II takes the dense feature map $F = \Phi(\mathbf{x})$ from a frozen segmentation backbone Φ (e.g., ResNet-101 or Swin-L). When learning a new task $\mathcal{T}_t$, we instantiate only new parameters: (i) per-class text-query prompts $\boldsymbol{p}_c \in \mathbb{R}^{N_p \times D}$ for each $c \in \mathcal{C}_t$, and (ii) binary segmentation heads H_c with parameters θ_c^{seg} for each $c \in \mathcal{C}_t$. All parameters from previous tasks $\mathcal{C}_{1:t-1}$, together with Φ_v, Φ_t, τ and Φ, remain strictly frozen.

Training. For task $\mathcal{T}_t$, we update only the newly instantiated parameters $\{\boldsymbol{p}_c, \theta_c^{seg}\}_{c \in \mathcal{C}_t}$. Phase I (text prompts) and Phase II (segmentation heads) are trained *separately* to avoid interference: Phase I optimizes Eq. (7), while Phase II uses a binary segmentation loss (see Appendix B.1). Gradients are computed only for $\{\boldsymbol{p}_c, \theta_c^{seg}\}_{c \in \mathcal{C}_t}$; all other parameters have zero gradient by design (SPI).

Inference. At test time, we first compute existence probabilities for all learned classes, then activate only the segmenters corresponding to detected classes, and finally fuse their binary masks into the final segmentation map. The complete inference pipeline is detailed in Appendix B.1.

4 Experiments

4.1 Experimental Setup

Datasets. Following prior work (Cermelli et al., 2020; Cha et al., 2021; Maracani et al., 2021; Yang et al., 2023), the proposed CogCaS was evaluated on two semantic segmentation datasets PASCAL VOC (Everingham et al., 2010) and ADE20K (Zhou et al., 2017) with different complexity levels. PASCAL VOC contains 20 object classes plus a background class, with 10,582 samples for training and 1,449 samples for validation, while the large-scale dataset ADE20K presents a more challenging scenario, containing 150 foreground classes and one background class, with 20,210 and 2,000 samples for training and validation, respectively.

CISS Settings. With each dataset, the widely used M-N setting is adopted, with M being the number of foreground classes in the first task and N the number of new foreground classes in each subsequent task. For example, in the VOC 10-1 setting, the model first learns to segment 10 classes, then incrementally learns one new class in each subsequent task.

Implementation details. The proposed CogCaS was trained using 8 NVIDIA GeForce RTX 4090 GPUs. We conducted training on models utilizing both ResNet-101 (He et al., 2016) and Swin-L (Liu et al., 2021) backbones. The Adam optimizer (Kingma & Ba, 2015) was employed for training, with each task being trained for 90 epochs. The initial learning rate was set to 1×10^{-5}. More details can be found in the Appendix B.1.

Table 1: Comparison with existing CISS methods on PASCAL VOC 2012 in mIoU (%). The best results are marked in **bold**. All methods adopt a ViT backbone of comparable scale to our SigLip2 backbone for fair comparison. †: unlike other methods, this one is based on Mask2Former (Cheng et al., 2021).

Method	**10-1** (11 tasks)			**15-5** (2 tasks)			**15-1** (6 tasks)		
	0-10	11-20	all	0-15	16-20	all	0-15	16-20	all
Ours	**76.1**	**75.7**	**75.9**	78.3	**74.9**	**77.8**	78.4	**72.5**	**76.9**
Joint Deeplab-v3 (Chen et al., 2017)	81.4	78.4	79.9	80.8	77.3	79.9	80.8	77.3	79.9
Joint Ours	82.7	80.9	81.8	81.3	83.4	81.8	81.3	83.4	81.8
MicroSeg (Zhang et al., 2022)	73.5	53.0	63.8	**81.9**	54.0	75.2	**80.5**	40.8	71.0
MiB (Cermelli et al., 2020)	35.7	14.8	26.7	74.3	45.1	67.3	48.7	19.5	41.7
MiB+NeST (Xie et al., 2024)	41.3	24.1	33.1	77.8	50.1	71.2	63.2	23.5	53.7
PLOP (Douillard et al., 2021)	47.2	18.4	33.5	79.2	50.2	72.3	67.6	25.2	57.6
PLOP+NeST (Xie et al., 2024)	49.2	19.8	35.2	81.6	55.8	75.4	72.2	33.7	63.1
BARM (Zhang & Gao, 2024)	74.2	53.8	64.4	77.8	72.1	76.4	79.3	48.1	71.8
SSUL (Cha et al., 2021)	74.3	51.0	63.2	79.7	55.3	73.9	78.1	33.4	67.5
CoMasTRe† (Gong et al., 2024)	—	—	—	79.7	51.9	73.1	69.8	43.6	63.5
CoMFormer† (Cermelli et al., 2023)	—	—	—	74.7	54.3	71.1	70.8	32.2	61.6

Table 2: Comparison with existing CISS methods on ADE20K using Swin-L backbone. †: unlike other methods, this one is based on Mask2Former (Cheng et al., 2021).

Method	**100-50** (2 tasks)			**100-10** (6 tasks)			**100-5** (11 tasks)		
	0-100	*101-150*	*all*	*0-100*	*101-150*	*all*	*0-100*	*101-150*	*all*
Ours	41.2	**32.4**	38.2	**42.3**	**31.1**	**38.6**	40.1	**30.9**	**37.6**
Joint Deeplab-v3 (Chen et al., 2017)	47.2	31.8	42.1	47.2	21.8	42.1	47.2	21.8	42.1
Joint Ours	47.8	38.7	44.7	47.8	38.7	44.7	47.8	38.7	44.7
MiB (Cermelli et al., 2020)	39.0	16.7	31.2	36.6	9.8	27.7	34.7	4.8	24.7
MiB+NeST (Xie et al., 2024)	38.8	23.1	33.5	38.8	19.1	32.2	35.2	13.6	28.1
PLOP (Douillard et al., 2021)	40.4	13.4	31.5	39.4	12.6	30.1	36.9	6.2	26.7
PLOP+NeST (Xie et al., 2024)	40.8	22.8	34.8	39.4	20.5	33.2	38.3	15.4	30.7
BARM (Zhang & Gao, 2024)	42.0	23.0	35.7	41.1	23.1	35.2	40.5	21.2	34.1
FALCON (Truong et al., 2025)	**45.9**	29.1	**40.3**	41.1	23.2	35.2	**40.8**	18.9	33.5
CoMFormer† (Cermelli et al., 2023)	44.7	26.2	38.4	40.6	15.6	32.3	39.5	13.6	30.9
CoMasTRe† (Gong et al., 2024)	45.7	26.0	39.2	42.3	18.4	34.4	**40.8**	15.8	32.6

4.2 Main Results

Experimental results demonstrate the efficacy of the proposed CogCaS method on the PASCAL VOC 2012 and ADE20K datasets, as presented in Table 1 and Table 2 respectively.

Our CogCaS as a non-replay method was compared with basic and state-of-the-art non-replay baselines. As Table 1 shows, on the PASCAL VOC 2012 dataset, CogCaS exhibited superior performance across various incremental learning configurations. For example, in the VOC 10-1 setting (11 tasks), CogCaS achieved the highest mean Intersection over Union (mIoU) of 75.7% for new classes (11-20) and a leading overall mIoU of 75.9%. The superiority of CogCaS was more pronounced on the complex ADE20K dataset across all evaluated incremental settings (Table 2). Figure 3 visually confirms the superior performance of our method. These results consistently support the efficacy of our CogCaS in learning new knowledge and preserving old knowledge in both small-scale and large-scale incremental scenarios.

We also compare with recent large-model-based CISS methods (e.g., SAM-based DecoupleCSS (Guo et al., 2026) with $\sim$632M parameters) in Appendix C.1. Our architecture is orthogonal to backbone choice and offers a favorable balance between performance and efficiency. Additional experiments on task-shared parameters and LoRA are provided in Appendix C.2.

