EXEMPLAR-FREE CROSS-CATEGORY INCREMENTAL KEYPOINT DETECTION: WHEN NEW CATEGORIES BRING HETEROGENEOUS SCHEMAS

Anonymous authors
Paper under double-blind review

ABSTRACT

Object categories often have heterogeneous keypoint schemas differing in keypoint number, order, and semantics, yet keypoint detectors are typically trained for fixed categories with predefined schemas, making continual expansion to new categories challenging. Under the standard top-down setting, we study exemplar-free *Cross-Category Incremental Keypoint Detection* (CCIKD). At each incremental stage, the model learns only from new categories while preserving previous keypoint detection performance. This setting exposes a fundamental output-space challenge: assigning a separate identity to every category-keypoint causes unbounded output growth, while shared heads reuse identities through fixed local indices, causing *Output-Identity Collision* (OIC), where incompatible keypoints are forced to share parameters. During continual updates, such collisions cause semantic ambiguity, leading to greater forgetting of previously learned knowledge. CCIKD therefore requires preserving old knowledge while organizing an evolving output space under heterogeneous schemas. To address these challenges, we propose **POISE**, *prototype-guided Persistent Output Identity Space Expansion*, which builds visual–geometric prototypes and an append-only identity bank. POISE selectively reuses compatible identities, transfers related output parameters for initialization, and expands the bank for unmatched keypoints, while decoupling output-identity organization from preservation objectives. Experiments on IKDD and MP100 across diverse orders, protocols, and continual-learning strategies demonstrate improved performance and reduced forgetting, with increasingly larger gains on longer and more heterogeneous streams. In particular, with EWC on MP100-8stage, POISE boosts FinalPCK from 45.51 to 70.66.

1 INTRODUCTION

Keypoint detectors are typically trained on fixed categories with predefined keypoint schemas (Toshev & Szegedy, 2014; Newell et al., 2016; Zou et al., 2019). Under the standard top-down pipeline (Sun et al., 2019; Xu et al., 2022b), an upstream detector localizes an instance and provides its category, followed by category-specific keypoint prediction. This static paradigm becomes problematic as new categories with heterogeneous schemas arrive: without old data, joint retraining is unavailable, while per-category models incur growing storage and maintenance costs. Existing incremental-learning methods for classification (Tong et al., 2025), detection (Wang et al., 2025), and segmentation (Zhang et al., 2025) address catastrophic forgetting when output identities are explicit. Keypoint detection is further complicated by heterogeneous schemas, where cross-category output identities are not predefined and must evolve as new schemas arrive. Thus, incremental keypoint detection must jointly preserve old knowledge and organize an evolving output space.

Existing incremental keypoint methods address restricted evolution scenarios, typically evolving one dimension while constraining the other (Fig. 1). Incremental animal pose estimation (Nayak et al., 2021) expands the category space under an aligned keypoint space, whereas IKL (Liang et al., 2025) and PoseAdapt (Khan & Stricker, 2026) expand keypoint schemas within a fixed category or semantic domain. These paradigms therefore evolve either the category space or the keypoint schema, leaving their joint evolution in a single model unexplored. CCIKD studies this setting, where a single model continually incorporates new categories with heterogeneous schemas.

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

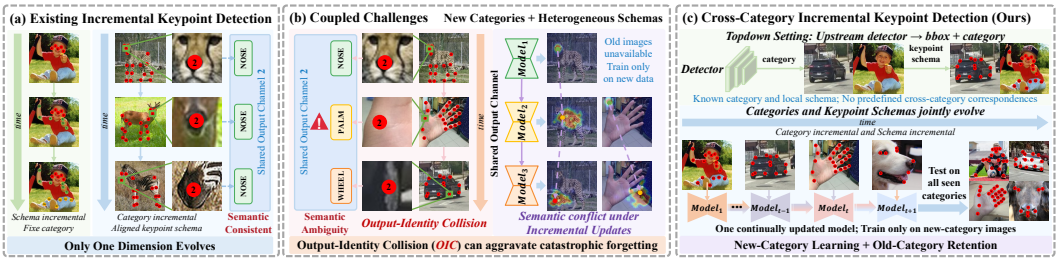


Figure 1: **CCIKD and its coupled challenges.** (a) Existing incremental keypoint settings evolve only one dimension: they either introduce new categories with aligned schemas or expand schemas within a fixed category. (b) A shared head binds semantically different keypoints with the same local index to one output row, causing *Output-Identity Collision* (OIC) during updates and semantic conflicts under shared parameters. (c) CCIKD introduces new categories with heterogeneous schemas in a top-down setting, where one model learns only from new-category data and evaluates all seen categories. **Output-identity organization and knowledge retention are coupled challenges.**

Motivated by this gap, we formulate *Cross-Category Incremental Keypoint Detection* (CCIKD) within the standard top-down pipeline in an exemplar-free setting (Fig. 1). At incremental stage t, the model learns new categories $\mathcal{C}_t$ using their training data, without old-category data or cross-category keypoint correspondences, while preserving keypoint detection on previously learned categories. This exposes a fundamental output-space design problem: per-category readouts avoid cross-category sharing but grow with the stream and do not reuse output parameters, whereas a fixed Shared-$K_{\max}$ head binds category-keypoints to output rows by local keypoint order. It can therefore force keypoints with low cross-category compatibility to share readout parameters, causing *Output-Identity Collision* (OIC). Meanwhile, training on new-category data leaves shared representations vulnerable to catastrophic forgetting. CCIKD therefore requires both preserving old-category knowledge and organizing an evolving output-identity space as heterogeneous schemas arrive.

Building on this diagnosis, we propose **POISE**, *prototype-guided Persistent Output Identity Space Expansion*, which organizes outputs by cross-category keypoint compatibility rather than category-local indices. POISE represents each category-keypoint with a visual–geometric prototype and matches new category-keypoints to persistent identities. Keypoints with sufficiently high prototype compatibility reuse existing identities, whereas others are assigned new identities to expand the output space beyond fixed local-index sharing. For related keypoints not assigned for permanent sharing, POISE transfers output parameters to initialize new identities for independent learning. As new categories arrive, existing assignments remain fixed while the identity space expands to accommodate heterogeneous schemas. POISE thus decouples *persistent output-identity construction* from *functional preservation*: it organizes evolving category-keypoint identities, enabling continual-learning strategies to preserve prediction functions under persistent correspondence.

We evaluate POISE on IKDD and MP100 across diverse incremental orders and protocols, ranging from relatively aligned schemas to long, heterogeneous category streams. Across continual-learning strategies, POISE improves continual performance and reduces forgetting, with larger gains on longer and more heterogeneous streams. For example, with EWC on MP100-8stage, POISE improves FinalPCK from 45.51 to 70.66, a gain of 25.15 percentage points. These results show that persistent output organization complements knowledge-preservation mechanisms for CCIKD.

Our main contributions are summarized as follows:

(1) We formulate CCIKD, an exemplar-free continual keypoint learning setting under the standard top-down pipeline, where new categories arrive sequentially with heterogeneous schemas. We identify its coupled challenges of *Output-Identity Collision* and catastrophic forgetting.

(2) We propose POISE, which uses visual–geometric prototypes to estimate cross-category keypoint compatibility and build a persistent output-identity space by selectively reusing existing identities, transferring related output parameters, and creating new identities.

(3) We decouple *persistent output-identity construction* from *functional preservation*, making POISE compatible with diverse continual-learning strategies. Experiments on IKDD and MP100 across orders, protocols, and backbones demonstrate broad improvements over Shared-$K_{\max}$, with larger gains on longer and more heterogeneous streams.

2 Related Work

Keypoint Detection. Keypoint detection has advanced through coordinate regression (Toshev & Szegedy, 2014), heatmap-based prediction (Newell et al., 2016), and coordinate classification (Li et al., 2022), alongside high-resolution (Sun et al., 2019) and Transformer-based architectures (Xu et al., 2022b). Most methods assume fixed categories and keypoint schemas. Category-Agnostic Pose Estimation (CAPE) (Xu et al., 2022a; Shi et al., 2023; Nguyen et al., 2024) generalizes to unseen categories using annotated support examples, while promptable methods (Yang et al., 2024; Rusanovsky et al., 2025; Kim et al., 2025; Lu et al., 2026) specify target keypoints through visual or textual prompts. These methods focus on support- or prompt-conditioned generalization across categories rather than sequential category-incremental learning. In contrast, CCIKD studies sequential learning of new categories with heterogeneous schemas while retaining performance on previously learned categories. Under the standard top-down setting, it requires a single model to continually learn from new-category data without access to historical images or test-time keypoint prompts.

Incremental Learning. Continual learning mitigates catastrophic forgetting through parameter regularization (Kirkpatrick et al., 2017), knowledge distillation (Li & Hoiem, 2016), and exemplar replay (Rebuffi et al., 2017). Parameter-efficient methods further reduce interference through prompt tuning (Wang et al., 2022b;a; Smith et al., 2023) and low-rank adaptation (Liang & Li, 2024). Vision–language continual learning (Radford et al., 2021; Zheng et al., 2023; Ni et al., 2024) additionally leverages pretrained semantic knowledge to preserve prior capabilities while learning new classes or tasks. However, most methods target classification (Tong et al., 2025), object detection (Wang et al., 2025), or segmentation (Zhang et al., 2025), with keypoint detection remaining less explored. Moreover, text-based methods rely on explicit semantics, whereas standard keypoint detection typically lacks natural-language descriptions for individual keypoints; promptable interfaces have only recently emerged (Yang et al., 2024; Kim et al., 2025). Existing incremental keypoint methods address restricted forms of evolution: they either add categories under an aligned keypoint schema (Nayak et al., 2021) or expand keypoint schemas within a fixed category or semantic domain (Liang et al., 2025; Khan & Stricker, 2026). Thus, prior paradigms typically evolve either the category space or the keypoint schema while constraining the other. CCIKD instead considers the joint continual evolution with heterogeneous category-specific keypoint schemas.

3 Understanding Output Identity in CCIKD

3.1 Problem Setup

We consider exemplar-free CCIKD over incremental stages $t = 1, \dots, T$ in the standard top-down setting. Let $\mathcal{C}_t$ and $\mathcal{D}_t$ denote the categories and training data introduced at stage t, with $\mathcal{C}_t \cap \mathcal{C}_s = \varnothing$ for $t \neq s$. Training at stage t accesses only $\mathcal{D}_t$, while previous data $\mathcal{D}_{<t} = \bigcup_{s<t} \mathcal{D}_s$ are unavailable. Each category c is introduced once with a complete category-specific keypoint schema $\mathcal{K}_c = \{k_{c,1}, k_{c,2}, \dots, k_{c,K_c}\}$ that remains fixed thereafter; the keypoint number and semantics may differ across categories. We define a persistent output identity as a cross-category correspondence among category-keypoints with compatible visual–geometric evidence. To handle heterogeneous schemas, we consider a Shared-$K_{\max}$ baseline with $K_{\max} = \max_c K_c$, which assigns each category-keypoint $k_{c,j}$ to output j. This mapping implicitly treats category-local indices as global output identities. To analyze cross-category compatibility, we construct a prototype $\mathbf{p}_{c,j}$ for each category-keypoint (c, j) from training observations and measure compatibility using a visual–geometric score $s((c_a, j), (c_b, k))$, where higher values indicate stronger prototype compatibility.

3.2 Can Local Index Define a Global Output Identity?

To test whether local index j defines a reliable global output identity, we identify the highest-scoring target keypoint for each ordered pair of distinct categories (c_a, c_b) and source keypoint (c_a, j):

$$k^\star(c_a, j, c_b) = \arg \max_{1 \le k \le K_{c_b}} s\big((c_a, j), (c_b, k)\big). \tag{1}$$

We retain only queries with an available same-index target ($j \le K_{c_b}$) and construct the heatmap

$$P_{j,k} = \Pr(k^\star(c_a, j, c_b) = k \mid j)\,, \tag{2}$$

where each row gives the empirical distribution of the highest-scoring target index k for a given source index j. A reliable global identity would concentrate $P_{j,k}$ along the diagonal $j = k$.

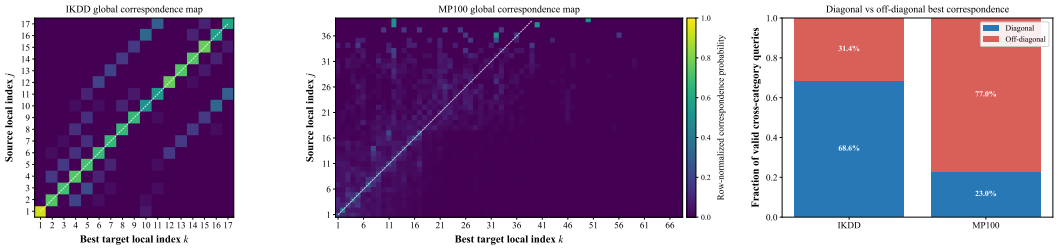


Figure 2: **Prototype compatibility as a proxy for keypoint identity consistency.** Row-normalized maps $P_{j,k}$ show the highest-compatibility target index k for each source index j. On IKDD with aligned keypoint schemas, diagonal patterns emerge without correspondence supervision, whereas MP100 exhibits diverse non-diagonal matches under heterogeneous schemas.

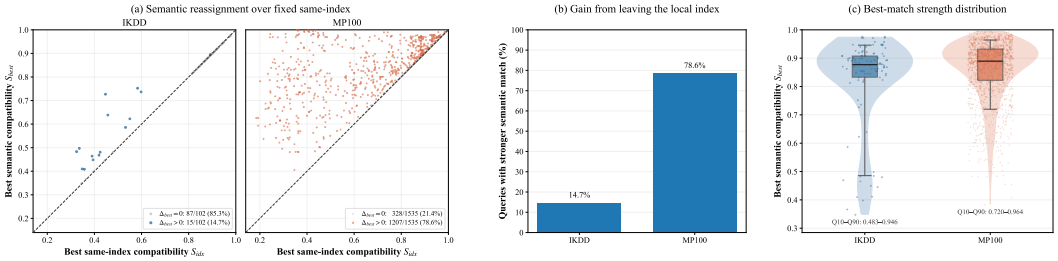


Figure 3: **Compatibility-based matching supports selective identity reuse.** (a) Strongest same-index versus unrestricted cross-category matches; points above the diagonal satisfy $\Delta_{\text{best}} > 0$. (b) Fraction of queries with a stronger match after relaxing the same-index constraint. (c) Distribution of unrestricted best-match strength S_{best}. Together, reassignment gain and match strength motivate selective reuse while keeping the output space expandable for weaker matches.

Figure 2 investigates whether local indices define reliable global output identities. Prototype compatibility serves as an evidence-based proxy for cross-category keypoint identity consistency. IKDD with manually aligned schemas shows a strong diagonal pattern, while heterogeneous MP100 yields 77.0% non-identical matches among queries with valid same-index candidates. **These results indicate that local indices reflect schema-specific ordering rather than stable cross-category identities, motivating compatibility-based persistent output identity organization.**

3.3 Is the Highest-Scoring Match Always Reusable?

The preceding results motivate compatibility-based matching beyond local indices. We examine the gain from relaxing the same-index constraint and the strength of the resulting match. For an evaluable query $q = (c, j)$ with a valid cross-category same-index candidate, we define

$$S_{\text{idx}}(q) = \max_{\substack{c' \neq c \\ K_{c'} \geq j}} s\big(q, (c', j)\big), \qquad S_{\text{best}}(q) = \max_{\substack{c' \neq c \\ 1 \leq k \leq K_{c'}}} s\big(q, (c', k)\big). \tag{3}$$

Here, both maxima are taken over candidates with valid prototypes. S_{idx} selects the strongest same-index match across other categories, whereas S_{best} searches all keypoints. Their difference $\Delta_{\text{best}}(q) = S_{\text{best}}(q) - S_{\text{idx}}(q) \geq 0$ measures the gain from leaving the local index.

