PAST AS GUIDANCE, NOT JUST CONSTRAINT: CROSS-LEVEL KNOWLEDGE REORGANIZATION FOR CLASS-INCREMENTAL SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

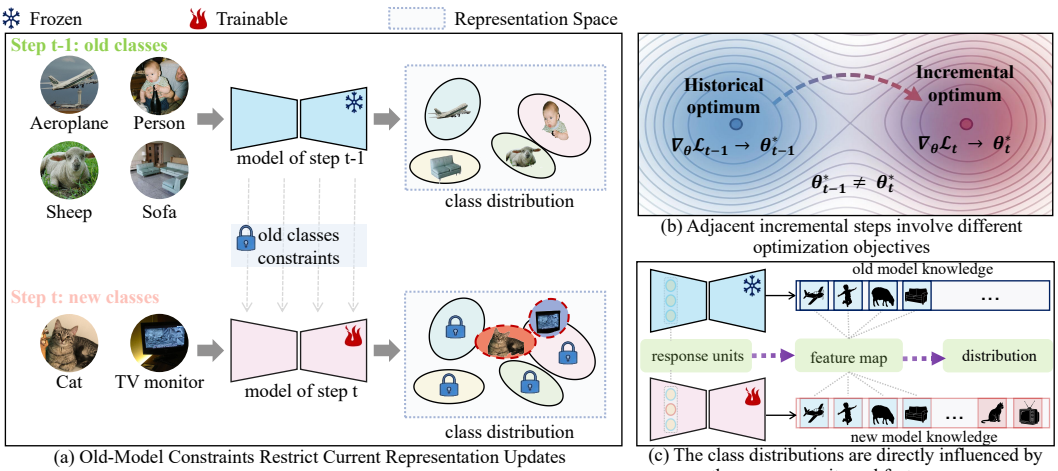
Class-incremental semantic segmentation (CISS) requires a model to learn new classes while retaining acquired knowledge without full access to historical training data. Existing regularization strategies alleviate catastrophic forgetting by preserving knowledge from previous steps. However, under fixed model capacity, newly introduced classes alter the optimization objective and require the representation to be reorganized. Our optimization analysis shows that adjacent incremental objectives generally correspond to different stationary states, while inherited constraints can restrict current-task-aligned adaptation. Insufficient representation reorganization can consequently increase class distribution overlap. We further empirically reveal a cross-level representation propagation pathway through controlled perturbation analysis: response-unit variations are systematically reflected in downstream feature variations, which are further associated with changes in class-conditional distributions. Together, the optimization analysis provides a rationale for representation reorganization as incremental objectives evolve, while the empirical evidence supports coordinating this reorganization across representation levels. Motivated by these findings, we propose Adaptive Knowledge Inheritance (AKI), a framework that coordinates knowledge reorganization across the response, feature, and distribution levels. Response Capacity Adaptation (RCA) constructs a compact response support and enforces prediction consistency between the complete-response and compact-support branches, reducing excessive dependence on the complete response space. Reusable Projection Alignment (RPA) preserves historical semantics along reusable directions identified from historical discriminability and observed current adaptation. Distribution Entanglement Penalization (DEP) models current old- and new-class distributions and suppresses their pairwise overlap. Extensive experiments on Pascal VOC and ADE20K across nine incremental settings validate the effectiveness of AKI and the complementary roles of its components in balancing historical preservation and current adaptation.

1 INTRODUCTION

Under the closed-set assumption, semantic segmentation has achieved remarkable progress, with semantic categories predefined during training Liao & Kong (2025); Qiao et al. (2025). In real-world deployment, however, new semantic concepts continually emerge, while historical images and annotations may be inaccessible due to storage, privacy, or communication constraints Yang et al. (2023); Bao et al. (2026). Retraining with all old and new data is therefore impractical, whereas learning only from newly available data can cause catastrophic forgetting. Class-incremental semantic segmentation (CISS) Cha et al. (2021); Maracani et al. (2021) addresses this challenge by learning new classes while retaining acquired knowledge without access to the complete historical training set. This capability is important for long-term deployed systems encountering unseen categories.

Existing CISS methods Douillard et al. (2021); Yuan & Zhao (2024) mitigate catastrophic forgetting through architectural adaptation, replay, or regularization. Architecture-based approaches Yoon et al. (2017); Qin et al. (2021) expand model capacity to accommodate successive knowledge, but require structural growth over incremental steps. Replay-based approaches Cha et al. (2021); Chen et al. (2025) retain or generate historical information, introducing additional storage or data-

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.



(a) Old-Model Constraints Restrict Current Representation Updates

(b) Adjacent incremental steps involve different optimization objectives

(c) The class distributions are directly influenced by the response units and feature maps

Figure 1: **Motivation of AKI.** (a) Under fixed model capacity, inherited constraints can restrict the representation reorganization required by current supervision, increasing old–new distribution overlap. (b) Cross-step optimization analysis shows that adjacent incremental objectives generally correspond to different stationary states, $\boldsymbol{\theta}^*_{t-1} \neq \boldsymbol{\theta}^*_t$, while inheritance constraints can restrict current-task-aligned adaptation. (c) Empirical analysis reveals cross-level coupling among response, feature, and distribution variations, motivating coordinated knowledge reorganization across three levels.

management requirements. Under fixed-capacity and replay-free settings, regularization-based approaches are particularly attractive. Representative techniques include knowledge distillation, prototype regularization, and contrastive learning Zhao et al. (2023); Wang et al. (2024); Yang et al. (2022); Shang et al. (2023); Baek et al. (2022); Cong et al. (2023). Recent methods further improve this paradigm by adaptively regulating historical knowledge preservation. Despite recent advances in adaptive knowledge preservation, insufficient representation reorganization can still leave class distributions entangled in the current representation space, as illustrated in Fig. 1(a).

We revisit this problem from a cross-step optimization perspective under fixed model capacity. As illustrated in Fig. 1(b), the model at step $t-1$ is optimized toward a stationary solution $\boldsymbol{\theta}^*_{t-1}$ associated with the previously observed classes. Once new classes are introduced at step t, the optimization objective changes and generally corresponds to a different stationary state $\boldsymbol{\theta}^*_t$. Our optimization-based analysis further shows that inherited constraints can restrict updates aligned with current adaptation when their optimization directions conflict with the current objective. These properties provide optimization guidance for knowledge reorganization: the cross-step objective shift motivates sufficient adaptation flexibility, while the interaction between inheritance and current optimization motivates selective regulation of historical constraints.

Importantly, the required reorganization is not isolated at a single representation level, as shown in Fig. 1(c). Our controlled perturbation analysis reveals a systematic cross-level propagation pathway: variations in response units are reflected in downstream feature representations, while feature variations are further associated with changes in class distributions (see Sec. 3.1 for details). The diagnostic deliberately fixes model parameters to isolate this forward propagation pathway from concurrent parameter changes during optimization. Together with the cross-step optimization analysis, these observations motivate coordinated regulation across the response, feature, and distribution levels rather than treating each representation level independently.

Motivated by these optimization properties and empirical findings, we propose **A**daptive **K**nowledge **I**nheritance (AKI), a framework for knowledge reorganization at the response, feature, and distribution levels. At the response level, *Response Capacity Adaptation (RCA)* constructs a compact response support and enforces prediction consistency between the complete-response and compact-support branches of the current model, reducing excessive dependence on the complete response space. At the feature level, *Reusable Projection Alignment (RPA)* identifies reusable semantic directions by considering historical discriminability and observed current adaptation, and preserves historical semantics only along these directions rather than constraining the complete feature space. At the distribution level, *Distribution Entanglement Penalization (DEP)* models current old- and new-class distributions and directly penalizes their pairwise overlap. Together, RCA, RPA, and DEP instantiate

knowledge reorganization at three coupled representation levels under the guidance of cross-step optimization analysis. The main contributions are summarized as follows:

- We establish an optimization-guided view of replay-free CISS under fixed model capacity, showing that objective shifts generally induce different stationary states, while inherited constraints can restrict current adaptation. We further empirically reveal a cross-level propagation pathway among response, feature, and distribution variations through controlled perturbation analysis.
- Based on these findings, we propose *Adaptive Knowledge Inheritance (AKI)*. RCA regulates dependence on the complete response space through compact-support consistency, RPA preserves reusable semantic directions, and DEP suppresses class distribution overlap in the current representation space.
- Experiments across multiple protocols validate the effectiveness of AKI and the complementary roles of its components through controlled analyses.

2 Related Work

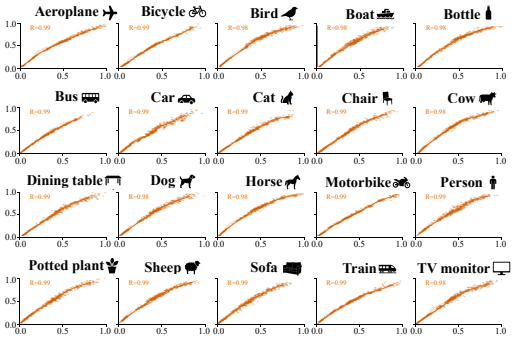
Class-incremental semantic segmentation (CISS) learns new classes while retaining prior segmentation knowledge with limited access to historical data. Existing methods mainly rely on replay, architectural adaptation, and regularization.

Replay-based Methods. Replay methods preserve historical knowledge via stored exemplars, generated samples, or feature statistics Cha et al. (2021); Maracani et al. (2021); Chen et al. (2025); Yu et al. (2024); Oh et al. (2022). RECALL Maracani et al. (2021) and TiKP Yu et al. (2024) recover historical content through generation or retrieval, while Adapter Zhu et al. (2025b) and CoGaMiD Zhu et al. (2025a) adapt stored prototypes or distributions. These approaches make historical information explicitly available during incremental training, but introduce additional requirements for storing, generating, or managing the content used for replay.

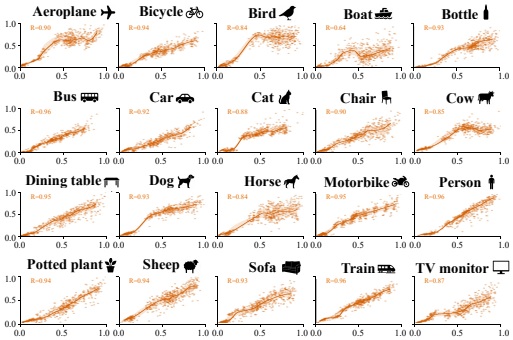
Dynamic Architecture Methods. Architectural adaptation provides another strategy for continual learning Yoon et al. (2017); Qin et al. (2021); Aljundi et al. (2017); Yan et al. (2021). BNS Qin et al. (2021) dynamically constructs architectures for successive tasks, while Expert Gate Aljundi et al. (2017) uses task-specific experts. Such methods accommodate new knowledge through additional capacity, but maintaining an expanding architecture can increase model storage and deployment requirements as incremental learning proceeds.

Regularization Methods. Regularization-based CISS preserves historical knowledge through constraints on model components Michieli & Zanuttigh (2019); Zhao et al. (2023); Yang et al. (2023). Strategies can be grouped by their primary targets. Prediction-level methods regulate output-space supervision. MiB Cermelli et al. (2020) reformulates classification to alleviate background semantic shift, while MBS Park et al. (2024) selectively transfers historical predictions and separates background from new classes. Feature-level methods constrain representations. PLOP Douillard et al. (2021) preserves spatial feature relations, LSKD Wang et al. (2024) adapts distillation across feature levels, and SPPA Lin et al. (2022) aligns historical representations in a learned projected space. Prototype-level methods organize class representations through semantic anchors. CoinSeg Zhang et al. (2023) promotes inter- and intra-class discrimination, while FairCL Truong et al. (2023) employs prototype-based organization. Distribution-level methods characterize class knowledge beyond feature centers. CoGaMiD Zhu et al. (2025a) models historical classes with evolving Gaussian mixtures for pseudo-feature replay. At the parameter level, AWT Goswami et al. (2023) transfers attribution-relevant classifier weights for new-class initialization.

Our Perspective. Our method is replay-free and does not require dynamic architecture expansion. In contrast to regularization methods that primarily preserve historical knowledge, AKI emphasizes coordinated knowledge reorganization for the current incremental objective. This design is motivated by cross-step optimization analysis and controlled cross-level empirical analysis. The former characterizes the optimization requirements arising from objective shifts and inheritance conflicts, while the latter reveals systematic associations across response, feature, and distribution representations. Together, they motivate coordinated knowledge reorganization across three representation levels.



(a) Variations in response units (x-axis) exhibit an overall positive correlation with variations in feature maps (y-axis).



(b) Variations in feature maps (x-axis) exhibit an overall positive correlation with changes in class distributions (y-axis).

Figure 2: **Empirical analysis of cross-level representation propagation under controlled perturbations.** R denotes the class-wise Pearson correlation coefficient Pearson (1897). (a) Response-unit variations are associated with downstream feature variations across classes. (b) Feature variations are associated with class distribution changes under controlled response perturbations. These results provide empirical evidence of cross-level representation propagation and motivate coordinated knowledge reorganization across the response, feature, and distribution levels.

3 METHOD

3.1 CROSS-STEP OPTIMIZATION AND EMPIRICAL ANALYSES UNDER FIXED ARCHITECTURE

We consider replay-free CISS at step t, where frozen f_{t-1} guides f_t under the fixed-capacity setting defined in Appendix A. Here, $\boldsymbol{\theta}$, $\mathcal{L}_t$, and $\boldsymbol{\theta}_t^\star$ denote model parameters, the current objective, and a corresponding stationary state, respectively.

Property 1. A cross-step objective shift requires a different stationary state when it induces a nonzero aggregate gradient at the previous stationary state. At step $t-1$, $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{t-1}(\boldsymbol{\theta}_{t-1}^*) = \mathbf{0}$. Introducing new classes changes supervision; for a new class c, $\partial \ell_t / \partial s_c = p_c - 1 \neq 0$ for finite logits, where s_c and p_c denote its logit and probability. If the objective shift induces a nonzero aggregate gradient,

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_t(\boldsymbol{\theta}_{t-1}^*) \neq \mathbf{0}, \qquad \nabla_{\boldsymbol{\theta}}\mathcal{L}_t(\boldsymbol{\theta}_t^*) = \mathbf{0} \Rightarrow \boldsymbol{\theta}_{t-1}^* \neq \boldsymbol{\theta}_t^*. \tag{1}$$

Property 2. Inherited constraints can restrict current-task-aligned adaptation. We write

$$\mathcal{L}_t^{\text{train}} = \mathcal{L}_t^{\text{cur}} + \Omega_t, \qquad \mathbf{g}_t^{\text{cur}} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\text{cur}}, \quad \mathbf{g}_t^{\text{inh}} = \nabla_{\boldsymbol{\theta}}\Omega_t, \tag{2}$$

where Ω_t denotes the inheritance constraint. With $\Delta\boldsymbol{\theta}_t = -\eta(\mathbf{g}_t^{\text{cur}} + \mathbf{g}_t^{\text{inh}})$,

$$(-\mathbf{g}_t^{\text{cur}})^\top \Delta\boldsymbol{\theta}_t = \eta\|\mathbf{g}_t^{\text{cur}}\|_2^2 + \eta(\mathbf{g}_t^{\text{cur}})^\top \mathbf{g}_t^{\text{inh}}. \tag{3}$$

Thus, when $(\mathbf{g}_t^{\text{cur}})^\top \mathbf{g}_t^{\text{inh}} < 0$, inheritance reduces current-task-aligned updates. Detailed derivations are provided in Appendix A.1 and Appendix A.2.

Key empirical observation. Fig. 2 provides a controlled diagnostic of cross-level representation propagation. Across the tested perturbation levels, response-unit perturbations are reflected in downstream feature variations, which are further associated with class-distribution changes (see Appendix B.1). Fixing the parameters isolates how response variations propagate to downstream features and distributions without interference from simultaneous parameter updates. Together, the above two optimization properties and this key empirical observation motivate coordinated knowledge reorganization across the response, feature, and distribution levels.

3.2 OVERVIEW OF AKI

Motivated by the above analyses, we propose Adaptive Knowledge Inheritance (AKI), a framework guided by cross-step optimization properties and a cross-level empirical observation for response-, feature-, and distribution-level knowledge reorganization, as illustrated in Fig. 3.

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \underbrace{\mathcal{L}_{\text{RCA}} + \mathcal{L}_{\text{RPA}} + \mathcal{L}_{\text{DEP}}}_{\mathcal{L}_{\text{AKI}}}. \tag{4}$$

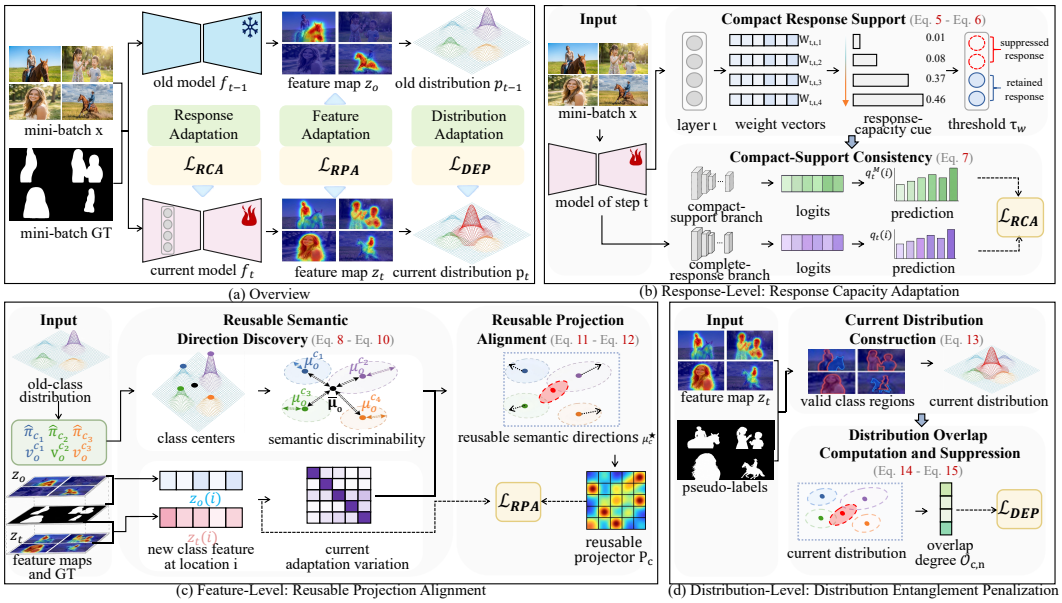


Figure 3: **Overview of AKI.** (a) Three-level knowledge reorganization. (b) RCA constructs compact response support and enforces response consistency. (c) RPA discovers reusable semantic directions for selective alignment. (d) DEP suppresses current old–new distribution overlap.

Here, $\mathcal{L}_{\text{CE}}$ denotes cross-entropy Mao et al. (2023). Following standard CISS practice Cermelli et al. (2020), annotations supervise new classes, while frozen f_{t-1} provides old-class pseudo labels to mitigate background shift. All losses use fixed coefficients of 1 and are averaged over valid elements, avoiding additional balancing hyperparameters and reducing sensitivity to sample counts.

3.3 Response-Level: Response Capacity Adaptation

Property 1 characterizes the stationary-state shift induced by changes in the incremental objective, which motivates retaining sufficient response flexibility for current adaptation. Guided by this requirement, RCA provides a realization by reducing excessive dependence on the complete response space through compact-support consistency.

Compact response support. At incremental step t, consider the l-th trainable linear projection $\mathbf{W}_{t,l} \in \mathbb{R}^{d_l^{\text{out}} \times d_l^{\text{in}}}$, where d_l^{in} and d_l^{out} denote its input and output dimensions, respectively. RCA is applied to the trainable linear projections in the attention and MLP blocks. For output coordinate i, $\mathbf{w}_{t,l,i}^\top$ denotes the i-th row of $\mathbf{W}_{t,l}$, and $\mathbf{z}_{t,l-1} \in \mathbb{R}^{d_l^{\text{in}}}$ denotes the corresponding input feature. The magnitude of its weight-induced response satisfies

$$|w_{t,l,i}^\top z_{t,l-1}| \le \|w_{t,l,i}\|_2 \|z_{t,l-1}\|_2. \tag{5}$$

Under bounded features, $\|\mathbf{w}_{t,l,i}\|_2$ provides an upper-bound cue for potential response magnitude rather than a complete importance measure. We define

$$M_{t,l,i} = \mathbf{1}[\|w_{t,l,i}\|_2 \ge \tau_w], \tag{6}$$

where $M_{t,l,i} = 1$ retains unit i. The mask suppresses selected responses without altering parameters or residual paths, and is recomputed with gradients detached.

Compact-support consistency. We use $\mathbf{q}_t(i)$ and $\mathbf{q}_t^M(i)$ for predictions from the complete-response and compact-support branches, respectively. Both branches share the current model parameters and process the same input, with the response mask applied only in the compact-support branch. RCA minimizes

$$\mathcal{L}_{\text{RCA}} = \frac{1}{|\Omega|} \sum_{i \in \Omega} D_{\text{JS}}(\mathbf{q}_t(i) \| \mathbf{q}_t^M(i)), \tag{7}$$

where Ω contains all output locations. RCA regularizes the model's dependence on the complete response space without removing or freezing units; its preservation and adaptation effects are evaluated experimentally.

3.4 FEATURE-LEVEL: REUSABLE PROJECTION ALIGNMENT

Property 2 motivates selective rather than indiscriminate feature inheritance when historical constraints conflict with current adaptation. Guided by this principle, RPA provides one viable realization by identifying reusable directions that retain historical discriminability while exhibiting limited observed current adaptation.