To further confirm the robustness of our method, experiments under more challenging conditions were performed in which tasks are more numerous with fewer classes to be learned within each task. As Table 3a shows, our CogCaS significantly outperforms traditional knowledge distillation methods (MiB, PLOP, and NeST variants) and parameter-isolation strategies (SSUL, IPSeg). For example, in the VOC 1-1 setting (totally 20 tasks), our method achieves 70.9% overall mIoU versus only 7.3% for PLOP+NeST and 33.2% for IPSeg. In the VOC 2-2 setting (10 tasks), our method reaches 71.5% compared to 27.8% (MiB+NeST) and 51.4% (IPSeg). To verify statistical robustness, we re-evaluated the VOC 1-1 setting using three independent random seeds, yielding an average mIoU of 71.1% ($\pm$ 0.8%), which is highly consistent with the reported 70.9%. This confirms that

Table 3: Left: Comparison with CISS baselines under more challenging continual learning settings. Right: Sensitivity of mask fusion on VOC, reporting the all-class mIoU (%) of the model trained at the last step under different task settings.

(a) Comparison with CISS baselines under more challenging continual learning settings.

Method	VOC 1-1 (20 tasks)			VOC 2-1 (19 tasks)			VOC 2-2 (10 tasks)		
	0-1	2-20	all	0-2	3-20	all	0-2	3-20	all
Ours	**79.4**	**70.1**	**70.9**	**76.3**	**71.2**	**71.9**	**75.9**	**70.8**	**71.5**
MiB (Cermelli et al., 2020)	27.3	6.4	8.3	23.6	7.9	10.14	41.1	23.4	25.9
PLOP (Douillard et al., 2021)	25.4	4.2	6.2	19.4	6.2	8.1	39.7	22.8	25.2
MiB+NeST (Xie et al., 2024)	28.1	6.8	8.7	24.5	8.1	10.4	40.4	25.8	27.8
PLOP+NeST (Xie et al., 2024)	32.5	4.6	7.3	20.1	7.9	10.5	38.1	23.5	25.5
SSUL (Cha et al., 2021)	60.1	29.6	32.5	59.6	34.7	38.2	60.3	40.6	44.0
IPSeg (Yu et al., 2025)	61.8	30.2	33.2	60.1	32.6	36.2	64.7	49.5	51.4

(b) Sensitivity of mask fusion on VOC.(mIoU, %)

Strategy	10-1	15-5	15-1
Logits	71.4	73.3	73.4
Random	71.9	73.2	73.9
Strict	70.9	72.8	73.0
Dist. (Ours)	72.1	74.1	74.4
Loose	**72.8**	**74.9**	**75.2**

the substantial performance gap over IPSeg (33.2%) is statistically stable. The low variance stems from the inherent stability of our SPI design: increasing task count only adds training cost without affecting previously learned knowledge.

Mask Fusion Strategy Analysis. To handle overlapping predictions in segmentation masks, we evaluate five fusion strategies: (1) *Logits-based*: selects the class with highest confidence; (2) *Random*: randomly chooses among overlapping predictions; (3) *Strict*: assigns overlapping pixels to background; (4) *Distributed*: prioritizes rare categories to small objects; and (5) *Loose*: accepts predictions containing the ground truth category. As shown in Table 3b, while different strategies varies slightly in performance, they all outperform strong baselines, supporting that our approach is largely robust the choice of mask fusion strategy.

4.3 Ablation Studies

To assess the role of the classification head at inference, we compare three configurations: the **Full Model**; **Segmentation Only**, which deactivates the classification head and relies solely on the segmentation heads at test time; and **Oracle**, which replaces the classifier's output with ground-truth labels to isolate its error contribution.

Table 4: Ablation study with respect to the classification head under different task settings

Method	VOC 1-1 (20 tasks)			VOC 2-1 (19 tasks)			VOC 2-2 (10 tasks)			ADE 100-5 (11 tasks)		
	0-1	2-20	all	0-2	3-20	all	0-2	3-20	all	0-100	101-150	all
Segmentation Only	18.2	13.1	13.5	14.3	17.2	16.7	13.8	16.5	16.1	6.7	9.5	7.6
Full Model	79.4	70.1	70.9	76.3	71.2	71.9	75.9	70.8	71.5	40.1	30.9	37.6
Oracle	80.2	70.3	71.2	77.4	71.5	72.3	77.4	71.8	72.4	48.6	31.8	43.0

As shown in Table 4, on PASCAL VOC the Oracle and Full Model perform nearly identically (e.g., 70.9% vs. 71.2%), indicating that the classifier handles the 20 categories reliably; the drop under Segmentation Only nonetheless shows that foreground–background separation remains hard even with near-OOD data in training. On ADE20K (150 classes), Oracle (43.0%) substantially outperforms the Full Model (37.6%), and the gap widens as the class count grows: the classifier becomes the bottleneck, while on VOC its few errors mostly fall on small or occluded objects that the segmentation head would also miss. Overall, the classification head is critical to our decoupled framework, particularly in many-class settings.

5 Conclusion

This study provides a theoretical analysis to deeply understand the limitations within existing CISS methods and introduces a novel dual-phase CISS framework in which the segmentation task is decomposed into two disentangled stages. Crucially, the SPI strategy inherent in our design enables the framework to potentially achieve a zero-forgetting rate for knowledge learned in previous tasks. Consequently, the model's performance is not significantly affected by the number of tasks in the CISS setting, showing notable robustness in challenging, long-sequence scenarios where other methods falter. A primary limitation is that it demands substantial training resource. Future work includes investigation of class-specific fine-tuning of the feature encoder for other imaging modalities (e.g., medical images) and applying foundational models (e.g., SAM) as the segmentation heads.

AI USE STATEMENT

We used large language model assistants for support work. They helped us write and debug parts of the experiment code, move the manuscript into the ICLR LaTeX template and lay out its tables, and re-check the algebra in the proofs of Appendix A. One use touches the method itself: the unified text template fed to the Phase I text encoder (Section 3.3) was generated once by a language model. We read and corrected it by hand. It only contains generic class-name descriptions and could have been written manually.

The research questions, the design of CogCaS, the theoretical analysis, and the experimental design are our own, and no AI tool produced any claim or result in this paper. We checked all AI-assisted code, formatting, and derivations ourselves, and every number, table, and figure comes from our own training and evaluation runs. We did not use AI tools to review or evaluate other researchers' work for this submission.

ETHICS STATEMENT

All experiments use two public benchmarks, PASCAL VOC 2012 and ADE20K, under their original terms, together with publicly released pretrained models. The work involves no human subjects and collects no new data. Near-OOD samples are cut from the same training images, so no outside data enter training. Since CogCaS does not replay old samples, a deployed model can learn new classes without keeping earlier training images, which helps when those images are private. We want to flag two risks. The frozen vision-language backbone may carry biases from its pretraining data, and when the existence classifier misses a class, that class disappears from the output map without warning. Anyone applying this approach in safety-critical settings such as driving or medical imaging should validate it on their own data and keep a human in the loop.

REPRODUCIBILITY STATEMENT

Full proofs and assumptions for the theory in Section 3 are in Appendix A. Section 3.3 states which parameters are trained or frozen at each task, and Section 4.1 and Appendix B.1 give the training configuration, near-OOD construction, losses, and inference procedure. All benchmarks and M-N protocols follow prior CISS work. We will upload the complete code to the supplementary material during the rebuttal period.

REFERENCES

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9233–9242, 2020.

Fabio Cermelli, Matthieu Cord, and Arthur Douillard. CoMFormer: Continual learning in semantic and panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3010–3020, 2023.

Sungmin Cha, Beomyoung Kim, YoungJoon Yoo, and Taesup Moon. SSUL: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Advances in neural information processing systems*, 34:10919–10930, 2021.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875, 2021.

Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. PLOP: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4040–4050, 2021.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

Jinchao Ge, Bowen Zhang, Akide Liu, Vu Minh Hieu Phan, Qi Chen, Yangyang Shu, and Yang Zhao. CIT: Rethinking class-incremental semantic segmentation with a class independent transformation. *Pattern Recognition*, 167:111707, 2025.

Yizheng Gong, Siyue Yu, Xiaoyang Wang, and Jimin Xiao. Continual segmentation with disentangled objectness learning and class recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3848–3857, 2024.

Yifu Guo, Yuquan Lu, Wentao Zhang, Zishan Xu, Dexia Chen, Siyu Zhang, Yizhe Zhang, and Ruixuan Wang. Decoupling continual semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4476–4484, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15190–15200, 2023.

Beomyoung Kim, Joonsang Yu, and Sung Ju Hwang. ECLIPSE: Efficient continual learning in panoptic segmentation with visual prompt tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3346–3356, 2024.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical study on solving continual learning. *Advances in neural information processing systems*, 35:5065–5079, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. RECALL: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7026–7035, 2021.

Gilhan Park, WonJun Moon, SuBeen Lee, Tae-Young Kim, and Jae-Pil Heo. Mitigating background shift in class-incremental semantic segmentation. In *European Conference on Computer Vision*, pp. 71–88. Springer, 2024.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 82–91, 2021.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021.

Nicol N Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural computation*, 14(7):1723–1738, 2002.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Sidak Pal Singh, Gregor Bachmann, and Thomas Hofmann. Analytic insights into structure and rank of neural network Hessian maps. *Advances in Neural Information Processing Systems*, 34: 23914–23927, 2021.

Yuanzhi Su, Siyuan Chen, and Yuan-Gen Wang. Balanced residual distillation learning for 3D point cloud class-incremental semantic segmentation. *Expert Systems with Applications*, 269:126399, 2025.

Thanh-Dat Truong, Utsav Prabhu, Bhiksha Raj, Jackson Cothren, and Khoa Luu. FALCON: Fairness learning via contrastive attention approach to continual semantic scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15065–15075, 2025.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Huyong Wang, Huisi Wu, and Jing Qin. Incremental nuclei segmentation from histopathological images via future-class awareness and compatibility-inspired distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11408–11417, 2024.

Zhengyuan Xie, Haiquan Lu, Jia-wen Xiao, Enguang Wang, Le Zhang, and Xialei Liu. Early preparation pays off: New classifier pre-tuning for class incremental semantic segmentation. In *European conference on computer vision*, pp. 183–201. Springer, 2024.

Ze Yang, Ruibo Li, Evan Ling, Chi Zhang, Yiming Wang, Dezhao Huang, Keng Teck Ma, Minhoe Hur, and Guosheng Lin. Label-guided knowledge distillation for continual semantic segmentation on 2D images and 3D point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18601–18612, 2023.

Xiao Yu, Yan Fang, Yao Zhao, and Yunchao Wei. IPSeg: image posterior mitigates semantic drift in class-incremental segmentation. *arXiv preprint arXiv:2502.04870*, 2025.

Anqi Zhang and Guangyu Gao. Background adaptation with residual modeling for exemplar-free class-incremental semantic segmentation. In *European Conference on Computer Vision*, pp. 166–183. Springer, 2024.

Zekang Zhang, Guangyu Gao, Zhiyuan Fang, Jianbo Jiao, and Yunchao Wei. Mining unseen classes via regional objectness: A simple baseline for incremental segmentation. *Advances in neural information processing systems*, 35:24340–24353, 2022.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022.

Yuchen Zhu, Cheng Shi, Dingyou Wang, Jiajin Tang, Zhengxuan Wei, Yu Wu, Guanbin Li, and Sibei Yang. Rethinking query-based transformer for continual image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4595–4606, 2025.

A THEORETICAL PROOFS

A.1 FORGETTING RATE ANALYSIS

For the forgetting rate in Definition 3.2, the average forgetting rate $\mathcal{E}_\tau(\theta)$is defined as the arithmetic average of the forgetting rates of historical tasks:

$$\mathcal{E}_\tau(\theta) = \mathcal{L}_\tau^{val}(\theta) - \mathcal{L}_\tau^{val}(\theta_\tau^*)$$

Applying Taylor expansion to the above equation gives us:

$$\mathcal{E}_\tau(\theta) = (\theta - \theta_\tau^*)^\intercal \nabla \mathcal{L}_\tau^{val}(\theta_\tau^*) + \frac{1}{2}(\theta - \theta_\tau^*)^\intercal \mathbf{H}_\tau(\theta_\tau^*)(\theta - \theta_\tau^*) + O(\|\theta - \theta_\tau^*\|^3) \tag{10}$$

In Equation (10):

- First order $(\theta - \theta_\tau^*)^\intercal \nabla \mathcal{L}_\tau^{val}(\theta_\tau^*)$: said loss function near the optimal parameters of linear change;
- Second order terms $\frac{1}{2}(\theta - \theta_\tau^*)^\intercal \mathbf{H}_\tau(\theta_\tau^*)(\theta - \theta_\tau^*)$: by Hessian Matrix $\mathbf{H}_\tau(\theta_\tau^*)$ said the local curvature of loss function;
- high order events $O(\|\theta - \theta_\tau^*\|^3)$: said the higher order nonlinear effects.

In Assumption 3.1: $\nabla \mathcal{L}_\tau(\theta_\tau^*) \leq \epsilon$, including ϵ is a limitless tends to zero (in the optimal parameter θ_τ^* , The gradient of the loss function has gone to zero), so the first-order term in the Taylor expansion is ignored, and the forgetting rate is dominated by the second-order term:

$$\mathcal{E}_\tau(\theta) = \frac{1}{2}(\theta - \theta_\tau^*)^\intercal \mathbf{H}_\tau(\theta_\tau^*)(\theta - \theta_\tau^*) + O(\|\theta - \theta_\tau^*\|^3) \tag{11}$$

For two specific tasks $i, j$and the optimal parameters θ_j^*on the tasks j(assuming that the parameters vary within a range of δ, That's $\|\theta - \theta_t^*\| \leq \delta$) (we've added the range of the parameter to our assumption). We find that based on Equation (11), this forgetting rate can be simplified to Equation (12), and the simplified formula is as follows:

$$\begin{aligned}
\mathcal{E}_i(\theta_j^*) &= \frac{1}{2}(\theta_j^* - \theta_i^*)^\intercal \mathbf{H}_i(\theta_i^*)(\theta_j^* - \theta_i^*) + \mathcal{O}(\delta^3) \\
&= \frac{1}{2}\left(\sum_{\tau=i+1}^{j} \Delta_\tau\right)^\intercal \mathbf{H}_i(\theta_i^*)\left(\sum_{\tau=i+1}^{j} \Delta_\tau\right) + \mathcal{O}(\delta^3) \\
&= \frac{1}{2}\left(\Delta_j + \sum_{\tau=i+1}^{j-1} \Delta_\tau\right)^\intercal \mathbf{H}_i(\theta_i^*)\left(\Delta_j + \sum_{\tau=i+1}^{j-1} \Delta_\tau\right) + \mathcal{O}(\delta^3) \\
&= \underbrace{\frac{1}{2}\left(\sum_{\tau=i+1}^{j-1} \Delta_\tau\right)^\intercal \mathbf{H}_i(\theta_i^*)\left(\sum_{\tau=i+1}^{j-1} \Delta_\tau\right)}_{\mathcal{E}_i(\theta_{j-1}^*)} + \frac{1}{2}\Delta_j^\intercal \mathbf{H}_j(\theta_j^*)\Delta_j \\
&\quad + \frac{1}{2}\left(\sum_{\tau=i+1}^{j-1} \Delta_\tau\right)^\intercal \mathbf{H}_i(\theta_i^*)\Delta_j + \frac{1}{2}\Delta_j^\intercal \mathbf{H}_i(\theta_i^*)\left(\sum_{\tau=i+1}^{j-1} \Delta_\tau\right) + \mathcal{O}(\delta^3) \\
&= \mathcal{E}_i(\theta_{j-1}^*) + \frac{1}{2}\Delta_j^\intercal \mathbf{H}_i(\theta_i^*)\Delta_j + \left(\sum_{\tau=i+1}^{j-1} \Delta_\tau\right)^\intercal \mathbf{H}_i(\theta_i^*)\Delta_j + \mathcal{O}(\delta^3)
\end{aligned}$$