Figure 3(a) visualizes the reassignment gain, while Fig. 3(b) reports the fraction of queries with $\Delta_{\text{best}} > 0$. Relaxing the same-index constraint yields a stronger match for 14.7% of IKDD and 78.6% of MP100 queries, showing that fixed indexing misses higher-compatibility alternatives, especially on MP100. However, the highest-scoring candidate may still be too weak for identity reuse. Figure 3(c) shows substantial variation in S_{best}, with Q10–Q90 ranges of 0.483–0.946 on IKDD and 0.720–0.964 on MP100. **These results motivate selective identity reuse based on match strength while keeping the output space expandable for weaker matches.**

3.4 How Does Output Conflict Accumulate During Continual Learning?

The preceding analyses reveal static mismatch; we now quantify its accumulation across stages. Let $\mathcal{C}_{<t} = \bigcup_{u<t} \mathcal{C}_u$ denote the old categories. Under Shared-$K_{\max}$, each new keypoint (c_n, j) shares

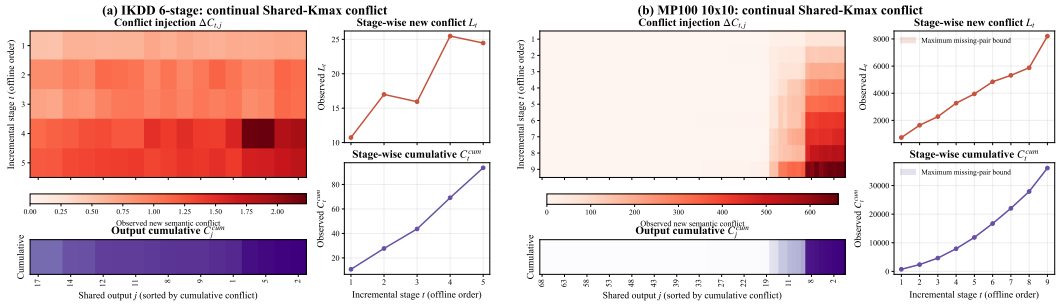


Figure 4: **Prototype-incompatibility load under Shared-$K_{\max}$.** Heatmaps show stage-wise load $\Delta C_{t,j}$, lower strips show output accumulation C_j^{cum}, and right plots show L_t and C_t^{cum}. Cumulative load grows by construction; mean load per forced pair enables cross-protocol comparison.

Table 1: **Forced-sharing incompatibility across continual protocols.** Final cumulative load sums forced new–old same-index relations and depends on category-pair count. Mean incompatibility per forced pair provides normalized cross-setting comparison. Values are averaged across orders.

Setting	Orders	Transitions with $L_t > 0$	Final cumulative conflict	Mean conflict per pair
IKDD 6-stage	3	15/15	93.60	0.367
MP100 10-stage	3	27/27	36,058.95	0.669
MP100 2-stage	5	5/5	16,884.78	0.669
MP100 superclass	3	21/21	28,583.33	0.758

output j with all old keypoints (c_o, j). We define the incompatibility load on output j at stage t as

$$\Delta C_{t,j} = \sum_{\substack{c_n \in \mathcal{C}_t \\ K_{c_n} \geq j}} \sum_{\substack{c_o \in \mathcal{C}_{<t} \\ K_{c_o} \geq j}} \left[1 - s\big((c_n, j), (c_o, j)\big)\right]. \tag{4}$$

Each term weights a forced new–old relation by its incompatibility under the prototype score. To summarize output-wise accumulation, stage-wise load, and cumulative load, respectively, we define

$$C_j^{\text{cum}} = \sum_u \Delta C_{u,j}, \qquad L_t = \sum_j \Delta C_{t,j}, \qquad C_t^{\text{cum}} = \sum_{u=1}^{t} L_u. \tag{5}$$

Figure 4 shows that forced local-index sharing accumulates greater incompatibility on MP100. Across 14 orders in Table 1, all transitions exhibit non-zero conflict load. Since cumulative load depends on pairwise incompatibility and the number of forced new–old assignments, the normalized mean load per forced pair provides a more informative comparison: MP100 achieves higher values than IKDD. This analysis characterizes the incompatibility introduced by fixed-index sharing, which becomes a source of continual interference under heterogeneous schemas, **motivating a persistent output space beyond local-index binding with prototype-guided reuse and expansion.**

4 METHOD

Motivated by the preceding analyses, we introduce **POISE**, *Prototype-Guided Persistent Output Identity Space Expansion*, a framework for incrementally organizing output identities under heterogeneous keypoint schemas. As shown in Fig. 5, Base Align constructs the initial identity space from training-only visual–geometric prototypes; subsequent category-keypoints either *Align* with historical identities or create new ones through *Transfer* or *Expand*. Established routes remain fixed, while new identities are appended as needed. POISE is compatible with diverse continual-learning strategies: it organizes the evolving output space, while the preservation strategy mitigates forgetting.

4.1 PROTOTYPE-GUIDED PERSISTENT IDENTITY ROUTING

Let $\mathcal{Q}_t = \{(c, j) \mid c \in \mathcal{C}_t,\ 1 \leq j \leq K_c\}$ denote the category-keypoints introduced at stage t, and let $\mathcal{Q}_{<t} = \bigcup_{s<t} \mathcal{Q}_s$ and $\mathcal{Q}_{\leq t} = \bigcup_{s \leq t} \mathcal{Q}_s$. POISE maintains an identity set $\mathcal{Z}_t = \{0, \ldots, M_t - 1\}$, a routing map $r_t : \mathcal{Q}_{\leq t} \to \mathcal{Z}_t$, and member sets $\mathcal{A}_z^t = \{q \mid r_t(q) = z\}$, subject to

$$\mathcal{Z}_{t-1} \subseteq \mathcal{Z}_t, \qquad r_t(q) = r_{t-1}(q)\ \ \forall q \in \mathcal{Q}_{<t}, \qquad r_t(c, j) \neq r_t(c, k)\ \ \forall (c, j), (c, k) \in \mathcal{Q}_{\leq t},\ j \neq k. \tag{6}$$

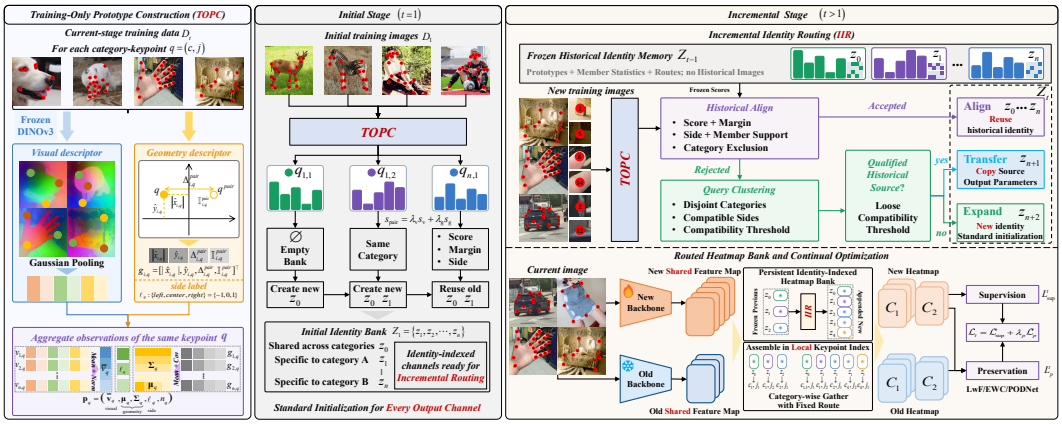


Figure 5: **Overview of POISE.** POISE first constructs training-only visual–geometric prototypes through TOPC and initializes the identity space at $t = 1$ via Base Align. For $t > 1$, frozen historical identity memory guides incremental routing: Historical Align reuses qualified identities, while rejected queries create new identities through Transfer or Expand. The routed heatmap bank keeps historical routes fixed and supports continual optimization with the selected preservation strategy.

These constraints make the identity space append-only, preserve historical assignments, and ensure category-wise injective routing. At $t = 1$, Base Align constructs the initial identity space from prototypes on $\mathcal{D}_1$; at $t > 1$, routing matches current-stage prototypes against the frozen previous-stage identity memory. Routes are fixed before optimization and remain unchanged during training, requiring no learnable routing module. POISE retains only aggregated prototypes and route metadata for routing, without historical images; the selected continual-learning strategy may maintain its own auxiliary state, such as a previous model, parameter-importance estimates, or adapter state.

Training-only prototype construction. For each visible occurrence of $q = (c, j)$, a frozen DINOv3 encoder (Siméoni et al., 2025) extracts a visual descriptor $\mathbf{v}_{i,q}$ via Gaussian-weighted pooling around its annotation. The geometry descriptor $\mathbf{g}_{i,q} = [|\widehat{x}_{i,q}|, \widehat{y}_{i,q}, \Delta^{\text{pair}}_{i,q}, \mathbb{I}^{\text{pair}}_{i,q}]^\top$ encodes normalized position, normalized horizontal left–right pair distance, and counterpart visibility. Absolute horizontal coordinates remove signs, while $\ell_q \in \{-1, 0, 1\}$ encodes left, center/unknown, and right.

Let $\mathcal{I}_q$ denote the visible training occurrences of q and $n_q = |\mathcal{I}_q|$. We aggregate them into

$$\mathbf{p}_q = \big(\underbrace{\overline{\mathbf{v}}_q}_{\text{visual}}, \underbrace{\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q}_{\text{geometry}}, \underbrace{\ell_q}_{\text{side}}, n_q\big), \ \overline{\mathbf{v}}_q = \text{norm}\Big(\sum_{i\in\mathcal{I}_q} \mathbf{v}_{i,q}\Big), \ (\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) = \text{MeanCov}\big(\{\mathbf{g}_{i,q}\}_{i\in\mathcal{I}_q}\big). \quad (7)$$

where MeanCov computes the mean $\boldsymbol{\mu}_q$ and regularized covariance $\boldsymbol{\Sigma}_q$ of geometry descriptors. Identity prototypes aggregate member visual and geometric statistics with sample-dependent weights, while side metadata is aggregated separately. Prototypes are computed before stage-wise optimization and serve as fixed routing evidence. No historical images or cross-category correspondence labels are required, and the frozen DINOv3 encoder is used only for prototype construction.

Prototype assignment and routing. POISE has three routing outcomes: *Align* reuses an established identity, *Transfer* creates a distinct new identity initialized from a related historical identity, and *Expand* creates a new identity from scratch. At stage $t > 1$, POISE first evaluates Align against frozen historical identities; rejected queries create new identities through Transfer or Expand.

For prototypes a and b, compatibility is defined as $s_{\text{pair}}(a, b) = \lambda_{\text{v}} s_{\text{v}}(a, b) + \lambda_{\text{g}} s_{\text{g}}(a, b)$, where s_{v} is raw visual cosine similarity, s_{g} is a side-weighted regularized bidirectional Mahalanobis similarity, and $\lambda_{\text{v}} + \lambda_{\text{g}} = 1$. The score ranges in $[-\lambda_{\text{v}}, 1]$. Side metadata adjusts the geometry term, while side-compatibility constraints apply during routing. For query q and historical identity z, $s(q, z)$ combines compatibility with the aggregate prototype $\mathbf{p}_z$ and support from best-matching members.

Base Align. At $t = 1$, Base Align constructs the initial identity space in canonical order. Each query considers identities from other categories; side metadata enters through the geometry term without a hard predicate. If the best candidate satisfies score and margin thresholds, it is aligned; otherwise, a new identity is initialized. Category exclusion preserves injectivity, and accepted assignments update summaries sequentially. The detailed predicate is provided in the appendix B.

Historical Align. At stage $t > 1$, candidates for query $q = (c, j)$ are identities in $\mathcal{Z}_{t-1}$ not already used by another keypoint of category c. Let $z^\star$ be the highest-scoring eligible identity and $m(q)$ its score gap over the runner-up. POISE routes q to $z^\star$ only if its score, margin, side compatibility, and evidence safeguards satisfy the Historical Align predicate. Within each category, POISE selects qualified queries by highest best-candidate score, with margin and canonical order as tie-breakers. The selected identity is then removed, and remaining candidates are updated accordingly. All compatibility values use the frozen previous-stage snapshot; accepted assignments affect availability but not snapshot scores. The complete predicate and tie-breaking rules are provided in the appendix B.

Transfer or Expand. Queries rejected by Historical Align begin as singleton clusters. POISE repeatedly merges the most compatible legal pair using the minimum cross-cluster s_{pair}; a merge requires disjoint categories, compatible sides, and compatibility of at least τ_B. Each final cluster creates one new identity. After category-wise occupancy filtering, each cluster member proposes its best eligible historical source. Among proposals satisfying the relaxed Transfer predicate, the highest-scoring one initializes the new identity from the source output parameters; if none qualifies, *Expand* uses standard initialization. A transferred identity remains distinct from its historical source. Complete linkage avoids weak transitive matches, while category disjointness preserves route injectivity. The detailed clustering and Transfer predicates are provided in the appendix B.

4.2 Routed Expandable Heatmap Bank

POISE expands only the final readout while retaining the backbone and heatmap tower of the underlying pose estimator. Let $\mathbf{U}_t(x)$ denote the D-channel feature map before the final readout. At stage t, the final convolution produces an identity-indexed bank of M_t heatmaps:

$$\mathbf{H}_t(x) = \mathbf{W}_t * \mathbf{U}_t(x) + \mathbf{b}_t, \qquad f_t(x, c) = \left[\mathbf{H}_{t, r_t(c,j)}(x)\right]_{j=1}^{K_c}. \tag{8}$$

Given category c, the routing map selects the K_c heatmaps indexed by $r_t(c, j)$ and assembles them in local order $j = 1, \ldots, K_c$. Each persistent identity is thus materialized as a global heatmap channel, while category-specific outputs remain independent of global channel order or contiguity.

When stage t is initialized from the stage-$(t-1)$ checkpoint, the first M_{t-1} output channels retain their indices. Historical Align reuses existing channels, whereas Transfer and Expand append new ones initialized from a selected historical source or standard initialization, respectively. Thus, growth is confined to the final readout, while the backbone and heatmap tower remain fixed-dimensional.

4.3 Continual Optimization

Persistent routing fixes identity correspondence, while parameter updates follow the selected continual-learning strategy. At stage t, POISE uses the generic objective

$$\mathcal{L}_t = \mathcal{L}_{\text{sup}}^t + \lambda_{\text{P}} \mathcal{L}_{\text{P}}^t, \tag{9}$$

where $\mathcal{L}_{\text{sup}}^t$ is the visibility-weighted heatmap loss on routed outputs for $\mathcal{D}_t$, and $\mathcal{L}_{\text{P}}^t$ is a strategy-dependent preservation term. It acts at different levels: LwF (Li & Hoiem, 2016) preserves outputs through distillation, EWC (Kirkpatrick et al., 2017) regularizes important parameters, and PODNet-style distillation (Douillard et al., 2020) preserves intermediate features. Parameter-efficient strategies update designated modules. All objectives share the same persistent output correspondence.

Routing determines which output rows category-keypoints use, while preservation controls prediction-function updates. By decoupling the two, POISE avoids fixed local-index binding and provides persistent correspondence for diverse continual-learning objectives.

5 Experiments

5.1 Datasets and Experimental Setup

Datasets and Evaluation Metrics. We evaluate POISE on our constructed IKDD benchmark and MP100 (Xu et al., 2022a). IKDD provides a controlled setting with six categories sharing a 17-keypoint schema, whereas MP100 evaluates heterogeneous schema evolution across 100 categories

Table 2: **Main results on IKDD and MP100.** Results (%) are averaged over category orders. Parenthesized values denote POISE gains over the corresponding Shared-$K_{\max}$ head for each continual-learning strategy. Red values indicate positive gains.

Method	Head	IKDD-6stage (3 orders)			MP100-2stage (5 orders)			MP100-10stage (3 orders)			MP100-8stage (3 orders)		
		AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
FT	Shared	81.39	74.75	-18.53	86.09	82.17	-11.37	73.01	64.04	-29.58	39.77	25.75	-66.05
	POISE	81.51 (+0.12)	75.56 (+0.81)	-17.56 (+0.97)	88.32 (+2.23)	86.19 (+4.02)	-6.20 (+5.17)	79.13 (+6.12)	73.62 (+9.58)	-18.97 (+10.61)	50.32 (+10.55)	36.00 (+10.25)	-55.99 (+10.06)
Joint	Shared	89.92	90.43	0.54	89.95	89.96	0.01	90.02	89.92	-0.24	88.85	89.90	-0.34
	POISE	89.97 (+0.05)	90.55 (+0.12)	0.59 (+0.05)	90.52 (+0.57)	90.60 (+0.64)	0.16 (+0.15)	90.53 (+0.51)	90.61 (+0.69)	0.11 (+0.35)	89.56 (+0.71)	90.45 (+0.55)	-0.34 (+0.00)
EWC	Shared	85.34	81.69	-10.37	88.79	87.65	-3.29	82.09	78.02	-12.94	62.28	45.51	-44.49
	POISE	87.24 (+1.90)	85.13 (+3.44)	-6.29 (+4.08)	90.16 (+1.37)	89.81 (+2.16)	-0.93 (+2.36)	87.25 (+5.16)	85.97 (+7.95)	-4.28 (+8.66)	77.40 (+15.12)	70.66 (+25.15)	-18.65 (+25.84)
LwF	Shared	86.04	83.15	-7.02	88.48	87.02	-4.25	77.02	69.36	-23.25	48.21	30.70	-60.88
	POISE	87.97 (+1.93)	86.19 (+3.04)	-4.22 (+2.80)	89.43 (+0.95)	88.43 (+1.41)	-2.93 (+1.32)	81.78 (+4.76)	75.07 (+5.71)	-17.21 (+6.04)	65.78 (+17.57)	47.44 (+16.74)	-44.25 (+16.63)
PODNet	Shared	77.28	68.79	-23.12	84.90	79.82	-14.17	69.02	57.40	-34.70	37.12	22.27	-67.85
	POISE	77.73 (+0.45)	70.54 (+1.75)	-20.98 (+2.14)	87.82 (+2.92)	85.20 (+5.38)	-7.14 (+7.03)	73.95 (+4.93)	66.33 (+8.93)	-24.87 (+9.83)	44.16 (+7.04)	31.05 (+8.78)	-59.53 (+8.32)

Table 3: **Parameter-efficient continual-learning results on IKDD and MP100.** Results (%) are averaged over category orders. Parenthesized values denote POISE gains over the corresponding Shared-$K_{\max}$ head for each PEFT method. Red values indicate positive gains.