Reusable direction discovery. For old classes $\mathcal{C}_{\text{old}} = \mathcal{C}_{1:t-1}$, statistics are accumulated across mini-batches from current-step pseudo-labeled regions predicted by frozen f_{t-1}; unobserved classes impose no RPA constraint. Here, $\mathcal{S}_c$ denotes the valid old-class locations contributing to the step-level accumulated statistics, $N_c = |\mathcal{S}_c|$, and $\mathbf{z}_o(i) \in \mathbb{R}^D$ denotes the corresponding previous-model feature. Since f_{t-1} is fixed at step t, all statistics share one parameter state. We compute

$$\hat{\pi}_c = \frac{N_c}{\sum_{k:N_k>0} N_k}, \quad \boldsymbol{\mu}_o^c = \frac{1}{N_c}\sum_{i\in\mathcal{S}_c} \mathbf{z}_o(i), \quad \mathbf{v}_o^c = \frac{1}{N_c}\sum_{i\in\mathcal{S}_c} (\mathbf{z}_o(i) - \boldsymbol{\mu}_o^c)^{\odot 2}, \tag{8}$$

representing class frequency, center, and feature-wise variance. The global historical center $\bar{\boldsymbol{\mu}}_o$, class contrast $\boldsymbol{\delta}_c$, and feature-wise historical dispersion $\boldsymbol{\Sigma}_{\text{hist}}$ are computed as

$$\bar{\boldsymbol{\mu}}_o = \sum_c \hat{\pi}_c \boldsymbol{\mu}_o^c, \qquad \boldsymbol{\delta}_c = \boldsymbol{\mu}_o^c - \bar{\boldsymbol{\mu}}_o, \qquad \boldsymbol{\Sigma}_{\text{hist}} = \sum_c \hat{\pi}_c \,\text{Diag}(\mathbf{v}_o^c). \tag{9}$$

Unlike historical statistics, $\boldsymbol{\Sigma}_{\text{ada}}$ depends on the current model and is recomputed per mini-batch. For new-class locations Ω_t^{new} with features $\mathbf{z}_t(i)$,

$$\Delta\mathbf{z}(i) = \mathbf{z}_t(i) - \mathbf{z}_o(i), \qquad \boldsymbol{\Sigma}_{\text{ada}} = \frac{1}{|\Omega_t^{\text{new}}|}\sum_{i\in\Omega_t^{\text{new}}} \Delta\mathbf{z}(i)\Delta\mathbf{z}(i)^\top. \tag{10}$$

when $|\Omega_t^{\text{new}}| > 0$; otherwise, $\boldsymbol{\Sigma}_{\text{ada}} = 0$. Since $\boldsymbol{\Sigma}_{\text{ada}}$ is a transient mini-batch statistic, we retain its matrix form to capture correlations across feature dimensions. Combining both statistics gives $\mathbf{A} = \boldsymbol{\Sigma}_{\text{hist}} + \boldsymbol{\Sigma}_{\text{ada}} \succeq 0$. To avoid unstable directions associated with zero or numerically negligible eigenvalues, we restrict the search to the numerically effective eigenspace $\mathcal{E}_r$ of $\mathbf{A}$ and use its corresponding truncated pseudoinverse $\mathbf{A}_r^\dagger$. For $\mathbf{u} \in \mathcal{E}_r \setminus \{\mathbf{0}\}$, we define

$$\mathcal{R}_c(\mathbf{u}) = \frac{(\mathbf{u}^\top \boldsymbol{\delta}_c)^2}{\mathbf{u}^\top \mathbf{A}\mathbf{u}}, \qquad \mathbf{u}_c^\star = \frac{\mathbf{A}_r^\dagger \boldsymbol{\delta}_c}{\|\mathbf{A}_r^\dagger \boldsymbol{\delta}_c\|_2}. \tag{11}$$

For classes satisfying $\mathbf{A}_r^\dagger \boldsymbol{\delta}_c \neq \mathbf{0}$, the resulting direction is used for RPA. The derivation and numerical details are provided in Appendix A.3.

Reusable projection alignment. For the current mini-batch, $\mathcal{S}_c$ denotes the valid locations of old class c, with $N_c = |\mathcal{S}_c|$. RPA is applied only to old classes that appear in the current mini-batch and have a valid reusable direction; these classes form $\mathcal{C}_{\text{valid}}$. For each $c \in \mathcal{C}_{\text{valid}}$, we construct the rank-one projector $\mathbf{P}_c = \text{sg}[\mathbf{u}_c^\star (\mathbf{u}_c^\star)^\top]$ onto the reusable direction $\mathbf{u}_c^\star$, where $\text{sg}[\cdot]$ stops gradients through the projector. RPA then minimizes the feature discrepancy only along the reusable direction:

$$\mathcal{L}_{\text{RPA}} = \frac{1}{\sum_{c\in\mathcal{C}_{\text{valid}}} N_c} \sum_{c\in\mathcal{C}_{\text{valid}}} \sum_{i\in\mathcal{S}_c} \|\mathbf{P}_c\,(\mathbf{z}_t(i) - \mathbf{z}_o(i))\|_2^2\,. \tag{12}$$

3.5 DISTRIBUTION-LEVEL: DISTRIBUTION ENTANGLEMENT PENALIZATION

Response- and feature-level reorganization do not explicitly remove old–new distribution overlap. DEP therefore regulates current class distributions. For class $k \in \mathcal{C}_{\text{old}} \cup \mathcal{C}_t^{\text{new}}$ with valid locations Ω_t^k in the current mini-batch, we estimate

$$\boldsymbol{\mu}_t^k = \frac{1}{|\Omega_t^k|}\sum_{i\in\Omega_t^k} \mathbf{z}_t(i), \qquad \mathbf{v}_t^k = \frac{1}{|\Omega_t^k|}\sum_{i\in\Omega_t^k} (\mathbf{z}_t(i) - \boldsymbol{\mu}_t^k)^{\odot 2}, \tag{13}$$

and model $p_t^k(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_t^k, \text{Diag}(\mathbf{v}_t^k))$. For old class c and new class n, overlap is measured by

$$\mathcal{O}_{c,n} = \text{BC}(p_t^c, p_t^n) = \int \sqrt{p_t^c(\mathbf{z}) p_t^n(\mathbf{z})}\, d\mathbf{z}, \tag{14}$$

where larger $\mathcal{O}_{c,n}$ indicates stronger overlap. For the set $\mathcal{P}_t$ of valid old–new pairs,

$$\mathcal{L}_{\text{DEP}} = \frac{1}{|\mathcal{P}_t|}\sum_{(c,n)\in\mathcal{P}_t} \mathcal{O}_{c,n}. \tag{15}$$

DEP directly suppresses old–new distribution overlap in the decoder feature space.

Table 1: **Quantitative comparison with previous methods on Pascal VOC 2012.** Methods are grouped by backbone family. **Bold** and underlined denote the best and second-best results, respectively, and † denotes our re-implementation results.

Method	15-5			19-1			15-1			2-2			10-1		
	0-15	16-20	All	0-19	20	All	0-15	16-20	All	0-2	3-20	All	0-10	11-20	All
CNN-based Methods															
MIB† Cermelli et al. (2020)	76.4	50.0	70.1	71.4	23.6	69.1	34.2	13.5	29.3	41.1	23.4	25.9	12.3	13.1	12.7
PLOP† Douillard et al. (2021)	75.7	51.7	70.0	75.4	37.4	73.6	65.1	21.1	54.6	24.1	11.9	13.6	44.0	15.5	30.4
SSUL-M Cha et al. (2021)	78.4	55.8	73.0	77.8	49.8	76.5	78.4	49.0	71.4	–	–	–	74.0	53.2	64.1
SPPA Lin et al. (2022)	78.1	52.9	72.1	76.5	36.2	74.6	66.2	23.3	56.0	–	–	–	–	–	–
MicroSeg-M Zhang et al. (2022)	82.0	59.2	76.6	79.3	62.9	78.5	81.3	52.5	74.4	60.0	50.9	52.2	77.2	57.2	67.7
AWT Goswami et al. (2023)	77.3	52.9	71.5	–	–	–	59.1	17.2	49.1	–	–	–	33.2	18.0	26.0
LGKD Yang et al. (2023)	79.5	56.1	73.9	77.3	42.9	75.7	70.6	30.9	61.1	–	–	–	–	–	–
CoMasTRe Gong et al. (2024)	79.7	51.9	73.1	75.1	69.5	74.8	69.8	43.6	63.6	–	–	–	–	–	–
Adapter Zhu et al. (2025b)	–	–	–	–	–	–	79.9	51.9	73.2	62.8	57.9	58.6	74.9	54.3	65.1
ReBac Gao et al. (2026)	79.0	58.0	74.0	78.3	47.7	76.8	78.5	45.0	70.5	–	–	–	73.3	50.5	62.4
BDMD Hao et al. (2026)	79.7	58.7	74.7	78.5	45.8	76.9	78.9	48.7	71.7	–	–	–	73.2	51.6	62.9
Transformer-based Methods															
MIB† Cermelli et al. (2020)	78.5	63.2	74.9	80.4	47.8	78.8	72.6	23.5	60.9	41.1	29.3	31.0	11.4	18.9	15.0
SSUL† Cha et al. (2021)	79.7	55.3	73.9	80.8	31.5	78.5	78.1	33.4	67.5	60.3	40.6	43.4	74.3	51.0	63.2
MicroSeg† Zhang et al. (2022)	81.9	54.0	75.3	79.0	25.3	76.4	80.5	40.8	71.0	64.8	43.4	46.5	73.5	53.0	63.7
FairCL Truong et al. (2023)	–	–	–	–	–	–	73.5	22.8	61.4	–	–	–	57.1	14.2	36.7
CoinSeg Zhang et al. (2023)	82.1	63.2	77.6	81.5	44.8	79.8	82.7	52.5	75.5	70.1	63.3	64.3	80.1	60.0	70.5
MBS† Park et al. (2024)	83.9	72.6	81.2	82.2	72.6	81.7	81.9	65.6	78.0	67.5	73.4	72.6	80.0	**72.9**	76.6
Nest Xie et al. (2024)	81.2	67.4	77.9	79.7	60.0	78.8	77.0	53.3	71.4	–	–	–	65.2	35.8	51.2
IPSeg Yu et al. (2025)	81.4	62.4	76.9	–	–	–	82.4	52.9	75.4	72.1	64.5	65.6	80.0	61.2	71.0
CoGaMiD Zhu et al. (2025a)	–	–	–	–	–	–	83.2	61.2	78.0	73.4	70.0	70.5	81.1	65.9	73.9
LCSKD Wang et al. (2026)	81.2	69.6	78.4	80.2	63.8	79.4	73.6	52.1	68.5	–	–	–	–	–	–
MRKD Zhou et al. (2026)	–	–	–	**83.1**	66.9	**82.3**	82.9	60.0	77.4	–	–	–	72.8	54.3	64.0
AKI (Ours)	**84.8**	**76.3**	**82.8**	82.7	**74.1**	**82.3**	**83.4**	**66.2**	**79.3**	**74.0**	**75.0**	**74.9**	**81.8**	72.6	**77.4**
Joint (Upper bound)	85.5	80.3	84.3	84.4	79.6	84.2	83.9	79.1	82.8	77.3	85.5	84.3	85.0	84.7	84.9

Table 2: **Quantitative comparison with previous methods on ADE20K.** Methods are grouped by backbone family. AKI achieves competitive overall mIoU across diverse settings while maintaining a favorable balance between stability and plasticity.

Method	100–50			50–50			100–10			100–5		
	0-100	101-150	All	0-50	51-150	All	0-100	101-150	All	0-100	101-150	All
CNN-based Methods												
MIB† Cermelli et al. (2020)	40.5	17.2	32.8	45.6	21.0	29.3	38.2	11.1	29.2	36.0	5.7	26.0
PLOP† Douillard et al. (2021)	41.9	14.9	33.0	48.8	21.0	30.4	40.5	13.6	31.6	39.1	7.8	28.7
SSUL-M Cha et al. (2021)	42.8	17.5	34.4	49.1	20.1	29.9	42.9	17.6	34.5	42.9	17.8	34.6
SPPA Lin et al. (2022)	42.9	19.9	35.3	49.8	23.9	32.6	41.0	12.5	31.6	–	–	–
MicroSeg-M Zhang et al. (2022)	43.4	20.9	35.9	49.8	22.0	31.4	43.7	22.2	36.6	43.6	22.4	36.6
AWT† Goswami et al. (2023)	40.9	24.7	35.5	46.6	26.9	33.6	39.1	21.3	33.2	38.6	16.0	31.1
LGKD Yang et al. (2023)	43.4	25.7	37.5	48.9	29.4	36.0	41.9	22.0	35.3	–	–	–
CoMasTRe Gong et al. (2024)	45.7	26.0	39.2	–	–	–	42.3	18.4	34.4	40.8	15.8	32.5
Adapter Zhu et al. (2025b)	43.1	23.6	36.6	49.3	27.3	34.7	42.9	19.9	35.3	42.6	18.0	34.5
ReBac Gao et al. (2026)	42.2	23.6	36.0	48.3	27.0	34.2	41.5	25.4	36.2	41.1	24.0	35.4
BDMD Hao et al. (2026)	43.5	26.5	37.9	49.6	29.2	36.1	42.1	23.8	36.0	–	–	–
Transformer-based Methods												
MIB† Cermelli et al. (2020)	46.4	35.0	42.6	52.2	35.6	41.2	43.0	30.8	39.0	40.2	26.6	35.7
SSUL† Cha et al. (2021)	41.9	20.1	34.7	49.5	21.3	30.8	40.7	19.0	33.5	41.3	16.0	32.9
MicroSeg† Zhang et al. (2022)	41.1	24.1	35.5	49.8	23.9	32.6	41.0	22.6	34.9	41.2	21.0	34.5
FairCL Truong et al. (2023)	43.6	25.5	37.6	49.6	27.8	35.2	42.2	21.9	35.5	–	–	–
CoinSeg Zhang et al. (2023)	41.6	26.7	36.7	49.0	28.9	35.7	42.1	24.5	36.3	43.1	24.1	36.8
MBS† Park et al. (2024)	**49.4**	37.5	45.5	55.6	39.8	45.1	48.1	35.2	43.8	45.7	22.7	38.1
Nest Xie et al. (2024)	42.8	27.8	37.8	49.7	29.3	36.2	41.8	23.8	35.8	40.5	19.9	33.7
IPSeg Yu et al. (2025)	43.2	29.0	38.5	49.3	33.0	38.5	42.5	27.8	37.6	43.1	26.2	37.5
CoGaMiD Zhu et al. (2025a)	43.9	27.3	38.4	49.9	29.8	36.6	43.7	26.5	38.0	43.6	25.8	37.7
LCSKD Wang et al. (2026)	42.9	28.1	38.0	48.4	29.8	36.1	41.1	26.0	36.1	40.6	24.3	35.2
MRKD Zhou et al. (2026)	48.0	34.2	43.4	–	–	–	45.8	28.9	40.2	44.4	25.9	38.3
AKI (Ours)	49.3	**39.9**	**46.2**	**56.6**	**40.5**	**45.9**	**48.7**	**37.3**	**44.9**	**47.2**	**30.0**	**41.5**
Joint (Upper bound)	52.9	42.6	49.5	58.9	44.7	49.5	52.7	42.4	49.3	52.6	42.6	49.3

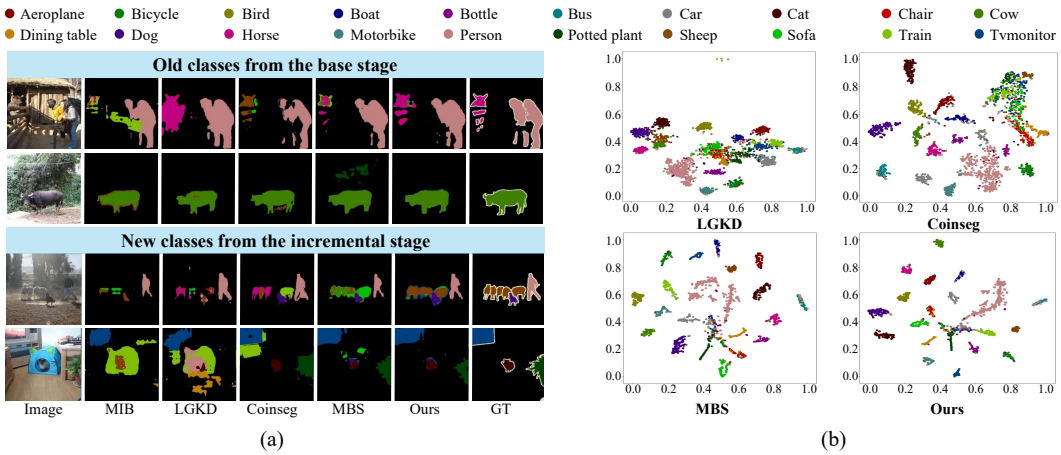


Figure 4: **Qualitative comparison under the Pascal VOC 15-1 setting.** (a) Segmentation predictions. (b) t-SNE visualization. AKI better preserves previously learned classes while improving the recognition of newly introduced classes. The learned features also exhibit more compact intra-class structures and clearer inter-class separation, indicating reduced class-distribution overlap.

4 Experiments

4.1 Experimental Setups

Datasets and protocols. Following standard CISS practice Cermelli et al. (2020); Yang et al. (2023); Park et al. (2024), we evaluate AKI on Pascal VOC 2012 Everingham et al. (2010) and ADE20K Zhou et al. (2017), two widely used benchmarks. Unless otherwise stated, experiments follow the widely adopted overlap setting.

Implementation details. We use ImageNet-1K-pretrained ViT-B/16 Dosovitskiy et al. (2021); Deng et al. (2009) as the backbone, with two transformer decoder blocks and 512×512 inputs. Models are trained for 64 epochs per step using SGD Ketkar (2017). The learning rate is 1×10^{-3} at the base step and 1×10^{-4} and 5×10^{-4} at incremental steps on Pascal VOC and ADE20K, respectively. We use a shared response threshold $\tau_w = 0.1$ across trainable linear mappings. Experiments are implemented in PyTorch and conducted on six NVIDIA RTX 3090 GPUs.

4.2 Comparison with Representative CISS Methods

As shown in Tab. 1, AKI achieves competitive performance across all five Pascal VOC settings. To account for backbone differences, methods are grouped into CNN- and Transformer-based architectures, with AKI remaining competitive or superior within the Transformer group. Tab. 2 shows a similar trend on ADE20K. These results demonstrate the stable effectiveness of AKI across

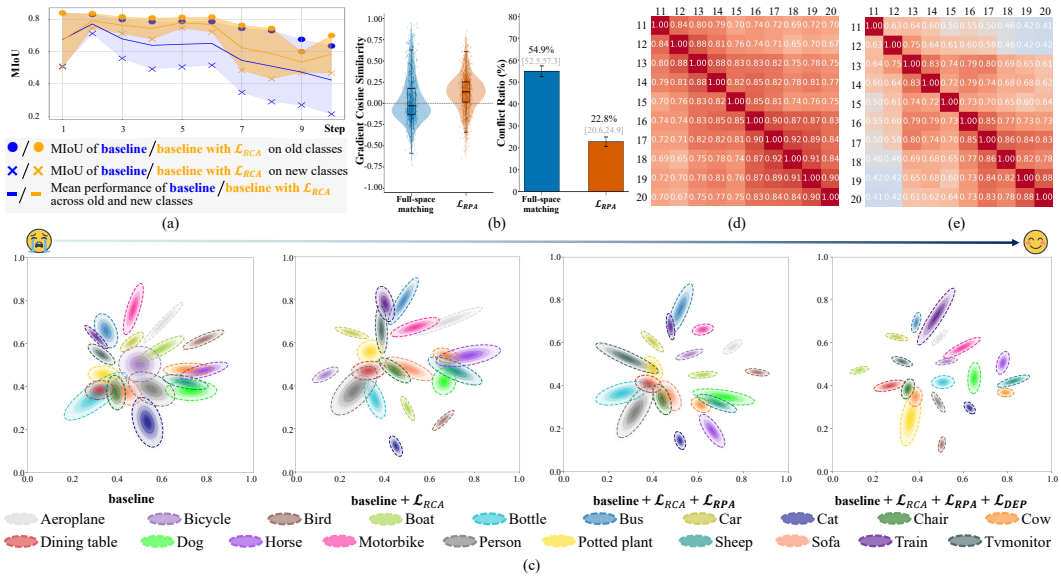


Figure 5: **Ablation analysis of AKI.** (a) RCA improves new-class adaptation with stable old-class performance. (b) RPA improves gradient alignment and reduces the conflict ratio from 54.9% to 22.8% compared with full-space matching. (c) DEP further reduces class distribution overlap after RCA and RPA; 10% peripheral features are removed for visualization. (d) Class-similarity matrix of the baseline model; (e) the corresponding matrix after introducing AKI, which reduces off-diagonal similarities and inter-class entanglement.

diverse incremental scenarios. Fig. 4(a) shows that competing methods tend to confuse the old class "horse" with the newly introduced class "sheep", whereas AKI better preserves old-class predictions while improving new-class segmentation. The t-SNE visualization in Fig. 4(b) also shows more compact intra-class clusters and clearer inter-class separation, supporting the improved class organization achieved by AKI.

4.3 Ablation Studies

Table 3: **Component ablation of AKI on Pascal VOC under 10-1 setting.**

Grp.	$\mathcal{L}_{\text{RCA}}$	$\mathcal{L}_{\text{RPA}}$	$\mathcal{L}_{\text{DEP}}$	10-1		
				0-10	11-20	All
1	×	×	×	63.5	21.3	43.4
2	✓			69.9	46.7	58.9
3		✓		72.7	45.2	59.6
4			✓	68.7	35.7	53.0
5	✓	✓		81.1	70.8	76.2
6	✓		✓	81.1	71.2	76.4
7		✓	✓	71.2	49.6	60.9
8	✓	✓	✓	**81.8**	**72.6**	**77.4**

Effectiveness of AKI components. As shown in Tab. 3, Grps. 2–4 consistently outperform the baseline in Grp. 1, demonstrating the individual effectiveness of RCA, RPA, and DEP. In particular, RCA and RPA improve new-class mIoU by 25.4 and 23.9 percentage points, respectively. Compared with the single-component variants, the pairwise combinations in Grps. 5–7 generally yield further improvements, indicating complementary effects across different adaptation levels. Grp. 8 achieves the best new-class and overall mIoU. Therefore, considering overall performance, we adopt the combination of RCA, RPA, and DEP as the final AKI configuration.

Effect of compact response reorganization. Fig. 5(a) shows that RCA improves new-class mIoU while maintaining comparable old-class performance. RCA enforces consistency between complete and compact-support responses; excessive reliance on responses outside the compact support therefore increases discrepancy and is penalized during optimization. This promotes compact response organization, preserving historical knowledge while retaining flexibility for current adaptation.

Effect of Reusable Projection Alignment. Fig. 5(b) examines gradient interaction between current adaptation and historical inheritance. The left panel shows improved gradient cosine similarity with RPA. As cosine similarity is magnitude-normalized, the improvement reflects directional compatibility rather than weaker inheritance gradients. The right panel reports the conflict ratio, the proportion of negative cosine-similarity cases, decreasing from 54.9% under full-space matching to 22.8% with RPA. This supports Property 2: full-space constraints can involve directions required by current learning, whereas RPA restricts alignment to reusable semantic directions with limited adaptation demand, leaving complementary feature directions available for adaptation to new classes.

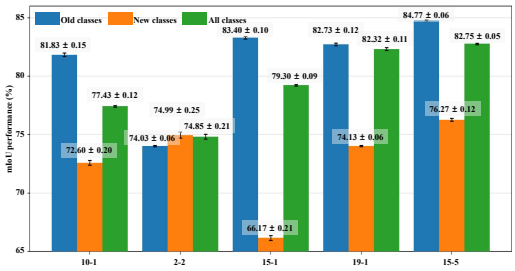


Figure 6: **Performance variability of AKI over three runs across five incremental settings.** The consistently small standard deviations indicate stable performance under different settings.