The Forgetting rate for task i and optimal parameter θ_j^* can be expressed as Equation (12)

$$\mathcal{E}_i(\theta_j^*) = \mathcal{E}_i(\theta_{j-1}^*) + \frac{1}{2}\Delta_j^\intercal \mathbf{H}_i(\theta_i^*)\Delta_j + \left(\sum_{\tau=i+1}^{j-1}\Delta_\tau\right)^\intercal \mathbf{H}_i(\theta_i^*)\Delta_j + \mathcal{O}(\delta^3) \tag{12}$$

In Equation (12), it consists of several parts: past forgotten rate: $\mathcal{E}_i(\theta_{j-1}^*)$, namely after the completion of the task $j-1$ for task i forgotten; independent effects of the current parameter update: $\frac{1}{2}\Delta_j^\intercal \mathbf{H}_i(\theta_i^*)\Delta_j$, directly caused by the parameter update $\Delta_j$of task j; interaction between historical and current updates : $\left(\sum_{\tau=i+1}^{j-1}\Delta_\tau\right)^\intercal \mathbf{H}_i(\theta_i^*)\Delta_j$, which reflects the nonlinear superposition effect of the parameter update sequence, and represents the inner product of the historical update and the current update. If the two directions are negatively correlated under the measure of $\mathbf{H}_i(\theta_i^*)$(e.g., orthogonal or reverse), forgetting may be alleviated. On the contrary, if the direction is consistent, the forgetting is aggravated.

Similarly, for the average forgetting rate $\bar{\mathcal{E}}_t(\theta_t^*) = \frac{1}{t-1}\sum_{\tau=1}^{t-1}\mathcal{E}_\tau(\theta_t^*)$, we can also simplify it through Equation (11)

$$\begin{aligned}
\bar{\mathcal{E}}_t(\theta_t^*) &= \frac{1}{t-1}\sum_{\tau=1}^{t-1}\left(\mathcal{E}_\tau(\theta_{t-1}^*) + \frac{1}{2}\Delta_t^\intercal \mathbf{H}_\tau(\theta_\tau^*)\Delta_t + \left(\sum_{o=\tau+1}^{t-1}\Delta_o\right)^\intercal \mathbf{H}_\tau(\theta_\tau^*)\Delta_t\right) + \mathcal{O}(\delta^3) \\
&= \frac{1}{t-1}\left(\sum_{\tau=1}^{t-2}\left(\mathcal{E}_\tau(\theta_{t-1}^*)\right) + \frac{1}{2}\sum_{\tau=1}^{t-1}\Delta_t^\intercal \mathbf{H}_\tau(\theta_\tau^*)\Delta_t + \sum_{\tau=1}^{t-1}\left(\sum_{o=\tau+1}^{t-1}\Delta_o\right)^\intercal \mathbf{H}_\tau(\theta_\tau^*)\Delta_t\right) + \mathcal{O}(\delta^3) \\
&= \frac{t-2}{t-1}\bar{\mathcal{E}}_{t-1}(\theta_{t-1}^*) + \frac{1}{2(t-1)}\Delta_t^\intercal\left(\sum_{\tau=1}^{t-1}\mathbf{H}_\tau(\theta_\tau^*)\right)\Delta_t + \frac{1}{t-1}\left(\underbrace{\sum_{\tau=1}^{t-1}(\theta_{t-1}^* - \theta_\tau^*)^\intercal \mathbf{H}_\tau(\theta_\tau^*)}_{v_t^\intercal}\right)\Delta_t \\
&+ \mathcal{O}(\delta^3) \\
&= \frac{1}{t-1}\left((t-2)\bar{\mathcal{E}}_{t-1}(\theta_{t-1}^*) + \frac{1}{2}\Delta_t^\intercal(\sum_{\tau=1}^{t-1}\mathbf{H}_\tau(\theta_\tau^*))\Delta_t + v_t^\intercal\Delta_t\right) + \mathcal{O}(\delta^3)
\end{aligned}$$

In the above simplification, each average forgetting rate has a specific v, which we denote as v_t, representing that it belongs to $\bar{\mathcal{E}}_t(\theta_t^*)$

$$\bar{\mathcal{E}}_t(\theta_t^*) = \frac{1}{t-1}\left((t-2)\cdot\bar{\mathcal{E}}_{t-1}(\theta_{t-1}^*) + \frac{1}{2}\Delta_t^\intercal(\sum_{o=1}^{t-1}\mathbf{H}_o^\star)\Delta_t + v_t^\intercal\Delta_t\right) + \mathcal{O}(\delta^3) \tag{13}$$

From this equation, we can observe that the average forgetting rate for task t, $\bar{\mathcal{E}}_t(\theta_t^*)$, is influenced by three principal terms (ignoring the higher-order term $\mathcal{O}(\delta^3)$) that are averaged:

- $(t-2)\cdot\bar{\mathcal{E}}_{t-1}(\theta_{t-1}^*)$ relates to the average forgetting rate of the immediately preceding task.
- $\frac{1}{2}\Delta_t^\intercal(\sum_{o=1}^{t-1}\mathbf{H}_o^\star)\Delta_t$ captures the impact of the parameter change Δ_t in conjunction with the cumulative Hessian matrices from all prior tasks (from $o=1$ to $t-1$).

Building on these observations, a further hypothesis can be formulated: It is proposed that if the average forgetting rate for each task from $k=2$ up to $k=t-1$ is zero (i.e., the first two items in $\bar{\mathcal{E}}_k(\theta_k^*)$ is zero for all $k\in\{2,3,\ldots,t-1\}$), then for the current t-th task, the component $v^{\mathrm{T}}\Delta_t$ is also hypothesized to be zero.

We will use mathematical induction to prove this hypothesis. At first, we suppose the first two terms of $\bar{\mathcal{E}}_2(\theta_2^*)$ are both 0, then we can get $v_3 = 0$:

$$\begin{aligned}\bar{\mathcal{E}}_3(\theta_3^*) &= \frac{1}{2}\left(1 \cdot \bar{\mathcal{E}}_2(\theta_2^*) + \frac{1}{2}\Delta_3^\intercal(H_2(\theta_2^*) + H_1(\theta_1^*))\Delta_3 + \sum_{t=1}^{2}(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_t)^\intercal \boldsymbol{H}_t^\star \boldsymbol{\Delta}_3\right) \\ &= 0 + \frac{1}{2}\Delta_3^\intercal(\frac{1}{2}H_2(\theta_2^*) + \frac{1}{2}H_1(\theta_1^*))\Delta_3 + \frac{1}{2}\underbrace{\Delta_2^\intercal H_1^1}_{=\mathbf{0}}\Delta_3 + \frac{1}{2}\underbrace{(\boldsymbol{\theta}_2^\intercal - \boldsymbol{\theta}_2)}_{=\mathbf{0}}H_t^\star\Delta_3\end{aligned}.$$

Then, we suppose the first tow terms of $\bar{\mathcal{E}}_{t-1}(\theta_{t-1}^*)$ is zero, we find that:

$$\begin{aligned}v_t - v_{t-1} &= \sum_{o=1}^{t-1}(\theta_{t-1}^* - \theta_o^*)^\intercal \mathbf{H}_o\left(\theta_o^*\right) - \sum_{o=1}^{t-2}(\theta_{t-2}^* - \theta_o^*)^\intercal \mathbf{H}_o\left(\theta_o^*\right) \\ &= \sum_{o=1}^{t-2}(\theta_{t-1}^* - \theta_o^*)^\intercal \mathbf{H}_o\left(\theta_o^*\right) - \sum_{o=1}^{t-2}(\theta_{t-2}^* - \theta_o^*)^\intercal \mathbf{H}_o\left(\theta_o^*\right) \\ &= \sum_{o=1}^{t-2}\left(\theta_{t-1}^* - \theta_o^* - \theta_{t-2}^* + \theta_o^*\right)^\intercal \mathbf{H}_o\left(\theta_o^*\right)\end{aligned}$$

Since in $\bar{\mathcal{E}}_{t-1}(\theta_{t-1}^*)$, $\frac{1}{2}\Delta_{t-1}^\intercal \sum_{o=1}^{t-2} H_o\left(\boldsymbol{\theta}_o^*\right)\Delta_{t-1} = 0$, So we find that: $v_t - v_{t-1} = 0$.