Method	Head	IKDD-6stage (3 orders)			MP100-2stage (5 orders)			MP100-10stage (3 orders)			MP100-8stage (3 orders)		
		AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
L2P	Shared	60.94	51.45	-29.70	70.84	62.48	-24.68	55.76	45.17	-39.87	33.32	17.21	-65.33
	POISE	61.10 (+0.16)	53.47 (+2.02)	-27.32 (+2.38)	74.28 (+3.44)	68.27 (+5.79)	-17.63 (+7.05)	61.79 (+6.03)	54.93 (+9.76)	-29.55 (+10.32)	38.90 (+5.58)	24.79 (+7.58)	-58.50 (+6.83)
InfLoRA	Shared	69.75	62.46	-24.47	78.68	70.65	-23.09	63.60	52.93	-37.04	37.93	23.40	-64.71
	POISE	70.11 (+0.36)	63.48 (+1.02)	-23.24 (+1.23)	81.57 (+2.89)	76.09 (+5.44)	-15.73 (+7.36)	68.85 (+5.25)	62.18 (+9.25)	-26.82 (+10.22)	44.40 (+6.47)	31.70 (+8.30)	-56.68 (+8.03)

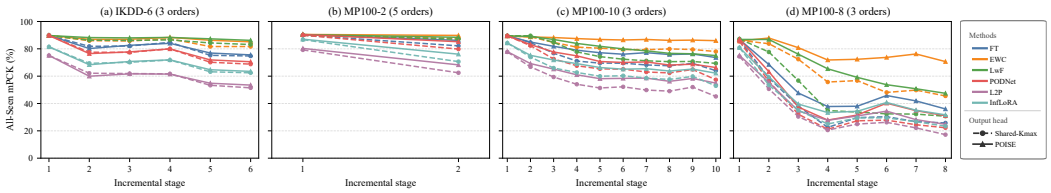


Figure 6: **Stage-wise All-Seen mPCK Results.** Curves are averaged over orders. POISE generally improves over Shared-$K_{\max}$, with larger gaps on the longer and more heterogeneous protocols.

with 8–68 keypoints. Under the standard top-down setting, we follow an exemplar-free protocol where each stage trains only on newly introduced categories and evaluates all seen categories. We consider four incremental protocols—IKDD-6stage, MP100-2stage, MP100-10stage, and MP100-8stage—under multiple category orders. Our primary metric is mPCK, averaged over PCK thresholds $\{0.05, 0.10, 0.15, 0.20\}$ and macro-averaged across categories. We report final All-Seen mPCK (*FinalPCK*), its stage average (*AIAPCK*), and backward transfer (*BWTPCK*). Further benchmark statistics, protocol details, category orders, and aggregation rules are provided in the appendix A.

Implementation Details. We compare fine-tuning (FT), EWC (Kirkpatrick et al., 2017), heatmap-level LwF (Li & Hoiem, 2016), and PODNet (Douillard et al., 2020). We further include L2P (Wang et al., 2022b) and InfLoRA (Liang & Li, 2024) to test compatibility with parameter-efficient continual learning. All strategies and the Joint reference use both the fixed local-index Shared-$K_{\max}$ head and the expandable POISE head. ViTPose (Xu et al., 2022b) is the main backbone, with HRNet (Sun et al., 2019) used for backbone ablation. Experimental details are provided in the appendix C.

5.2 Main Results

Table 2 compares multiple continual-learning strategies under the Shared-$K_{\max}$ and POISE heads. POISE improves AIAPCK, FinalPCK, and BWTPCK for FT, EWC, LwF, and PODNet on every protocol, yielding 48 positive order-averaged comparisons. The gains are generally larger on MP100 than on the aligned-schema IKDD and are particularly pronounced on the multi-stage MP100-8stage and MP100-10stage protocols. The stage-wise trajectories in Fig. 6 show the same pattern, with wider gaps between POISE and Shared-$K_{\max}$ on longer streams. Joint training with cumulative all-seen data avoids the accumulation of output-identity conflicts induced by incremental updates, highlighting the value of persistent output organization for exemplar-free incremental learning.

Table 3 shows that POISE improves L2P and InfLoRA across all reported metrics and protocols under the same-strategy comparison. The gains are generally larger on MP100, especially for the longer multi-stage protocols, where fixed output identities become more problematic. These results demonstrate that persistent output organization remains compatible with parameter-efficient continual-learning strategies and retains its benefits under restricted parameter updates.

Table 4: **HRNet backbone ablation.** Results are averaged over category orders. Parenthesized values denote POISE gains over the corresponding Shared-$K_{\max}$ head under the same continual-learning method. Red and green values indicate positive and negative gains, respectively.

Method	Head	IKDD-6-stage (3 orders)			MP100-2-stage (5 orders)			MP100-10-stage (3 orders)			MP100-8-stage (3 orders)		
		AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
FT	Shared	74.42	65.60	-25.35	86.20	83.20	-8.69	69.07	59.64	-31.77	37.23	22.77	-66.93
	POISE	73.90 (-0.52)	65.48 (-0.12)	-25.12 (+0.23)	88.32 (+2.12)	86.76 (+3.56)	-4.43 (+4.26)	75.08 (+6.01)	69.84 (+10.20)	-20.52 (+11.25)	43.86 (+6.63)	27.44 (+4.67)	-63.08 (+3.85)
Joint	Shared	87.59	89.07	1.53	89.37	89.54	0.29	88.79	89.43	0.56	88.22	89.43	-0.08
	POISE	87.66 (+0.07)	89.17 (+0.10)	1.53 (+0.00)	90.14 (+0.77)	90.42 (+0.88)	0.51 (+0.22)	89.49 (+0.70)	90.49 (+1.06)	1.12 (+0.56)	89.22 (+1.00)	90.27 (+0.84)	0.05 (+0.13)
EWC	Shared	76.93	69.02	-19.69	87.69	86.18	-3.30	75.63	70.93	-16.04	42.51	21.67	-64.10
	POISE	77.87 (+0.94)	72.62 (+3.60)	-15.33 (+4.36)	89.35 (+1.66)	88.82 (+2.64)	-0.49 (+2.81)	82.27 (+6.64)	80.61 (+9.68)	-5.56 (+10.48)	65.40 (+22.89)	54.92 (+33.25)	-31.37 (+32.73)
LwF	Shared	79.19	69.85	-18.77	88.08	86.96	-3.09	72.62	62.68	-28.69	39.48	24.36	-65.08
	POISE	79.31 (+0.12)	72.94 (+3.09)	-13.55 (+5.22)	89.29 (+1.21)	88.72 (+1.76)	-1.50 (+1.59)	73.38 (+0.76)	63.85 (+1.17)	-26.42 (+2.27)	47.25 (+7.77)	30.09 (+5.73)	-60.26 (+4.82)
PODNet	Shared	67.96	60.12	-19.15	86.15	83.10	-6.13	66.43	58.41	-20.34	36.99	22.90	-59.17
	POISE	71.89 (+3.93)	65.19 (+5.07)	-13.46 (+5.69)	88.32 (+2.17)	86.78 (+3.68)	-2.15 (+3.98)	73.20 (+6.77)	69.40 (+10.99)	-12.34 (+8.00)	48.79 (+11.80)	34.10 (+11.20)	-51.88 (+7.29)

Table 5: **Routing-action ablation.** Mean AIAPCK, final channel count, and head parameters on MP100-10stage. POISE balances accuracy and output growth.

Variant	AIAPCK↑	Channels↓	Final Output Params.↓
All-Align	78.96	111-111	28.527-28.527
All-Expand	**82.05**	184-1565	47.288-402.205
Align-Expand	79.76	111-825	28.527-212.025
Random	70.18	111-825	28.527-212.025
POISE	<u>81.78</u>	111-825	28.527-212.025

Table 6: **Prototype-evidence ablation.** AIAPCK on MP100-10stage. The complete POISE prototype outperforms all partial-evidence variants.

Variant	Visual only	Geometry only	w/o side comp.	POISE
AIAPCK↑	76.29	72.16	<u>78.25</u>	**81.78**

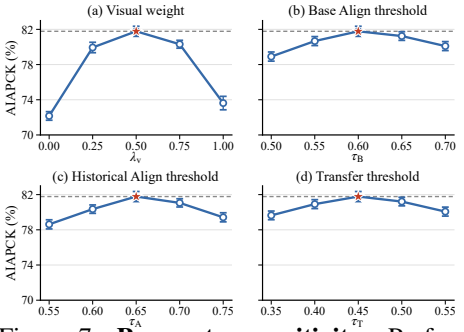


Figure 7: **Parameter sensitivity.** Performance is stable around the selected routing thresholds but degrades under extreme visual weights, highlighting the importance of balanced visual–geometric fusion.

5.3 Ablation Studies

Architecture. To evaluate architectural generality, we replace ViTPose with HRNet while retaining the same strategies and protocols. Table 4 shows that POISE maintains consistent improvements across continual-learning strategies and protocols, particularly on heterogeneous MP100 streams. Together with the ViTPose results, these observations show that POISE generalizes across backbone architectures and provides larger benefits as data streams become longer and more heterogeneous.

Routing and prototypes. Using ViTPose with LwF on MP100-10stage, Table 5 analyzes different routing choices. Prototype-guided POISE outperforms Matched-Random under the same output budget, demonstrating the benefit of compatibility-based allocation. All-Align and Align-Expand further validate the importance of selective reuse and Transfer initialization. All-Expand provides an unrestricted capacity reference by assigning separate identities to category-keypoints, achieving slightly higher accuracy with substantially more output channels and readout parameters. This reveals the accuracy–output-growth trade-off, while POISE achieves competitive performance through controlled identity expansion. Table 6 further shows that the complete visual–geometric–side prototype provides the strongest evidence combination. Figure 7 shows stable performance around the selected thresholds, highlighting the robustness of balanced prototype fusion.

6 Conclusion

We studied category-incremental keypoint detection with heterogeneous schemas, where category-local indices cannot provide stable cross-category output correspondence. We identified *Output-Identity Collision* (OIC) as a key challenge and proposed **POISE** to construct a persistent, prototype-guided output organization beyond fixed-index sharing. POISE enables selective identity reuse and controlled expansion of the output space while decoupling output correspondence construction from functional preservation. Across IKDD and MP100, POISE consistently improves performance over Shared-$K_{\max}$ across both full-parameter and parameter-efficient continual-learning strategies, with FinalPCK improving from 45.51 to 70.66 with EWC on MP100-8stage. Its compatibility with different backbones, including HRNet, demonstrates that prototype-guided output organization provides a reusable and scalable interface for continual keypoint learning under evolving schemas.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript writing and editing, as well as preliminary tasks including literature discovery, summarization, and organization. All AI-assisted content was carefully reviewed and revised by the authors, and relevant citations and factual claims were verified against original sources. The authors take full responsibility for the final content of this submission and all associated research artifacts.

REFERENCES

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pp. 86–102, 2020.

Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

Jinda Huang et al. Animal-pose: A novel large-scale dataset for animal pose estimation. In *CVPR*, 2020.

Muhammad Saif Ullah Khan and Didier Stricker. PoseAdapt: Sustainable human pose estimation via continual learning benchmarks and toolkit. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6840–6850, 2026.

Junho Kim, Hyungjin Chung, and Byung-Hoon Kim. CapeLLM: Support-free category-agnostic pose estimation with multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22889–22898, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. SimCC: A simple coordinate classification perspective for human pose estimation. In *European Conference on Computer Vision*, pp. 89–106, 2022.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pp. 614–629, 2016.

Mingfu Liang, Jiahuan Zhou, Xu Zou, and Ying Wu. Incremental object keypoint learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25399–25410, 2025.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Changsheng Lu, Yuxin Chen, Haokun Gui, Rong Wang, Jie Yang, Harry Yang, Anton van den Hengel, and Jiaya Jia. Gkdt: General keypoint detection transformer. In *European Conference on Computer Vision*. Springer, 2026.

Gaurav Kumar Nayak, Het Shah, and Anirban Chakraborty. Incremental learning for animal pose estimation using RBF k-DPP. In *British Machine Vision Conference*, 2021.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pp. 483–499, 2016.

Khoi Duc Nguyen, Chen Li, and Gim Hee Lee. ESCAPE: Encoding super-keypoints for category-agnostic pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23491–23500, 2024.

Bolin Ni, Hongbo Zhao, Chenghao Zhang, Ke Hu, Gaofeng Meng, Zhaoxiang Zhang, and Shiming Xiang. Enhancing visual continual learning with language-guided supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24068–24077, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

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

Matan Rusanovsky, Or Hirschorn, and Shai Avidan. CapeX: Category-agnostic pose estimation from textual point explanation. In *International Conference on Learning Representations*, 2025.

Min Shi, Zihao Huang, Xianzheng Ma, Xiaowei Hu, and Zhiguo Cao. Matching is not enough: A two-stage framework for category-agnostic pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7308–7317, 2023.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL `https://arxiv.org/abs/2508.10104`.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11909–11919, 2023.

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703, 2019.

Ruilin Tong, Yuhang Liu, Javen Qinfeng Shi, and Dong Gong. Coreset selection via reducible loss in continual learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Alexander Toshev and Christian Szegedy. DeepPose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, 2014.

Yanjie Wang, Liqun Chen, Tianming Zhao, Tao Zhang, Guodong Wang, Luxin Yan, Sheng Zhong, Jiahuan Zhou, and Xu Zou. High-dimension prototype is a better incremental object detection learner. In *International Conference on Learning Representations (ICLR)*, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pp. 631–648, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022b.

Lumin Xu, Sheng Jin, Wang Zeng, Wentao Liu, Chen Qian, Wanli Ouyang, Ping Luo, and Xiaogang Wang. Pose for everything: Towards category-agnostic pose estimation. In *European Conference on Computer Vision*, pp. 398–416, 2022a.

Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38571–38584, 2022b.

Jie Yang, Ailing Zeng, Ruimao Zhang, and Lei Zhang. X-Pose: Detecting any keypoints. In *European Conference on Computer Vision*, pp. 249–268, 2024.

Xinyue Zhang, Wanjia Luo, Guodong Wang, Sheng Zhong, Xu Zou, and Luxin Yan. Rethinking the influence of distribution adjustment in incremental semantic segmentation. *IEEE Transactions on Multimedia*, pp. 1–13, 2025.

Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19125–19136, 2023.

Xu Zou, Sheng Zhong, Luxin Yan, Xiangyun Zhao, Jiahuan Zhou, and Ying Wu. Learning robust facial landmark detection via hierarchical structured ensemble. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 141–150, 2019.

A Benchmark Construction and Evaluation Setup

This appendix details the benchmark construction, incremental protocols, evaluation metrics, and evaluation procedure used for CCIKD. We consider top-down keypoint estimation, where each instance is provided with an image crop, bounding box, and category label from an upstream detector. The model predicts only category-specific keypoints, so the reported results isolate incremental keypoint estimation without involving detector or category-recognition errors.

A.1 Benchmark Construction and Dataset Statistics

A.1.1 IKDD

IKDD is a six-category benchmark constructed in this work to provide a controlled low-heterogeneity setting for CCIKD. It integrates human pose data from COCO (Lin et al., 2014) and animal pose data from Animal-Pose (Huang et al., 2020) and PASCAL VOC (Everingham et al., 2015). To enable unified evaluation across object categories, we define a shared 17-keypoint vocabulary based on the human keypoint convention and manually establish corresponding animal keypoints with consistent semantic ordering. The resulting benchmark contains six categories, including person, cat, cow, dog, horse, and sheep, all represented with the same 17-keypoint schema. The detailed dataset statistics are summarized in Table 7.