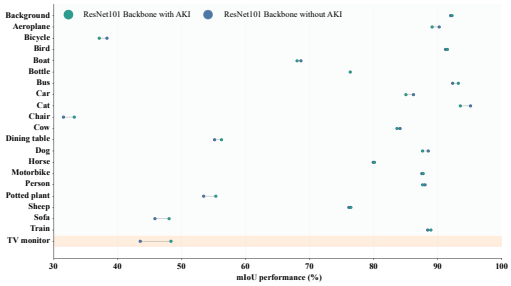


Figure 7: **Class-wise mIoU comparison with and without AKI using a ResNet101 backbone under the 19-1 setting.** This provides a CNN-based compatibility analysis of AKI.

Effect of distribution-overlap penalization. Fig. 5(c) shows organization of class distributions. Although RCA and RPA alleviate class entanglement, overlap remains. DEP produces compact distributions and clearer inter-class boundaries by directly penalizing pairwise old–new overlap. Unlike center-only separation, DEP jointly considers distribution means and feature-wise dispersion, capturing both class location and distributional spread. Diagnostic analysis of class-distribution overlap in a two-dimensional PCA space is provided in Appendix C.8. The reduced overlap after introducing DEP validates the benefit of distribution-level regulation.

Impact of AKI on class confusion. To further examine class-wise interference, we compute the class-token similarity matrix for the 10 incremental classes under the 10-1 setting. As shown in Fig. 5(d) and (e), the baseline exhibits several strong off-diagonal responses, indicating that representations of different classes remain highly correlated and are therefore more prone to confusion. After introducing AKI, these off-diagonal similarities are consistently reduced, while the class-specific structure becomes more distinct. This suggests that the coordinated response-, feature-, and distribution-level adaptation progressively suppresses cross-class entanglement and improves the separation of incrementally learned classes.

Table 4: **Analysis of the threshold τ_w on 10-1 setting.**

	0.05	**0.10**	**0.15**	**0.20**	**0.30**	**0.40**
0-10	83.1	81.8	81.3	81.6	81.1	76.9
11-20	70.5	72.6	72.9	72.1	64.5	49.4
All	77.1	77.4	77.3	77.1	73.2	63.8

Effect of τ_w. The threshold τ_w controls the compactness of the response support. As shown in Tab. 4, the overall mIoU remains stable for $\tau_w \in [0.05, 0.20]$, ranging from 77.1% to 77.4%. This indicates robustness within the tested moderate range. We therefore set $\tau_w = 0.1$, which achieves the best overall performance within the stable range. Further analysis is provided in Appendix C.9.

Experimental variability. We further evaluate the stability of AKI through three repeated runs across five incremental settings. As shown in Fig. 6, the standard deviation of the overall mIoU remains low across different protocols, ranging from 0.05 to 0.21, indicating low experimental variability and stable performance across repeated runs under the evaluated protocols.

Compatibility with a CNN backbone. To examine AKI beyond the Transformer-based setting, we integrate it into a CISS method Zhang et al. (2023) with a ResNet101 backbone under the 19-1 protocol. As illustrated in Fig. 7, AKI improves the newly introduced TV monitor class from 43.5% to 48.3% mIoU, while maintaining comparable performance across previously learned classes. These results indicate that AKI can be incorporated into a ResNet101-based CISS configuration and provide further performance gains over the corresponding baseline.

5 Conclusion

This work presents Adaptive Knowledge Inheritance (AKI) for replay-free class-incremental semantic segmentation under fixed model capacity. Our cross-step optimization analysis highlights the need to reorganize historical knowledge as incremental objectives evolve, while empirical evidence further reveals coupling across response, feature, and distribution representations. Guided by these findings, AKI coordinates response-capacity adaptation, reusable semantic inheritance, and class distribution disentanglement through RCA, RPA, and DEP, respectively. Comprehensive experiments on Pascal VOC 2012 and ADE20K validate the effectiveness and complementarity of these designs in balancing historical preservation and adaptation across diverse incremental settings.

AI USE STATEMENT

Generative AI tools were used solely for language polishing. All AI-assisted content was carefully reviewed by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

We adhere to the ICLR Code of Ethics throughout data use, experimentation, and manuscript preparation. The main ethical considerations associated with this work are summarized below.

Data and licensing. Our experiments are conducted on the publicly available Pascal VOC 2012 and ADE20K benchmarks following their established evaluation protocols. We do not collect new user data or construct new datasets, and the benchmark data are used in accordance with their original licenses.

Human subjects and privacy. This work does not involve human-subject experiments or direct interaction with individuals. No additional personally identifiable information is collected, annotated, or inferred beyond the content already contained in the public benchmark datasets.

Potential harms and deployment. The proposed method addresses class-incremental semantic segmentation and supports continuously updated perception systems. As with other semantic segmentation methods, prediction errors or accumulated errors during incremental learning may affect downstream decisions, particularly in safety-critical applications. Practical deployment should therefore include appropriate validation, monitoring, and application-specific risk assessment.

Bias and fairness. Our method is evaluated on standard semantic segmentation benchmarks and does not explicitly address demographic or societal bias. Any biases present in the underlying datasets may therefore also be reflected in the learned models. The reported results should be interpreted within the scope of the evaluated datasets and protocols.

REPRODUCIBILITY STATEMENT

We provide detailed experimental settings, optimization configurations, theoretical derivations, and additional diagnostic analyses to facilitate reproduction of the reported results.

Datasets and evaluation protocols. The datasets, incremental protocols, evaluation metrics, and overlapping and disjoint settings used in our experiments are described in the main paper and Appendix B. We evaluate on Pascal VOC 2012 and ADE20K using the standard CISS protocols adopted in prior work.

Implementation details. The backbone architecture, input resolution, optimizer, learning rates, training schedule, and response-capacity threshold are reported in Sec. 4.1 and Appendix B. Specifically, the main experiments use an ImageNet-pretrained ViT-B/16 backbone with 512×512 inputs and SGD optimization. The default response threshold is fixed to $\tau_w = 0.1$ across experiments.

Method specification and derivations. The complete formulations of RCA, RPA, and DEP are provided in Secs. 3.3–3.5. Appendix A gives the detailed derivations of the two cross-step optimization properties, the reusable semantic direction used by RPA, the construction of current-step statistics, and the numerically stable computation of DEP.

Ablation and diagnostic analyses. Appendix C provides additional experiments on component effectiveness, response-support construction, threshold sensitivity, directional selection in RPA, gradient interaction, distribution overlap, backbone compatibility, and computational efficiency. These analyses specify the corresponding controlled settings and evaluation procedures used to examine the individual mechanisms of AKI.

Experimental variability. We additionally report repeated experiments across multiple incremental settings in Fig. 6. The consistently small standard deviations of the reported mIoU results provide evidence that the observed performance is stable across independent runs.

Computational resources. The main experiments are conducted using NVIDIA GeForce RTX 3090 GPUs. For the computational-efficiency analysis, runtime and memory measurements are reported

under the same hardware and software environment for all compared methods, with the corresponding batch size, input resolution, and AMP setting specified in Appendix C.

A video overview of this work is provided in the supplementary material.

REFERENCES

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 3366–3375, 2017.

Donghyeon Baek, Youngmin Oh, Sanghoon Lee, Junghyup Lee, and Bumsub Ham. Decomposed knowledge distillation for class-incremental semantic segmentation. *Neural Information Processing Systems (NIPS)*, 35:10380–10392, 2022.

Kexin Bao, Yong Li, Dan Zeng, and Shiming Ge. Few-shot class-incremental learning via generative co-memory regularization. *International Journal of Computer Vision (IJCV)*, 134(4):164, 2026.

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 9233–9242, 2020.

Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Neural Information Processing Systems (NIPS)*, 34:10919–10930, 2021.

Jinpeng Chen, Runmin Cong, Yuxuan Luo, Horace Ho Shing Ip, and Sam Kwong. Replay without saving: Prototype derivation and distribution rebalance for class-incremental semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2025.

Wei Cong, Yang Cong, Jiahua Dong, Gan Sun, and Henghui Ding. Gradient-semantic compensation for incremental semantic segmentation. *IEEE Transactions on Multimedia (TMM)*, 26:5561–5574, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer vision and pattern recognition (CVPR)*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 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 (IJCV)*, 88: 303–338, 2010.

Guangyu Gao, Anqi Zhang, Jianbo Jiao, Fangkunhan Liu, Chi Harold Liu, and Yunchao Wei. Rebac: Current-to-past residual background correction for class-incremental semantic segmentation. *International Journal of Computer Vision (IJCV)*, 134(5):225, 2026.

Yizheng Gong, Siyue Yu, Xiaoyang Wang, and Jimin Xiao. Continual segmentation with disentangled objectness learning and class recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 3848–3857, 2024.

Dipam Goswami, René Schuster, Joost Van de Weijer, and Didier Stricker. Attribution-aware weight transfer: A warm-start initialization for class-incremental semantic segmentation. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3194–3203. IEEE, 2023.

Xuze Hao, Xuhao Jiang, Wenqian Ni, Weimin Tan, and Bo Yan. Text-guided bidirectional mapping distillation for continual semantic segmentation. *Science China Information Sciences (SCIS)*, 69 (7):172106, 2026.

Nikhil Ketkar. Stochastic gradient descent. In *Deep learning with Python: A hands-on introduction*, pp. 113–132. 2017.

Chenzhong Liao and Lingguo Kong. A comprehensive survey of semantic segmentation based on computer vision. In *Fourth International Conference on Computer Vision, Application, and Algorithm (CVAA 2024)*, volume 13486, pp. 235–250. SPIE, 2025.

Zihan Lin, Zilei Wang, and Yixin Zhang. Continual semantic segmentation via structure preserving and projected feature alignment. In *European Conference on Computer Vision (ECCV)*, pp. 345–361. Springer, 2022.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International Conference on Machine learning (ICML)*, pp. 23803–23828. pmlr, 2023.

Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *International conference on computer vision (ICCV)*, pp. 7026–7035, 2021.

Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *International Conference on Computer Vision Workshops (ICCVW)*, 2019.

Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1114–1124, 2021.

Youngmin Oh, Donghyeon Baek, and Bumsub Ham. Alife: Adaptive logit regularizer and feature replay for incremental semantic segmentation. *Neural Information Processing Systems (NIPS)*, 35: 14516–14528, 2022.

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 (ECCV)*, pp. 71–88. Springer, 2024.

Karl Pearson. Mathematical contributions to the theory of evolution. on the law of ancestral heredity. *Proceedings of the Royal Society of London*, 62:386–412, 1897.

Pengchong Qiao, Yu Wang, Chang Liu, Lei Shang, Baigui Sun, Zhennan Wang, Xiawu Zheng, Rongrong Ji, and Jie Chen. Adaptive fuzzy positive learning for annotation-scarce semantic segmentation. *International Journal of Computer Vision (IJCV)*, 133(3):1048–1066, 2025.

Qi Qin, Wenpeng Hu, Han Peng, Dongyan Zhao, and Bing Liu. Bns: Building network structures dynamically for continual learning. *Neural Information Processing Systems (NIPS)*, 34:20608–20620, 2021.

Chao Shang, Hongliang Li, Fanman Meng, Qingbo Wu, Heqian Qiu, and Lanxiao Wang. Incrementer: Transformer for class-incremental semantic segmentation with knowledge distillation focusing on old class. *Computer Vision and Pattern Recognition (CVPR)*, pp. 7214–7224, 2023.

Thanh-Dat Truong, Hoang-Quan Nguyen, Bhiksha Raj, and Khoa Luu. Fairness continual learning approach to semantic scene understanding in open-world environments. *Neural Information Processing Systems (NIPS)*, 36:65456–65467, 2023.

Bing Wang, Yanchuan Li, Shuo Yang, Taifeng Huang, Zutong Zhao, and Shiyin Zhang. Label-guided class similarity distillation for continual semantic segmentation. *International Journal of Machine Learning and Cybernetics*, 17(6):292, 2026.

Qilong Wang, Yiwen Wu, Liu Yang, Wangmeng Zuo, and Qinghua Hu. Layer-specific knowledge distillation for class incremental semantic segmentation. *IEEE Transactions on Image Processing (TIP)*, 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 (ECCV)*, pp. 183–201. Springer, 2024.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 3014–3023, 2021.

Guanglei Yang, Enrico Fini, Dan Xu, Paolo Rota, Mingli Ding, Moin Nabi, Xavier Alameda-Pineda, and Elisa Ricci. Uncertainty-aware contrastive distillation for incremental semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(2):2567–2581, 2022.

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 *International Conference on Computer Vision (ICCV)*, pp. 18601–18612, 2023.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

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.

Zhidong Yu, Wei Yang, Xike Xie, and Zhenbo Shi. Tikp: Text-to-image knowledge preservation for continual semantic segmentation. In *Association for the Advancement of Artificial Intelligence Conference (AAAI)*, volume 38, pp. 16596–16604, 2024.

Bo Yuan and Danpei Zhao. A survey on continual semantic segmentation: Theory, challenge, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

Zekang Zhang, Guangyu Gao, Zhiyuan Fang, Jianbo Jiao, and Yunchao Wei. Mining unseen classes via regional objectness: A simple baseline for incremental segmentation. *Neural Information Processing Systems (NIPS)*, 35:24340–24353, 2022.

Zekang Zhang, Guangyu Gao, Jianbo Jiao, Chi Harold Liu, and Yunchao Wei. Coinseg: Contrast inter- and intra- class representations for incremental segmentation. In *International Conference on Computer Vision (ICCV)*, pp. 843–853, 2023.

Danpei Zhao, Bo Yuan, and Zhenwei Shi. Inherit with distillation and evolve with contrast: Exploring class incremental semantic segmentation without exemplar memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(10):11932–11947, 2023.

Hanbin Zhao, Fengyu Yang, Xinghe Fu, and Xi Li. Rbc: Rectifying the biased context in continual semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pp. 55–72. Springer, 2022.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 633–641, 2017.

Hui Zhou, Huanjie Tao, Qiuhan Zhang, Chao Xie, and Benran Li. Mrkd-pbcl: Multi-level region-wise knowledge distillation and prototype balanced contrastive learning for class incremental semantic segmentation. *Neurocomputing*, pp. 133285, 2026.

Guilin Zhu, Runmin Wang, Yuanjie Shao, Weidong Yang, Nong Sang, and Changxin Gao. Continual gaussian mixture distribution modeling for class incremental semantic segmentation. *Neural Information Processing Systems (NIPS)*, 2025a.

Guilin Zhu, Dongyue Wu, Changxin Gao, Runmin Wang, Weidong Yang, and Nong Sang. Adaptive prototype replay for class incremental semantic segmentation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pp. 10932–10940, 2025b.

APPENDIX

A THEORETICAL ANALYSIS

This section expands the cross-step optimization analysis introduced in Sec. 3.1. The two properties below are conditional first-order observations rather than universal performance guarantees. Their purpose is to identify two optimization mechanisms relevant to fixed-capacity CISS: the shift of the current stationary condition after new supervision is introduced, and the directional interaction between current adaptation and historical inheritance. Their role is to identify optimization requirements that guide subsequent method design; they are not intended to deductively determine the specific response mask, projection criterion, or distribution regularizer used in AKI. Accordingly, these properties do not claim that RCA, RPA, or DEP are unique or theoretically optimal solutions. The specific designs are further supported by the analytical derivations in the following subsections and the controlled mechanism analyses in the supplementary experiments.

A.1 DERIVATION OF PROPERTY 1

Fixed capacity in this work refers to an unchanged shared network architecture across incremental steps, with its parameters remaining trainable. We distinguish routine label-space extensions, such as adding class-specific classifiers or tokens, from dynamic architectural expansion through additional feature branches, experts, or representation modules. This distinction is consistent with MBS, which introduces new class tokens within a regularization-based CISS framework Park et al. (2024). AKI follows this setting, adapting its parameters and learned representations within the existing shared architecture. For theoretical analysis, we express adjacent objectives in a common parameterization, with each objective depending only on the parameters used at its corresponding step.

Property 1. *Within the same fixed-capacity parameter space, a cross-step objective shift requires a different stationary state whenever it produces a nonzero aggregate gradient at the previous stationary state. At step t, newly introduced classes change the task-driven supervision and consequently alter the optimization objective. If the resulting aggregate objective shift produces a nonzero parameter gradient at the previous stationary state, that state is no longer stationary for the current objective.*

For clarity, quantities associated with the previous incremental objective are shown in blue, while those associated with the current objective are shown in red. In this property, we consider the task-driven objective induced by class supervision; the influence of historical inheritance constraints is analyzed separately in Property 2.

Previous stationary condition. At step $t-1$, let $\mathcal{L}_{t-1}^{\text{task}}$ denote the task-driven objective associated with the classes observed up to that step. A stationary solution $\boldsymbol{\theta}_{t-1}^{*}$ satisfies

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{t-1}^{\text{task}} \left(\boldsymbol{\theta}_{t-1}^{*} \right) = \mathbf{0}. \tag{16}$$

At step t, previously unseen classes introduce new supervision and thereby change the task-driven objective. Consider a training location belonging to a newly introduced class c. Its cross-entropy gradient with respect to the corresponding logit is

$$\frac{\partial \ell_t}{\partial s_c} = p_c - 1. \tag{17}$$

For finite logits under the standard softmax formulation, $0 < p_c < 1$, and hence

$$\boxed{\frac{\partial \ell_t}{\partial s_c} \neq 0}. \tag{18}$$

Thus, a newly introduced class provides a prediction-space optimization signal that is absent from the previous task objective.

Propagation to the parameter space. Let $\mathbf{s}(\boldsymbol{\theta})$ denote the logit vector and $\mathbf{J}_s(\boldsymbol{\theta})$ its Jacobian with respect to the model parameters:

$$\mathbf{J}_s(\boldsymbol{\theta}) = \frac{\partial \mathbf{s}}{\partial \boldsymbol{\theta}}. \tag{19}$$

By the chain rule,

$$\nabla_{\boldsymbol{\theta}} \ell_t = \mathbf{J}_s(\boldsymbol{\theta})^\top \nabla_{\mathbf{s}} \ell_t. \tag{20}$$

Therefore, whenever the prediction-space signal affects at least one trainable parameter direction,

$$\mathbf{J}_s(\boldsymbol{\theta})^\top \nabla_{\mathbf{s}} \ell_t \neq \mathbf{0}, \tag{21}$$

the corresponding new-class sample contributes a nonzero optimization signal in the parameter space.

Importantly, this sample-wise result alone does not imply that the gradient of the complete current objective is nonzero. Gradients contributed by different samples or supervision terms can point in different directions and may partially or, in a degenerate case, exactly cancel after aggregation. We therefore formulate the stationary-state analysis directly at the objective level.

At step t, the introduction of previously unseen classes provides supervision that is absent at step $t-1$. Consequently, the task-driven objective changes from $\mathcal{L}_{t-1}^{\text{task}}$ to $\mathcal{L}_t^{\text{task}}$. The preceding sample-wise analysis explains how newly introduced classes generate additional optimization signals. We next examine whether these changes remain nonzero after aggregation at the objective level.

Aggregate change of the task-driven objective. We define the change between two adjacent task-driven objectives as

$$\Delta\mathcal{L}_t^{\text{task}}(\boldsymbol{\theta}) = \mathcal{L}_t^{\text{task}}(\boldsymbol{\theta}) - \mathcal{L}_{t-1}^{\text{task}}(\boldsymbol{\theta}). \tag{22}$$

This quantity captures the aggregate change in supervision between adjacent incremental steps, including the contribution of newly introduced classes and any associated change in the task-driven training objective.

By definition,

$$\mathcal{L}_t^{\text{task}} = \mathcal{L}_{t-1}^{\text{task}} + \Delta\mathcal{L}_t^{\text{task}}. \tag{23}$$

Evaluating its gradient at the previous stationary state gives

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\text{task}} \left(\boldsymbol{\theta}_{t-1}^*\right) \\ &= \nabla_{\boldsymbol{\theta}} \mathcal{L}_{t-1}^{\text{task}} \left(\boldsymbol{\theta}_{t-1}^*\right) + \nabla_{\boldsymbol{\theta}} \Delta\mathcal{L}_t^{\text{task}} \left(\boldsymbol{\theta}_{t-1}^*\right). \end{aligned} \tag{24}$$

Using the previous stationary condition in Eq. (16), we obtain

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\text{task}} \left(\boldsymbol{\theta}_{t-1}^*\right) = \nabla_{\boldsymbol{\theta}} \Delta\mathcal{L}_t^{\text{task}} \left(\boldsymbol{\theta}_{t-1}^*\right). \tag{25}$$

Therefore, using the relation above, the aggregate non-cancellation condition directly implies that the previous stationary state is no longer stationary for the current task-driven objective:

$$\boxed{\nabla_{\boldsymbol{\theta}} \Delta\mathcal{L}_t^{\text{task}} \left(\boldsymbol{\theta}_{t-1}^*\right) \neq \mathbf{0} \quad \Longrightarrow \quad \nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\text{task}} \left(\boldsymbol{\theta}_{t-1}^*\right) \neq \mathbf{0}}. \tag{26}$$

The sample-wise cross-entropy analysis above explains why newly introduced classes contribute to the objective shift in Eq. (22), while the aggregate non-cancellation condition explicitly accounts for possible cancellation across samples and supervision terms.

Shift of the stationary state. Let $\boldsymbol{\theta}_t^*$ denote a stationary solution of the current task-driven objective. By definition,

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_t^{\text{task}} \left(\boldsymbol{\theta}_t^*\right) = \mathbf{0}. \tag{27}$$

Together with Eq. (26), this gives

$$\boxed{\boldsymbol{\theta}_t^* \neq \boldsymbol{\theta}_{t-1}^*}. \tag{28}$$

Property 1 therefore provides a conditional characterization of a nontrivial cross-step objective shift. The aggregate non-cancellation condition is not introduced as an independent guarantee that every change of supervision must alter the stationary state; rather, it identifies the case in which the changed supervision remains active after aggregation at the previous stationary state. When this condition holds, the previous state cannot remain stationary for the current objective. Under fixed model capacity, this motivates retaining sufficient parameter and representation flexibility to accommodate the resulting cross-step adaptation.

Implication for response-level adaptation. The stationary-state shift characterized above motivates preserving sufficient response flexibility when the incremental objective changes. Under fixed model

capacity, excessive dependence on the complete response space can restrict the reorganization available to the current model.

RCA addresses this requirement by constructing a compact response support and enforcing consistency between the complete and compact-support predictions. Rather than forcing the current model to rely on the complete response space, this training constraint encourages effective responses to be maintained under a compact support, thereby reducing unnecessary complete-space dependence and retaining flexibility for current adaptation.

Property 1 thus provides the optimization-level motivation for RCA: when the task-driven objective changes, the response organization should retain sufficient flexibility to adapt toward the current stationary state.