Also we can get the conclusion that:

$$v_t - v_{t-1} = \Delta_{t-1}^\intercal \sum_{\tau=1}^{t-2}\mathbf{H}_\tau(\theta_\tau^*) \tag{14}$$

We can say that if $\bar{\mathcal{E}}_\tau(\theta_\tau^*) = 0, \forall \tau < t$, then $\bar{\mathcal{E}}_t(\theta_t^*) = \frac{1}{2(t-1)}\Delta_t^\intercal\left(\sum_{i=1}^{t-1}\mathbf{H}_i(\theta_i^*)\right)\Delta_t$, which is show in Theorem 3.3 (1).

In Theorem 3.3 (2), we need to proof the statement: $\mathcal{E}_\tau(\theta_t^*) = 0, \forall \tau < t \iff \Delta_t^\intercal\left(\sum_{\tau'=1}^{t-1}\mathbf{H}_{\tau'}(\theta_{\tau'}^*)\right)\Delta_t = 0$. It is clear that when $\mathcal{E}_\tau(\theta_t^*) = 0, \forall \tau < t$, then $\bar{\mathcal{E}}_\tau(\theta_\tau^*) = 0, \forall \tau < t$, using conclusion in Theorem 3.3(1), we can get the result: $\Delta_t^\intercal\left(\sum_{\tau'=1}^{t-1}\mathbf{H}_{\tau'}(\theta_{\tau'}^*)\right)\Delta_t = 0$.

When $\Delta_t^\intercal\left(\sum_{\tau'=1}^{t-1}\mathbf{H}_{\tau'}(\theta_{\tau'}^*)\right)\Delta_t = 0$, using Assumption 3.1, all Hessian matrix $\mathbf{H}_\tau(\theta_\tau^*)$ is positive semi-definite, then we have: $\Delta_\tau^\intercal(\sum_{o=1}^{\tau-1}\mathbf{H}_o(\theta_o^*)) = 0, \forall \tau < t$, using Equation (14), we have: $v_2 \leq v_3 \cdots \leq v_t \leq v_{t+1}$. In definition, $\bar{\mathcal{E}}_2(\theta_2^*)$ is zero, and $\bar{\mathcal{E}}_3(\theta_3^*) = \frac{1}{8}\Delta_3^\intercal(\sum_{o=1}^{2}\mathbf{H}_o\theta_o^*)\Delta_3 = 0$, we have $\bar{\mathcal{E}}_t(\theta_t^*) = 0$ which can say that $\mathcal{E}_\tau(\theta_t^*) = 0, \forall \tau < t$

A.2 Other Zero Forgetting strategy

A.2.1 Orthogonal Gradient Method

We follow the setting of Farajtabar et al. (2020), where $\mathcal{L}$ is a non-negative loss function(CE or BCE loss). We also assume that the symbol f_θ represents the parameter θ used by the model ($f_\theta^c(x)$ means one of the output's channel which is about class c) and N_t represents the total amount of data used in the tasks from 1 to t.

In Orthogonal Gradient Method (Farajtabar et al., 2020), they want to address catastrophic forgetting in continual learning by keeping the updates for a new task orthogonal to the gradient directions associated with previous tasks' predictions. Formally, equal to Equation (15).

$$\langle \Delta_t^i, \nabla_{\theta_{t-1}^*} f_{\theta_{t-1}^*}^c(x_\tau)\rangle = 0 \quad \forall c \in \mathcal{C}_{[1:t-1]}, x_\tau \in D_\tau, \tau < t\,, \tag{15}$$

where Δ_t^i denotes the the i-th step for update.

Due to the previous studies (Schraudolph, 2002), the hessian matrix of the loss can be decomposed as two other matrices: the outer-product Hessian and the functional Hessian, and at the optimum parameter for loss function, the functional Hessian is negligible (Singh et al., 2021). So we can write the approximation of the Hessian matrix under the optimal parameters in task τ:

$$\mathbf{H}_\tau(\theta_\tau^*) = \frac{1}{N_\tau}\sum_{i=1}^{N_\tau} \nabla_{\theta_\tau^*} f_{\theta_\tau^*}(x_i)(\nabla_f^2 \mathcal{L}_\tau(x_i, y_i))\nabla_{\theta_\tau^*} f(x_i)^\intercal$$

When the parameter update follows Equation (15), it essentially satisfies the condition that the binomial is 0 mentioned in our zero forgetting condition in Theorem 3.3.

A.3 Why Parameters Isolation is Zero Forgetting

We find that when the SPI strategy is used, the sum of the historical Hessian matrix($\sum_{\tau=1}^{t-1} \mathbf{H}_\tau(\theta_\tau^*)$) is similar to a semi-positive definite block diagonal matrix to its parameter subspace:

$$\sum_{\tau=1}^{t-1} \mathbf{H}_\tau(\theta_\tau^*) = \begin{pmatrix} \mathbf{H}_1(\theta_1^*) & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2(\theta_2^*) & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{H}_{t-1}(\theta_{t-1}^*) & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{16}$$

Where:

- According to Assumption 3.1, Each $\mathbf{H}_\tau(\theta_\tau^*)$ on the diagonal is a square matrix representing the Hessian for task τ and is a semi-positive matrix. The dimensions of $\mathbf{H}_\tau(\theta_\tau^*)$ correspond to the number of new parameters introduced for task τ.
- The $\prime$ symbols represent zero, indicating that the Hessian components for parameters of different tasks are decoupled due to SPI strategy.

Just as expressed in Equation (16), the sum of the historical tasks' Hessian matrices $\sum_{\tau=1}^{t-1} \mathbf{H}_\tau(\theta_\tau^*)$ and the parameter update for new task Δ_t are not in the same subspace, and the product between them must be 0 which is the zero forgetting condition in Theorem 3.3.

A.4 Cascade Modeling

Variable Definitions. We first clarify the random variables used throughout this derivation:

- $Y_p \in \mathcal{C}_{[1:t]} \cup \{0\}$: the random variable representing the class label assigned to pixel p, where 0 denotes background.
- $Z_c \in \{0, 1\}$: a binary random variable indicating whether class c is present in image $\mathbf{x}$. Formally, $Z_c = 1 \Leftrightarrow \exists\, p : Y_p = c$.

We start from the task-level factorization introduced in Equation (4):

$$P(Y_p = c \mid \mathbf{x}) = P\big(Y_p = c \mid \mathbf{x}, \mathcal{T}_t\big)\, P\big(\mathcal{T}_t \mid \mathbf{x}\big), \quad c \in \mathcal{C}_t,$$

where $\mathcal{T}_t$ denotes the task that first introduced class c. Under SPI, classes are disjoint across tasks: $\mathcal{C}_\tau \cap \mathcal{C}_t = \varnothing$ for all $\tau \neq t$.