Each category uses COCO-style annotations with bounding boxes, keypoint locations, and visibility labels. For incremental evaluation, each category contains 600 training instances and 200 held-out test instances. Unlike MP100, where category-specific schemas vary substantially, IKDD keeps the keypoint schema aligned while expanding only the category space. It therefore provides a controlled reference for studying incremental output organization when schema heterogeneity is limited.

Table 7: Dataset statistics used in our experiments.

Dataset	Categories	Keypoint Schema	Train Images / Instances	Test Images / Instances
IKDD	6	17 shared keypoints	2,900/3,600	940/1,200
MP100	100	8–68 category-specific keypoints	14,198/15,231	4,727/5,060

A.1.2 MP100-to-CCIKD Reconstruction

MP100 (Xu et al., 2022a) is a large-scale category-agnostic pose estimation dataset containing 100 object categories with category-specific keypoint schemas. Unlike conventional pose benchmarks that define a fixed keypoint vocabulary within a single object domain, MP100 provides category-dependent landmark definitions, resulting in substantial variations in keypoint number, ordering, and semantics. These properties make MP100 suitable for evaluating CCIKD, where new categories arrive with heterogeneous schemas and cross-category keypoint correspondences are not predefined.

As shown in Table 8, the original MP100 benchmark targets category-agnostic generalization, where training and evaluation categories are separated to measure transfer to unseen categories. In contrast, CCIKD studies sequential category acquisition and retention: each category requires both training and held-out examples, while the available training data are determined by the stage at which the category is introduced. Therefore, we reorganize MP100 into a unified incremental benchmark while preserving the original images, keypoint locations, visibility annotations, and category identities.

The conversion consists of four main steps, as summarized in Table 9. First, we preserve each category's original keypoint schema, including its local keypoint number and ordering. Second, we augment the schemas with within-category metadata, including side and flip information required for geometry-aware prototype construction in POISE. These metadata describe within-category geometric relations and do not provide cross-category keypoint correspondences. Third, we validate and canonicalize the annotations to ensure deterministic data loading across incremental protocols. Finally, we construct category arrival schedules to form different continual-learning streams.

After conversion, the benchmark contains 100 categories and 1,565 category-keypoint entries, with category-specific schemas ranging from 8 to 68 keypoints. Unlike the original category-agnostic setting, CCIKD provides no cross-category keypoint correspondences; each category only exposes its local schema. POISE therefore discovers compatible output identities through prototype-based routing rather than predefined semantic alignment.

Table 8: Differences between the original MP100 task and the reconstructed CCIKD benchmark.

Aspect	Original MP100	CCIKD reconstruction
Learning goal	Category-agnostic generalization	Sequential category acquisition and retention
Category split	Disjoint training and testing categories	Same categories with image-disjoint splits
Training access	Base-category training data	Current-stage categories only
Evaluation	Unseen categories	All categories observed so far

Table 9: Summary of MP100 conversion for CCIKD.

Operation	Description
Schema preservation	Retain category-specific keypoint numbers and local semantic ordering.
Metadata augmentation	Add within-category side and flip metadata for geometry-aware prototype construction.
Data validation and canonicalization	Filter invalid annotations while preserving original coordinates and visibility labels, and generate deterministic records for reproducible incremental training and evaluation.
Stream construction	Organize categories into incremental streams for different continual-learning protocols.

Category-stratified image split. Unlike the original MP100 protocol, where training and evaluation categories are disjoint, the reconstructed CCIKD benchmark uses the same category set for training and evaluation. Each category is first divided into training and held-out images, after which category arrival schedules are constructed for different incremental protocols. All instances from the same original image are assigned to the same split to avoid image-level leakage.

This fixed within-category image split is shared across all incremental protocols, while only the category arrival schedule varies across different orders. Therefore, performance differences among orders reflect continual-learning dynamics rather than changes in training or evaluation samples.

Difference from the original MP100 task. The original MP100 benchmark evaluates category-agnostic generalization across disjoint training and testing categories. In contrast, the reconstructed CCIKD benchmark evaluates sequential learning, where a single model receives new categories over time, trains only on current-stage data, and is evaluated on all previously observed categories. Results under the reconstructed protocol should therefore be interpreted as CCIKD results.

A.2 Incremental Protocols

Let $\mathcal{C}_t$ denote the categories introduced at stage t, and let $\mathcal{S}_t = \bigcup_{u \le t} \mathcal{C}_u$ denote all categories observed up to stage t. All incremental protocols satisfy $\mathcal{C}_t \cap \mathcal{C}_s = \emptyset$ for $t \neq s$. We consider an exemplar-free setting, where incremental training has access only to the current training categories without replaying previous training examples. Evaluation is performed on all seen categories.

For each benchmark, let $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ denote the fixed within-category image-disjoint training and held-out test sets described in Section A.1. Given a category set $\mathcal{A}$, $\mathcal{D}[\mathcal{A}]$ denotes the subset containing annotations whose categories belong to $\mathcal{A}$. At stage t, the training and evaluation sets are defined as

$$\mathcal{D}_t^{\text{train}} = \mathcal{D}^{\text{train}}[\mathcal{C}_t], \qquad \mathcal{E}_t = \mathcal{D}^{\text{test}}[\mathcal{S}_t]. \tag{10}$$

Therefore, each stage learns only from new categories while preserving knowledge of previous categories, and is evaluated on all seen categories to measure both acquisition and retention.

We evaluate CCIKD under four incremental protocols with different category scales, stage numbers, and schema heterogeneity, as summarized in Table 10.

Definition of category order. A category order defines the temporal trajectory through which categories arrive during incremental learning. Specifically, an order π partitions the complete category set $\mathcal{C}$ into stage-wise subsets $\{\mathcal{C}_1^\pi, \mathcal{C}_2^\pi, \ldots, \mathcal{C}_T^\pi\}$, where each category appears exactly once. Different

Table 10: Incremental protocols used in our experiments.

Protocol	Stages	Categories/Stage	Orders	Setting
IKDD-6stage	6	1	3	Aligned schemas
MP100-2stage	2	70/30	5	Original MP100 category partitions
MP100-10stage	10	10	3	Randomized streams
MP100-8stage	8	Unequal	3	Semantic superclasses

Table 11: Category orders for IKDD-6stage.

Order	Stage sequence
Order 1	person → cat → cow → dog → horse → sheep
Order 2	cat → cow → dog → horse → sheep → person
Order 3	cat → cow → dog → horse → person → sheep

Table 12: Construction of MP100-10stage category orders.

Order	Construction rule
Order 1	Sort category IDs from 1 to 100 and divide them into ten consecutive groups.
Order 2	Use a random permutation of the 100 categories and divide them into ten groups.
Order 3	Use another random permutation of the 100 categories and divide them into ten groups.

Table 13: Semantic superclass arrival orders for MP100-8stage.

Order	Stage sequence
Order 1	human hand → human face → furniture → vehicle → clothes → animal face → animal body → human body
Order 2	animal body → animal face → clothes → furniture → human body → human face → human hand → vehicle
Order 3	animal body → animal face → clothes → furniture → vehicle → human face → human hand → human body

orders share the same category set and within-category image split, but expose categories in different sequences. Multiple orders are evaluated because continual-learning performance depends not only on the final category set but also on the trajectory through which new categories arrive.

IKDD-6stage. IKDD-6stage introduces one category at each stage. Since all categories share the same 17-keypoint schema, it provides a controlled reference setting for studying incremental learning under aligned schemas. We evaluate three manually constructed category arrival orders, as summarized in Table 11, to reduce the dependence on a specific category trajectory and evaluate the robustness of continual learning under different arrival sequences.

MP100-2stage. MP100-2stage follows the original MP100 category split protocol. Each original split contains 70 training categories, 10 validation categories, and 20 test categories. We convert each split into a two-stage CCIKD stream by introducing the original 70 training categories at stage one and the remaining 30 categories, obtained by merging the original validation and test categories, at stage two. Therefore, the five original MP100 splits naturally define five category orders while sharing the same within-category image split.

MP100-10stage. MP100-10stage evaluates long-horizon continual learning by dividing the 100 categories into ten stages, with ten categories introduced at each stage. We construct three category orders with different arrival trajectories. The first order follows the canonical category ID ordering, where categories 1–100 are divided into ten consecutive groups, while the other two orders are generated using independent random permutations of the category set before forming the stage groups. All orders preserve the same category space and within-category image split while exposing the model to different heterogeneous category arrival sequences.

MP100-8stage. MP100-8stage is designed to evaluate continual learning under semantic category trajectories. Following the original semantic superclass organization provided by MP100, we group the 100 categories into eight predefined semantic groups: animal body (45 categories), animal face (30 categories), clothes (13 categories), furniture (5 categories), vehicle (3 categories), human face (2 categories), human body (1 category), and human hand (1 category). These predefined groups are not manually redefined for CCIKD; instead, they preserve the original semantic structure of MP100 and naturally contain diverse object domains with heterogeneous keypoint schemas.

Three category orders are constructed by permuting the arrival sequence of these fixed semantic groups, allowing evaluation under different semantic category trajectories while preserving the same group composition. The exact group-level trajectories are summarized in Table 13.

A.3 Evaluation Metrics

For an instance i evaluated at stage t, with bounding-box width and height (w_i, h_i), predicted keypoint $\hat{\mathbf{y}}_{ij}$, ground-truth keypoint $\mathbf{y}_{ij}$, and visibility indicator $v_{ij} \in \{0, 1\}$, where $v_{ij} = 1$ denotes a valid labeled keypoint, the normalized keypoint distance is

$$d_{ij} = \frac{\|\hat{\mathbf{y}}_{ij} - \mathbf{y}_{ij}\|_2}{\max(w_i, h_i)}. \tag{11}$$

The PCK score of category c at stage t and threshold τ is

$$p_c^{(t)}(\tau) = 100 \frac{\sum_{i:c_i=c} \sum_j v_{ij} \mathbb{I}[d_{ij} < \tau]}{\sum_{i:c_i=c} \sum_j v_{ij}}, \tag{12}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function, which equals 1 when the condition inside the brackets is satisfied and 0 otherwise. The factor 100 converts the ratio into a percentage.

Following the main paper, we average over four PCK thresholds $\tau \in \{0.05, 0.10, 0.15, 0.20\}$:

$$\mathrm{mPCK}_c^{(t)} = \frac{1}{4} \sum_{\tau \in \{0.05, 0.10, 0.15, 0.20\}} p_c^{(t)}(\tau). \tag{13}$$

For a set of seen categories $\mathcal{S}$, the All-Seen score is computed by category-level macro averaging:

$$\mathrm{mPCK}^{(t)}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{c \in \mathcal{S}} \mathrm{mPCK}_c^{(t)}. \tag{14}$$

This category-level aggregation assigns equal importance to each category and prevents categories with more instances or longer schemas from dominating the evaluation. At stage t, $\mathcal{S}_t$ denotes all categories observed up to the current stage, and the All-Seen performance is denoted as

$$A_t = \mathrm{mPCK}^{(t)}(\mathcal{S}_t). \tag{15}$$

FinalPCK. The final All-Seen performance after the last incremental stage is

$$\mathrm{FinalPCK} = A_T. \tag{16}$$

FinalPCK measures the overall knowledge retained by the model after completing all incremental learning stages. Specifically, it evaluates the model on all categories observed throughout the entire incremental process and reflects the final performance after learning all introduced categories. Unlike intermediate-stage metrics, FinalPCK focuses only on the final state of the model and therefore captures the combined effect of category acquisition and knowledge retention at the end of continual learning. A higher FinalPCK indicates that the model can better preserve previously learned categories while successfully incorporating newly introduced categories.

AIAPCK. The average incremental performance over all stages is

$$\mathrm{AIAPCK} = \frac{1}{T}\sum_{t=1}^{T} A_t. \tag{17}$$

AIAPCK measures the average All-Seen performance throughout the entire incremental learning trajectory. Instead of evaluating only the final model state, AIAPCK considers the performance at every incremental stage and provides an overall assessment of learning stability during the sequential acquisition process. This metric reflects how effectively the model balances learning new categories and retaining previously acquired knowledge over time. A higher AIAPCK indicates that the model maintains stronger performance throughout the incremental process rather than achieving good performance only at the final stage.

BWTPCK. Let $b(c)$ denote the stage where category c is first introduced, and let $\mathcal{S}_{T-1}$ denote all categories introduced before the final stage. We compute backward transfer as

$$\mathrm{BWTPCK} = \frac{1}{|\mathcal{S}_{T-1}|}\sum_{c\in\mathcal{S}_{T-1}} \left(\mathrm{mPCK}_c^{(T)} - \mathrm{mPCK}_c^{(b(c))}\right). \tag{18}$$

BWTPCK measures the influence of later learning stages on previously acquired categories relative to their initial acquisition stage, rather than the drop from the best historical checkpoint. Positive values indicate improved performance after subsequent learning, whereas negative values indicate forgetting. All BWT values and performance differences are reported in percentage points.

A.4 Training and Evaluation Protocol

All incremental experiments train each stage using only the current-stage training categories, without access to previous-stage training images or stored exemplars. Evaluation is performed on cumulative seen-category test sets containing all categories observed up to the current stage. The Joint reference accesses cumulative seen-category training data and therefore serves as an offline reference rather than an exemplar-free continual-learning method.

All incremental protocols share the same within-category image split, while only the category arrival schedule changes across different orders. Therefore, performance differences among category orders reflect continual-learning dynamics rather than changes in training or evaluation samples. Unless otherwise specified, all reported results use the final checkpoint of each incremental stage. Results averaged over category orders use the arithmetic mean across the corresponding streams.

For a protocol with R category orders, the reported order-averaged metric is computed as

$$\overline{M} = \frac{1}{R}\sum_{r=1}^{R} M^{(r)}, \tag{19}$$

where $M^{(r)}$ denotes the metric obtained from order r. Thus, each category order contributes equally to the final reported result.

B Detailed POISE Formulation and Identity Routing Framework

This appendix provides a formal description of POISE, including the persistent identity state, visual–geometric prototype construction, compatibility estimation, and incremental routing rules. We first define the evolving identity space and prototype representations, then describe the compatibility-based routing procedure and identity transition mechanisms. Finally, we detail the routed heatmap-bank interface and explain how POISE separates output-identity organization from continual-learning optimization. Output correspondence is determined before each stage's optimization, while the selected continual-learning strategy controls subsequent parameter updates.

B.1 Persistent Identity State

Let $\mathcal{Q}_t = \{(c, j) \mid c \in \mathcal{C}_t,\ 1 \le j \le K_c\}$ denote the category–keypoints introduced at stage t, with $\mathcal{Q}_{<t} = \bigcup_{u<t} \mathcal{Q}_u$ and $\mathcal{Q}_{\le t} = \bigcup_{u \le t} \mathcal{Q}_u$. Categories arrive in disjoint sets, and each category preserves its local keypoint schema after introduction. POISE represents evolving output correspondence through a persistent identity state, including an identity set $\mathcal{Z}_t = \{0, \dots, M_t - 1\}$, a routing map $r_t : \mathcal{Q}_{\le t} \to \mathcal{Z}_t$, and identity member sets $\mathcal{A}_z^t = \{q \in \mathcal{Q}_{\le t} \mid r_t(q) = z\}$.

The persistent identity state satisfies

$$\begin{aligned} &\mathcal{Z}_{t-1} \subseteq \mathcal{Z}_t, \\ &r_t(q) = r_{t-1}(q), \quad q \in \mathcal{Q}_{<t}, \\ &r_t(c, j) \neq r_t(c, k), \quad \forall (c, j), (c, k) \in \mathcal{Q}_{\le t},\ j \neq k. \end{aligned} \tag{20}$$

The first constraint enforces append-only identity growth, the second preserves established output correspondence, and the third guarantees category-wise injective routing. Different categories may share one identity when their keypoints are sufficiently compatible, while keypoints within the same category must remain assigned to distinct output identities. Therefore, persistent identities define output correspondence independently of numerical parameter values: an existing output channel can still be optimized by the selected continual-learning strategy.

The routing memory stores member prototypes, identity-level summaries, membership information, and route metadata. Member statistics are required for identity-level compatibility evaluation, while a single aggregate prototype is insufficient to represent the diversity of members within an identity. The stored memory contains only aggregated training statistics rather than historical images or individual image descriptors. During each incremental stage, the previous routing memory is frozen for identity assignment, and the resulting routing map remains fixed during optimization. Thus, POISE performs deterministic stage-wise output organization without introducing a jointly trained routing module or requiring access to future categories.

B.2 Training-only Prototype Construction

B.2.1 Local Visual Observations

For each visible occurrence of $q = (c, j)$ in the current-stage training split, POISE extracts a local visual descriptor around the annotated keypoint using a frozen visual encoder (DINOv3). The encoder is used only to provide routing evidence, while prototype extraction does not update the pose estimator or require historical images.