A.2 Derivation of Property 2

Property 2. *When the gradient induced by historical inheritance conflicts with the gradient of the current adaptation objective, the inheritance constraint reduces the parameter-update component aligned with current adaptation.*

At incremental step t, training needs to accommodate newly introduced supervision while preserving previously acquired knowledge. We decompose the training objective as

$$\mathcal{L}_t^{\text{train}}(\boldsymbol{\theta}) = \underbrace{\mathcal{L}_t^{\text{cur}}(\boldsymbol{\theta})}_{\text{current adaptation}} + \underbrace{\Omega_t(\boldsymbol{\theta};\boldsymbol{\theta}_{t-1})}_{\text{historical inheritance}}, \tag{29}$$

where $\mathcal{L}_t^{\text{cur}}$ represents the objective that drives adaptation to the current incremental supervision, while Ω_t denotes the constraint introduced to preserve historical knowledge. Quantities related to current adaptation are shown in red, whereas those related to historical inheritance are shown in blue.

Their parameter gradients are

$$\mathbf{g}_t^{\text{cur}} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_t^{\text{cur}}, \qquad \mathbf{g}_t^{\text{inh}} = \nabla_{\boldsymbol{\theta}}\Omega_t. \tag{30}$$

Alignment with current adaptation. Without historical inheritance, the update driven by the current adaptation objective is

$$\Delta\boldsymbol{\theta}_t^{\text{cur}} = -\eta\mathbf{g}_t^{\text{cur}}, \tag{31}$$

where η is the learning rate. Its alignment with the descent direction preferred by the current objective is

$$(-\mathbf{g}_t^{\text{cur}})^\top\Delta\boldsymbol{\theta}_t^{\text{cur}} = \eta\|\mathbf{g}_t^{\text{cur}}\|_2^2. \tag{32}$$

After introducing historical inheritance, the update becomes

$$\Delta\boldsymbol{\theta}_t = -\eta\left(\mathbf{g}_t^{\text{cur}} + \mathbf{g}_t^{\text{inh}}\right). \tag{33}$$

Its alignment with the current descent direction is therefore

$$\begin{aligned}(-\mathbf{g}_t^{\text{cur}})^\top\Delta\boldsymbol{\theta}_t &= (-\mathbf{g}_t^{\text{cur}})^\top\left[-\eta(\mathbf{g}_t^{\text{cur}} + \mathbf{g}_t^{\text{inh}})\right] \\ &= \eta\|\mathbf{g}_t^{\text{cur}}\|_2^2 + \eta(\mathbf{g}_t^{\text{cur}})^\top\mathbf{g}_t^{\text{inh}}.\end{aligned} \tag{34}$$

The effect of historical inheritance on the current-aligned update is thus determined by the interaction between the two gradients. When

$$\boxed{(\mathbf{g}_t^{\text{cur}})^\top\mathbf{g}_t^{\text{inh}} < 0}, \tag{35}$$

we have

$$\boxed{(-\mathbf{g}_t^{\text{cur}})^\top\Delta\boldsymbol{\theta}_t < (-\mathbf{g}_t^{\text{cur}})^\top\Delta\boldsymbol{\theta}_t^{\text{cur}}}. \tag{36}$$

Therefore, a conflicting inheritance gradient reduces the component of the parameter update aligned with the current adaptation objective.

The interaction can equivalently be expressed as

$$(\mathbf{g}_t^{\text{cur}})^\top\mathbf{g}_t^{\text{inh}} = \|\mathbf{g}_t^{\text{cur}}\|_2\|\mathbf{g}_t^{\text{inh}}\|_2\cos\phi_t, \tag{37}$$

where ϕ_t denotes the angle between the two gradients. Hence, $\cos\phi_t < 0$ corresponds directly to the conflict condition in Eq. (35), which also motivates the gradient-cosine diagnostic used in our experiments.

Importantly, Property 2 does not imply that historical inheritance is detrimental to the overall incremental objective. Historical constraints can remain beneficial for preserving previously learned classes, even when they reduce part of the update favored by current supervision. The property instead characterizes a local optimization trade-off: preservation and current adaptation can request partially opposing parameter updates.

Property 2 therefore identifies when historical inheritance limits current-task-aligned adaptation, rather than claiming that such inheritance necessarily degrades overall incremental performance. This observation motivates regulating how historical information is inherited: rather than preserving the complete representation space, RPA realizes this principle at the feature level by selectively constraining directions with limited observed adaptation demand. The resulting feature-space inheritance signal is subsequently mapped to model parameters through back-propagation, as discussed in Sec. A.3.

A.3 Derivation of Reusable Semantic Directions

We provide a detailed derivation of the reusable semantic direction used in RPA. Property 2 motivates reducing historical inheritance that interferes with current adaptation. RPA realizes this principle in feature space, where historical semantics and observed adaptation can be characterized explicitly.

Let $\mathbf{z}(\boldsymbol{\theta})$ denote the feature representation and $\mathbf{J}_z(\boldsymbol{\theta}) = \partial \mathbf{z}/\partial \boldsymbol{\theta}$ its Jacobian with respect to the model parameters. For the feature-level inheritance objective, back-propagation gives

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{RPA}} = \mathbf{J}_z(\boldsymbol{\theta})^\top \nabla_{\mathbf{z}} \mathcal{L}_{\mathrm{RPA}}. \tag{38}$$

Thus, selecting reusable directions in feature space regulates the inheritance signal before it is propagated to parameter space. Because the Jacobian need not preserve directional angles, the feature-space criterion is not an exact reformulation of the parameter-space conflict condition in Property 2. Instead, it provides a feature-level realization of its optimization principle, whose induced parameter-space interaction is evaluated directly through the gradient diagnostics in our experiments.

The objective is therefore to identify a feature direction that preserves strong historical class discriminability while incurring limited historical dispersion and current adaptation demand.

For an old class c, $\boldsymbol{\delta}_c$ characterizes its historical class contrast, $\boldsymbol{\Sigma}_{\mathrm{hist}}$ describes historical within-class dispersion, and $\boldsymbol{\Sigma}_{\mathrm{ada}}$ measures the observed feature variation required by current adaptation. The directional reusability criterion is defined over directions supported by the observed dispersion and adaptation statistics:

$$\mathcal{R}_c(\mathbf{u}) = \frac{(\mathbf{u}^\top \boldsymbol{\delta}_c)^2}{\mathbf{u}^\top (\boldsymbol{\Sigma}_{\mathrm{hist}} + \boldsymbol{\Sigma}_{\mathrm{ada}}) \mathbf{u}}. \tag{39}$$

Directions in the null space of the combined scatter are excluded from the criterion, since their zero denominator provides no finite basis for comparing reusability and is not interpreted as infinite semantic reliability. The numerator measures the squared class contrast projected onto $\mathbf{u}$, whereas the denominator measures the historical dispersion and current adaptation demand along the same direction.

For compact notation, the combined dispersion matrix is written as

$$\mathbf{A} = \boldsymbol{\Sigma}_{\mathrm{hist}} + \boldsymbol{\Sigma}_{\mathrm{ada}} \succeq \mathbf{0}. \tag{40}$$

Equation 39 then becomes

$$\mathcal{R}_c(\mathbf{u}) = \frac{\left(\mathbf{u}^\top \boldsymbol{\delta}_c\right)^2}{\mathbf{u}^\top \mathbf{A} \mathbf{u}}. \tag{41}$$

Scale invariance. The criterion depends only on the direction of $\mathbf{u}$, rather than its magnitude. For any nonzero scalar α,

$$\begin{aligned} \mathcal{R}_c(\alpha \mathbf{u}) &= \frac{\alpha^2 (\mathbf{u}^\top \boldsymbol{\delta}_c)^2}{\alpha^2 \mathbf{u}^\top \mathbf{A} \mathbf{u}} \\ &= \mathcal{R}_c(\mathbf{u}). \end{aligned} \tag{42}$$

Therefore, maximizing $\mathcal{R}_c$ amounts to finding an optimal *direction*; its magnitude can be normalized afterwards.

Transforming the denominator. The main difficulty in Eq. (41) is the quadratic denominator $\mathbf{u}^\top \mathbf{A}\mathbf{u}$. On the effective subspace associated with the positive eigenvalues of $\mathbf{A}$, the matrix admits $\mathbf{A}^{1/2}$ and $\mathbf{A}^{-1/2}$. We introduce the transformed variable

$$\mathbf{v} = \mathbf{A}^{1/2}\mathbf{u}. \tag{43}$$

This transformation is introduced specifically to convert the quadratic denominator into an ordinary Euclidean norm. Since $\mathbf{u} = \mathbf{A}^{-1/2}\mathbf{v}$ on this subspace,

$$\begin{aligned}\mathbf{u}^\top \mathbf{A}\mathbf{u} &= (\mathbf{A}^{-1/2}\mathbf{v})^\top \mathbf{A}(\mathbf{A}^{-1/2}\mathbf{v}) \\ &= \mathbf{v}^\top \mathbf{A}^{-1/2}\mathbf{A}\mathbf{A}^{-1/2}\mathbf{v} \\ &= \mathbf{v}^\top \mathbf{v} = \|\mathbf{v}\|_2^2.\end{aligned} \tag{44}$$

The numerator is transformed in the same manner:

$$\begin{aligned}\left(\mathbf{u}^\top \boldsymbol{\delta}_c\right)^2 &= \left[(\mathbf{A}^{-1/2}\mathbf{v})^\top \boldsymbol{\delta}_c\right]^2 \\ &= \left[\mathbf{v}^\top \mathbf{A}^{-1/2}\boldsymbol{\delta}_c\right]^2,\end{aligned} \tag{45}$$

where the symmetry of $\mathbf{A}^{-1/2}$ is used.

Substituting Eqs. (44) and (45) into Eq. (41) gives

$$\mathcal{R}_c = \frac{\left[\mathbf{v}^\top \mathbf{A}^{-1/2}\boldsymbol{\delta}_c\right]^2}{\|\mathbf{v}\|_2^2}. \tag{46}$$

Upper bound from the Cauchy–Schwarz inequality. The Cauchy–Schwarz inequality states that, for two vectors $\mathbf{a}$ and $\mathbf{b}$,

$$(\mathbf{a}^\top \mathbf{b})^2 \le \|\mathbf{a}\|_2^2 \|\mathbf{b}\|_2^2, \tag{47}$$

with equality when the two vectors are linearly dependent.

Applying this inequality to $\mathbf{a} = \mathbf{v}$ and $\mathbf{b} = \mathbf{A}^{-1/2}\boldsymbol{\delta}_c$ gives

$$\left[\mathbf{v}^\top \mathbf{A}^{-1/2}\boldsymbol{\delta}_c\right]^2 \le \|\mathbf{v}\|_2^2 \left\|\mathbf{A}^{-1/2}\boldsymbol{\delta}_c\right\|_2^2. \tag{48}$$

Dividing both sides by $\|\mathbf{v}\|_2^2$ yields the upper bound

$$\mathcal{R}_c \le \left\|\mathbf{A}^{-1/2}\boldsymbol{\delta}_c\right\|_2^2. \tag{49}$$

Therefore, maximizing $\mathcal{R}_c$ is equivalent to reaching the equality condition of the Cauchy–Schwarz inequality. Equality is achieved when $\mathbf{v}$ and $\mathbf{A}^{-1/2}\boldsymbol{\delta}_c$ are linearly dependent, i.e.,

$$\boxed{\mathbf{v} \propto \mathbf{A}^{-1/2}\boldsymbol{\delta}_c}. \tag{50}$$

Here, $\propto$ means that the two vectors have the same direction up to a nonzero scalar factor.

Recovering the optimal direction in the original space. According to Eq. (43), $\mathbf{v} = \mathbf{A}^{1/2}\mathbf{u}$. Substituting this relation into Eq. (50) gives

$$\mathbf{A}^{1/2}\mathbf{u} \propto \mathbf{A}^{-1/2}\boldsymbol{\delta}_c. \tag{51}$$

Multiplying both sides by $\mathbf{A}^{-1/2}$ gives

$$\boxed{\mathbf{u} \propto \mathbf{A}^{-1}\boldsymbol{\delta}_c}. \tag{52}$$

This result provides a direct interpretation of the reusable direction: $\boldsymbol{\delta}_c$ favors directions with strong historical class contrast, whereas $\mathbf{A}^{-1}$ suppresses directions associated with large historical dispersion or large current adaptation demand.

Singular case and effective-subspace optimization. Because $\mathbf{A} \succeq 0$ can be singular or numerically ill-conditioned, we optimize the directional reusability objective over its numerically effective subspace. This restriction is part of the definition of the RPA criterion: only directions associated with eigenvalues above the numerical threshold are compared, whereas null-space and numerically

negligible directions are excluded rather than assigned an infinite reusability score. Let

$$A = U\Lambda U^\top \tag{53}$$

be the eigendecomposition of A. We retain eigenvalues satisfying

$$\lambda_j > \tau_A, \qquad \tau_A = D\,\epsilon_{\text{mach}}\,\lambda_{\max}(A), \tag{54}$$

where D is the feature dimension and ϵ_{mach} denotes machine precision. Let U_r and Λ_r contain the retained eigenvectors and eigenvalues, respectively. We define the numerically effective subspace as

$$\mathcal{E}_r = \text{span}(U_r),$$

and the corresponding truncated pseudoinverse as

$$A_r^\dagger = U_r\Lambda_r^{-1}U_r^\top.$$

Eigenvalues below τ_A are treated as zero, preventing near-null directions from being excessively amplified.

Numerical eigendecomposition and effective rank. The eigendecomposition of $\mathbf{A}$ and the construction of $\mathbf{A}_r^\dagger$ are performed in FP32, independently of the mixed-precision forward pass used during network training. The effective rank is determined by

$$r = \sum_{j=1}^{D} \mathbf{1}[\lambda_j > \tau_A], \tag{55}$$

and is therefore a numerical rank rather than a predefined projection dimension or a tunable hyperparameter. Since $\mathbf{A} = \mathbf{\Sigma}_{\text{hist}} + \mathbf{\Sigma}_{\text{ada}}$ combines step-level historical statistics with the current adaptation statistics, r is recomputed whenever $\mathbf{A}$ is updated and can vary across mini-batches and incremental steps. The threshold τ_A only removes numerically unsupported near-null directions. Importantly, this effective rank should not be confused with the rank-one projector $\mathbf{P}_c = \mathbf{u}_c^\star(\mathbf{u}_c^\star)^\top$ used by RPA: $\mathcal{E}_r$ defines the supported search space from which one reusable direction is selected for each valid class.

Any feasible direction can therefore be written as

$$\mathbf{u} = \mathbf{U}_r\mathbf{a}, \qquad \mathbf{a} \neq \mathbf{0}. \tag{56}$$

Within this effective subspace, $\mathbf{\Lambda}_r$ is positive definite, and the reusability objective becomes

$$\mathcal{R}_c(\mathbf{u}) = \frac{\left(\mathbf{a}^\top\mathbf{U}_r^\top\boldsymbol{\delta}_c\right)^2}{\mathbf{a}^\top\mathbf{\Lambda}_r\mathbf{a}}. \tag{57}$$

Applying the same Cauchy–Schwarz argument as above gives the maximizing direction in the effective subspace as

$$\mathbf{u}_c^\star \propto \mathbf{U}_r\mathbf{\Lambda}_r^{-1}\mathbf{U}_r^\top\boldsymbol{\delta}_c. \tag{58}$$

Since

$$\mathbf{A}_r^\dagger = \mathbf{U}_r\mathbf{\Lambda}_r^{-1}\mathbf{U}_r^\top, \tag{59}$$

the solution can be compactly written using the truncated pseudoinverse. For a valid class satisfying

$$\mathbf{A}_r^\dagger\boldsymbol{\delta}_c \neq \mathbf{0}, \tag{60}$$

the normalized maximizing direction is

$$\mathbf{u}_c^\star = \frac{\mathbf{A}_r^\dagger\boldsymbol{\delta}_c}{\left\|\mathbf{A}_r^\dagger\boldsymbol{\delta}_c\right\|_2}. \tag{61}$$

The corresponding maximum over the effective subspace is finite and equals

$$\max_{\mathbf{u}\in\mathcal{E}_r\setminus\{\mathbf{0}\}} \mathcal{R}_c(\mathbf{u}) = \boldsymbol{\delta}_c^\top\mathbf{A}_r^\dagger\boldsymbol{\delta}_c. \tag{62}$$

If $\mathbf{A}_r^\dagger\boldsymbol{\delta}_c = \mathbf{0}$, the projection of $\boldsymbol{\delta}_c$ onto the numerically retained subspace is zero, and $\mathcal{R}_c(\mathbf{u}) = 0$ for every feasible direction in this subspace.

Thus, $\mathbf{u}_c^\star$ is the optimal reusable direction within the numerically effective subspace $\mathcal{E}_r$, rather than an unconditional optimum over the complete feature space.

A.4 Current-Step Distribution Construction for RPA and DEP

All statistics are constructed from current-step data $\mathcal{D}_t$. Old-class regions are recovered only from background-labeled locations using predictions from frozen f_{t-1}, whereas new-class regions follow

the current ground-truth annotations. These regions support historical reference construction for RPA and current distribution estimation for DEP, without accessing historical training samples.

Historical statistics and update order for RPA. For old class c, $\mathcal{S}_c$ indexes the valid observations processed so far within step t, with $N_c = |\mathcal{S}_c|$. At each iteration, newly observed old-class locations are incorporated into $\mathcal{S}_c$, and their detached previous-model features $\mathbf{z}_o(i)$ are included before recomputing the statistics in Eq. 8. Thus, each update combines the current mini-batch with previously processed observations rather than replacing the accumulated statistics. Although all features are extracted under the same frozen model state, the statistical estimates change as observations accumulate.

The accumulated observations and statistics are retained across mini-batches and epochs within step t and discarded at its completion. An old class absent from the current mini-batch retains its contribution to $\bar{\boldsymbol{\mu}}_o$ and $\boldsymbol{\Sigma}_{\mathrm{hist}}$, but contributes no direct alignment term in that batch. Direct RPA alignment requires both current-batch locations and a valid reusable direction. Classes not yet observed have no available historical statistics.

Current-model statistics for RPA and DEP. RPA additionally recomputes $\boldsymbol{\Sigma}_{\mathrm{ada}}$ from current- and previous-model features at new-class locations in each mini-batch, following Sec. 3.4. This statistic is not accumulated across parameter updates and is set to $\mathbf{0}$ when no valid new-class location is available.

For DEP, Ω_t^k contains the valid locations of class k in the current mini-batch. Both old- and new-class distributions are estimated from current-model features $\mathbf{z}_t$ according to Sec. 3.5, with numerical stabilization described in Appendix A.5. They are recomputed at each iteration rather than inherited from earlier batches or steps. Gradients propagate through their means and variances to f_t, allowing both old- and new-class representations to adapt. Only pairs with two valid class distributions contribute to DEP; the loss is zero when no valid old–new pair is available.

Synchronization under distributed training. Across workers, per-mini-batch observation counts, feature sums, and element-wise squared-feature sums are aggregated before updating the RPA historical statistics. New-class counts and sums of feature-displacement outer products are likewise aggregated before normalizing $\boldsymbol{\Sigma}_{\mathrm{ada}}$. All workers therefore construct the same historical and adaptation statistics for reusable direction selection.

Robustness to pseudo-supervision errors. Discarding step-local observations and statistics avoids directly carrying their estimation errors into subsequent steps through persistent statistical buffers. However, pseudo-label errors may still affect the learned parameters and hence subsequent previous models; step-wise reconstruction does not eliminate this possibility.

The long-horizon experiments provide evidence of old-class retention under the evaluated protocols. On VOC 10–1 and 2–2, AKI achieves old-class mIoUs of 81.8% and 74.0%, compared with 85.0% and 77.3% for joint training, respectively. On ADE20K 100–5, the corresponding values are 47.2% and 52.6%. These results support historical preservation in the tested settings rather than establishing immunity to pseudo-supervision errors.

A.5 Stable Computation of DEP

DEP models only classes with valid feature locations in the current incremental step. For class k, statistics are computed when $|\Omega_t^k| > 0$; otherwise, the class is excluded from the corresponding DEP computation. Therefore, no artificial distribution is introduced for classes that are absent from the available current-step regions.

Feature representation used by DEP. DEP operates on the decoder patch features before ℓ_2 normalization, which are the same representations used in the cross-level analysis in Appendix B.1. Specifically, $\mathbf{z}_t(i) \in \mathbb{R}^D$ denotes the decoder feature at spatial location i, with $D = 768$ in our implementation. DEP directly constructs class-conditional statistics in this original D-dimensional feature space; the two-dimensional PCA projections used elsewhere are only for diagnostic visualization and are not involved in DEP optimization.

To avoid degenerate diagonal Gaussian distributions and excessively small variance denominators, we use

$$\widetilde{v}_t^k = v_t^k + \epsilon_v \mathbf{1}, \qquad \epsilon_v = 10^{-6}, \tag{63}$$

where ϵ_v is fixed for numerical stability in all experiments and is not treated as a tunable hyperparameter. The resulting distribution is

$$p_t^k(\mathbf{z}) = \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_t^k, \mathrm{Diag}(\widetilde{\mathbf{v}}_t^k)\right). \tag{64}$$

For a valid old–new class pair (c, n), define

$$\bar{v}_{c,n} = \frac{\widetilde{v}_t^c + \widetilde{v}_t^n}{2}. \tag{65}$$

The variance floor ϵ_v ensures that all diagonal covariance entries and the corresponding denominators below remain strictly positive. For diagonal Gaussian distributions, the log Bhattacharyya coefficient is evaluated in closed form as

$$\begin{aligned}\log \mathrm{BC}_{c,n} = &- \frac{1}{8}\sum_{d=1}^{D} \frac{(\mu_{t,d}^c - \mu_{t,d}^n)^2}{\bar{v}_{c,n,d}} \\ &- \frac{1}{2}\sum_{d=1}^{D} \log\left(\frac{\bar{v}_{c,n,d}}{\sqrt{\widetilde{v}_{t,d}^c \widetilde{v}_{t,d}^n}}\right).\end{aligned} \tag{66}$$

We then obtain

$$O_{c,n} = \exp(\log \mathrm{BC}_{c,n}). \tag{67}$$

For numerical stability, the high-dimensional coefficient is evaluated through the log-domain expression above rather than by directly multiplying dimension-wise overlap terms; exponentiation is applied only to the resulting scalar log coefficient.

Range and normalization of the DEP objective. For any valid Gaussian pair,

$$0 < \mathrm{BC}(p_t^c, p_t^n) \le 1, \qquad \log \mathrm{BC}(p_t^c, p_t^n) \le 0. \tag{68}$$

Accordingly, each pairwise DEP term is bounded in $(0, 1]$. Moreover, $\mathcal{L}_{\mathrm{DEP}}$ averages the Bhattacharyya coefficients over valid old–new class pairs rather than summing them. Its numerical scale therefore does not increase directly with the number of valid class pairs in a mini-batch. The log-domain expression is used only for numerically stable evaluation, whereas the exponentiated coefficient $\mathcal{O}_{c,n}$ constitutes the optimization objective. All auxiliary objectives retain the unit coefficients defined in Eq. 4; no pair-count-dependent rescaling is introduced.