Lemma A.1 (Task Prior Decomposition)**.** *For any image* $\mathbf{x}$*,*

$$P(\mathcal{T}_t \mid \mathbf{x}) = P\left(\bigvee_{c\in\mathcal{C}_t} Z_c = 1 \;\middle|\; \mathbf{x}\right) \approx \sum_{c\in\mathcal{C}_t} P(Z_c = 1 \mid \mathbf{x}), \tag{17}$$

where $P(Z_c = 1 \mid \mathbf{x})$ *is the probability that class* c *exists somewhere in image* $\mathbf{x}$*, and the approximation holds when class co-occurrence within a single task is rare.*

Proof. Since classes in $\mathcal{C}_t$ are mutually exclusive and task $\mathcal{T}_t$ is active if and only if at least one class from $\mathcal{C}_t$ appears in $\mathbf{x}$, we have

$$P(\mathcal{T}_t \mid \mathbf{x}) = P\left(\bigvee_{c \in \mathcal{C}_t} Z_c = 1 \;\middle|\; \mathbf{x}\right) \approx \sum_{c \in \mathcal{C}_t} P(Z_c = 1 \mid \mathbf{x})$$

by the inclusion-exclusion principle (approximated when inter-class overlap is negligible). □

From Task-Level to Class-Level Factorization. The task-level factorization necessarily reduces to a simpler class-level form that eliminates the problematic task prior $P(\mathcal{T}_t \mid \mathbf{x})$.

Theorem A.2 (Cascade Factorization)**.** *For any pixel p, class c, and image $\mathbf{x}$:*

$$P(Y_p = c \mid \mathbf{x}) = P\big(Y_p = c \mid \mathbf{x}, Z_c = 1\big) \cdot P(Z_c = 1 \mid \mathbf{x}). \tag{18}$$

Proof. We derive this using the law of total probability over the class existence variable Z_c:

$$\begin{aligned} P(Y_p = c \mid \mathbf{x}) = {} & P(Y_p = c \mid \mathbf{x}, Z_c = 1) \cdot P(Z_c = 1 \mid \mathbf{x}) \\ & + P(Y_p = c \mid \mathbf{x}, Z_c = 0) \cdot P(Z_c = 0 \mid \mathbf{x}). \end{aligned}$$

The key insight: if class c is not present in the image ($Z_c = 0$), then no pixel can be labeled as c. Therefore, $P(Y_p = c \mid \mathbf{x}, Z_c = 0) = 0$. This eliminates the second term, yielding:

$$P(Y_p = c \mid \mathbf{x}) = P(Y_p = c \mid \mathbf{x}, Z_c = 1) \cdot P(Z_c = 1 \mid \mathbf{x}).$$

□

Normalization over All Classes. The final pixel-level prediction requires normalization over all candidate classes. Let $\mathcal{C}_{\text{pred}} = \{c : P(Z_c = 1 \mid \mathbf{x}) \geq \alpha\}$ be the set of predicted classes after thresholding. The normalized probability is:

$$P(Y_p = c \mid \mathbf{x}) = \frac{P(Y_p = c \mid \mathbf{x}, Z_c = 1) \cdot P(Z_c = 1 \mid \mathbf{x})}{\sum_{c' \in \mathcal{C}_{\text{pred}} \cup \{0\}} P(Y_p = c' \mid \mathbf{x}, Z_{c'} = 1) \cdot P(Z_{c'} = 1 \mid \mathbf{x})}, \tag{19}$$

where $c' = 0$ corresponds to the background class with $P(Z_0 = 1 \mid \mathbf{x}) = 1$ (background is always present).

Implications for CogCaS Architecture. Theorem A.2 proves that under SPI, the cascade factorization in Equation (18) is the unique probabilistically consistent decomposition. This directly motivates our two-phase CogCaS design:

- **Phase I**: Multi-label classifier estimates class existence probabilities $P(Z_c = 1 \mid \mathbf{x})$
- **Phase II**: Binary segmentation heads model conditional segmentation $P(Y_p = c \mid \mathbf{x}, Z_c = 1)$

This decomposition eliminates the ill-defined task prior, enables complete parameter isolation for zero-forgetting, and aligns with optimal Bayesian factorization principles.

A.5 Sufficient and Necessary Conditions for CISS via Cascade Decoupling

The cascade factorization in Theorem A.2 is a probabilistically consistent decomposition, but probabilistic consistency by itself does not explain why decoupling Phase I and Phase II is the right design choice for CISS. In this subsection we close that gap with two complementary results. The first, a *sufficiency* result, shows that the Phase I and Phase II cross-entropies together form an additive decomposition of the CISS cross-entropy, so improving either phase monotonically tightens an upper bound on the CISS loss. The second, a *necessity* result, shows that every CISS predictor – including monolithic softmax classifiers that never explicitly model class existence – implicitly contains a class-existence detector, and its loss is bounded below by the loss of this implicit detector. Together, the two statements rule out the possibility that ignoring the routing sub-problem is harmless: it can be optimized explicitly (as in CogCaS) or left to degrade silently, but it cannot be eliminated. The cross-entropy framework we adopt below is adapted from the image-level class-incremental analysis of Kim et al. (2022); the pixel-level CISS structure permits sharper, often equality-form statements, which we prove directly.

Setup. We work with two random variables, both already used in the cascade modeling above. For an image $\mathbf{x}$ and a pixel p:

- Y_p is the (unknown) class label of pixel p, taking values in $\mathcal{C}_{1:t} \cup \{0\}$;
- $Z_c \in \{0, 1\}$ indicates whether class c appears anywhere in $\mathbf{x}$, with $Z_c = 1$ if and only if $Y_p = c$ for at least one pixel p.

A simple but key observation: if pixel p has true label c^*, then class c^* appears in the image, so $Z_{c^*} = 1$ holds with probability one. We refer to this fact as the *deterministic shadowing* of Z by Y and use it repeatedly below.

To compare predictors of different forms (cascade or monolithic) we measure their quality by the cross-entropy at the ground truth. For an image $\mathbf{x}$ and a pixel p with ground-truth class c^*, define

$$H_{\mathrm{CISS}}(\mathbf{x}, p) := -\log P(Y_p = c^* \mid \mathbf{x}), \quad \text{(full CISS predictor at the true class)} \tag{20}$$

$$H_{\mathrm{II}}(\mathbf{x}, p) := -\log P(Y_p = c^* \mid \mathbf{x}, Z_{c^*} = 1), \quad \text{(conditional segmenter at the true class)} \tag{21}$$

$$H_{\mathrm{I}}(\mathbf{x}, c^*) := -\log P(Z_{c^*} = 1 \mid \mathbf{x}). \quad \text{(class-existence detector at the true class)} \tag{22}$$

Smaller H means a better-calibrated prediction at the ground truth.

Sufficiency: Phase I and Phase II are additive.

Theorem A.3 (Pixelwise loss decomposition)**.** *For any image $\mathbf{x}$ and any pixel p with ground-truth class c^*,*

$$H_{\mathrm{CISS}}(\mathbf{x}, p) \;=\; H_{\mathrm{II}}(\mathbf{x}, p) \;+\; H_{\mathrm{I}}(\mathbf{x}, c^*). \tag{23}$$

In particular, $H_{\mathrm{I}}(\mathbf{x}, c^*) \le \delta$ and $H_{\mathrm{II}}(\mathbf{x}, p) \le \varepsilon$ jointly imply $H_{\mathrm{CISS}}(\mathbf{x}, p) \le \delta + \varepsilon$.

Proof. Theorem A.2 gives the cascade factorization $P(Y_p = c^* \mid \mathbf{x}) = P(Y_p = c^* \mid \mathbf{x}, Z_{c^*} = 1)\, P(Z_{c^*} = 1 \mid \mathbf{x})$. Taking $-\log$ on both sides yields equation 23. The implication on bounded cross-entropies is immediate. □

We emphasise that equation 23 is an exact equality, not an upper bound: deterministic shadowing forces $Z_{c^*} = 1$ with probability one at the ground-truth pixel, so the law of total probability collapses to a single term and no slack arises from the $Z_{c^*} = 0$ branch. The CISS loss at a true-class pixel therefore decomposes exactly into the loss the conditional segmenter incurs on that pixel plus the loss the class-existence detector incurs on the corresponding class. The two contributions do not overlap, and improvements to either phase translate one-for-one into improvements of the CISS upper bound, regardless of the quality of the other.