Let $\mathbf{F}_i(\mathbf{u})$ denote the feature at patch location $\mathbf{u}$ and $\widetilde{\mathbf{y}}_{i,q}$ denote the annotated keypoint location in the feature-grid coordinates. Gaussian-weighted pooling followed by ℓ_2 normalization produces the observation-level visual descriptor:

$$\begin{aligned} a_{i,q}(\mathbf{u}) &= \exp\left(-\frac{\|\mathbf{u} - \widetilde{\mathbf{y}}_{i,q}\|_2^2}{2\sigma_e^2}\right), \qquad \sigma_e = 0.75, \\ \mathbf{v}_{i,q} &= \mathrm{norm}\left(\frac{\sum_{\mathbf{u}} a_{i,q}(\mathbf{u}) \mathbf{F}_i(\mathbf{u})}{\sum_{\mathbf{u}} a_{i,q}(\mathbf{u})}\right). \end{aligned} \tag{21}$$

The Gaussian width is measured in feature-grid units. Let $\mathcal{I}_q$ denote the valid training occurrences of q and $n_q = |\mathcal{I}_q|$. The aggregated visual prototype is computed as

$$\overline{\mathbf{v}}_q = \mathrm{norm}\left(n_q^{-1} \sum_{i \in \mathcal{I}_q} \mathbf{v}_{i,q}\right). \tag{22}$$

The resulting descriptor summarizes the appearance evidence of a category-keypoint from repeated training observations. Since normalization removes scale differences, averaging normalized descriptors or normalizing their sum gives the same direction. The frozen visual encoder only provides prototype evidence and is not involved in pose-model optimization or inference.

B.2.2 GEOMETRY AND SIDE METADATA

For crop dimensions (W_i, H_i) and transformed keypoint coordinates $(x_{i,q}, y_{i,q})$, we define normalized coordinates as $\widehat{x}_{i,q} = (x_{i,q} - W_i/2)/W_i$ and $\widehat{y}_{i,q} = (y_{i,q} - H_i/2)/H_i$. Let j' denote the schema-provided left–right counterpart of keypoint j, and let $\mathbb{I}^{\text{pair}}_{i,q}$ indicate whether $j' \neq j$ and the counterpart is visible. The geometry descriptor is

$$\Delta^{\text{pair}}_{i,q} = \begin{cases} |x_{i,(c,j)} - x_{i,(c,j')}|/W_i, & \mathbb{I}^{\text{pair}}_{i,q} = 1, \\ 0, & \mathbb{I}^{\text{pair}}_{i,q} = 0, \end{cases} \qquad \mathbf{g}_{i,q} = \left[|\widehat{x}_{i,q}|, \widehat{y}_{i,q}, \Delta^{\text{pair}}_{i,q}, \mathbb{I}^{\text{pair}}_{i,q} \right]^\top. \tag{23}$$

The descriptor captures category-specific geometric evidence: absolute horizontal position removes left–right ambiguity, pair distance encodes local structure, and the visibility indicator distinguishes an unavailable counterpart from an observed zero distance. The side label $\ell_q \in \{-1, 0, 1\}$ represents left, center/unknown, and right keypoints. Both partner indices and side labels are schema metadata within each category and do not provide cross-category keypoint correspondence.

For $n_q > 1$, the geometry statistics are computed as

$$\boldsymbol{\mu}_q = \frac{1}{n_q} \sum_{i \in \mathcal{I}_q} \mathbf{g}_{i,q}, \qquad \boldsymbol{\Sigma}_q = \frac{1}{n_q - 1} \sum_{i \in \mathcal{I}_q} (\mathbf{g}_{i,q} - \boldsymbol{\mu}_q)(\mathbf{g}_{i,q} - \boldsymbol{\mu}_q)^\top + \epsilon \mathbf{I}, \qquad \epsilon = 10^{-4}. \tag{24}$$

For a single observation, the covariance is set to $\epsilon\mathbf{I}$, providing a regularized uncertainty estimate while avoiding degenerate covariance estimation caused by insufficient samples. Entries without valid observations lack sufficient geometric evidence to support reliable identity reuse and therefore cannot satisfy the reuse criteria. Such cases are excluded from prototype-based matching and are instead handled through the new identity discovery mechanism.

The resulting category–keypoint prototype is

$$\mathbf{p}_q = \left(\underbrace{\overline{\mathbf{v}}_q}_{\text{visual}}, \underbrace{\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q}_{\text{geometry}}, \underbrace{\ell_q}_{\text{side}}, n_q \right). \tag{25}$$

The visual and geometric components summarize repeated observations of the same category–keypoint, while side information remains schema metadata. The observation count n_q controls statistical aggregation rather than semantic confidence. All prototypes are computed before stage-wise optimization using only the currently available training split.

B.2.3 IDENTITY-LEVEL AGGREGATION

For an identity z containing a set of category–keypoint members $q \in \mathcal{A}_z$, POISE aggregates member-level prototypes into an identity-level representation. This aggregation allows an identity to be described by multiple compatible category–keypoints rather than a single schema-specific observation. Let $w_q = \max(n_q, 1)$ denote the observation support of member q, and define the normalized aggregation weight as $\pi_q = w_q / \sum_{u \in \mathcal{A}_z} w_u$. We omit the stage superscript when the context is clear. The identity-level summary is computed as

$$\overline{\mathbf{v}}_z = \text{norm}\left(\sum_{q \in \mathcal{A}_z} \pi_q \overline{\mathbf{v}}_q \right), \quad \boldsymbol{\mu}_z = \sum_{q \in \mathcal{A}_z} \pi_q \boldsymbol{\mu}_q, \quad \boldsymbol{\Sigma}_z = \sum_{q \in \mathcal{A}_z} \pi_q [\boldsymbol{\Sigma}_q + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_z)(\boldsymbol{\mu}_q - \boldsymbol{\mu}_z)^\top]. \tag{26}$$

The geometry summary follows the law of total covariance, combining within-member uncertainty and between-member variation. Therefore, the identity representation can capture both the intrinsic variability of each category–keypoint and the structural differences among compatible members. The identity side label ℓ_z is derived from member-side metadata, while the accumulated observation support $n_z = \sum_{q \in \mathcal{A}_z} n_q$ is retained for statistical aggregation.

Individual member prototypes are preserved with the aggregated identity summary, as member-level evidence is required for compatibility evaluation. During Base Align, identity summaries are updated online as new members are assigned. In later incremental stages, historical summaries remain fixed during routing and are updated only after assignment decisions are finalized.

B.3 Prototype Compatibility and Member Evidence

B.3.1 Pairwise Visual–Geometric Score

For prototypes a and b, let $\boldsymbol{\delta}_{ab} = \boldsymbol{\mu}_a - \boldsymbol{\mu}_b$. The pairwise compatibility combines a visual cosine similarity and a covariance-aware geometric similarity:

$$\begin{aligned} s_{\mathrm{v}}(a,b) &= \frac{\overline{\mathbf{v}}_a^{\top}\overline{\mathbf{v}}_b}{\|\overline{\mathbf{v}}_a\|_2\|\overline{\mathbf{v}}_b\|_2}, \\ d_a^2(a,b) &= \boldsymbol{\delta}_{ab}^{\top}(\boldsymbol{\Sigma}_a + \epsilon\mathbf{I})^{+}\boldsymbol{\delta}_{ab}, \\ g_0(a,b) &= \frac{1}{2}\left[\exp\left(-\frac{d_a^2(a,b)}{2T_g}\right) + \exp\left(-\frac{d_b^2(a,b)}{2T_g}\right)\right]. \end{aligned} \tag{27}$$

Here $(\cdot)^{+}$ denotes the Moore–Penrose pseudoinverse, and $d_b^2(a,b)$ is computed by replacing $\boldsymbol{\Sigma}_a$ with $\boldsymbol{\Sigma}_b$ in the same expression. We set $T_g = 1$. The bidirectional formulation makes g_0 symmetric even when two prototypes have different covariance estimates.

Side metadata adjusts the geometric similarity through

$$\omega(a,b) = \begin{cases} 1, & \ell_a = \ell_b, \\ \omega_0, & \ell_a \neq \ell_b \text{ and } (\ell_a = 0 \text{ or } \ell_b = 0), \\ 0, & \text{otherwise}, \end{cases} \qquad \omega_0 = 0.5. \tag{28}$$

The final pairwise score is

$$s_{\mathrm{g}}(a,b) = \omega(a,b)g_0(a,b), \qquad s_{\mathrm{pair}}(a,b) = \lambda_{\mathrm{v}}s_{\mathrm{v}}(a,b) + \lambda_{\mathrm{g}}s_{\mathrm{g}}(a,b), \tag{29}$$

where $\lambda_{\mathrm{v}} + \lambda_{\mathrm{g}} = 1$ and both weights are set to 0.5 in the reported configuration. We retain the raw visual cosine without shifting or clipping, resulting in $s_{\mathrm{pair}} \in [-\lambda_{\mathrm{v}}, 1]$. Base Align incorporates side information through this score, whereas Historical Align, clustering, and Transfer additionally enforce positive side compatibility.

B.3.2 Aggregate and Member Support

For a query q and identity z, compatibility is evaluated using both the identity-level summary and individual member prototypes. Let $S_z(q) = s_{\mathrm{pair}}(q,z)$ denote the aggregate score from the identity summary, and let $u_a(q) = s_{\mathrm{pair}}(q,a)$ denote the score between q and member $a \in \mathcal{A}_z$. We define the adaptive support set as the members whose scores are close to the best-matching member:

$$\begin{aligned} \mathcal{M}_z(q) &= \left\{ a \in \mathcal{A}_z \mid u_a(q) \geq \max_{b\in\mathcal{A}_z} u_b(q) - \delta_m \right\}, \qquad \delta_m = 0.05, \\ U_z(q) &= \frac{1}{|\mathcal{M}_z(q)|}\sum_{a\in\mathcal{M}_z(q)} u_a(q). \end{aligned} \tag{30}$$

POISE combines the identity summary and adaptive member support as

$$s(q,z) = \alpha S_z(q) + (1-\alpha)U_z(q), \qquad \alpha = 0.4. \tag{31}$$

Member coverage is evaluated over the complete identity rather than only the support subset:

$$\kappa_z(q) = \frac{1}{|\mathcal{A}_z|}\sum_{a\in\mathcal{A}_z} \mathbb{I}[u_a(q) \geq \tau_A - e], \qquad e = 0.15. \tag{32}$$

The support set is adaptive rather than fixed-size: its cardinality is determined by member compatibility. Given the selected Align threshold τ_A, the coverage score $\kappa_z(q)$ measures the fraction of identity members whose compatibility scores exceed the relaxed threshold $\tau_A - e$, where $e = 0.15$. Thus, $\kappa_z(q)$ distinguishes broadly supported matches from matches relying on only a subset of members. The aggregate–member gap $|S_z - U_z|$ and modality gap $|s_{\mathrm{v}} - s_{\mathrm{g}}|$ provide additional consistency checks. These statistics estimate compatibility confidence rather than establish ground-truth cross-category semantic correspondence.

B.4 Deterministic Identity Routing

POISE separates identity assignment from parameter initialization. Align reuses an existing identity, whereas rejected queries create new identities through Transfer or Expand.

B.4.1 Base Align

At $t = 1$, the identity bank is empty and queries are processed in canonical category–index order. For query $q = (c, j)$, identities assigned to category c are excluded to preserve category-wise injectivity. If no candidate exists, a new identity is created; otherwise, s_1 and s_2 denote the best and runner-up eligible scores ($s_2 = 0$ for a single candidate). Base Align accepts the best identity when

$$s_1 \geq \tau_B = 0.60, \qquad s_1 - s_2 \geq m_B = 0.05. \tag{33}$$

An accepted query is added to the selected identity and updates its summary before processing the next query; otherwise, a new identity is created. Side information is incorporated through the compatibility score without an additional hard constraint. Since no historical model exists at this stage, all output rows use standard initialization.

B.4.2 Historical Align

At $t > 1$, all query-to-identity scores are evaluated using the frozen stage-$(t-1)$ routing memory. Let $\mathcal{O}_c$ contain the historical identities already assigned to current keypoints of category c, and let $\mathrm{Cat}(\mathcal{A}) = \{c' \mid (c', j') \in \mathcal{A}\}$. The eligible identity set is

$$\mathcal{E}_c = \left\{ z \in \mathcal{Z}_{t-1} \mid c \notin \mathrm{Cat}(\mathcal{A}_z^{t-1}) \right\} \setminus \mathcal{O}_c. \tag{34}$$

The first constraint ensures that an identity cannot be reused by multiple keypoints from the same category, while $\mathcal{O}_c$ prevents assigning the same historical identity to multiple queries within the current category. In contrast, keypoints from different categories may still share one historical identity when their prototypes are sufficiently compatible.

For each query with available candidates, let $z^\star$ and z_2 denote the best and runner-up identities, respectively. We set $s(q, z_2) = 0$ when only one candidate exists and define the margin as

$$m(q) = s(q, z^\star) - s(q, z_2). \tag{35}$$

Queries without eligible identities remain unassigned and are handled by identity discovery.

Writing S_z, U_z, and κ_z for the query-dependent evidence statistics, Historical Align accepts the best candidate $z^\star$ when the compatibility score, margin, and evidence consistency criteria are satisfied:

$$s(q, z^\star) \geq \tau_A = 0.65, \qquad m(q) \geq m_A = 0.05. \tag{36}$$

The evidence consistency criteria include sufficient member support, coverage, modality agreement, and side compatibility as defined in Sec. B.3.

Within each category, POISE greedily accepts the qualified query with the largest tuple $(s(q, z^\star), m(q), \mathrm{id}(q))$. After acceptance, $z^\star$ is removed from the remaining candidate sets, and the procedure repeats until no query satisfies the predicate.

All compatibility scores are evaluated on the frozen historical snapshot. During routing, accepted assignments only affect candidate availability, while the underlying evidence remains unchanged for subsequent queries within the same stage.

B.4.3 Within-stage Identity Discovery

Unassigned queries are initialized as singleton clusters for within-stage identity discovery. A merge between clusters G and H is permitted only if their category sets are disjoint, all cross-cluster pairs satisfy side compatibility, and the complete-linkage score exceeds the threshold:

$$\begin{aligned} \mathrm{Cat}(G) \cap \mathrm{Cat}(H) &= \emptyset, \\ \omega(a, b) &> 0, \qquad \forall a \in G,\ b \in H, \\ s_{\mathrm{cl}}(G, H) &= \min_{a \in G, b \in H} s_{\mathrm{pair}}(a, b) \geq \tau_B = 0.60. \end{aligned} \tag{37}$$

Among valid merge candidates, POISE greedily merges the pair with the largest s_{cl}. The mean cross-pair score is defined as

$$\bar{s}_{\mathrm{cl}}(G, H) = \frac{1}{|G||H|} \sum_{a \in G} \sum_{b \in H} s_{\mathrm{pair}}(a, b). \tag{38}$$

The mean score and the canonical cluster key are used for deterministic tie breaking. Merging continues until no valid pair remains.

Complete linkage prevents weak pairs from being concealed by locally strong matches, while category disjointness ensures at most one keypoint per category in each final cluster. Each cluster receives a new identity, with initialization determined by Transfer or Expand.

B.4.4 Transfer or Expand

For each member $q = (c, j)$ in a final cluster G, POISE searches eligible historical sources using the frozen routing memory. A qualified source leads to Transfer; otherwise, the new identity is initialized by Expand. A source proposal from q to historical identity z is accepted when

$$n_q \geq 1, \qquad s(q, z) \geq \tau_T = 0.45, \qquad m(q) \geq m_T = 0.05, \qquad \omega(q, z) > 0. \tag{39}$$

Among qualified proposals, the highest-scoring source determines the initialization, with margin and canonical query order resolving ties. Transfer uses a relaxed criterion because it only transfers initialization parameters and does not establish persistent identity reuse.

If the selected source is z_s and the new identity is z_n, Transfer copies the output parameters:

$$\mathbf{W}^{\mathrm{init}}_{t,z_n} \leftarrow \mathbf{W}_{t-1,z_s}, \qquad b^{\mathrm{init}}_{t,z_n} \leftarrow b_{t-1,z_s}. \tag{40}$$

The source and destination retain separate identity indices and parameters after initialization; Transfer does not modify source membership. If no qualified source exists, Expand initializes the new identity using the standard head initializer:

$$(\mathbf{W}^{\mathrm{init}}_{t,z_n}, b^{\mathrm{init}}_{t,z_n}) \leftarrow \mathrm{Initialize}(). \tag{41}$$

Both Transfer and Expand create new identities for final clusters, with the only difference being how the corresponding output parameters are initialized. Historical Align, in contrast, reuses existing identities without expanding the output space.