Gradient signal for overlapping class pairs. The pairwise objective satisfies $\mathcal{O}_{c,n} = \exp(\log \mathrm{BC}_{c,n})$. For feature dimension d, its derivative with respect to the old-class mean is

$$\frac{\partial \mathcal{O}_{c,n}}{\partial \mu_{t,d}^c} = -\frac{\mathcal{O}_{c,n}}{4\bar{v}_{c,n,d}}\left(\mu_{t,d}^c - \mu_{t,d}^n\right). \tag{69}$$

Therefore, high overlap does not by itself eliminate the analytical mean-separation signal: an overlapping class pair has a nonzero contribution through this term whenever its means differ. Gradients additionally propagate through $\widetilde{\mathbf{v}}_t^c$ and $\widetilde{\mathbf{v}}_t^n$, so pairs with different dispersions also receive a distribution-shaping signal. If two Gaussian distributions are exactly identical in both mean and variance, their Bhattacharyya coefficient reaches its maximum and the configuration is stationary. DEP does not assume such exact coincidence in general.

Mean and dispersion contributions. The negative log Bhattacharyya coefficient admits the decomposition

$$-\log \mathrm{BC}_{c,n} = D_\mu(c, n) + D_v(c, n), \tag{70}$$

where

$$D_\mu(c, n) = \frac{1}{8}\sum_{d=1}^{D} \frac{\left(\mu_{t,d}^c - \mu_{t,d}^n\right)^2}{\bar{v}_{c,n,d}}, \tag{71}$$

and

$$D_v(c,n) = \frac{1}{2}\sum_{d=1}^{D} \log \frac{\bar{v}_{c,n,d}}{\sqrt{\widetilde{v}^c_{t,d}\widetilde{v}^n_{t,d}}}. \tag{72}$$

Here, D_μ measures class-center separation relative to the pooled feature-wise dispersion, whereas D_v characterizes differences in class-specific dispersion. DEP therefore regulates both the location and spread of the current class distributions instead of operating only on their centers. We do not attribute the effectiveness of DEP to either term alone; both jointly determine the pairwise Bhattacharyya overlap optimized by the full objective.

Relation to center-based separation. Consider the special case in which an old–new class pair shares the same diagonal covariance,

$$\widetilde{\mathbf{v}}^c_t = \widetilde{\mathbf{v}}^n_t = \mathbf{v}. \tag{73}$$

Then $\bar{\mathbf{v}}_{c,n} = \mathbf{v}$ and $D_v(c,n) = 0$, such that

$$-\log \mathrm{BC}_{c,n} = \frac{1}{8}\sum_{d=1}^{D} \frac{\left(\mu^c_{t,d} - \mu^n_{t,d}\right)^2}{v_d}. \tag{74}$$

Thus, under shared covariance, DEP reduces to a variance-normalized center-separation objective. The full formulation generalizes this special case by retaining class-specific feature-wise variances, allowing both class-center separation and distributional spread to participate in overlap regulation.

Sparse class occurrences. DEP uses only class regions observed in the current mini-batch. For a class k, the statistic is constructed only when $|\Omega^k_t| > 0$; otherwise, no distribution is introduced for that class. Likewise, an old–new pair contributes to $\mathcal{P}_t$ only when both class distributions are available. The variance floor ϵ_v ensures that the resulting diagonal Gaussian and Bhattacharyya computation remain numerically well-defined even when a class has relatively few valid feature locations. This stabilization addresses numerical degeneracy rather than imposing or assuming a minimum statistical support size. DEP uses these Gaussian parameters as differentiable mini-batch statistics for overlap regularization rather than as persistent density estimates stored across incremental steps.

DEP averages $O_{c,n}$ only over valid old–new class pairs. If no valid pair is available, the DEP term is skipped for that optimization step. Since the class statistics are computed from current-model features $\mathbf{z}_t$, gradients from $\mathcal{L}_{\mathrm{DEP}}$ propagate through $\boldsymbol{\mu}^k_t$ and $\widetilde{\mathbf{v}}^k_t$ to the current representation and model parameters.

Thus, DEP remains well-defined for sparse class occurrences while directly optimizing overlap between valid old- and new-class distributions in the current representation space.

Numerical precision. Although automatic mixed precision is used during network training, the class means, feature-wise variances, variance stabilization, log-domain Bhattacharyya coefficient, and final scalar exponentiation used by DEP are evaluated in FP32. The resulting DEP loss remains differentiable with respect to the current-model features, and gradients are propagated through the class statistics by standard back-propagation. Using FP32 together with the log-domain formulation reduces numerical underflow risk and precision loss in the original D-dimensional feature space.

A.6 Scope of the Theoretical Analysis and Design Implications

Properties 1 and 2 characterize two optimization phenomena arising in fixed-capacity CISS. Property 1 shows that changes in incremental supervision generally require movement away from the previous stationary state, whereas Property 2 shows that historical inheritance can contain gradient components that oppose the direction favored by current adaptation.

These properties provide the optimization motivation of AKI, but they are not intended to prove that a particular response mask, projection criterion, or distribution penalty is the unique or theoretically optimal realization of this principle. RCA, RPA, and DEP are concrete mechanisms designed to address the corresponding optimization requirements, and their specific forms are further justified by their respective formulations and empirical analyses.

Response-level implication. Property 1 indicates that the model needs sufficient flexibility to move toward the stationary state associated with the current objective. RCA instantiates this requirement at the response level by reducing dependence on the complete response space through compact-support consistency. The weight-norm-based capacity cue is a practical criterion for constructing this support; its effectiveness is evaluated separately through the threshold, compact-support, and component ablations.

Feature-level implication. Property 2 motivates avoiding unnecessary historical constraints in feature directions involved in current adaptation. RPA realizes this principle at the feature level by selecting reusable semantic directions according to historical discriminability and observed adaptation demand, rather than constraining the entire representation space. Through back-propagation, the resulting feature-space inheritance signal induces the corresponding parameter-space gradient. The particular reusability criterion is not a closed-form consequence of Property 2, but a feature-level realization of its optimization principle; its analytic solution is derived in Sec. A.3, while its induced parameter-space compatibility is evaluated through the gradient-conflict analyses.

Distribution-level implication. Response- and feature-level adaptation ultimately changes the organization of current class representations. However, Properties 1 and 2 alone do not guarantee that the resulting old- and new-class distributions are sufficiently separated. DEP therefore provides a complementary distribution-level objective that directly penalizes their residual overlap. Its formulation is analyzed separately through the Bhattacharyya-coefficient interpretation and the distribution-overlap experiments.

The optimization analysis therefore provides design guidance for adaptive inheritance, while the specific designs of RCA, RPA, and DEP constitute complementary realizations whose effectiveness is established by their individual derivations and empirical evaluations.

A.7 From Optimization and Empirical Analyses to Adaptive Knowledge Inheritance

The preceding analysis characterizes how changes in incremental supervision and historical inheritance interact with current optimization under fixed model capacity. Property 1 identifies the need for sufficient adaptation flexibility when the task-driven objective changes, while Property 2 characterizes when historical inheritance reduces the update component aligned with current adaptation.

Importantly, these properties are not intended to establish RCA, RPA, and DEP as unique or theoretically necessary solutions. Instead, they provide the optimization motivation for reconsidering how previously acquired knowledge should participate in current learning. AKI translates this motivation into three complementary mechanisms with different operating scopes: RCA regulates the response organization of the current model, RPA explicitly controls cross-step feature inheritance, and DEP reorganizes old–new class distributions within the current representation space.

Distinction from closely related adaptive CISS strategies. Recent CISS methods have already introduced different forms of adaptation, and our contribution should therefore not be interpreted as introducing "adaptivity" itself. CoinSeg Zhang et al. (2023) emphasizes plasticity through flexible parameter tuning together with discriminative inter- and intra-class representation learning. MBS Park et al. (2024) selectively transfers reliable historical knowledge through selective pseudo-labeling and adaptive feature distillation. CoGaMiD Zhu et al. (2025a) explicitly addresses the mismatch between stored historical distributions and evolving representations by dynamically updating Gaussian mixture models and constraining new representations against historical distributions.

AKI differs from these approaches in both its optimization perspective and the objects being regulated. Rather than treating parameter flexibility, reliable distillation regions, or historical-distribution updating as separate mechanisms, AKI distinguishes three different questions arising during incremental optimization: *how strongly should the current prediction depend on its complete response space, which historical feature directions remain reusable for the current objective, and how should the resulting current old–new distributions be organized?* These questions lead to RCA, RPA, and DEP, respectively.

From Property 1 to current response-space regulation. Property 1 shows that, when the aggregate change of the task-driven objective produces a nonzero gradient at the previous stationary state, the stationary state required by the current objective differs from the previous one:

$$\boxed{\underline{\boldsymbol{\theta}_t^* \neq \boldsymbol{\theta}_{t-1}^*}}. \tag{75}$$

The current model therefore needs sufficient flexibility to reorganize its parameters and responses toward the current objective. RCA instantiates this requirement entirely within the current model. Different from explicit parameter release or structural pruning, RCA does not permanently remove, freeze, or allocate particular parameters. Instead, it constructs an auxiliary compact response support and regularizes prediction consistency between the complete and compact-support responses. Its specific role is therefore to regulate the dependence of the current predictive function on the complete response space while retaining response-level flexibility for current adaptation.

From Property 2 to selective cross-step feature inheritance. Property 2 characterizes a different issue: historical inheritance can reduce the update component aligned with current adaptation when the two gradients conflict:

$$\boxed{\underline{(\mathbf{g}_t^{\mathrm{cur}})^\top \mathbf{g}_t^{\mathrm{inh}} < 0}}. \tag{76}$$

This result does not imply that historical preservation is detrimental to the overall incremental objective; instead, it reveals a local optimization trade-off between preservation and current adaptation. RPA is the explicit cross-step inheritance mechanism in AKI. Different from confidence-based feature distillation that primarily determines *where* historical information should be transferred, RPA determines *which feature directions* remain reusable. It jointly considers historical discriminability and current adaptation demand, and constrains only the projected reusable components. The complementary directions remain flexible for current optimization. Accordingly, Property 2 provides the parameter-space motivation for selective inheritance, while RPA realizes this principle through direction-aware regulation in feature space; the actual parameter-space interaction is determined after back-propagation and is evaluated empirically through gradient alignment.

The novelty of RPA therefore lies in changing feature inheritance from full-space or region-wise preservation to direction-aware semantic reuse, rather than simply adjusting the strength of conventional feature distillation.

From current representation adaptation to distribution organization. RCA and RPA regulate current learning at the response and feature levels, respectively, and their effects are ultimately reflected in the organization of class-conditional representations. Response organization influences feature formation, while feature representations determine the location and dispersion of class distributions.

However, response flexibility and selective feature inheritance do not guarantee sufficient separation between old and new classes in the current representation space. Residual distributional entanglement can therefore remain. DEP provides an explicit current-step mechanism for addressing this consequence. Different from methods that store, replay, or dynamically update historical distribution models, DEP does not attempt to recover or anchor the current representation to historical distribution geometry. Instead, it models both previously learned and newly introduced classes using their *current* representations and directly penalizes their pairwise Bhattacharyya overlap. This allows old-class representations to reorganize together with new classes while explicitly encouraging current old–new distribution separation.

Innovation scope of AKI. The distinction of AKI therefore lies neither in parameter plasticity, selective distillation, nor adaptive distribution modeling alone, all of which have been explored from different perspectives in previous CISS methods. Instead, AKI formulates the interaction between historical inheritance and current adaptation at the optimization level, and realizes this formulation through three different intervention granularities: current response-space dependence, reusable cross-step feature directions, and current old–new distribution overlap.

The resulting design logic can be summarized as

$$\boxed{\begin{aligned} \text{Changed incremental objective} &\Longrightarrow \text{response-space regulation (RCA)} \\ \text{Potential inheritance conflict} &\Longrightarrow \text{direction-aware feature inheritance (RPA)} \\ \text{old–new overlap} &\Longrightarrow \text{distribution regulation (DEP).} \end{aligned}} \tag{77}$$

The contribution of AKI therefore lies in using cross-step optimization analysis to coordinate three distinct forms of knowledge reorganization: RCA regulates current response-space dependence, RPA selectively inherits reusable historical feature directions, and DEP directly regulates old–new distribution overlap in the current representation space.

B Details of experimental setups

Dataset. Following previous works Cermelli et al. (2020); Yang et al. (2023); Zhang et al. (2023); Park et al. (2024), our method is evaluated on the Pascal VOC 2012 Everingham et al. (2010) and ADE20K Zhou et al. (2017) datasets. The Pascal VOC dataset Everingham et al. (2010), a widely used benchmark for incremental segmentation, comprises 10,582 training images and 1,449 validation images across 20 categories. Additionally, the more challenging ADE20K dataset Zhou et al. (2017) includes 150 categories, with 20,210 training images and 2,000 validation images.

Implementation details. Following established practices Park et al. (2024); Zhang et al. (2023), we adopt an ImageNet-1K-pretrained ViT as the backbone. The decoder consists of two Transformer blocks, and the training crop size is 512×512. Each GPU uses a batch size of 6; with six GPUs, the total batch size is 36. The model is optimized using SGD with momentum 0.9, Nesterov acceleration, and weight decay 1×10^{-5}. Each base and incremental step is trained for 64 epochs. The learning rate is set to 1×10^{-3} at the base step and to 1×10^{-4} and 5×10^{-4} at incremental steps on Pascal VOC and ADE20K, respectively.

All internally trained variants use the same data-processing pipeline. Training images are randomly scaled by a factor in $[0.5, 2.0]$, randomly cropped to 512×512 with padding when necessary, horizontally flipped with probability 0.5, and normalized using the ImageNet mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$. The compact-response threshold in RCA is fixed to $\tau_w = 0.1$ throughout all experiments. All experiments are implemented in PyTorch and conducted on six NVIDIA GeForce RTX 3090 GPUs with an Intel(R) Xeon(R) Gold 6226R CPU.

Controlled comparison protocol. For the comparison with MBS, AKI uses the same network architecture, optimization settings, data augmentation, pseudo-labeling strategy, training schedule, and inference pipeline; the difference lies only in the training objectives. The component ablation in Table 3 uses the same base configuration and pseudo supervision, with only different subsets of $\mathcal{L}_{\mathrm{RCA}}$, $\mathcal{L}_{\mathrm{RPA}}$, and $\mathcal{L}_{\mathrm{DEP}}$ enabled.

Metrics. In this paper, the performance of incremental semantic segmentation is evaluated using the commonly adopted mean Intersection over Union (mIoU) metric, consistent with previous methods. In the supplemental material, we additionally report class accuracy for class-wise analysis, defined as

$$\mathrm{Acc} = \frac{TP_c}{TP_c + FN_c}, \tag{78}$$

where TP_c and FN_c denote the true-positive and false-negative pixels of class c, respectively. Thus, Acc measures the recall over ground-truth pixels of class c. The mean class accuracy (mAcc) is obtained by averaging Acc over the evaluated classes. In our terminology, "Old" refers to the mIoU of old classes from the base stage, "New" denotes the mIoU of new classes introduced in incremental stages, and "All" represents the mIoU across all classes, including background.

Baselines. In CISS, most current methods Cermelli et al. (2020); Yang et al. (2023); Zhang et al. (2023); Park et al. (2024) focus on optimizing incremental learning strategies rather than the architectural design. The primary architectural difference among existing methods lies in the choice of backbone—typically either ResNet101 or ViT. In this paper, we compare our method with both ResNet101-based and ViT-based approaches on the VOC and ADE20K datasets. To demonstrate that our method is not limited to ViT-based architectures, we replace the backbone in CoinSeg with ResNet101 and apply our proposed AKI strategy for performance validation. We also report re-implementation results of existing methods. Additionally, we provide the performance of our archi-

tecture under joint training across multiple incremental configurations, which is currently considered the upper bound of incremental learning for models in this field Yang et al. (2023); Baek et al. (2022).

Incremental configurations. In addition to the widely explored incremental configurations on the VOC dataset—15-1, 15-5, and 19-1—we also conduct experiments on settings with more incremental steps, such as 10-1 and 2-2. Furthermore, we evaluate our method on four configurations on ADE20K: 100-50, 50-50, 100-10, and 100-5. These diverse experimental setups allow for a comprehensive assessment of the robustness of our approach under various incremental learning settings. For example, the 19-1 setting represents a typical setting where most foreground classes (19 in total, excluding background) are learned in the base stage, with only one new class introduced in the incremental stage. In the 2-2 configuration, the model is initially trained on 2 foreground classes, with 2 additional classes added at each of the 9 subsequent incremental steps—culminating in 18 new classes. This setup reflects a representative case of a substantial number of classes to be acquired incrementally.

B.1 Controlled Analysis of Cross-Level Representation Propagation

To examine whether changes at different representation levels are systematically propagated through the model, we conduct a controlled perturbation analysis on a fixed trained model rather than comparing different training checkpoints. This choice is deliberate: during incremental optimization, both upstream representations and downstream model parameters can change simultaneously, making it difficult to isolate whether an observed downstream variation is induced by the upstream representation change itself or by concurrent parameter updates. Fixing the model removes the latter factor and allows us to specifically characterize the forward propagation of controlled response variations to downstream feature and distribution representations. Specifically, we use the ViT-B/16 Segmenter trained on Pascal VOC under the 10-1 protocol at the final incremental step (step 10), and perform inference on the official validation set without any additional training. The analysis contains 1,449 validation images, among which 1,437 contain at least one eligible foreground class. All perturbations are applied only to the encoder response, while the downstream decoder remains unchanged.

Response and feature representations. For an image x, let $\mathbf{r}(x) \in \mathbb{R}^{HW\times 768}$ denote the encoder patch responses after removing the special tokens, which are directly fed into the decoder. Let $\mathbf{z}(x) \in \mathbb{R}^{HW\times 768}$ denote the decoder patch features before ℓ_2 normalization. The latter corresponds to the same feature level used for the distribution modeling in DEP. The ground-truth segmentation mask is resized to the patch resolution by nearest-neighbor interpolation. For semantic class c, we denote its patch locations in image x by $\Omega_{x,c}$ and retain image–class pairs containing at least four valid patches.

Controlled response perturbation. We first estimate the global response magnitude of encoder channel j over the validation set as

$$s_j = \frac{1}{N}\sum_x \sqrt{\sum_{i=1}^{HW} r_{i,j}(x)^2}, \qquad j = 1, \ldots, 768, \tag{79}$$

where N is the number of validation images. For a perturbation ratio ρ, the channels with the largest $\mathrm{round}(768\rho)$ values of s_j are set to zero, producing the perturbed response $\mathbf{r}^{(\rho)}$. The remaining channels retain their original values from the unperturbed response $\mathbf{r}^{(0)}$. The perturbed response is then passed through the unchanged decoder to obtain $\mathbf{z}^{(\rho)}$; the encoder is not re-executed.

We consider

$$\rho \in \left\{\frac{1}{768}, \frac{2}{768}, \frac{4}{768}, \frac{8}{768}, 0.02, 0.05, 0.08, 0.10, 0.15, 0.20, 0.30, 0.40, 0.50\right\}. \tag{80}$$

The unperturbed case $\rho = 0$ serves only as the reference and is not shown as a scatter point.

Response–feature association. For each valid triplet (x, c, ρ), response and feature variations are measured over the same class-specific patch locations:

$$D_{\mathrm{resp}}(x, c, \rho) = \frac{\left\|\mathbf{r}^{(\rho)}[\Omega_{x,c}] - \mathbf{r}^{(0)}[\Omega_{x,c}]\right\|_F}{\left\|\mathbf{r}^{(0)}[\Omega_{x,c}]\right\|_F}, \tag{81}$$

and

$$D_{\mathrm{feat}}(x,c,\rho)=\frac{\left\|\mathbf{z}^{(\rho)}[\Omega_{x,c}]-\mathbf{z}^{(0)}[\Omega_{x,c}]\right\|_F}{\left\|\mathbf{z}^{(0)}[\Omega_{x,c}]\right\|_F}. \tag{82}$$

Thus, each scatter point in Fig. 2(a) corresponds to one image–class–perturbation triplet (x,c,ρ), rather than an individual pixel, a class pair, or a pair of training checkpoints. In total, 26,286 such triplets are evaluated. The horizontal axis measures the relative change in encoder responses, while the vertical axis measures the corresponding change in downstream decoder features.

Feature–distribution association. To quantify how feature variations are associated with distributional changes, we construct a fixed two-dimensional PCA space from the unperturbed validation features. At most eight patches are sampled per image and class, and at most 200 patches are retained per class. The same projection is then used for all images and all perturbation ratios.

For each valid (x,c,ρ), the projected features of class c before and after perturbation are modeled as

$$\mathcal{N}_c^{(0)}=\mathcal{N}(\boldsymbol{\mu}_c^{(0)},\boldsymbol{\Sigma}_c^{(0)}),\qquad \mathcal{N}_c^{(\rho)}=\mathcal{N}(\boldsymbol{\mu}_c^{(\rho)},\boldsymbol{\Sigma}_c^{(\rho)}). \tag{83}$$

A diagonal ridge of 10^{-6} is added to the covariance for numerical stability. The distribution variation is measured using the Gaussian 2-Wasserstein distance:

$$\begin{aligned}W_2^2\left(\mathcal{N}_c^{(0)},\mathcal{N}_c^{(\rho)}\right)=&\left\|\boldsymbol{\mu}_c^{(\rho)}-\boldsymbol{\mu}_c^{(0)}\right\|_2^2\\&+\mathrm{Tr}\left[\boldsymbol{\Sigma}_c^{(0)}+\boldsymbol{\Sigma}_c^{(\rho)}-2\left((\boldsymbol{\Sigma}_c^{(0)})^{1/2}\boldsymbol{\Sigma}_c^{(\rho)}(\boldsymbol{\Sigma}_c^{(0)})^{1/2}\right)^{1/2}\right].\end{aligned} \tag{84}$$

Importantly, this quantity compares the *same semantic class* before and after perturbation. It is therefore $W_2(\mathcal{N}_c^{(0)},\mathcal{N}_c^{(\rho)})$, rather than a distance between two different classes c and c'. Accordingly, Fig. 2(b) plots $D_{\mathrm{feat}}(x,c,\rho)$ on the horizontal axis and the corresponding within-class distribution change on the vertical axis.