Image-level lift. Summing the per-pixel equality equation 23 over all pixels of an image gives

$$\sum_{p} H_{\mathrm{CISS}}(\mathbf{x}, p) = \sum_{p} H_{\mathrm{II}}(\mathbf{x}, p) + \sum_{c \in \mathcal{C}_{1:t}} n_c(\mathbf{x})\, H_{\mathrm{I}}(\mathbf{x}, c), \tag{24}$$

where $n_c(\mathbf{x}) = |\{p : Y_p = c\}|$ counts the ground-truth pixels of class c. The Phase I term is therefore weighted by class frequency at the image level. Rare classes contribute little weight to the standard image-averaged loss, which motivates the use of an asymmetric multi-label objective in $\mathcal{L}_{\mathrm{align}}$ (see Appendix B.1): the per-class formulation has to compensate explicitly for the long-tailed distribution of n_c.

Phase I is, by construction, per-class existence detection. The next observation is direct enough to state as a corollary.

Corollary A.4 (Phase I as per-class binary detection)**.** *Phase I is, by construction, a collection of independent binary detectors, one per class, each predicting whether its class is present in the input image. The Phase I cross-entropy summed over classes is therefore exactly a sum of per-class binary detection cross-entropies:*

$$\sum_{c \in \mathcal{C}_{1:t}} H\big(\mathbb{1}[Z_c = 1],\, P(Z_c = 1 \mid \mathbf{x})\big) = \sum_{c} H_{\mathrm{exist},c}(\mathbf{x}), \tag{25}$$

where $H_{\mathrm{exist},c}(\mathbf{x}) := H\big(\mathbb{1}[Z_c = 1],\, P(Z_c = 1 \mid \mathbf{x})\big)$ is the standard binary cross-entropy of an existence detector for class c.

Proof. $P(Z_c = 1 \mid \mathbf{x})$ is a Bernoulli prediction by construction, and the cross-entropy of a Bernoulli prediction against its $0/1$ ground truth is exactly the binary detection cross-entropy on the right-hand side. The equality is by definition. □

The equality is structural rather than approximate, because Phase I is instantiated as a multi-label predictor with one Bernoulli output per class – as opposed to, say, a single categorical softmax that would only admit such a relation up to constant factors. Combined with Theorem A.3, Corollary A.4 formalizes the modular-improvability claim in our introduction: any external advance in per-class existence detection, OOD detection, or open-vocabulary recognition plugs directly into our framework and tightens the CISS upper bound.

Necessity: every CISS predictor contains a Phase I detector. The sufficiency result describes what *can* be achieved by the cascade factorization. The next, and structurally more interesting, result describes what *cannot be avoided* by any CISS predictor whatsoever. The key construction is to read off an implicit class-existence detector from any pixel-level posterior. The appropriate pixel-to-image lift here is $\max_p$ aggregation: a class is present in the image as soon as any pixel is predicted to belong to it.

Theorem A.5 (Hidden routing in any CISS predictor)**.** *Let $P(Y_p = c \mid \mathbf{x})$ be any CISS predictor, regardless of how it is implemented. Define the induced class-existence detector by max-pooling the per-pixel posterior of class c,*

$$\widehat{P}(Z_c = 1 \mid \mathbf{x}) := \max_p P(Y_p = c \mid \mathbf{x}), \tag{26}$$

and the induced conditional segmenter by

$$\widehat{P}(Y_p = c \mid \mathbf{x}, Z_c = 1) := \frac{P(Y_p = c \mid \mathbf{x})}{\widehat{P}(Z_c = 1 \mid \mathbf{x})}. \tag{27}$$

Let $\widehat{H}_{\mathrm{I}}$ and $\widehat{H}_{\mathrm{II}}$ denote the cross-entropies of the induced detector and segmenter at the ground truth. Then for every pixel p with ground-truth class c^*,

$$\widehat{H}_{\mathrm{I}}(\mathbf{x}, c^*) \le H_{\mathrm{CISS}}(\mathbf{x}, p), \qquad \widehat{H}_{\mathrm{II}}(\mathbf{x}, p) \le H_{\mathrm{CISS}}(\mathbf{x}, p). \tag{28}$$

Equivalently, by contraposition: if every class-existence detector that one can read off from a CISS predictor satisfies $H_{\mathrm{I}} > \eta$, then the predictor itself satisfies $H_{\mathrm{CISS}} > \eta$.

Proof. For the Phase I bound, equation 26 gives $\widehat{P}(Z_{c^*} = 1 \mid \mathbf{x}) \ge P(Y_p = c^* \mid \mathbf{x})$ for every pixel p, so taking $-\log$ flips the inequality and yields $\widehat{H}_{\mathrm{I}}(\mathbf{x}, c^*) \le H_{\mathrm{CISS}}(\mathbf{x}, p)$. For the Phase II bound, $\widehat{P}(Z_{c^*} = 1 \mid \mathbf{x}) \le 1$ implies $\widehat{P}(Y_p = c^* \mid \mathbf{x}, Z_{c^*} = 1) = P(Y_p = c^* \mid \mathbf{x}) / \widehat{P}(Z_{c^*} = 1 \mid \mathbf{x}) \ge P(Y_p = c^* \mid \mathbf{x})$, hence $\widehat{H}_{\mathrm{II}}(\mathbf{x}, p) \le H_{\mathrm{CISS}}(\mathbf{x}, p)$. □

The contrapositive form is the substantive claim. A CISS predictor cannot achieve cross-entropy better than η unless some class-existence detector readable from its outputs already achieves cross-entropy at most η. Conventional softmax-based CISS methods do not train such a detector explicitly; they only hope a usable one emerges from the joint softmax over all class logits. When it does not, no amount of within-class refinement can lower the CISS loss below the loss of the implicit detector. The empirical manifestation of this ceiling is the cross-task logit miscalibration shown in Figure 2(A): with task-specific heads trained in isolation under SPI, the logits live on incompatible scales, so the implicit detector read off through a softmax is uniformly poor. CogCaS responds to Theorem A.5 by making the implicit detector explicit: Phase I is trained directly on the class-existence label, with its own loss and its own SPI-protected parameters, and the cascade composition rule recovers the CISS posterior by exact multiplication rather than by softmax-induced competition.

Synthesis. Theorems A.3 and A.5, together with Corollary A.4, give a tight characterization of CISS under SPI:

1. **Sufficiency.** The Phase I and Phase II cross-entropies add up to the CISS cross-entropy at every true-class pixel, so improving either phase monotonically improves the CISS upper bound.

2. **Modularity.** Phase I is by construction a per-class binary detector, so any external advance in OOD detection or open-vocabulary recognition plugs directly into our framework.
3. **Necessity.** Every CISS predictor implicitly contains a class-existence detector via max-pixel pooling, and the predictor's cross-entropy is lower-bounded by the cross-entropy of this implicit detector; a CISS method that does not train such a detector is structurally constrained by it.

CogCaS is the minimal architecture that operationalizes all three: it instantiates Phase I and Phase II as separate SPI-protected modules, trains each with a phase-appropriate objective, and recovers the CISS prediction by exact cascade composition.

B Implementation Details

B.1 Training and Inference

CISS Training. For every new task $\mathcal{T}_t$ and each new class $c \in \mathcal{C}_t$, we add a learnable text prompt $\boldsymbol{p}_c$ for Phase I and a class-specific binary segmentation head H_c (Deeplab-v3's ASPP module + 1×1 conv) for Phase II; all previous parameters stay frozen (SPI, Section 3.3).

Near-OOD. Furthermore, during the training stage, we manually construct near-out-of-distribution (near-OOD) data based on the data available for the current task to enhance the model's robustness. As illustrated in Figure 4, this process is divided into two distinct stages: Phase I and Phase II.