B.5 Complete Stage Procedure and Structural Guarantees

Given current training data $\mathcal{D}_t$ and, for $t > 1$, the previous routing memory, POISE follows the stage-wise identity routing procedure summarized in Algorithm 1.

The identity space grows in an append-only manner: new identities receive fresh indices, while existing identities are never removed. Historical persistence is preserved because previous routes remain frozen and only current-stage queries receive new assignments. Category-wise injectivity is guaranteed through category exclusion in Base Align, occupancy constraints in Historical Align, and category-disjoint clustering during identity discovery.

These guarantees concern the structure of output correspondence rather than the semantic correctness of individual prototype matches. They do not imply invariance to arbitrary query ordering or eliminate parameter forgetting. Instead, compatibility safeguards prevent unsupported identity sharing, while the continual-learning objective preserves the associated prediction functions.

B.6 Routed Heatmap Bank and Continual Optimization

Let $\mathbf{U}_t(x) \in \mathbb{R}^{D \times H \times W}$ denote the feature map before the final readout. POISE keeps the backbone and heatmap tower unchanged and assigns each persistent identity to one output row of the final convolution. The identity-indexed heatmap bank and category-conditioned output are

$$\begin{aligned} \mathbf{H}_t(x) &= \mathbf{W}_t * \mathbf{U}_t(x) + \mathbf{b}_t, \\ f_t(x, c) &= \left[\mathbf{H}_{t, r_t(c,j)}(x)\right]_{j=1}^{K_c}. \end{aligned} \tag{42}$$

Algorithm 1 Stage-wise Routing and Optimization in POISE

Require: Current-stage data $\mathcal{D}_t$; previous routing memory (if $t > 1$)
Ensure: Updated identity bank, routing map, and model state
1: Extract category–keypoint prototypes from $\mathcal{D}_t$ using the frozen encoder and geometry statistics.
2: **if** $t = 1$ **then**
3: Construct the initial identity bank using Base Align.
4: **else**
5: Freeze the previous routing memory.
6: Perform Historical Align to reuse qualified historical identities.
7: Apply within-stage identity discovery to unassigned queries using valid complete-linkage merges.
8: Initialize newly created identities with Transfer if a qualified historical source exists; otherwise use Expand.
9: **end if**
10: Commit routing assignments, update identity summaries, and construct the identity-indexed output bank while preserving historical routes.
11: Optimize the detector with the selected continual-learning objective under the fixed routing map, and update the model and retained states for the next stage.

Given category c, the routing map determines which identity-indexed heatmap channels correspond to its local keypoint schema. Specifically, for each local keypoint index j, the route $r_t(c, j)$ selects one persistent identity channel from the global heatmap bank, and the selected channels are gathered and reordered according to the category-specific schema:

$$f_t(x, c) = \left[\mathbf{H}_{t,r_t(c,1)}(x), \mathbf{H}_{t,r_t(c,2)}(x), \ldots, \mathbf{H}_{t,r_t(c,K_c)}(x)\right]. \tag{43}$$

Therefore, categories with different keypoint numbers and local orderings can construct their own prediction outputs from the shared identity-indexed heatmap bank. The global identity indices only define persistent output correspondence and do not constrain category-local ordering or require contiguous channels. During stage transitions, historical identity rows preserve their weights and indices, while newly created identities append additional readout rows.

After routing determines the output correspondence, parameter optimization follows the selected continual-learning strategy. The training objective is

$$\mathcal{L}_t = \mathcal{L}^t_{\text{sup}} + \lambda_{\text{P}} \mathcal{L}^t_{\text{P}}, \tag{44}$$

where $\mathcal{L}^t_{\text{sup}}$ denotes the visibility-weighted heatmap loss on current-stage routed outputs and $\mathcal{L}^t_{\text{P}}$ denotes the preservation objective of the selected continual-learning strategy. Different strategies preserve knowledge through different mechanisms: output distillation maintains prediction functions, parameter regularization constrains important parameters, feature-level distillation preserves intermediate representations, and parameter-efficient learners update only designated modules. Therefore, POISE determines output correspondence independently of the specific optimization strategy.

The preservation strategy operates on the routed outputs without modifying identity assignment. We use each continual-learning method with its original objective, configuration, and hyperparameters, while POISE only determines the correspondence between category–keypoints and persistent output identities. The auxiliary states required by different strategies, such as previous models, parameter-importance statistics, or task-specific modules, are maintained independently from the POISE routing memory. Thus, identity correspondence provides a consistent output interface for continual optimization, rather than enforcing shared parameters or a specific preservation mechanism.

B.7 Capacity and Retained State

Let $Q_t = \sum_{c \in \mathcal{C}_t} K_c$ denote the number of category–keypoints introduced at stage t, and let G_t denote the number of final clusters formed from queries not assigned by Historical Align. Each final cluster contributes one new identity, regardless of whether its initialization uses Transfer or Expand:

$$M_t = M_{t-1} + G_t, \qquad 0 \le G_t \le Q_t, \qquad M_{t-1} \le M_t \le M_{t-1} + Q_t. \tag{45}$$

The identity growth of POISE is governed by the number of newly created clusters. Complete identity reuse introduces no additional identities, whereas assigning each new category–keypoint a

separate identity yields maximal expansion. This relation characterizes output-space growth rather than performance ordering: a larger identity bank does not necessarily improve accuracy. The realized expansion depends on the trade-off between historical reuse and within-stage identity sharing.

For a readout with D input channels, a $k \times k$ kernel, and one bias term per output row, the parameter count is $M_t(Dk^2 + 1)$. Therefore, appending G_t identities introduces $G_t(Dk^2 + 1)$ additional parameters, while the backbone and heatmap tower remain unchanged. The complete retained state includes member prototypes, identity summaries, routing metadata, and the auxiliary state required by the selected continual-learning strategy.

Between stages, POISE retains each member's visual prototype, geometry statistics, side metadata, observation count, identity membership, and routing assignments. It does not store historical images, crops, or per-instance visual descriptors. At inference, the pose model and category-conditioned routing map are sufficient; prototype construction and compatibility evaluation are unnecessary.

C Implementation Details

This appendix describes the architectural configurations, optimization schedules, continual-learning objectives, parameter-efficient settings, and retained states used for reproducing the experiments. Unless otherwise specified, Shared-$K_{\max}$ and POISE share the same detector architecture, data preprocessing, optimizer, augmentation, stage schedule, and preservation strategy. The only difference is the mapping from category-local keypoints to output rows.

C.1 Pose Architectures and Common Training Schedule

All experiments are conducted using NVIDIA GeForce RTX 3090 GPUs. For fair comparison, all methods are trained and evaluated under the same hardware and software environment. Unless otherwise specified, Shared-$K_{\max}$ and POISE use the same pose architecture, data preprocessing, augmentation, optimizer, and training schedule. We use ViTPose as the primary pose estimator and HRNet for backbone ablation. Both architectures adopt the same output-bank interface for Shared-$K_{\max}$ and POISE, where the only difference lies in the mapping from category-local keypoints to output identities. Under the same backbone and continual-learning strategy, Shared-$K_{\max}$ and POISE share identical training configurations. The complete settings are summarized in Table 14.

Table 14: **Common training settings.** Shared-$K_{\max}$ and POISE use identical settings under the same backbone and continual-learning strategy. Spatial dimensions are reported as width $\times$ height.

Component	Setting
Pose input / heatmap	192×256 / 48×64; UDP encoding with Gaussian $\sigma = 2$
Stage schedule	210 epochs; evaluation every 10 epochs; 500-iteration warm-up
Batch size	32
ViTPose optimizer	AdamW, learning rate 5×10^{-4}, weight decay 0.1
HRNet optimizer	Adam, learning rate 5×10^{-4}
Augmentation / inference	Horizontal flip, half-body, box transform, UDP crop, and flip testing

The supervised objective is a visibility-weighted heatmap mean-squared error with valid-mean reduction. For heterogeneous schemas, only valid category-specific keypoints contribute to the loss, while padded positions are ignored. This formulation keeps the optimization interface unchanged across Shared-$K_{\max}$ and POISE despite their different output organizations.

Output-head implementation. For Shared-$K_{\max}$, the final heatmap head contains a fixed number of $K_{\max} = 68$ output channels, and each category-local keypoint $k_{c,j}$ is directly mapped to the j-th output row. Therefore, keypoints with the same local index across different categories share the same output parameters regardless of their cross-category compatibility. For POISE, the final heatmap head contains an expandable identity-indexed output bank. Each category-keypoint is assigned to a persistent identity through the routing map, and category-specific predictions are reconstructed by gathering the corresponding identity channels according to the local keypoint order.

C.2 CONTINUAL-LEARNING STRATEGIES

Fine-tuning (FT) minimizes only the current-stage supervised loss. Joint is an offline reference trained with cumulative seen-category data.

Heatmap-level preservation. For LwF, a frozen previous-stage teacher supervises historical identity rows using current-stage images. Let $H^S_{i,z}$ and $H^T_{i,z}$ denote the student and teacher heatmaps for image i and historical identity z. The preservation term is

$$\mathcal{L}_{\mathrm{HM}} = \frac{1}{BM_{t-1}H_hW_h} \sum_{i=1}^{B} \sum_{z=0}^{M_{t-1}-1} \left\| H^S_{i,z} - H^T_{i,z} \right\|_F^2 . \tag{46}$$

The distillation coefficient is set to 1. Distillation applies only to the M_{t-1} historical rows; appended rows lack teacher counterparts and learn from current-stage supervision only.

Parameter regularization. EWC applies parameter importance constraints to historical parameters. For the historical parameter set $\mathcal{P}_{t-1}$, the regularization term is

$$\mathcal{L}_{\mathrm{EWC}} = \frac{1000}{2|\mathcal{P}_{t-1}|} \sum_{k \in \mathcal{P}_{t-1}} F_k \left(\theta_k - \theta_k^{t-1}\right)^2 . \tag{47}$$

The importance values are estimated from squared minibatch gradients of the supervised loss in evaluation mode, normalized by the selected-parameter mean, and accumulated online with factor one. When an output tensor expands, only its historical prefix participates in the regularization term.

Feature-level preservation. PODNet-style spatial distillation extracts normalized spatial descriptors from exposed feature maps and matches teacher and student descriptors with original coefficient. This implementation adapts feature-level distillation to keypoint detection.

Parameter-efficient strategies. L2P and InfLoRA evaluate whether persistent output organization remains effective under parameter-efficient continual learning. For each PEFT method, Shared-$K_{\max}$ and POISE use the same initialization, trainable modules, stage schedule, and preservation configuration. L2P selects prompts using a frozen global-average feature. InfLoRA retains historical adapters and subspaces while learning current low-rank updates and the expanded output bank.

The prescribed pretrained initializations are retained for each method: standard ViTPose uses MAE initialization, L2P uses supervised ImageNet-21K ViT-B/16 weights, and InfLoRA uses AugReg ImageNet-21K weights. Therefore, paired Shared–POISE comparisons isolate the effect of output organization within each continual-learning strategy, while absolute differences across PEFT methods also reflect their prescribed initialization and trainable parameterization.

C.3 RETAINED ROUTING STATE

Retained-state definition. The exemplar-free setting assumes that no historical training examples are available during incremental optimization. POISE does not retain historical images or per-instance descriptors. Instead, between incremental stages, it maintains aggregated routing state, including category–keypoint prototypes, identity-level summaries, identity membership information, and persistent routing metadata. Each member prototype consists of a visual descriptor, a four-dimensional geometry mean, a 4×4 geometry covariance, a side label, and an observation count. The visual component is extracted using a frozen DINOv3 encoder with *768-dimensional* descriptors. We quantify the retained payload under this representation.

Memory accounting. We quantify the logical numerical payload required for routing. We use float32 storage for visual descriptors and geometry statistics, int32 for side labels, and int64 for observation counts. Since the visual and geometry observation counts are identical in our implementation, they share a single stored count. Therefore, for a visual descriptor dimension d, one packed member prototype requires

$$b(d) = \underbrace{4d}_{\text{visual}} + \underbrace{4(4+4^2)}_{\text{geometry}} + \underbrace{4}_{\text{side}} + \underbrace{8}_{\text{count}} = 4d + 92 \text{ bytes}. \tag{48}$$

Table 15: Final retained routing-state memory footprint across evaluated protocols. The member-prototype payload is reported under the 768-dimensional DINOv3 descriptors used in our experiments. The upper bound corresponds to the worst-case accounting where each retained member has an independent identity summary. All capacities are reported in decimal MB.

Protocol	Category-keypoints	Member prototypes	Routing-state upper bound
IKDD-6stage	102	0.323	0.645
MP100-2stage	1565	4.952	9.903
MP100-10stage	1565	4.952	9.903
MP100-8stage	1565	4.952	9.903

Table 16: Stage-wise retained routing-state memory for the representative MP100-10stage order (order2). Q_t, N_t, and M_t denote the numbers of newly introduced category–keypoints, retained category–keypoint members, and persistent identities, respectively. B_{old} and B_{new} represent the historical and current-stage member-prototype payloads, while B_{mem} denotes the cumulative member-prototype payload. $B_{\mathrm{all}}^{\mathrm{ub}}$ is a worst-case upper bound obtained by assuming one identity summary per retained member. Member, Summary, and Total denote the cached member-prototype state, identity-summary state, and total retained routing state. All capacities are reported in decimal MB.

Stage	Q_t	N_t	M_t	B_{old}	B_{new}	B_{mem}	$B_{\mathrm{all}}^{\mathrm{ub}}$	Member	Summary	Total(768)
1	184	184	111	0.000	0.582	0.582	1.164	0.582	0.351	0.933
2	140	324	169	0.582	0.443	1.025	2.050	1.025	0.535	1.560
3	207	531	304	1.025	0.655	1.680	3.360	1.680	0.962	2.642
4	136	667	367	1.680	0.430	2.110	4.221	2.110	1.161	3.272
5	143	810	458	2.110	0.452	2.563	5.126	2.563	1.449	4.012
6	141	951	524	2.563	0.446	3.009	6.018	3.009	1.658	4.667
7	160	1111	617	3.009	0.506	3.515	7.030	3.515	1.952	5.467
8	142	1253	659	3.515	0.449	3.964	7.929	3.964	2.085	6.050
9	131	1384	715	3.964	0.414	4.379	8.758	4.379	2.262	6.641
10	181	1565	825	4.379	0.573	4.952	9.903	4.952	2.610	7.562

For the 768-dimensional descriptors used in our experiments, this corresponds to $b(768) = 3{,}164$ bytes per member prototype. Let $Q_t = |\mathcal{Q}_t|$ denote the number of category–keypoints introduced at stage t, and let $N_t = \sum_{u \le t} Q_u$ denote the cumulative number of retained category–keypoint members. The cumulative member-prototype payload retained after stage t is

$$B_{\mathrm{mem}}(t; d) = \frac{N_t b(d)}{10^6} \ \mathrm{MB}. \tag{49}$$

This payload scales with the cumulative schema size rather than the number of historical training images. As summarized in Table 15, IKDD contains 102 category–keypoint members and requires only $0.323\,\mathrm{MB}$ for member prototypes, while all MP100 protocols contain 1,565 category–keypoint members and require $4.952\,\mathrm{MB}$ under the same representation.

Identity summary overhead. POISE maintains identity-level summaries to support compatibility evaluation at both identity and member levels. If M_t identity summaries are additionally cached, the total retained routing-state payload is

$$B_{\mathrm{total}}(t; d) = \frac{(N_t + M_t) b(d)}{10^6}, \tag{50}$$

where N_t and M_t denote the numbers of retained category–keypoint members and persistent identities, respectively. Since each identity contains at least one member, $M_t \le N_t$, yielding the following upper bound:

$$B_{\mathrm{total}}(t; d) \le 2 B_{\mathrm{mem}}(t; d). \tag{51}$$

Table 16 provides a stage-wise decomposition of the retained routing state for the representative MP100-10stage order 2. The retained state grows with the cumulative number of introduced category–keypoints rather than the number of historical training examples. At the final stage, POISE

Table 17: **Comparison of research settings between CAPE and CCIKD.** The two paradigms address heterogeneous keypoint schemas under different learning assumptions.

Paradigm	New-category data	Test-time support	Previous-category support	Main focus
CAPE	✓	✓	✓	Category Generalization
CCIKD (POISE)	✓	×	×	Exemplar-free Continual Learning

retains 1,565 category–keypoint members and 825 persistent identities, resulting in $7.562\,\mathrm{MB}$ of cached routing state, including identity summaries. The corresponding worst-case upper bound is $9.903\,\mathrm{MB}$ when every retained member is assumed to maintain an independent identity summary.