Correlation and visualization. For each semantic class c, we collect all valid image–perturbation observations (x,ρ). The correlation value shown in each subplot is the Pearson correlation coefficient

$$R_c=\frac{\sum_{m=1}^{M_c}(X_m-\bar{X})(Y_m-\bar{Y})}{\sqrt{\sum_{m=1}^{M_c}(X_m-\bar{X})^2}\sqrt{\sum_{m=1}^{M_c}(Y_m-\bar{Y})^2}}, \tag{85}$$

where M_c is the number of valid observations for class c. For Fig. 2(a), $X_m=D_{\mathrm{resp}}(x,c,\rho)$ and $Y_m=D_{\mathrm{feat}}(x,c,\rho)$. For Fig. 2(b), $X_m=D_{\mathrm{feat}}(x,c,\rho)$ and $Y_m=W_2(\mathcal{N}_c^{(0)},\mathcal{N}_c^{(\rho)})$.

Importantly, R_c is computed from the original unnormalized measurements. Min–max normalization to $[0,1]$ is applied only to the displayed axes and does not affect the reported correlation coefficient. When more than 500 observations are available, at most 500 points are randomly displayed to reduce visual clutter, whereas all valid observations are used to compute R_c. The smooth curve is obtained by quantile binning the horizontal variable, computing the median vertical value within each bin, and applying PCHIP interpolation; the shaded region represents the corresponding 25th–75th percentile interval.

Relation to incremental optimization. The fixed-model perturbation is intended to isolate one propagation component of representation evolution during incremental learning, rather than to reproduce the complete training trajectory. Let $r_t(x)$ denote an upstream response representation at incremental step t, and let $D_t(\cdot)$ denote the downstream mapping from this response representation to the feature representation, such that

$$z_t(x)=D_t(r_t(x)). \tag{86}$$

The feature change between two adjacent incremental states can be decomposed as

$$\begin{aligned}z_t-z_{t-1}=&\underbrace{D_t(r_t)-D_t(r_{t-1})}_{\text{variation induced by response change}}\\&+\underbrace{D_t(r_{t-1})-D_{t-1}(r_{t-1})}_{\text{variation induced by downstream parameter change}}.\end{aligned} \tag{87}$$

Therefore, feature evolution during incremental learning can arise from both changes in the upstream response and changes in the downstream mapping. The controlled perturbation analysis in Fig. 2 deliberately fixes the downstream mapping and evaluates the first component in isolation. This design prevents simultaneous parameter updates from confounding the measured effect of response variation.

For a differentiable downstream mapping, a local response variation Δr induces

$$\Delta z = D_t(r + \Delta r) - D_t(r) \approx J_{D_t}(r)\Delta r, \tag{88}$$

where $J_{D_t}(r)$ denotes the Jacobian of the downstream mapping with respect to the response representation. Whenever Δr contains a nonzero component along an effective direction of $J_{D_t}(r)$, the response variation is transmitted to the downstream feature representation. Fig. 2(a) empirically evaluates the magnitude and consistency of this transmission across semantic classes and perturbation levels.

The class-conditional distributions considered in our analysis are functions of the corresponding feature samples. For class c with N_c feature samples, its mean is

$$\mu^c = \frac{1}{N_c} \sum_{i=1}^{N_c} z_i^c. \tag{89}$$

Accordingly, feature variations Δz_i^c induce

$$\Delta\mu^c = \frac{1}{N_c} \sum_{i=1}^{N_c} \Delta z_i^c. \tag{90}$$

The covariance or feature-wise variance is likewise determined by the same feature samples. Therefore, changes in feature representations can modify both the location and dispersion of the corresponding class-conditional distribution. Fig. 2(b) quantitatively evaluates the resulting distributional change using the Gaussian 2-Wasserstein distance.

Interpretation. The consistently high positive correlations in Fig. 2(a) show that controlled variations in encoder responses are systematically reflected in downstream feature representations across the tested perturbation levels. Fig. 2(b) further shows that these feature variations are accompanied by changes in the corresponding class-conditional distributions. Because the same perturbation magnitude is applied across the measured levels, the observed correlations also reflect this shared perturbation factor; we therefore interpret them as a controlled diagnostic rather than a perturbation-independent estimate of coupling. Together with the structural relations above, the results provide empirical evidence for a cross-level propagation pathway from response representations to feature representations and subsequently to class-distribution geometry under controlled perturbations.

We emphasize that this diagnostic should not be interpreted as reproducing the complete optimization trajectory of incremental learning, nor as showing that all feature or distribution changes during training are caused exclusively by upstream response changes. During actual incremental optimization, representation changes and parameter updates occur jointly. The purpose of fixing model parameters is precisely to isolate one component of this process, namely

$$\Delta r \;\longrightarrow\; \Delta z \;\longrightarrow\; \Delta\mathcal{P}, \tag{91}$$

where Δr, Δz, and $\Delta\mathcal{P}$ denote variations in the response, feature, and class-conditional distribution representations, respectively.

The shared perturbation magnitude can contribute to the observed correlations; therefore, the correlation analysis is not interpreted as either a perturbation-independent coupling estimate or an independent causal proof of the complete incremental optimization process. Instead, it quantifies how controlled upstream representation changes are associated with downstream representation changes when other model factors are held fixed. Combined with the cross-step optimization analysis, which explains why representation adaptation is required when incremental objectives evolve, this controlled diagnostic motivates coordinated knowledge reorganization across the response, feature, and distribution levels in AKI.

C Additional Experiments

C.1 Performance comparison in the Disjoint setting

As shown in Tab. 5, in addition to the commonly explored overlap setting, we also conduct a quantitative analysis on the VOC dataset using an additional disjoint setting. In the overlap setting, training images may contain old, current, and future classes, while old and future classes are labeled as background. In the disjoint setting, the background excludes classes required for future steps. Considering all classes, our method achieves the best overall performance across all three incremental configurations under the disjoint setting. In cases with fewer incremental steps and fewer new classes to learn, our method demonstrates better learning of new classes. Specifically, in the 19-1 task, our method improves the new-class mIoU by 5.2 percentage points. In the 15-1 task, with more incremental steps, our method achieves a 4.0-percentage-point improvement in new-class mIoU. This demonstrates that our method is not only effective for the overlap setting mentioned in Tab. 1 in the main paper, but also excels in CISS under the disjoint setting, showing both resistance to forgetting old classes and adaptability to new ones.

Table 5: Performance comparison of the disjoint setting on the VOC dataset. Unlike the overlapping setting widely explored in Tab. 1 of the main paper, the disjoint setting contains only the current classes C_t and previously learned classes $C_{1:t-1}$, without future classes. The best and second-best results are shown in **bold** and underline, respectively. The † symbol denotes results reproduced using publicly released code. Our method achieves improvements of 5.2 and 4.0 percentage points in new-class mIoU over MBS under the 19–1 and 15–1 settings, respectively.

Method	Publication	Backbone	19-1 (2 steps)			15-5 (2 steps)			15-1 (6 steps)		
			Old	New	All	Old	New	All	Old	New	All
Joint	–	ViT	84.5	80.6	84.3	85.3	80.7	84.2	85.3	80.4	84.1
MiB Cermelli et al. (2020)	CVPR 2020	ResNet101	69.6	25.6	67.5	71.8	43.3	65.0	46.2	12.9	38.3
MiB† Cermelli et al. (2020)	CVPR 2020	ViT	80.6	45.2	78.9	75.0	59.9	71.4	66.7	26.3	57.1
SDR Michieli & Zanuttigh (2021)	CVPR 2021	ResNet101	69.9	37.3	68.3	73.5	47.3	67.3	59.2	12.9	48.2
PLOP Douillard et al. (2021)	CVPR 2021	ResNet101	75.4	38.9	73.7	71.0	42.8	64.3	57.9	13.7	47.4
RBC Zhao et al. (2022)	ECCV 2022	ResNet101	76.4	45.8	74.9	75.1	49.7	69.1	61.7	19.5	51.7
RBC† Zhao et al. (2022)	ECCV 2022	ViT	80.9	42.1	79.1	77.7	59.1	73.3	69.0	28.4	59.3
MBS† Park et al. (2024)	ECCV 2024	ViT	**84.4**	70.8	83.8	82.7	**68.6**	79.3	**81.0**	62.0	76.5
Ours	–	ViT	**84.4**	**76.0**	**84.0**	**82.8**	68.5	**79.4**	80.9	**66.0**	**77.4**

C.2 Compatibility with the ResNet101-based CISS architecture.

To further evaluate the generalizability of AKI beyond ViT-based models, we apply it to a mainstream CISS architecture using ResNet101 as the backbone. Specifically, we follow the ResNet101-based implementation of CoinSeg Zhang et al. (2023) and conduct experiments under the 19-1 setting with the same backbone configuration. In Tab. 6, *Origin* denotes the performance of the original ResNet101-based CoinSeg model, while *Origin+AKI* denotes the same model equipped with our proposed AKI framework. This comparison isolates the contribution of AKI without changing the underlying backbone.

As reported in Tab. 6, the original model achieves an mIoU of 43.5% on the newly introduced TV monitor class, whereas Origin+AKI improves it to 48.3%, yielding a 4.8 percentage-point gain. Averaged over all foreground classes and the background class, AKI further improves the overall mIoU from 73.9% to 74.2% and the mean class accuracy (mAcc) from 85.7% to 86.2%. For TV monitor, the higher mIoU despite lower class accuracy reflects the different sensitivities of the two metrics: class accuracy measures recall, whereas IoU jointly accounts for false positives and false negatives.

These results indicate that AKI is not tied to the ViT-based architecture used in the main experiments. Its response-, feature-, and distribution-level adaptation remains effective when incorporated into a ResNet101-based CISS model, supporting the compatibility of the proposed knowledge inheritance framework with a CNN-based configuration.

C.3 Compatibility with existing CISS frameworks.

To further evaluate whether AKI can be incorporated into existing CISS frameworks, we integrate it into CoinSeg Zhang et al. (2023) with a ViT backbone under the 19-1 setting while retaining the

Table 6: Performance comparison with and without AKI in a ResNet101-based CISS configuration under the 19-1 protocol. Incorporating AKI increases the overall mIoU from 73.9% to 74.2% and the mIoU of the newly introduced TV monitor class from 43.5% to 48.3%.

Class Name	Origin (ResNet101)		Origin + AKI (ResNet101)	
	mIoU	Acc	mIoU	Acc
Background	92.0	95.6	92.2	95.7
Aeroplane	90.2	95.8	89.1	96.1
Bicycle	38.3	89.2	37.1	90.5
Bird	91.2	95.9	91.5	96.3
Boat	68.6	88.5	68.0	90.2
Bottle	76.3	92.8	76.3	93.2
Bus	92.3	95.6	93.2	96.3
Car	86.2	92.8	85.0	93.1
Cat	95.1	98.6	93.5	98.0
Chair	31.5	40.8	33.2	43.1
Cow	84.1	91.2	83.6	90.7
Dining table	55.1	62.4	56.2	65.4
Dog	88.5	95.0	87.6	94.1
Horse	79.9	88.8	80.1	90.4
Motorbike	87.5	95.5	87.7	95.9
Person	88.0	91.1	87.6	91.5
Potted plant	53.4	68.9	55.3	72.7
Sheep	76.1	89.9	76.4	89.6
Sofa	45.8	53.8	48.0	57.0
Train	88.4	92.7	88.9	93.3
Tv monitor	43.5	85.8	48.3	77.0
Average	73.9	85.7	74.2	86.2

original learning pipeline. In Tab. 7, *Origin* denotes the original CoinSeg model, whereas *Origin+AKI* denotes the same framework equipped with our proposed AKI.

The results show that AKI substantially improves current-class adaptation without degrading the overall performance on previously learned categories. For the newly introduced TV monitor class, the mIoU increases from 38.6% to 58.8%, yielding a 20.2 percentage-point gain. Although its class accuracy decreases from 86.1% to 82.8%, this does not contradict the mIoU improvement, since class accuracy measures recall whereas IoU also penalizes false positives. Averaged over all foreground classes and the background class, AKI further improves the overall mIoU from 79.9% to 81.2% and the mean class accuracy (mAcc) from 88.9% to 89.5%. These results demonstrate that AKI can be effectively incorporated into an existing CISS framework, supporting the compatibility of its adaptive knowledge inheritance strategy.

We further evaluate AKI on the remote-sensing dataset iSAID as an additional cross-domain analysis. We follow the same backbone, input resolution, optimization strategy, and training schedule as in the main experiments, and adopt the 10-5 incremental setting on iSAID. Joint training, used only as an upper-bound reference, obtains mIoUs of 45.2% and 52.3% on old and new classes, respectively, whereas AKI achieves 44.6% and 49.3%. The small gaps to joint training indicate that AKI remains effective in a substantially different remote-sensing domain. This experiment is intended as an additional cross-domain evaluation, while comprehensive comparisons with existing CISS methods are conducted on Pascal VOC and ADE20K following standard CISS evaluation protocols.

C.4 Further Analysis of Response Capacity Adaptation

Role of response consistency. The consistency between complete and compact-support predictions should not be interpreted as an independent guarantee of historical preservation, because both predictions are produced by the current model and can evolve during incremental optimization. Instead, $\mathcal{L}_{\mathrm{RCA}}$ regularizes how the predictive function is distributed over the response space. If a prediction depends strongly on responses excluded from the compact support, masking these responses produces a larger discrepancy and therefore a stronger consistency penalty.

Table 7: Compatibility of AKI with an existing ViT-based CISS framework under the 19-1 setting. *Origin* denotes our reproduction of CoinSeg Zhang et al. (2023) with the same ViT backbone and training protocol, while *Origin+AKI* incorporates AKI into this reproduced baseline. AKI improves the newly introduced *TV monitor* class by 20.2 percentage points in mIoU.

Class Name	Origin (ViT)		Origin+AKI (ViT)	
	mIoU	Acc	mIoU	Acc
Background	93.0	96.3	94.0	97.2
Aeroplane	88.1	92.1	90.5	95.0
Bicycle	42.9	86.6	42.7	89.7
Bird	95.1	97.1	95.1	97.4
Boat	74.9	88.0	75.6	89.1
Bottle	85.3	94.6	85.1	94.8
Bus	95.0	97.3	95.6	97.5
Car	90.5	92.4	90.3	92.4
Cat	96.2	98.3	96.2	98.3
Chair	48.1	59.5	49.0	62.3
Cow	93.8	97.7	93.7	98.0
Dining table	58.4	60.1	59.1	61.0
Dog	92.9	97.7	92.6	97.8
Horse	91.7	94.6	92.2	95.2
Motorbike	92.4	95.9	92.5	96.1
Person	89.7	92.6	89.8	93.0
Potted plant	68.6	85.0	68.6	85.8
Sheep	93.2	95.3	92.5	95.6
Sofa	59.5	67.9	60.1	68.6
Train	90.8	92.2	90.9	92.4
TV monitor	38.6	86.1	58.8	82.8
Average	79.9	88.9	81.2	89.5

Historical preservation is achieved by the overall AKI objective, while RCA specifically regulates response-space dependence. RPA and DEP provide complementary regulation at the feature and distribution levels, respectively.

Empirical evidence for functional preservation and adaptation. We first isolate the effect of the compact response support before incremental adaptation. At the default $\tau_w = 0.1$, approximately 6% of response units are masked on average. Under the 19-1 setting, the RCA support mask is applied to the step-0 model and both branches are evaluated on the same previously learned classes without additional training. As shown in Tab. 8, the average mIoU decreases only from 82.7% for the complete response to 81.7% for the compact support, while the corresponding accuracy changes from 91.3% to 89.8%. Thus, replacing the complete response by the compact support causes only 1.0 and 1.5 percentage-point decreases in mIoU and accuracy, respectively.

The class-wise results provide a more detailed view. For 16 of the 19 observed classes, the mIoU difference between the complete and compact responses is no larger than 1.5 percentage points. Several classes, including Boat, Bus, Horse, and Train, even exhibit negligible positive changes after applying the compact support. Larger deviations occur only for a small number of classes, such as Chair and Sheep. A similar pattern is observed for accuracy: most classes remain close to the complete-response branch, while a few classes, including Chair and Potted plant, are more sensitive to response compaction. These results indicate that the compact support preserves most of the existing predictive function, while also confirming that it is not trivially identical to the complete response.

The component ablation further evaluates whether this functional preservation is accompanied by useful current adaptation. Under the 10-1 setting, introducing RCA increases new-class mIoU from 21.3% to 46.7%; under 15-1, it improves new-class mIoU from 38.3% to 42.4%. Combined with the limited degradation observed in Tab. 8, these results support the intended role of RCA: the compact support retains most previously acquired predictive behavior, while consistency regularization reduces excessive dependence on the complete response space and facilitates current adaptation.

Importantly, these observations do not imply that units outside the compact support are explicitly repurposed for new classes, nor that parameter norm is a complete indicator of unit importance. Rather, they show that the default compact-support perturbation preserves most of the learned predictive function while providing a nontrivial response-space perturbation, making compact-support consistency a meaningful mechanism for regulating response-space dependence during incremental learning.

Table 8: Functional preservation under compact response support. Applying the RCA support mask to the learned model causes only a minor performance change on previously learned classes, indicating that the retained support preserves most of the existing predictive function.

Class Name	Compact Support		Complete Response	
	mIoU	Acc	mIoU	Acc
Aeroplane	93.2	97.5	94.3	97.6
Bicycle	43.9	89.6	45.2	90.5
Bird	93.1	96.2	94.8	97.6
Boat	78.5	91.0	78.2	91.8
Bottle	86.4	95.4	87.7	96.2
Bus	92.8	95.5	92.7	95.5
Car	88.8	95.7	89.8	96.3
Cat	94.4	97.1	95.4	97.4
Chair	51.7	67.4	55.7	74.6
Cow	95.0	97.2	95.3	97.7
Dining table	62.6	65.1	63.9	66.5
Dog	93.2	97.1	94.3	97.9
Horse	92.7	95.2	92.6	94.9
Motorbike	85.9	91.5	86.7	91.7
Person	88.5	93.6	89.3	94.4
Potted plant	71.7	85.8	72.5	96.4
Sheep	91.4	93.9	94.4	96.6
Sofa	57.2	63.2	57.6	63.2
Train	90.8	98.3	90.6	98.3
TV monitor	Unseen class		Unseen class	
Average	81.7	89.8	82.7	91.3

C.5 Further Evaluation of AKI Component Effectiveness

To further examine the individual contribution and complementarity of the three AKI components, we extend the ablation study to the 10-1 and 15-1 settings, which represent different incremental horizons. Response Capacity Adaptation (RCA), Reusable Projection Alignment (RPA), and Distribution Entanglement Penalization (DEP) operate at the response, feature, and distribution levels, respectively. We evaluate both individual and pairwise configurations to determine whether these components provide redundant or complementary effects.

Individual contribution. As shown in Tab. 9, each component independently improves the overall mIoU over the baseline under both settings. Under 10-1, RCA, RPA, and DEP increase the overall mIoU from 43.4% to 58.9%, 59.6%, and 53.0%, respectively. RCA and RPA particularly improve new-class mIoU from 21.3% to 46.7% and 45.2%, suggesting that response-level adaptation and directional feature inheritance are beneficial for long-horizon incremental learning.

Under 15-1, the corresponding overall mIoUs increase from 66.2% to 72.4%, 70.4%, and 74.6%. DEP provides the strongest individual gain in this setting, improving new-class mIoU from 38.3% to 50.1%. RPA exhibits a different behavior: when used alone, it substantially improves old-class mIoU from 74.9% to 81.0%, while the new-class mIoU slightly decreases to 36.6%. These results indicate that the three components affect historical preservation and current adaptation differently rather than providing interchangeable improvements.

Complementarity of different representation levels. The pairwise configurations provide stronger evidence of complementarity. Under 10-1, RCA+RPA and RCA+DEP achieve 76.2% and 76.4%

Table 9: Extended ablation study of the AKI components across different incremental settings. RCA, RPA, and DEP provide complementary improvements under both long- and short-horizon incremental learning protocols.

Grp.	$\mathcal{L}_{\text{RCA}}$	$\mathcal{L}_{\text{RPA}}$	$\mathcal{L}_{\text{DEP}}$	10-1 (11 Steps)			15-1 (6 Steps)		
				0-10	11-20	All	0-15	16-20	All
1	✗	✗	✗	63.5	21.3	43.4	74.9	38.3	66.2
2	✓			69.9	46.7	58.9	81.8	42.4	72.4
3		✓		72.7	45.2	59.6	81.0	36.6	70.4
4			✓	68.7	35.7	53.0	82.2	50.1	74.6
5	✓	✓		81.1	70.8	76.2	82.8	64.0	78.3
6	✓		✓	81.1	71.2	76.4	83.3	66.0	79.2
7		✓	✓	71.2	49.6	60.9	81.2	39.9	71.4
8	✓	✓	✓	**81.8**	**72.6**	**77.4**	**83.4**	**66.2**	**79.3**

overall mIoU, respectively, substantially exceeding their corresponding single-component variants. The same pattern is observed under 15-1, where RCA+RPA reaches 78.3% and RCA+DEP reaches 79.2%. These large gains over the individual components indicate that regulating one representation level cannot fully substitute for optimization at another level.

The RPA+DEP configuration further highlights the role of response-level adaptation. Without RCA, its overall mIoU is only 60.9% under 10-1 and 71.4% under 15-1, considerably below the configurations containing RCA. This suggests that feature inheritance and distribution regulation alone cannot reproduce the improvement obtained when response-level adaptation is also considered.

Effect of combining all three components. The complete RCA+RPA+DEP configuration achieves the best results under both incremental protocols. Under 10-1, it reaches 81.8%, 72.6%, and 77.4% mIoU on old, new, and all classes, respectively, outperforming all pairwise variants. Compared with RCA+RPA, introducing DEP further improves the overall mIoU from 76.2% to 77.4%; compared with RCA+DEP, introducing RPA improves it from 76.4% to 77.4%.

The same tendency is observed under 15-1. The complete configuration achieves 83.4% old-class, 66.2% new-class, and 79.3% overall mIoU, again providing the best performance across all three metrics. Although the gain over RCA+DEP is relatively small in this shorter protocol (79.2% to 79.3% overall mIoU), the complete configuration remains consistently superior rather than favoring only one particular incremental setting.

Across the two protocols, the average overall mIoU of the complete configuration is 78.35%, compared with 77.80% for RCA+DEP and 77.25% for RCA+RPA. More importantly, the complete AKI simultaneously achieves the best old-, new-, and all-class performance under both settings. This consistent behavior supports the complementary roles of response-level adaptation, feature-level inheritance, and distribution-level regulation.

Overall, the ablation results show that RCA, RPA, and DEP provide distinct but complementary contributions. RCA regulates dependence on the complete response space, RPA preserves reusable historical semantics along selected feature directions, and DEP directly suppresses old–new distribution overlap. Their joint use enables coordinated knowledge reorganization across response, feature, and distribution levels, yielding a more consistent balance between historical preservation and current adaptation across different incremental horizons.