The construction of near-OOD samples follows a generic and straightforward augmentation strategy rather than a carefully curated or hand-crafted procedure. Specifically, given a training image, we randomly sample patches from the background regions (i.e., regions not belonging to any current foreground class) and use them to fully overlay (i.e., completely replace the pixels of) patches located within the labeled foreground regions. This patch-level replacement perturbs the original semantic content while preserving the overall image statistics, thereby producing samples that lie close to but outside the in-distribution manifold of the current task. We emphasize that this procedure requires no additional annotation, no external data, and no task-specific tuning, making it a widely applicable augmentation scheme for CISS.

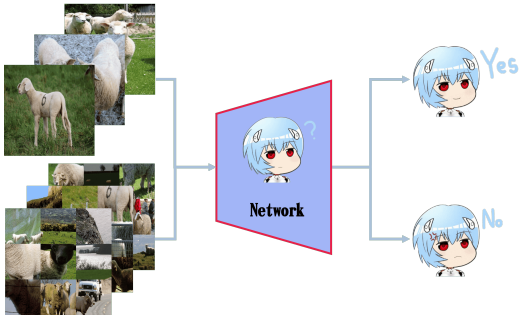


Figure 4: The figure illustrates near-OOD samples and the construction process.

Joint Training. In the joint training setting, we first train the classifier in Phase I. After the training for Phase I is complete, because we have access to the full dataset, we then select corresponding out-of-distribution data for each class-specific segmentation head in Phase II. This data is chosen from the dataset at a 1:1 ratio, and we ensure that for a specific class, its corresponding OOD samples do not contain any information about that class in any single pixel.

Parameter Overhead. Each additional class contributes ≈ 1.1 M parameters (4.2 MB at FP32), corresponding to 1.5 % of a ViT backbone.

Loss Functions. Phase I optimizes Eq. (7). Phase II is trained separately with the following segmentation loss:

$$\mathcal{L}_{\text{seg}} = \tfrac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \sum_{c \in \mathcal{C}_t} \Big[\alpha \, \text{Focal}(M_c, \hat{M}_c) + \beta \, \text{Dice}(M_c, \hat{M}_c) \Big]$$

$$\begin{aligned} \text{Focal}(M_c, \hat{M}_c) = -\tfrac{1}{N} \sum_{i=1}^{N} \Big[& \alpha_f \, M_{c,i} (1 - \hat{M}_{c,i})^{\gamma} \log(\hat{M}_{c,i}) \\ & + (1 - \alpha_f)(1 - M_{c,i})(\hat{M}_{c,i})^{\gamma} \log(1 - \hat{M}_{c,i}) \Big] \end{aligned}$$

$$\text{Dice}(M_c, \hat{M}_c) = \frac{2 \sum_i M_{c,i} \hat{M}_{c,i} + \varepsilon}{\sum_i M_{c,i} + \sum_i \hat{M}_{c,i} + \varepsilon}, \quad \varepsilon = 10^{-6}$$

Notation. $\mathcal{B}$: mini-batch; $\mathcal{C}_{1:t}$: all classes learned up to task t; $\mathcal{C}_t$: classes introduced at task t; $M_c, \hat{M}_c$: predicted / ground-truth masks; i: spatial index.

Optimisation Schedule. As stated in Section 4.1, each task is trained with Adam for 90 epochs with an initial learning rate of 1×10^{-5}, using weight decay 10^{-4} and batch size 20. The additional ResNet-101 experiment reported in Table 5 uses a different initial learning rate of 5×10^{-3} with cosine decay.

Algorithm 1 Inference Pipeline (per image)

Require: SigLIP2 image encoder Φ_v, text queries $\{\boldsymbol{t}_c\}$, segmentation backbone Φ, binary heads $\{H_c\}$, threshold α

$\boldsymbol{v} \leftarrow \text{Pool}(\Phi_v(x))$ {Phase I image embedding}
$P(Z_c{=}1 \mid x) \leftarrow \sigma(\langle \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{t}}_c \rangle / \tau)$ for all learned c {class existence}
$\mathcal{C}_{\text{pred}} \leftarrow \{\, c \mid P(Z_c{=}1 \mid x) \geq \alpha \,\}$
$F \leftarrow \Phi(x)$ {Phase II feature map}
for all $c \in \mathcal{C}_{\text{pred}}$ **do**
  $M_c \leftarrow H_c(F)$ {binary mask for class c}
end for
$M_{\text{final}} \leftarrow \mathcal{U}\big(\{M_c\}_{c \in \mathcal{C}_{\text{pred}}}\big)$ {mask fusion}
return M_{final}

Inference. Because only $|\mathcal{C}_{\text{pred}}|$ segmentation heads are activated, inference cost scales with the number of present classes rather than the total number of learned classes.

C Additional Experiments

C.1 Comparison with Large-Model-Based Methods

We compare CogCaS with recent large-model-based CISS methods in Table 5. SAM-based methods (e.g., DecoupleCSS (Guo et al., 2026)) leverage a foundation model with ∼632M parameters pretrained on 11M images, representing a fundamentally different resource regime compared to our standard backbones (ResNet-101: 44.5M; Swin-L: 197M). Our "Cognitive Cascade" architecture and theoretical zero-forgetting guarantee are orthogonal to backbone choice and can be readily integrated with SAM or similar models. CogCaS offers a favorable balance between performance and efficiency, particularly for resource-constrained deployment scenarios. The rows marked "Ours○" in Table 5 come from an additional experiment with a ResNet-101 backbone.

Table 5: Comparison with large-model-based CISS methods. ⋆: SAM-based (∼632M params). ‡: Mask2Former-based. ○: our additional experiment with a ResNet-101 backbone.

Method	Setting	Old	New	All
PASCAL VOC 2012				
DecoupleCSS (Guo et al., 2026)⋆	19-1	82.9	83.7	84.0
DecoupleCSS Guo et al. (2026)⋆	15-1	83.8	82.1	83.4
Ours○	15-1	75.5	71.4	74.4
ADE20K				
DecoupleCSS Guo et al. (2026)⋆	100-10	58.2	52.0	56.9
DecoupleCSS Guo et al. (2026)⋆	100-5	57.5	55.6	56.9
CIT-M2Former Ge et al. (2025)‡	100-10	56.9	38.2	50.7
CIT-M2Former Ge et al. (2025)‡	100-5	55.5	32.2	47.7
SimCIS Zhu et al. (2025)‡	100-10	49.7	27.4	42.3
SimCIS Zhu et al. (2025)‡	100-5	46.7	22.8	38.7
Ours○	100-10	49.2	44.0	47.4
Ours○	100-5	49.5	43.6	47.3

C.2 Effect of Task-Shared Parameters

Due to limited resources and time, our further investigations focused on the challenging 2-2 settings on the VOC dataset. To further validate our method, we unfroze the Encoder's final bottleneck layer, making it task-shared and trainable, as shown in Figure 5. Manually setting these task-shared parameters did not affect new class learning (matching baseline performance). However, this configuration, when combined with SPI, resulted in catastrophic forgetting, evidenced by a 37% drop in mIoU. By intentionally disrupting the SPI settings, we observed that while new classes learned normally, old classes experienced catastrophic forgetting. This observation, in reverse, further substantiates the correctness of our proposed method. What's more, adding class-specific LoRA to the backbone, the improvement obtained by our method is not significant (only improve 1% mIoU), but the parameters need to be changed with the task, which will significantly increase the inference time.

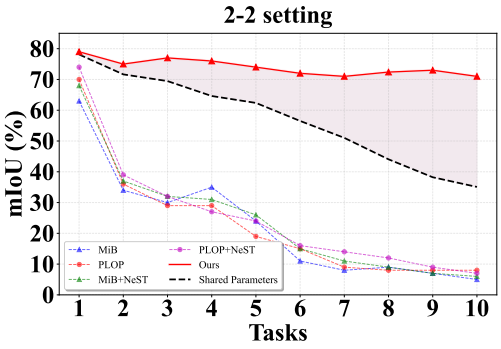


Figure 5: Impact on model performance when the task-shared parameters are set, evaluated on the VOC 2-2 setting.