The exemplar-free constraint concerns the unavailability of historical training examples during incremental optimization. Individual continual-learning strategies may maintain auxiliary states, such as previous models for distillation, Fisher estimates for EWC, prompts for L2P, or adapters and feature subspaces for InfLoRA. These strategy-specific states are independent of the POISE routing state and excluded from this accounting. The expandable output identity bank and readout parameters represent model capacity rather than retained routing memory and are analyzed in Sec. D.5.2.

D Additional Experiments and Analysis

This appendix complements the main results with cross-paradigm comparisons, preservation-signal comparisons, order-level measurements, stage trajectories, and diagnostic analyses of output-identity collisions. Unless stated otherwise, localization metrics are percentages computed on the cumulative all-seen test set at the final stage and use the mPCK convention from Appendix A.3; loads, channel counts, and parameter counts retain their natural units. Aggregate metric entries are order means, while the per-order tables expose the underlying final-stage values.

D.1 Comparison with Category-Agnostic Pose Estimation

Category-agnostic pose estimation provides an alternative paradigm for handling heterogeneous keypoint schemas, where category information is introduced through support-conditioned inference. Methods such as CapeFormer (Shi et al., 2023) and SCAPE (Nguyen et al., 2024) use annotated support examples at inference time to identify category-specific keypoints, whereas POISE studies exemplar-free continual keypoint learning, where historical training examples are unavailable during incremental updates and a single model progressively acquires new categories.

We compare POISE with CAPE-family methods to investigate whether persistent output organization enables effective continual adaptation without test-time support examples. Although both paradigms address heterogeneous keypoint schemas, they differ in their learning assumptions and objectives. Table 17 summarizes the two research settings. To enable a consistent incremental evaluation, CAPE-family methods are adapted to the same category orders and incremental splits defined in Appendix A. Unlike standard category-agnostic evaluation settings that focus on disjoint category generalization, this comparison follows the CCIKD protocol and evaluates final performance over cumulative seen categories.

For CAPE-family methods, we consider two stage-wise initialization policies. *Independent pretraining* ("-I") initializes each stage from an ImageNet-pretrained model, whereas *sequential previous* ("-S") initializes each stage from the previous completed checkpoint with a new optimizer. For POISE, we evaluate exemplar-free continual learning with multiple preservation strategies while keeping the persistent output organization unchanged. All methods use the same category orders, cumulative test sets, and evaluation metrics described in Appendix A.3.

Table 18 reports the final cumulative results averaged over category orders. The comparison reflects two approaches for handling heterogeneous schemas: support-conditioned inference and exemplar-free continual acquisition. While CAPE-family methods exploit test-time support examples for category-conditioned prediction, POISE maintains persistent output identities across evolving category streams without additional inference-time support.

On IKDD, CAPE-family methods benefit from sequential initialization, indicating that retaining previous-stage parameters can transfer useful representations between stages. However, support-conditioned methods require category-specific support examples when applied to continual streams,

Table 18: **Comparison with CAPE-methods.** Results (%) are averaged over category orders at the final cumulative stage. Each entry reports AIAPCK, FinalPCK, and BWTPCK. CAPE-family suffix "-I" denotes independent pretraining and "-S" denotes sequential previous-stage initialization.

Method	IKDD-6stage (3 orders)			MP100-2stage (5 orders)			MP100-10stage (3 orders)			MP100-8stage (3 orders)		
	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
FT (POISE)	81.51	75.56	-17.56	88.32	86.19	-6.20	79.13	73.62	-18.97	50.32	36.00	-55.99
Joint (POISE)	89.97	90.55	0.59	90.52	90.60	0.16	90.53	90.61	0.11	89.56	90.45	-0.34
EWC (POISE)	87.24	85.13	-6.29	**90.16**	**89.81**	**-0.93**	**87.25**	**85.97**	**-4.28**	**77.40**	**70.66**	**-18.65**
LwF (POISE)	**87.97**	**86.19**	**-4.22**	89.43	88.43	-2.93	81.78	75.07	-17.21	65.78	47.44	-44.25
PODNet (POISE)	77.73	70.54	-20.98	87.82	85.20	-7.14	73.95	66.33	-24.87	44.16	31.05	-59.53
CapeFormer-I	43.99	35.75	-24.65	74.14	66.08	-21.58	53.72	46.48	-27.15	35.08	15.39	-64.83
CapeFormer-S	56.21	54.19	-14.07	82.87	81.79	-6.83	68.20	67.18	-15.69	47.81	35.62	-50.96
SCAPE-I	57.13	46.09	-32.66	77.42	69.08	-23.09	58.01	48.27	-35.35	35.37	15.11	-70.21
SCAPE-S	68.16	61.94	-20.45	82.89	79.97	-8.87	71.41	66.64	-20.57	49.11	37.11	-50.16

Table 19: Detailed correspondence statistics. Directed rates use ordered category-pair queries (510 and 118,512); reassignment rates use valid same-index prototype comparisons (102 and 1,535).

Quantity	IKDD	MP100
Category–keypoint entries	102	1,565
Evaluable prototypes	102	1,564
Paired reassignment queries	102	1,535
Directed same-index queries	510	118,512
Same-index best-target rate	68.63%	23.01%
Off-diagonal best-target rate	31.37%	76.99%
Uniform-target diagonal reference	5.88%	7.06%
Unrestricted score strictly higher	15/102 (14.7%)	1,207/1,535 (78.6%)
S_{best} Q10–Q90	0.483–0.946	0.720–0.964

whereas POISE maintains persistent output identities without additional inference-time support. Across the more heterogeneous MP100 protocols, POISE maintains persistent output identities while learning new categories sequentially, demonstrating the benefit of compatibility-guided output organization under exemplar-free continual learning.

D.2 Additional Diagnostic Analysis of Identity Conflict

The main paper identifies local-index identity conflict as a key challenge in category-incremental pose estimation and visualizes representative correspondence maps and prototype-incompatibility analysis. This section provides additional quantitative statistics to further characterize the mismatch between category-local keypoint indices and persistent identities. All prototypes are extracted from training observations only and are not constructed using test samples.

The correspondence statistics quantify whether category-local indices provide reliable identity alignment. The same-index best-target rate measures how often the local index agrees with the highest-compatibility target, while the off-diagonal rate captures cases where another identity provides stronger compatibility. The uniform-target diagonal reference indicates the expected same-index agreement under random assignment. The unrestricted comparison measures whether the highest-compatibility cross-category match differs from the fixed local index.

These measurements show that local indices often fail to represent persistent cross-category identities, especially under heterogeneous MP100 schemas. However, they only measure prototype-based compatibility and should not be interpreted as human-annotated semantic correspondence.

The incompatibility load measures the conflict introduced when heterogeneous category keypoints are forced to share fixed output rows. Since the load accumulates non-negative pairwise incompatibilities, larger values indicate greater exposure to potentially conflicting parameter updates. The normalized load is higher for MP100 than IKDD, reflecting its broader schema diversity. The order-level correlation with final performance varies across protocols, suggesting that incompatibility load is a structural indicator of identity conflict rather than a universal predictor of forgetting.

Together, these analyses provide additional evidence that fixed local-index sharing can impose inconsistent output semantics across categories, motivating persistent output identities and compatibility-guided routing for heterogeneous keypoint streams.

Table 20: **Prototype-incompatibility load induced by fixed local-index sharing.** Final load, mean load per pair, and Spearman correlation are order means; transition counts are pooled across the listed orders. The association is descriptive and is not a causal estimate.

Protocol	Trans.	Final load	Mean/pair	ρ
IKDD-6stage	15	93.60	0.367	0.124
MP100-2stage	5	16,884.78	0.669	0.300
MP100-10stage	27	36,058.95	0.669	-0.600
MP100-8stage	21	28,583.33	0.758	0.333

Table 21: **Distillation location analysis under POISE on MP100-10stage.** Results are AIAPCK (%) averaged over three category orders. The comparison evaluates output-, feature-, and combined-level preservation objectives with fixed POISE output organization.

Objective	IKDD-6	MP100-2	MP100-10	MP100-8
FT	81.51	88.32	79.13	50.32
FullFeat	76.43	87.45	73.01	42.71
LwF	87.97	89.43	81.78	65.78
FullFeat+LwF	85.95	89.33	78.36	61.33

Table 22: **Main PCK@0.2 results on IKDD and MP100.** Results (%) are averaged over category orders using PCK@0.2 at every stage. Parenthesized values denote POISE gains over the corresponding Shared-$K_{\max}$ head; red and green denote positive and negative gains, respectively.

Method	Head	IKDD-6stage (3 orders)			MP100-2stage (5 orders)			MP100-10stage (3 orders)			MP100-8stage (3 orders)		
		AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
FT	Shared	91.29	87.24	-10.89	92.59	89.47	-9.15	81.93	74.49	-24.14	48.09	35.11	-62.13
	POISE	91.14 (-0.15)	87.99 (+0.76)	-9.98 (+0.91)	95.05 (+2.46)	93.94 (+4.47)	-3.29 (+5.86)	89.38 (+7.45)	85.92 (+11.43)	-11.40 (+12.73)	61.20 (+13.11)	47.84 (+12.74)	-49.52 (+12.60)
Joint	Shared	96.31	96.48	0.36	95.66	95.67	-0.07	95.88	95.66	-0.25	94.16	95.63	-0.21
	POISE	96.35 (+0.04)	96.52 (+0.04)	0.34 (-0.02)	96.22 (+0.56)	96.27 (+0.60)	0.07 (+0.14)	96.35 (+0.47)	96.27 (+0.61)	0.06 (+0.31)	94.94 (+0.78)	96.16 (+0.52)	-0.20 (+0.01)
EWC	Shared	94.06	92.66	-4.41	94.82	93.99	-2.52	89.84	86.67	-10.17	69.73	54.37	-41.83
	POISE	94.86 (+0.80)	94.17 (+1.51)	-2.64 (+1.76)	96.10 (+1.28)	96.00 (+2.00)	-0.34 (+2.17)	94.80 (+4.95)	94.25 (+7.58)	-1.86 (+8.31)	86.05 (+16.32)	81.47 (+27.10)	-14.32 (+27.51)
LwF	Shared	93.76	91.82	-4.20	94.63	93.62	-3.05	85.48	79.13	-18.70	56.98	40.40	-56.64
	POISE	95.13 (+1.37)	93.98 (+2.16)	-2.34 (+1.86)	95.62 (+0.98)	95.10 (+1.48)	-1.58 (+1.47)	91.10 (+5.61)	86.81 (+7.68)	-10.34 (+8.35)	77.95 (+20.97)	62.13 (+21.73)	-34.92 (+21.72)
PODNet	Shared	87.27	80.73	-16.89	91.60	87.49	-11.62	78.62	68.61	-29.27	45.21	31.87	-64.35
	POISE	87.45 (+0.18)	82.73 (+2.00)	-14.43 (+2.46)	94.64 (+3.04)	93.13 (+5.64)	-4.11 (+7.52)	85.01 (+6.39)	79.49 (+10.88)	-17.26 (+12.02)	53.80 (+8.59)	41.78 (+9.91)	-54.87 (+9.48)

D.3 Distillation Location Analysis

Beyond heatmap-level preservation, we further investigate how the location of the preservation signal affects continual learning under POISE. Different preservation objectives constrain different components of the model. FT serves as the supervised-learning baseline without additional preservation. LwF performs heatmap-level distillation by preserving historical output responses. FullFeat applies feature-level distillation by matching intermediate feature representations, while FullFeat+LwF combines feature-level and heatmap-level preservation.

Table 21 compares these preservation choices on MP100-10stage. The results show that preservation location substantially affects continual keypoint learning under the same POISE output organization. Heatmap-level LwF achieves strong performance in this setting, suggesting that preserving historical output responses is well aligned with the identity-indexed prediction space maintained by POISE. In contrast, FullFeat alone performs worse than FT across the evaluated protocols, indicating that directly transferring feature-level preservation from general continual learning may not be optimal for keypoint detection. This observation suggests that intermediate representations may not directly preserve the output-level semantics required by evolving keypoint identities.

Combining feature-level and heatmap-level preservation introduces an additional optimization trade-off. FullFeat+LwF improves over FullFeat alone but does not consistently outperform output-level preservation alone, suggesting that feature constraints require careful task-specific design for continual keypoint learning. These results highlight the complementary roles of output organization and functional preservation: POISE maintains stable identity correspondence across evolving categories, while the preservation objective determines how previously learned functions are retained. Future continual keypoint learning methods could further explore task-aware feature preservation objectives that explicitly consider evolving output identities and keypoint-specific semantics.

D.4 Additional Results

The main paper reports order-averaged results using mPCK. This section provides additional evaluation details from two perspectives: PCK@0.20 performance and per-order mPCK variation. Table 22 reports the corresponding PCK@0.20 results, while Tables 23, 24, 25, and 26 provide the detailed mPCK results for individual category orders.

The PCK@0.20 results follow the same trend as the mPCK evaluation. Across continual-learning strategies, POISE generally improves over Shared-$K_{\max}$, with larger gains on more heterogeneous

Table 23: **Per-order mPCK on IKDD-6stage.** Each order reports AIAPCK, FinalPCK, and BWTPCK (mPCK, %). Parenthesized values are POISE minus Shared-$K_{\max}$ for the same order and method; red and green denote positive and negative gains.

Method	Head	order1			order2			order3		
		AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
FT	Shared	79.70	80.16	-11.97	82.52	60.74	-35.40	81.95	83.34	-8.21
	POISE	78.78 (-0.92)	80.52 (+0.35)	-11.51 (+0.46)	82.93 (+0.41)	62.28 (+1.54)	-33.55 (+1.85)	82.83 (+0.88)	83.88 (+0.54)	-7.64 (+0.57)
Joint	Shared	90.21	90.39	0.48	89.75	90.44	0.69	89.79	90.46	0.44
	POISE	90.26 (+0.05)	90.54 (+0.15)	0.60 (+0.12)	89.81 (+0.06)	90.46 (+0.02)	0.55 (-0.13)	89.83 (+0.04)	90.65 (+0.19)	0.62 (+0.18)
EWC	Shared	85.11	84.83	-6.53	85.57	72.47	-21.46	85.33	87.76	-3.11
	POISE	87.34 (+2.23)	86.90 (+2.07)	-4.20 (+2.33)	86.87 (+1.29)	79.69 (+7.22)	-12.74 (+8.72)	87.52 (+2.19)	88.80 (+1.04)	-1.94 (+1.17)
LwF	Shared	84.67	82.12	-7.66	86.90	82.30	-8.33	86.56	85.02	-5.08
	POISE	88.26 (+3.59)	86.96 (+4.84)	-3.20 (+4.45)	87.83 (+0.93)	85.26 (+2.95)	-5.35 (+2.99)	87.81 (+1.25)	86.36 (+1.34)	-4.11 (+0.97)
PODNet	Shared	73.15	74.36	-15.99	79.77	54.31	-40.62	78.92	77.71	-12.74
	POISE	73.35 (+0.20)	76.02 (+1.66)	-13.99 (+2.01)	79.96 (+0.18)	56.43 (+2.12)	-38.05 (+2.57)	79.89 (+0.97)	79.18 (+1.47)	-10.90 (+1.84)

Table 24: **Per-order mPCK on MP100-2stage.** Each order reports AIAPCK, FinalPCK, and BWTPCK (mPCK, %). Parenthesized values are POISE minus Shared-$K_{\max}$ for the same order and method; red and green denote positive and negative gains.