C.6 Further Analysis of Directional Selection in RPA

To determine whether the improvement of RPA arises from selecting appropriate semantic directions rather than merely weakening feature distillation through rank-one projection, we conduct controlled comparisons using the same RCA-equipped configuration with DEP disabled. All variants use identical pseudo supervision, candidate old-class regions, optimizer, data augmentation, and training schedule; only the feature-level inheritance objective is replaced.

Table 10: Controlled comparison of feature-level inheritance strategies under the 15-1 setting. All variants share the same RCA-equipped baseline with DEP disabled and use identical pseudo supervision and candidate old-class regions. MKD performs full-space feature matching, whereas CKD applies reliability-guided selective feature distillation. Random-R1 and RPA impose the same rank-one constraint.

Feature Inheritance	Constraint	Old	New	All
MKD	Full-space	82.1	48.4	74.1
CKD	Confidence-selected	82.0	52.8	75.0
Random-R1	Rank-1	81.4	54.0	74.9
$\boldsymbol{\delta}_c$ only	Rank-1	82.4	59.2	76.9
w/o $\boldsymbol{\Sigma}_{\text{ada}}$	Rank-1	82.6	62.1	77.7
Full RPA	Rank-1	**82.8**	**64.0**	**78.3**

Table 11: Gradient interaction and equal-update analysis at the final incremental step of the Pascal VOC 10–1 setting. All variants use the same best RCA checkpoint after full training, with DEP disabled and identical evaluation mini-batches. Gradients are computed with respect to the same shared feature parameters. For the one-step test, inheritance gradients are normalized to unit norm and updated with $\eta = 1 \times 10^{-4}$.

Feature Inheritance	$\|\mathbf{g}_{\text{inh}}\|_2/\|\mathbf{g}_{\text{cur}}\|_2$	Cosine	Conflict Ratio	$\Delta\mathcal{L}_{\text{new}}$	$\Delta\mathcal{L}_{\text{old}}$
MKD	434.0	0.036	0.549	$+0.41$	-1.18
CKD	40.4	0.038	0.541	$+0.39$	-1.12
Random-R1	52.6	0.061	0.418	$+0.18$	-0.72
$\boldsymbol{\delta}_c$ only	44.8	0.094	0.312	-0.12	-0.64
w/o $\boldsymbol{\Sigma}_{\text{ada}}$	41.7	0.112	0.261	-0.24	-0.58
Full RPA	**40.4**	**0.131**	**0.228**	$\mathbf{-0.38}$	$\mathbf{-0.51}$

Controlled inheritance strategies. We first clarify the distinction between the conventional distillation baselines. MKD performs feature matching over all candidate old-class locations:

$$\mathcal{L}_{\text{MKD}} = \frac{1}{|\Omega_{\text{old}}|} \sum_{i \in \Omega_{\text{old}}} \|\mathbf{z}_t(i) - \mathbf{z}_o(i)\|_2^2, \tag{92}$$

where Ω_{old} denotes the candidate old-class locations, and $\mathbf{z}_t(i)$ and $\mathbf{z}_o(i)$ are the current- and previous-model features, respectively.

CKD instead performs confidence-based selective feature distillation. For each candidate location i, let

$$\kappa_i = \max_{c \in \mathcal{C}_{\text{old}}} q_{t-1}^c(i), \qquad a_i = \mathbb{1}[\kappa_i \geq 0.7], \tag{93}$$

where $q_{t-1}^c(i)$ is the posterior probability predicted by the frozen previous model for old class c. CKD is then defined as

$$\mathcal{L}_{\text{CKD}} = \frac{\sum_{i \in \Omega_{\text{old}}} a_i \, \|\mathbf{z}_t(i) - \mathbf{z}_o(i)\|_2^2}{\sum_{i \in \Omega_{\text{old}}} a_i}. \tag{94}$$

Therefore, CKD is not obtained by multiplying MKD by a positive scalar. Instead, it changes the spatial support of the inheritance objective by retaining only locations whose previous-model prediction confidence exceeds the fixed threshold 0.7. Consequently, MKD and CKD can induce different gradient directions and therefore different cosine similarities and conflict ratios.

Evaluation criteria. Table 10 reports old-, new-, and all-class mIoU. The comparison between Random-R1 and RPA is particularly informative because both constrain exactly one feature direction. Their difference therefore isolates the effect of *which direction is selected* from the reduction in constraint dimensionality.

For the optimization analysis in Table 11, we use the best checkpoint obtained after completing training at the final incremental step (step 10) of the Pascal VOC 10–1 setting. All variants are evaluated using the same mini-batches with a per-GPU batch size of 6. Here, θ_f denotes the trainable

parameters of the shared feature extractor that produces $\mathbf{z}_t$ and is common to all inheritance strategies. For evaluation batch b, we compute

$$\mathbf{g}_{\mathrm{cur}}^{(b)} = \nabla_{\boldsymbol{\theta}_f} \mathcal{L}_{\mathrm{cur}}^{(b)}, \qquad \mathbf{g}_{\mathrm{inh}}^{(m,b)} = \nabla_{\boldsymbol{\theta}_f} \mathcal{L}_{\mathrm{inh}}^{(m,b)}, \tag{95}$$

where m indexes the inheritance strategy. The relative inheritance strength is measured by

$$r_g^{(m)} = \frac{1}{B} \sum_{b=1}^{B} \frac{\|\mathbf{g}_{\mathrm{inh}}^{(m,b)}\|_2}{\|\mathbf{g}_{\mathrm{cur}}^{(b)}\|_2}, \tag{96}$$

where B is the number of identical evaluation mini-batches used for all variants.

Directional compatibility is measured by the cosine similarity

$$\mathrm{Cos}^{(m,b)} = \frac{(\mathbf{g}_{\mathrm{cur}}^{(b)})^\top \mathbf{g}_{\mathrm{inh}}^{(m,b)}}{\|\mathbf{g}_{\mathrm{cur}}^{(b)}\|_2 \|\mathbf{g}_{\mathrm{inh}}^{(m,b)}\|_2}, \tag{97}$$

and the conflict ratio is

$$\mathrm{CR}^{(m)} = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}\left[(\mathbf{g}_{\mathrm{cur}}^{(b)})^\top \mathbf{g}_{\mathrm{inh}}^{(m,b)} < 0\right]. \tag{98}$$

Thus, r_g measures the relative magnitude of the unweighted inheritance gradient, whereas cosine similarity and conflict ratio characterize its directional interaction with current optimization. The reported gradient-norm ratio is a diagnostic quantity rather than a loss-balancing coefficient; all training objectives retain the unit coefficients defined in Eq. 4.

Equal-update one-step test. Because cosine similarity is invariant to gradient magnitude, we further control the inheritance-update strength explicitly. For each strategy m, its inheritance gradient is normalized to unit norm:

$$\widetilde{\mathbf{g}}_{\mathrm{inh}}^{(m,b)} = \frac{\mathbf{g}_{\mathrm{inh}}^{(m,b)}}{\|\mathbf{g}_{\mathrm{inh}}^{(m,b)}\|_2}. \tag{99}$$

The controlled one-step update is

$$\Delta\boldsymbol{\theta}_f^{(m,b)} = -\eta \left(\mathbf{g}_{\mathrm{cur}}^{(b)} + \widetilde{\mathbf{g}}_{\mathrm{inh}}^{(m,b)}\right), \tag{100}$$

where the current-task gradient is kept unchanged and the inheritance gradient is normalized to the same unit norm across variants. We use the same incremental learning rate as training, $\eta = 1 \times 10^{-4}$, for this one-step evaluation. We then measure

$$\Delta\mathcal{L}_{\mathrm{new}} = \mathcal{L}_{\mathrm{new}}(\boldsymbol{\theta}_f + \Delta\boldsymbol{\theta}_f) - \mathcal{L}_{\mathrm{new}}(\boldsymbol{\theta}_f), \tag{101}$$

and analogously $\Delta\mathcal{L}_{\mathrm{old}}$. Here, $\mathcal{L}_{\mathrm{new}}$ and $\mathcal{L}_{\mathrm{old}}$ denote the unscaled mean segmentation cross-entropy over new-class and old-class locations, respectively. New-class locations are supervised by the current-step annotations, while old-class locations use pseudo labels from the frozen previous model. A negative value therefore indicates that the corresponding objective decreases after the controlled update. All strategies use the same checkpoint and evaluation batches, so the comparison isolates differences in inheritance direction under matched inheritance-update magnitude.

Effect of directional selection. Table 10 first confirms that reducing the constrained feature subspace can itself increase plasticity. Replacing MKD with Random-R1 improves new-class mIoU from 48.4% to 54.0%. However, rank reduction alone does not explain the full improvement of RPA. Random-R1 and RPA impose the same rank-one constraint, whereas RPA improves old-, new-, and overall mIoU by 1.4, 10.0, and 3.4 percentage points, respectively. This comparison directly separates directional selection from constraint dimensionality.

The intermediate variants further examine the selection criterion. Using only $\boldsymbol{\delta}_c$ improves new-class and overall mIoU to 59.2% and 76.9%, respectively. Incorporating historical dispersion while excluding $\boldsymbol{\Sigma}_{\mathrm{ada}}$ further increases them to 62.1% and 77.7%. The complete RPA reaches 64.0% and 78.3%. These progressive gains indicate that class-specific historical discriminability, historical dispersion, and observed current adaptation provide complementary information for selecting reusable semantic directions.

Separating directional compatibility from inheritance strength. Table 11 provides complementary optimization-level evidence. MKD produces a much larger inheritance-to-current gradient ratio

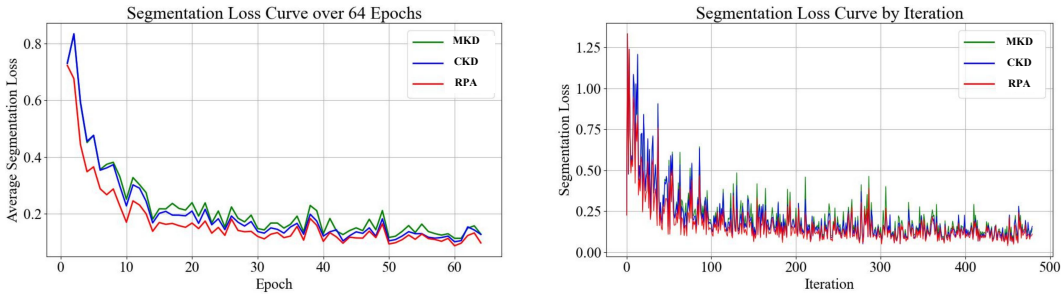


Figure 8: Optimization behavior of different feature-level inheritance strategies. With RCA fixed and DEP disabled, RPA yields a lower cross-entropy loss and a more stable optimization trajectory than the MKD- and CKD-based variants.

(434.0). Although cosine similarity and conflict ratio characterize directional interaction, they do not quantify the finite-step effect of inheritance-gradient magnitude.

A more informative comparison is obtained from CKD and RPA, which exhibit the same gradient-norm ratio of 40.4 under the unweighted diagnostic. Nevertheless, their directional behavior differs substantially: CKD achieves a cosine similarity of only 0.038 and a conflict ratio of 54.1%, whereas RPA reaches 0.131 and 22.8%, respectively. Since CKD changes the spatial support rather than merely rescaling MKD, these different conflict statistics are expected. More importantly, the comparison shows that the reduced conflict of RPA cannot be explained simply by a weaker inheritance gradient.

The rank-one variants provide an even stricter control of dimensionality. From Random-R1 to $\boldsymbol{\delta}_c$ only, then to RPA without $\boldsymbol{\Sigma}_{\text{ada}}$, and finally to full RPA, cosine similarity increases from 0.061 to 0.094, 0.112, and 0.131, while the conflict ratio decreases from 41.8% to 31.2%, 26.1%, and 22.8%. Because all four variants constrain exactly one feature direction, this consistent trend provides additional evidence that the direction selection criterion, rather than rank reduction alone, improves compatibility with current optimization.

Equal-update verification. The controlled one-step experiment removes differences in inheritance-gradient magnitude by normalizing all inheritance gradients to unit norm. Under this matched-magnitude setting, MKD and CKD increase the new-class loss by $+0.41$ and $+0.39$, respectively, and Random-R1 increases it by $+0.18$, although all three reduce the old-class loss. Their inheritance directions therefore favor historical preservation while partially restricting the current new-class objective.

In contrast, the direction-aware variants reduce both objectives. The new-class loss changes by -0.12 with $\boldsymbol{\delta}_c$ only, -0.24 without $\boldsymbol{\Sigma}_{\text{ada}}$, and -0.38 with the complete RPA, while full RPA also reduces the old-class loss by -0.51. Thus, even when inheritance-update magnitude is explicitly controlled, RPA produces a more favorable old–new optimization direction. Together with the matched-rank comparisons, this result shows that the benefit of RPA cannot be attributed solely to weaker or lower-dimensional feature inheritance.

C.7 Optimization Behavior under Different Inheritance Strategies

We further analyze how different feature-level inheritance strategies affect the optimization process. Following the controlled comparison above, RCA and DEP are kept unchanged, while the feature-level inheritance component is implemented using MKD, CKD, or the proposed RPA. We record the segmentation cross-entropy loss during step 1 of the 19-1 incremental setting, as shown in Fig. 8. The epoch-wise curves in the left panel show that all three variants rapidly reduce the segmentation loss during the early training stage, suggesting comparable initial convergence behavior. However, the variant equipped with RPA subsequently maintains a consistently lower cross-entropy loss than those using MKD or CKD. This difference indicates that restricting historical alignment to reusable semantic directions allows the current model to fit newly introduced supervision more effectively while still retaining the historical constraints provided by AKI. The iteration-wise curves in the right panel provide a finer-grained view of the optimization dynamics. All variants exhibit relatively large



Figure 9: Class-wise distribution overlap under progressive incorporation of AKI components. Pairwise Bhattacharyya coefficients (BC) are computed in the two-dimensional PCA space using all test features without outlier removal, where lower values indicate better distribution separation. The mean BC progressively decreases from 0.354 for the baseline to 0.222 with RCA, 0.214 with RCA+RPA, and 0.193 with the complete RCA+RPA+DEP configuration.

fluctuations during the initial iterations, which is expected when the model begins adapting to newly introduced supervision. As training proceeds, the RPA-based variant enters a lower-loss region and exhibits a comparatively stable optimization trajectory, whereas the MKD- and CKD-based variants remain at higher loss levels with more pronounced fluctuations. Importantly, these trajectories are obtained under the same unit-coefficient setting used in Eq. (4), indicating that the observed gradient-scale difference does not require additional loss reweighting for stable optimization.

C.8 Further Analysis of Inter-Class Distribution Overlap

To examine the class-distribution organization learned by different AKI configurations, we conduct a post-training analysis on the test set. For each configuration, we extract decoder features from the trained model while keeping its parameters fixed. The test features are projected into a two-dimensional PCA space, where class-conditional distributions are estimated and their pairwise Bhattacharyya coefficients (BC) are computed. All test feature samples are retained without outlier removal. A larger BC indicates stronger overlap between the estimated distributions in this projected space.

Table 12: Ablation study of the response threshold τ_w under the long-horizon 10-1 and short-horizon 19-1 settings. We report old-class, new-class, and overall mIoU to examine the robustness of RCA to different degrees of response compaction. Moderate thresholds yield stable performance in both settings, whereas increasingly large thresholds induce stronger response compaction and corresponding performance degradation. We therefore adopt $\tau_w = 0.1$ for all experiments as a moderate operating point within the stable region.

	10-1						**19-1**					
	$\tau_w = 0.05$	$\tau_w = 0.1$	$\tau_w = 0.15$	$\tau_w = 0.2$	$\tau_w = 0.3$	$\tau_w = 0.4$	$\tau_w = 0.05$	$\tau_w = 0.1$	$\tau_w = 0.15$	$\tau_w = 0.2$	$\tau_w = 0.3$	$\tau_w = 0.4$
Old	83.1	81.8	81.3	81.6	81.1	76.9	83.1	82.7	82.6	82.4	82.4	1.1
New	70.5	72.6	72.9	72.1	64.5	49.4	65.7	74.1	72.8	72.3	71.9	3.1
All	77.1	77.4	77.3	77.1	73.2	63.8	82.3	82.3	82.1	81.9	81.9	1.2

This analysis compares the representations learned by different component configurations after training, rather than successive states along a single optimization trajectory. The PCA projection and overlap computation are used only for diagnostic evaluation, with no model updates. They are separate from DEP training, which operates on the original D-dimensional decoder features.

As shown in Fig. 9, the baseline exhibits substantial inter-class entanglement, with a mean BC of 0.354. Introducing RCA reduces the mean BC to 0.222, indicating that response-level adaptation already facilitates a considerable reorganization of the representation space. After further introducing RPA, the mean BC decreases to 0.214. This suggests that preserving historical knowledge only along reusable semantic directions further alleviates unnecessary coupling between class representations while retaining flexibility in the complementary feature components. Finally, incorporating DEP reduces the mean BC to 0.193, the lowest among all configurations. Compared with the baseline, the complete AKI framework reduces the mean inter-class overlap by 0.161.

The class-wise matrices provide further insight beyond the averaged metric. The baseline contains numerous high-overlap off-diagonal class pairs, indicating substantial entanglement among different semantic categories. RCA removes a considerable portion of these strong overlaps, while RPA further improves the organization of several remaining class pairs. DEP provides an additional distribution-level constraint by directly penalizing old–new distribution overlap and further suppresses residual high-overlap pairs. These progressive changes support the complementary roles of the three components: RCA adapts the response organization, RPA regulates feature-level inheritance, and DEP explicitly refines class distributions.

Relation to the distribution visualization in the main paper. The distribution visualization in the main paper and the overlap matrices in Fig. 9 provide complementary views of class organization. The former is intended to qualitatively illustrate the dominant geometry of class representations in a two-dimensional space. Since a small number of peripheral samples can disproportionately enlarge the displayed distributional extent after dimensionality reduction and compress the high-density regions in the visualization, we exclude the farthest 10% of feature points from each class when constructing this figure. The same trimming protocol is uniformly applied to all classes and all compared configurations, ensuring a consistent basis for visual comparison. This processing is used exclusively for visualization and is not involved in training or quantitative evaluation.

In contrast, Fig. 9 quantitatively measures class-distribution overlap in the two-dimensional PCA space using all test features without trimming. All projected feature points, including peripheral samples, are retained when estimating the class-conditional distributions and computing pairwise BC. Therefore, the estimated overlap reflects the full set of test features after projection. Accordingly, the two analyses provide complementary views of class organization: the main-paper visualization emphasizes the dominant geometric structure after removing peripheral samples, whereas the BC matrices quantitatively evaluate distribution overlap using all projected features. Thus, the visual separation and the corresponding BC values need not exhibit an exact one-to-one correspondence.

C.9 Compact-Support Construction in RCA

Interpretation of the response-capacity cue. For an output unit associated with weight row $\mathbf{w}_{t,l,i}^\top$, the weight-induced response to input $\mathbf{z}_{t,l-1}$ satisfies

$$\left|\mathbf{w}_{t,l,i}^\top \mathbf{z}_{t,l-1}\right| \leq \|\mathbf{w}_{t,l,i}\|_2 \|\mathbf{z}_{t,l-1}\|_2. \tag{102}$$

Table 13: Matched comparison of compact-support construction strategies under the Pascal VOC 10–1 and 19–1 settings. Random support masks exactly the same number of weight-induced response coordinates as RCA within each trainable linear mapping, while all other components and training settings remain unchanged.

	10–1			19–1		
Support construction	Old	New	All	Old	New	All
Random support	4.6	1.0	2.9	45.8	25.9	44.9
Response-capacity support	81.8	72.6	77.4	82.7	74.1	82.3

Under bounded input feature magnitudes, $\|\mathbf{w}_{t,l,i}\|_2$ provides a feature-independent upper-bound cue for the potential magnitude of the weight-induced response. It is not interpreted as a calibrated importance score across different layer types. RCA uses this cue to construct an auxiliary compact response support, whose predictive effect is regulated through consistency with the complete-response branch.

Implementation of the two RCA branches. The complete-response and compact-support branches are two forward configurations of the same current model f_t and share all trainable parameters. For a trainable linear mapping with weight $\mathbf{W}_{t,l}$ and optional bias $\mathbf{b}_{t,l}$, the complete-response branch uses

$$\mathbf{h}_{t,l} = \mathbf{W}_{t,l}\mathbf{z}_{t,l-1} + \mathbf{b}_{t,l}, \tag{103}$$

whereas the compact-support branch uses

$$\mathbf{h}^{M}_{t,l} = \mathbf{M}_{t,l} \odot (\mathbf{W}_{t,l}\mathbf{z}_{t,l-1}) + \mathbf{b}_{t,l}. \tag{104}$$

The binary mask therefore suppresses only the weight-induced response, while leaving the bias term unchanged. This operation is applied to the output coordinates of the trainable attention and MLP linear projections included in the response-capacity construction. Residual identity paths remain unchanged.

Both branches participate in back-propagation through the shared parameters, and neither prediction is detached when computing the consistency loss. Stochastic layers use synchronized random realizations across the two forward configurations so that their discrepancy reflects response masking rather than unrelated stochastic variation. The compact-support branch is used only during training. Suppressed units are neither permanently removed nor frozen and remain available in the complete-response branch. RCA thus regularizes dependence on the complete response space without structural pruning or explicit allocation of units to new classes.

Shared threshold and realized compaction. We apply the same threshold τ_w to all trainable linear mappings considered by RCA, without enforcing a predefined retention ratio. For threshold values $\tau_w \in \{0.05, 0.10, 0.15, 0.20, 0.30, 0.40, 0.50\}$, the corresponding masked-unit ratios, averaged over the 10–1 and 19–1 settings, are $\{4\%, 6\%, 7\%, 10\%, 26\%, 60\%, 82\%\}$, respectively. The default $\tau_w = 0.1$ therefore produces moderate compaction, suppressing approximately 6% of response units, while larger thresholds progressively increase the degree of compaction. This trend is consistent with the stable performance at moderate thresholds and the degradation under excessively strong compaction reported in Table 12.

Matched random-support control. To isolate the effect of response-capacity-based support construction from generic response masking and consistency regularization, we introduce a matched random-support variant under the Pascal VOC 10–1 and 19–1 settings, as shown in Table 13. At the beginning of each incremental step, we record the number of output coordinates suppressed by RCA within each trainable linear mapping at $\tau_w = 0.1$. The random variant then masks exactly the same number of coordinates in each mapping and keeps this random support fixed throughout that step, whereas the RCA support is recomputed during optimization. The compact-support consistency objective, RPA, DEP, and all other training settings remain unchanged.

Different random-mask seeds are used across incremental steps, producing different fixed support realizations for different steps rather than resampling the mask at each optimization iteration. These step-wise seeds define the random supports used within the incremental sequence and are not treated as independent repeated experimental runs. The stochastic operations of the complete-response and compact-support branches remain synchronized, as in the original RCA configuration. Replacing the

Table 14: Computational cost comparison with MBS under the same runtime environment. Runtime and memory are measured over complete training iterations on a single NVIDIA GeForce RTX 3090 GPU with batch size 2 and 512×512 inputs, with AMP enabled for both methods. This batch size is used only for efficiency profiling and is independent of the main training batch size. ↓ indicates lower is better.

Method	Params (M) ↓	GFLOPs ↓	Train Time ↓ (ms/iter)	Peak Memory ↓ (GiB)	Inference Time ↓ (ms/img)
MBS	102.41	103.15	171.4	9.42	15.0
AKI (Ours)	102.41	103.15	**121.3**	**6.40**	15.0

response-capacity-based support with matched random support causes substantial degradation under both settings. Under 10–1, old-, new-, and all-class mIoU decrease from 81.8%, 72.6%, and 77.4% to 4.6%, 1.0%, and 2.9%, respectively. Under 19–1, the corresponding performance decreases from 82.7%, 74.1%, and 82.3% to 45.8%, 25.9%, and 44.9%. These results show that the effectiveness of RCA does not arise merely from masking the same number of responses and enforcing prediction consistency. Instead, the response-capacity cue is critical for constructing a compact support that preserves effective responses during incremental adaptation.

Boundary cases of the compact response support. RCA does not enforce a predefined compaction ratio. If all response units satisfy $\|\mathbf{w}_{t,l,i}\|_2 \geq \tau_w$, the mask becomes all-one and the compact-support branch coincides with the complete-response branch, yielding $\mathcal{L}_{\mathrm{RCA}} = 0$. Conversely, an excessively large threshold could in principle produce a highly sparse or even empty support for an individual mapping. In our evaluated range, however, no trainable linear mapping exhibits empty support at $\tau_w = 0.40$, despite an average masked-unit ratio of 60%. Therefore, the severe performance degradation at this threshold cannot be attributed to complete removal of all weight-induced responses in any layer, but is associated with excessively strong response compaction. Moreover, the bias terms and residual identity paths remain unchanged in the compact-support branch, so strong response masking does not imply an all-zero network representation.

C.10 Computational Efficiency

Computational efficiency. We further evaluate the computational cost of AKI in terms of parameter size, GFLOPs, end-to-end training iteration time, peak GPU memory, and inference latency. All measurements are conducted on a single NVIDIA GeForce RTX 3090 GPU with a profiling batch size of 2 and an input resolution of 512×512. Both MBS and AKI use the same underlying architecture and are evaluated under the same runtime environment using PyTorch 1.10.1 with CUDA 11.1 and automatic mixed precision (AMP) enabled.

The reported training time covers one complete optimization iteration, including the frozen previous-model forward pass, all current-model forward passes, auxiliary-statistic construction and feature decomposition, loss computation, back-propagation, and the optimizer update. Thus, for AKI, the measurement includes the additional compact-support forward pass and the RPA/DEP computations rather than timing only the main network forward pass.

As shown in Table 14, AKI introduces no additional deployed model parameters or inference-time computation compared with MBS. Specifically, both methods contain 102.41M parameters and require 103.15 GFLOPs, while their measured inference latency is identical at 15.0 ms per image. This is because RCA, RPA, and DEP are training-time knowledge-reorganization mechanisms and do not alter the deployed network architecture.

Under the matched architecture and runtime configuration, the measured end-to-end training iteration time decreases from 171.4 ms/iter for MBS to 121.3 ms/iter for AKI, corresponding to a reduction of approximately 29.2%. Meanwhile, the measured peak GPU memory decreases from 9.42 GiB to 6.40 GiB, corresponding to a reduction of approximately 32.1%. Since the network architecture and inference pipeline are identical, these differences reflect the training-time implementations of the respective learning objectives under the evaluated configuration.

These results indicate that the proposed knowledge-reorganization strategy does not increase deployment complexity and maintains favorable end-to-end training efficiency under the evaluated

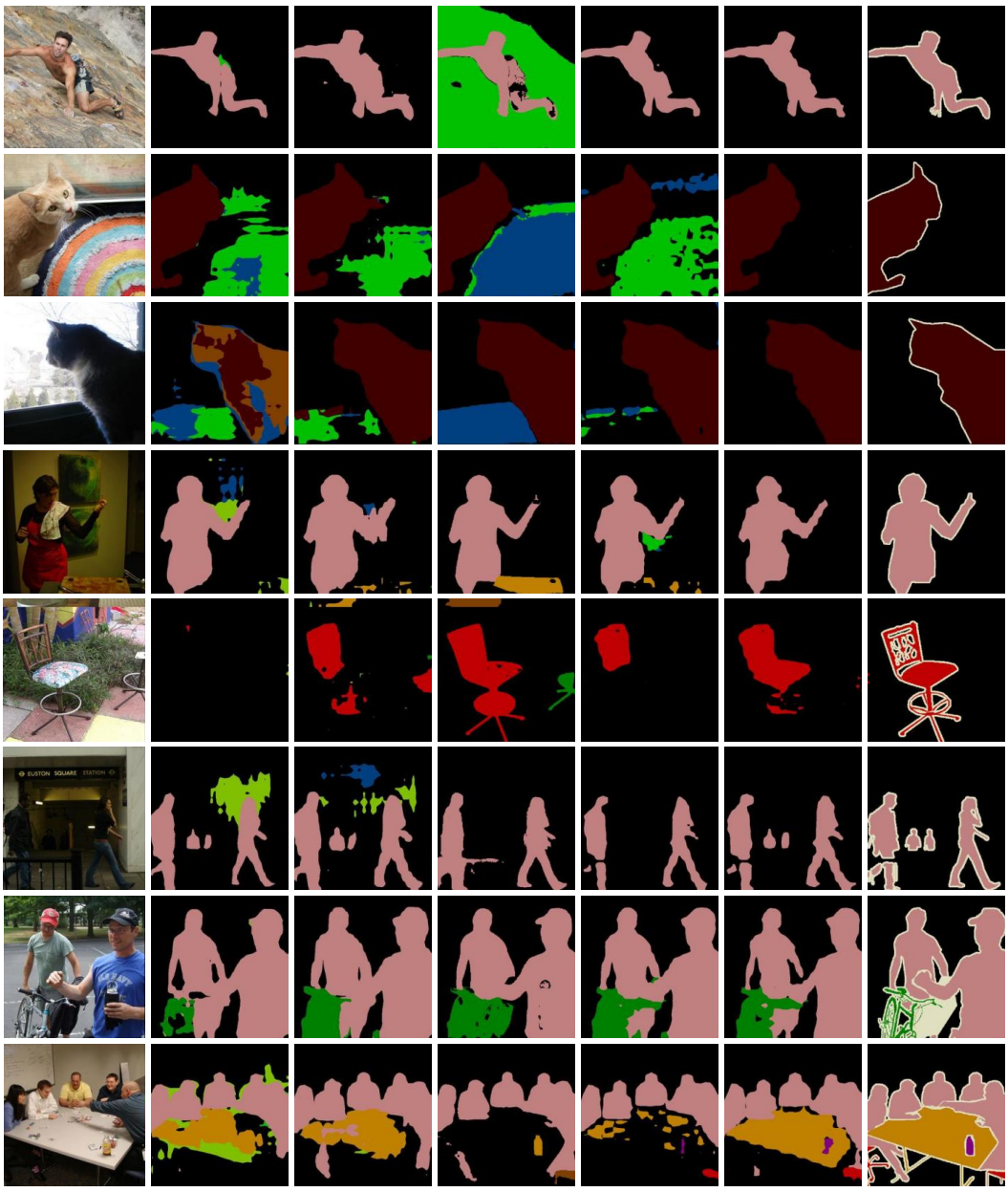


Figure 10: Qualitative analysis under the 15-1 setting. Our method better preserves old-class segmentation while reducing new-class misclassification, demonstrating improved stability and plasticity.

configuration. The reported runtime and memory results reflect the complete training pipeline; we do not attribute the measured cost to individual AKI components without separate operator-level profiling.

D MORE QUALITATIVE ANALYSIS

To further validate the effectiveness of our proposed method in improving pixel-level semantic classification accuracy, we present more qualitative comparisons with recent state-of-the-art approaches in Figs. 10-13 under the 15-1 setting. As illustrated in the second row of Fig. 10, existing methods

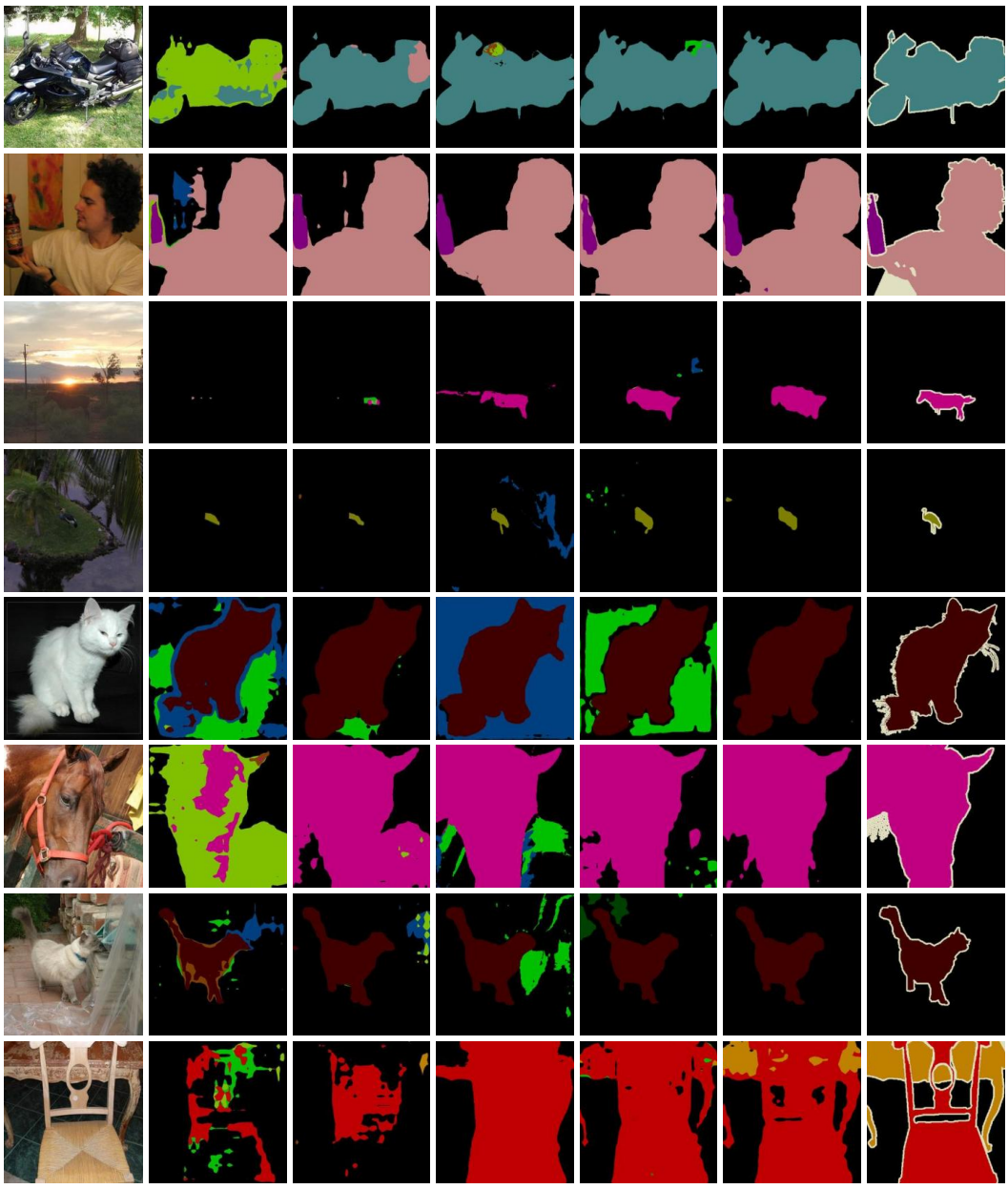


Figure 11: Qualitative analysis under the 15-1 setting. Our method better preserves old-class segmentation while reducing new-class misclassification, demonstrating improved stability and plasticity.

tend to confuse the background "cloth" with the "sofa" class after learning sofa-related knowledge in the incremental stage. In the third row, "windows" are mistakenly segmented as "TV monitor" or "sofa", indicating that these methods struggle to distinguish visually similar semantic categories when integrating new knowledge. Our method maintains the stability of previously learned classes while effectively acquiring new concepts, demonstrating improved plasticity in adapting to new classes. Fig. 11 provides additional results, where the first row shows a case of "motorcycle" being largely forgotten after five incremental steps. Some methods fail to retain the object's contour and misclassify it as other irrelevant categories. A similar confusion is more evident in Fig. 12 in the background of the first row. Our approach successfully preserves fine-grained class distinctions, significantly

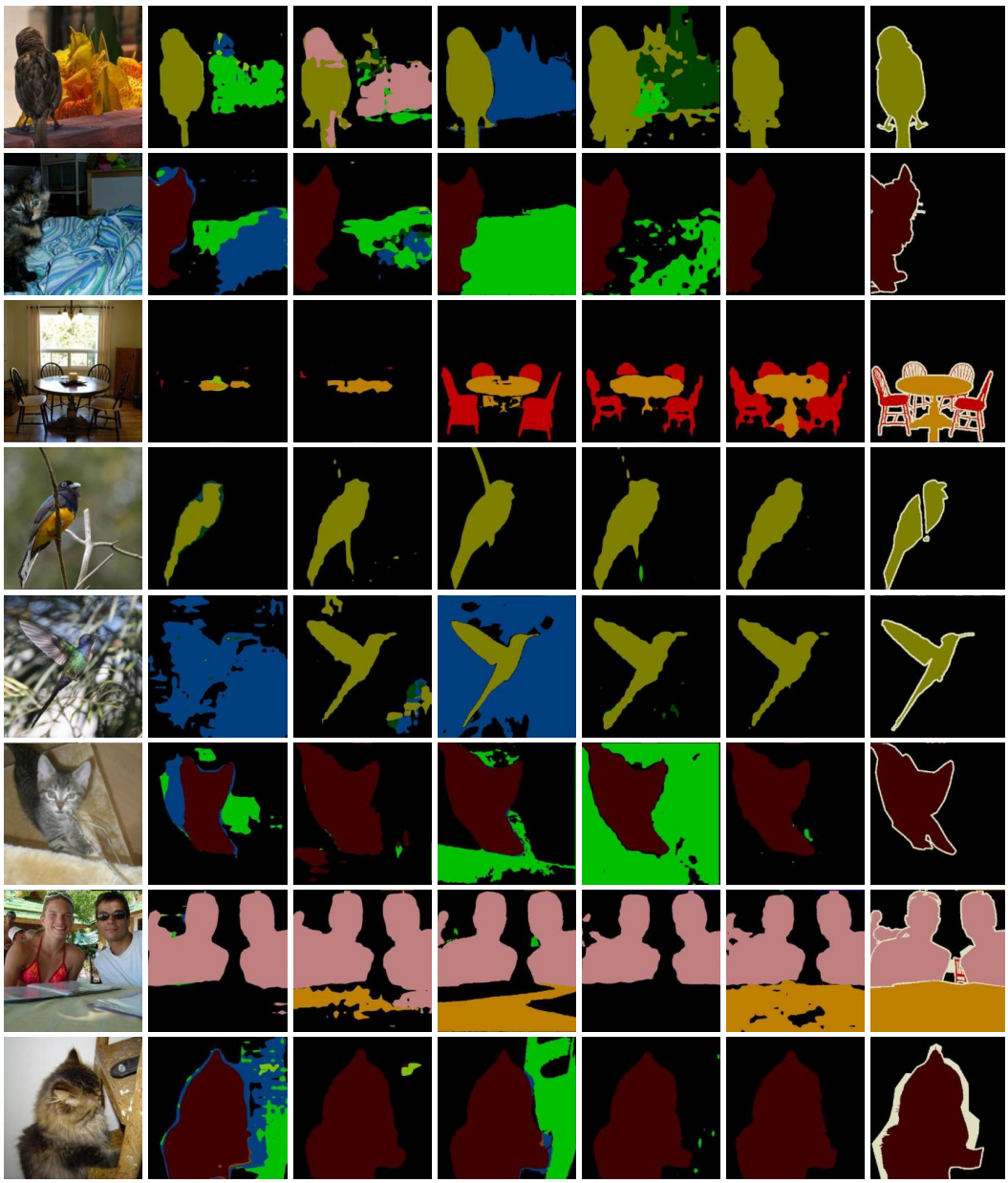


Figure 12: Qualitative analysis under the 15-1 setting. Our method better preserves old-class segmentation while reducing new-class misclassification, demonstrating improved stability and plasticity.

reducing pixel-level misclassification. In the final row of Fig. 13, existing methods confuse the "chair" occupied by a baby with a "sofa". Our method achieves much more precise segmentation, with only a small portion misclassified. These qualitative results clearly demonstrate that our approach substantially enhances pixel-wise classification accuracy in class-incremental semantic segmentation, while effectively mitigating class confusion during continual learning.

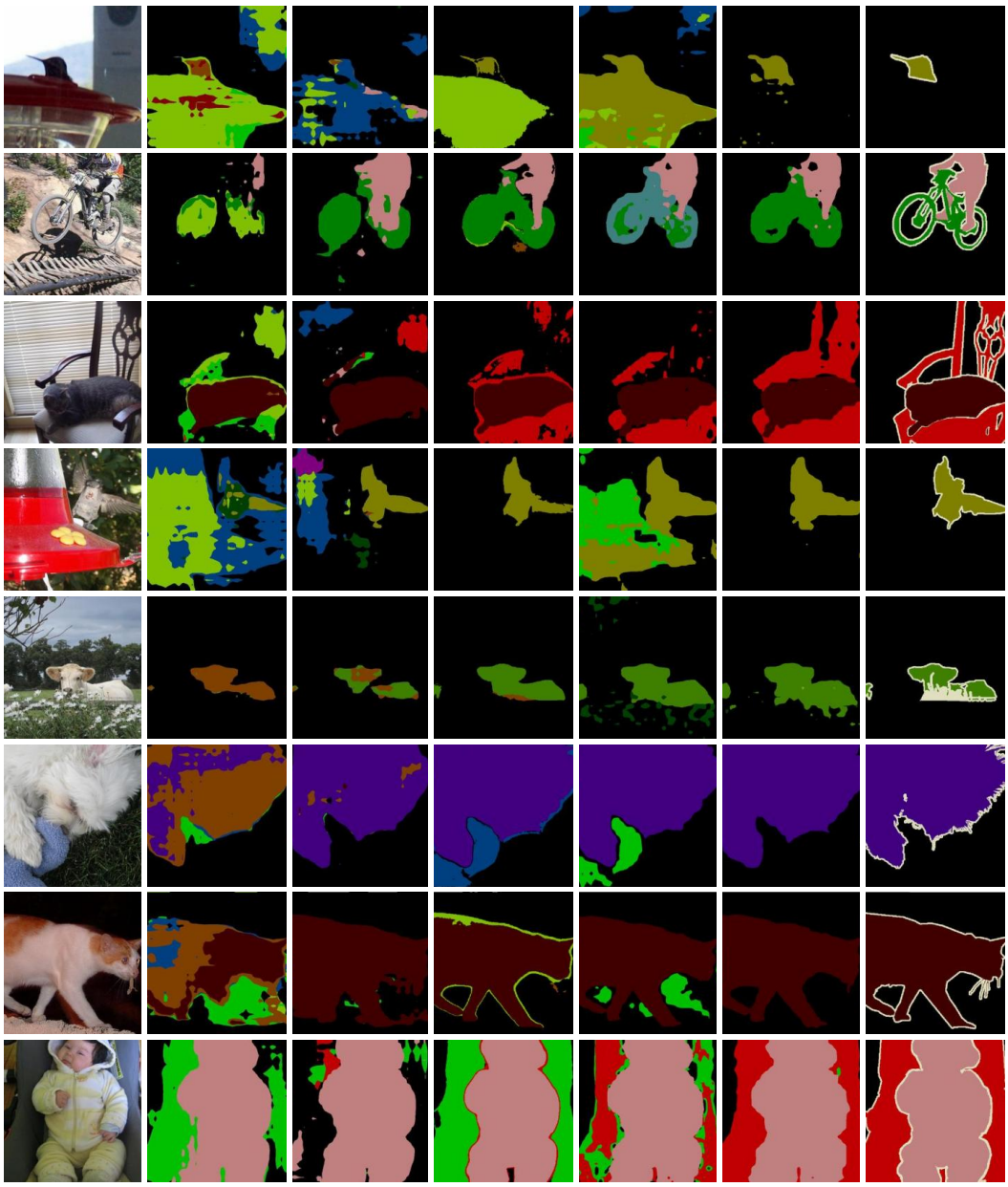


Figure 13: Qualitative analysis under the 15-1 setting. Our method better preserves old-class segmentation while reducing new-class misclassification, demonstrating improved stability and plasticity.

E DISCUSSION ABOUT LIMITATIONS AND FUTURE WORK

The present work focuses on fixed-capacity, replay-free CISS and investigates how knowledge inherited from the immediately preceding model can be adapted to the current incremental objective. This setting allows us to study cross-step optimization under a consistent parameter space and to coordinate knowledge inheritance at the response, feature, and distribution levels. Following mainstream CISS protocols, AKI uses the immediately preceding model f_{t-1} as the source of historical knowledge for optimizing f_t. A limitation of this formulation is that knowledge from earlier incremental steps is represented only implicitly through f_{t-1}, rather than being explicitly

distinguished and aggregated according to its step-wise origin. Consequently, the current framework does not explicitly model the long-range evolution of historical knowledge across multiple incremental stages.

An important future direction is therefore to extend AKI from adjacent-step inheritance to longer-range historical knowledge aggregation under the same fixed-capacity setting. Rather than relying only on the immediately preceding model, future work will investigate how information accumulated across multiple historical states can be selectively organized, evaluated, and incorporated into current optimization. In particular, historical knowledge from different incremental steps can exhibit different degrees of relevance to the current objective, making adaptive selection and aggregation across multiple states an important extension of the current inheritance mechanism. Such an extension will provide a broader formulation of adaptive knowledge inheritance while remaining consistent with the fixed-capacity assumptions underlying our cross-step optimization analysis.

F BROADER IMPACTS

Replay-free class-incremental semantic segmentation is relevant to real-world systems where historical data cannot be continuously stored or reused due to privacy, storage, or data-governance constraints. AKI can support continuously updated perception systems in applications such as autonomous driving, medical image analysis, remote sensing, and environmental monitoring. At the same time, errors accumulated during incremental learning may affect downstream decisions, especially in safety-critical scenarios. Therefore, practical deployment should include appropriate validation, monitoring, and domain-specific risk assessment.