Method	Head	split1			split2			split3			split4			split5		
		AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
FT	Shared	85.77	81.99	-11.51	86.73	82.49	-11.05	84.89	80.45	-13.58	86.25	82.44	-10.86	86.79	83.49	-9.85
	POISE	88.26 (+2.48)	86.47 (+4.48)	-5.54 (+5.97)	88.74 (+2.00)	86.25 (+3.76)	-6.20 (+4.85)	87.41 (+2.52)	84.79 (+4.33)	-8.12 (+5.46)	88.61 (+2.35)	86.73 (+4.29)	-5.43 (+5.43)	88.57 (+1.78)	86.69 (+3.20)	-5.70 (+4.15)
Joint	Shared	89.76	89.98	0.01	90.36	89.93	-0.00	89.61	90.01	0.28	89.93	89.88	0.03	90.08	90.00	-0.29
	POISE	90.34 (+0.58)	90.50 (+0.52)	0.16 (+0.15)	90.90 (+0.55)	90.61 (+0.68)	0.09 (+0.10)	90.34 (+0.73)	90.57 (+0.57)	0.04 (-0.24)	90.52 (+0.60)	90.63 (+0.75)	0.31 (+0.27)	90.50 (+0.41)	90.67 (+0.66)	0.18 (+0.48)
EWC	Shared	88.62	87.62	-3.36	89.32	87.92	-2.74	88.39	87.46	-3.43	88.54	87.15	-3.82	89.08	88.09	-3.08
	POISE	90.06 (+1.44)	89.89 (+2.26)	-0.83 (+2.53)	90.49 (+1.16)	89.63 (+1.71)	-1.10 (+1.64)	89.94 (+1.55)	89.69 (+2.23)	-1.18 (+2.26)	90.07 (+1.54)	89.72 (+2.56)	-0.89 (+2.93)	90.27 (+1.19)	90.15 (+2.05)	-0.65 (+2.43)
LwF	Shared	88.03	86.57	-4.84	88.71	86.68	-4.60	87.77	86.19	-5.34	89.07	88.05	-2.62	88.83	87.62	-3.87
	POISE	89.26 (+1.23)	88.42 (+1.85)	-2.79 (+2.04)	89.81 (+1.10)	88.28 (+1.60)	-3.31 (+1.30)	88.92 (+1.15)	87.83 (+1.64)	-3.70 (+1.64)	89.51 (+0.43)	88.65 (+0.60)	-2.53 (+0.09)	89.63 (+0.80)	88.95 (+1.33)	-2.31 (+1.56)
PODNet	Shared	84.30	79.06	-15.22	85.47	80.03	-13.86	83.81	78.25	-16.17	85.37	80.75	-12.74	85.54	81.02	-12.86
	POISE	87.87 (+3.58)	85.70 (+6.65)	-6.27 (+8.95)	88.35 (+2.89)	85.40 (+5.37)	-6.96 (+6.90)	86.62 (+2.80)	83.22 (+4.97)	-9.92 (+6.25)	88.05 (+2.69)	85.65 (+4.91)	-6.38 (+6.35)	88.20 (+2.66)	86.01 (+4.99)	-6.18 (+6.68)

Table 25: **Per-order mPCK on MP100-8stage.** Each order reports AIAPCK, FinalPCK, and BWTPCK (mPCK, %). Parenthesized values are POISE minus Shared-$K_{\max}$ for the same order and method; red and green denote positive and negative gains.

Method	Head	order1			order2			order3		
		AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
FT	Shared	54.97	42.30	-49.51	32.17	13.43	-79.19	32.19	21.54	-69.45
	POISE	65.53 (+10.57)	55.73 (+13.43)	-36.19 (+13.32)	42.77 (+10.61)	20.47 (+7.04)	-72.32 (+6.87)	42.65 (+10.46)	31.81 (+10.28)	-59.44 (+10.01)
Joint	Shared	85.72	89.94	-0.11	90.40	89.93	-0.31	90.41	89.83	-0.59
	POISE	87.18 (+1.46)	90.35 (+0.41)	-0.20 (-0.09)	90.76 (+0.36)	90.51 (+0.57)	-0.37 (-0.06)	90.75 (+0.33)	90.50 (+0.67)	-0.45 (+0.14)
EWC	Shared	71.42	54.69	-33.55	58.76	34.80	-56.73	56.65	47.06	-43.19
	POISE	80.60 (+9.17)	73.88 (+19.20)	-12.76 (+20.78)	76.53 (+17.77)	63.99 (+29.20)	-26.97 (+29.76)	75.08 (+18.43)	74.12 (+27.06)	-16.21 (+26.98)
LwF	Shared	64.99	55.57	-35.87	40.34	15.33	-77.09	39.28	21.19	-69.69
	POISE	78.81 (+13.82)	74.70 (+19.14)	-16.93 (+18.94)	61.07 (+20.73)	34.55 (+19.22)	-57.72 (+19.36)	57.46 (+18.18)	33.06 (+11.86)	-58.11 (+11.58)
PODNet	Shared	52.71	36.68	-52.55	29.51	12.41	-78.97	29.12	17.70	-72.03
	POISE	60.60 (+7.89)	50.39 (+13.70)	-39.11 (+13.44)	35.59 (+6.07)	18.15 (+5.74)	-73.72 (+5.25)	36.29 (+7.17)	24.63 (+6.92)	-65.75 (+6.28)

Table 26: **Per-order mPCK on MP100-10stage.** Each order reports AIAPCK, FinalPCK, and BWTPCK (mPCK, %). Parenthesized values are POISE minus Shared-$K_{\max}$ for the same order and method; red and green denote positive and negative gains.

Method	Head	order1			order2			order3		
		AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
FT	Shared	74.07	64.64	-28.99	71.87	67.10	-26.19	73.08	60.38	-33.57
	POISE	80.64 (+6.57)	74.42 (+9.77)	-18.02 (+10.96)	78.07 (+6.20)	73.23 (+6.13)	-19.39 (+6.80)	78.68 (+5.60)	73.20 (+12.82)	-19.51 (+14.06)
Joint	Shared	90.70	89.94	-0.30	89.07	89.94	-0.21	90.30	89.87	-0.22
	POISE	91.25 (+0.56)	90.62 (+0.68)	0.08 (+0.37)	89.78 (+0.71)	90.64 (+0.70)	0.11 (+0.32)	90.56 (+0.26)	90.57 (+0.70)	0.13 (+0.35)
EWC	Shared	83.48	78.60	-12.09	80.82	78.86	-12.18	81.98	76.58	-14.55
	POISE	87.64 (+4.16)	85.60 (+6.99)	-4.45 (+7.63)	86.75 (+5.93)	86.12 (+7.26)	-4.24 (+7.94)	87.35 (+5.38)	86.20 (+9.62)	-4.15 (+10.41)
LwF	Shared	77.88	68.02	-24.66	75.25	69.34	-23.35	77.94	70.72	-21.73
	POISE	82.77 (+4.89)	75.24 (+7.22)	-17.01 (+7.65)	80.63 (+5.38)	73.23 (+3.89)	-19.27 (+4.08)	81.93 (+3.99)	76.74 (+6.02)	-15.36 (+6.37)
PODNet	Shared	70.81	60.35	-31.35	67.32	60.14	-31.69	68.93	51.70	-41.07
	POISE	75.77 (+4.97)	68.11 (+7.76)	-22.93 (+8.43)	72.57 (+5.25)	64.18 (+4.04)	-27.20 (+4.49)	73.51 (+4.58)	66.71 (+15.01)	-24.48 (+16.59)

MP100 protocols. The improvements become more pronounced for longer streams, where fixed local-index sharing accumulates greater output conflicts. For example, on MP100-8stage, POISE substantially improves FinalPCK and BWTPCK under EWC and LwF, demonstrating improved localization and retention under challenging schema heterogeneity.

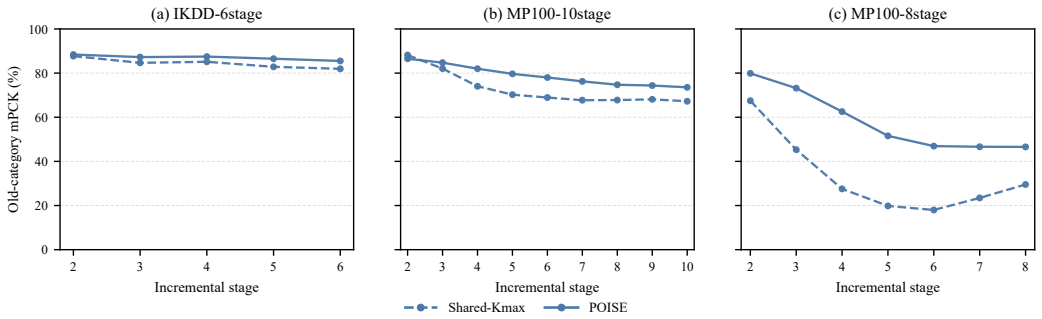


Figure 8: Stage-wise old-category retention analysis for ViTPose with LwF. Dashed and solid curves denote Shared-$K_{\max}$ and POISE, respectively. Results are averaged over category orders within each protocol. POISE maintains higher old-category performance as the incremental stream grows, especially on more heterogeneous long-horizon protocols.

Table 27: Stage-wise mPCK (%) for one representative MP100-10stage order (order2). A, O, and N denote all-seen, old, and new categories.

Stage	Shared			POISE		
	A	O	N	A	O	N
1	87.80	–	87.80	87.81	–	87.81
2	88.75	85.74	91.77	88.60	85.11	92.09
3	83.13	79.64	90.11	86.14	83.46	91.49
4	74.15	70.85	84.05	82.09	82.29	84.20
5	70.45	65.48	90.32	80.59	78.07	90.69
6	69.24	65.04	90.24	78.94	76.70	90.13
7	70.40	66.53	93.61	77.97	75.36	93.67
8	71.49	68.64	91.48	75.06	72.79	91.01
9	67.74	64.73	91.85	75.18	73.05	92.25
10	69.34	66.79	92.32	73.23	71.10	92.39

Table 28: Stage-wise PCK@0.20 (%) for the same representative order (order2). A, O, and N denote all-seen, old, and new categories.

Stage	Shared			POISE		
	A	O	N	A	O	N
1	94.78	–	94.78	94.84	–	94.84
2	95.30	93.51	97.09	95.62	93.85	97.38
3	91.24	89.02	95.70	94.08	92.56	97.12
4	83.08	80.52	90.76	91.81	92.13	90.85
5	79.33	75.08	96.36	90.50	89.03	96.40
6	78.84	75.20	97.06	89.56	88.07	96.99
7	80.18	77.27	97.58	88.71	87.24	97.50
8	81.18	78.97	96.68	86.89	85.51	96.53
9	77.76	75.32	97.22	86.94	85.61	97.60
10	79.34	77.46	96.19	85.37	84.17	96.21

The per-order mPCK results further evaluate the robustness of these improvements across different category arrival trajectories. Across IKDD-6stage, MP100-2stage, MP100-10stage, and MP100-8stage, POISE maintains consistent advantages over Shared-$K_{\max}$ across most orders and continual-learning strategies. The gains are smaller on shorter or less heterogeneous protocols, but increase on longer streams with diverse schemas, where fixed-index sharing faces stronger output conflicts.

These results complement the order-averaged comparisons in the main paper by showing that the improvements are not driven by a particular category ordering. Instead, persistent output organization provides stable benefits across diverse arrival trajectories and continual-learning strategies.

D.5 Analysis of Persistent Output Organization

D.5.1 Retention and Acquisition Behavior

Final-stage performance alone does not reveal how models preserve previously learned categories throughout the incremental process. We therefore analyze the stage-wise retention trajectory by tracking old-category performance after each incremental update. Figure 8 summarizes the old-category retention curves averaged over category orders, while Tables 27 and 28 provide a representative MP100-10stage order with detailed stage-wise results.

The retention trajectories show that POISE provides more stable preservation of previously learned categories than Shared-$K_{\max}$. This behavior is consistent with the output-conflict analysis in Fig. 2, Fig. 3, and Fig. 4: heterogeneous schemas introduce cross-category correspondence mismatch, causing fixed local-index sharing to accumulate incompatible output assignments during continual updates. The difference is limited on IKDD-6stage, where schemas are relatively aligned and such conflicts are weaker, but becomes increasingly pronounced on MP100-10stage and MP100-8stage as incompatible sharing accumulates over longer streams. These results indicate that Output-Identity Collision under fixed-index sharing increases continual interference with previously learned output

Table 29: Single-order retention decomposition on MP100-10stage for one representative order (order2). AIAPCK and FinalPCK follow the definitions in Appendix A.3. Old-category and New-category report final-stage mPCK on previously learned and current-stage category subsets, respectively. Identity bank size and readout parameters indicate the final output capacity.

Variant	AIAPCK↑	FinalPCK↑	Old-category mPCK	New-category mPCK	Final identities	Readout Params.
Shared-$K_{\max}$	75.25	69.34	66.79	92.32	68–68	17.476–17.476
All-Align	76.78	68.86	66.27	92.19	111–111	28.527–28.527
Matched-Random	72.51	59.65	55.97	92.71	111–825	28.527–212.025
Align-Expand	80.83	73.34	71.23	92.35	111–825	28.527–212.025
POISE	80.63	73.23	71.10	92.39	111–825	28.527–212.025
All-Expand	82.47	74.91	72.96	92.39	184–1,565	47.288–402.205

Table 30: POISE route composition for the representative MP100-10stage order. "Within" denotes a legal merge among current-stage queries; the bank count is cumulative.

Stage	Queries	Old Align	Within	Transfer	Expand	Bank
1	184	0	73	0	111	111
2	140	36	46	27	31	169
3	207	44	28	44	91	304
4	136	47	26	44	19	367
5	143	23	29	40	51	458
6	141	34	41	42	24	524
7	160	50	17	46	47	617
8	142	54	46	36	6	659
9	131	48	27	32	24	715
10	181	44	27	51	59	825

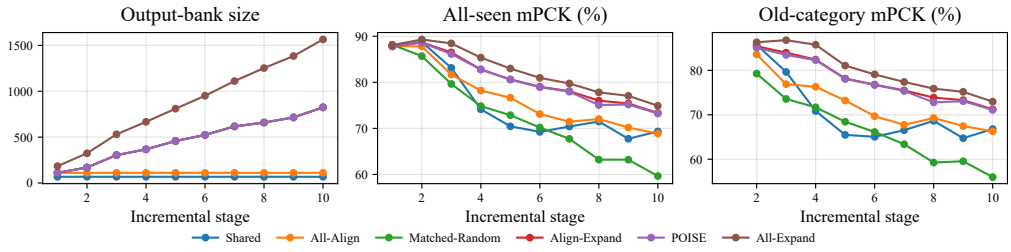


Figure 9: Capacity–accuracy trade-off on the representative MP100-10stage order. Curves compare continual performance under different identity growth strategies using the same cumulative category sets at each stage. The output-bank size denotes the number of persistent identities, and Matched-Random provides the matched-capacity routing baseline.

functions, leading to degraded old-category retention, while compatibility-guided output organization helps mitigate such interference by maintaining more stable cross-category correspondence.

Tables 27 and 28 provide a detailed stage-wise decomposition for a representative MP100-10stage order. At the final stage, POISE improves old-category mPCK from 66.79 to 71.10 while maintaining comparable current-stage performance (92.32 versus 92.39). The same trend is observed under PCK@0.20, where POISE preserves substantially higher old-category accuracy in later stages without reducing adaptation to newly introduced categories.

Overall, the trajectory and stage-wise analyses indicate that persistent output organization primarily mitigates interference among previously learned identities caused by evolving output assignments. Rather than improving current-stage fitting alone, POISE provides a more stable mechanism for maintaining existing output identities throughout the incremental process.

D.5.2 Routing Behavior and Capacity

POISE organizes the output space through compatibility-guided routing rather than unrestricted expansion. Beyond the aggregate routing ablation in the main paper, we further analyze how routing decisions affect retention, acquisition, and output growth on a representative MP100-10stage order.

Table 29 provides a final-stage decomposition of different allocation strategies. At the same output expansion budget of 825 identities, POISE achieves higher FinalPCK than Matched-Random (73.23 vs. 59.65) while maintaining comparable New-category performance (92.39 vs. 92.71).

Table 31: Prototype encoder sensitivity for compatibility estimation. The pose backbone, continual-learning strategy, and routing procedure are fixed, while only the visual encoder used for prototype extraction is changed. Results report AIAPCK, FinalPCK, and BWTPCK (%) on one representative order from IKDD-6stage (order1) and MP100-10stage (order2).

Prototype encoder	IKDD-6stage			MP100-10stage		
	AIAPCK↑	FinalPCK↑	BWTPCK↑	AIAPCK↑	FinalPCK↑	BWTPCK↑
MAE	86.94	85.71	-4.02	78.82	71.84	-21.36
CLIP	85.23	84.94	-4.11	76.65	70.42	-22.84
DINOv3	88.26	86.96	-3.20	80.63	73.23	-19.27

The gain mainly comes from improved Old-category retention (71.10 vs. 55.97), indicating that compatibility-guided routing improves identity allocation rather than simply increasing output capacity. Compared with All-Expand, POISE achieves comparable performance with fewer identities (825 vs. 1,565), demonstrating controlled expansion of the persistent identity space.

Table 30 further illustrates how POISE evolves the identity bank throughout the incremental stream. Rather than relying exclusively on identity reuse or expansion, POISE dynamically combines Historical Align, Within-stage merging, Transfer, and Expand according to incoming category schemas. This behavior enables selective reuse of compatible identities while allocating new identities when existing assignments are insufficient.

Figure 9 visualizes the trade-off between output growth and continual performance. All-Expand provides an upper-bound accuracy reference by assigning separate identities to category-keypoints, but its output bank grows rapidly throughout the stream. In contrast, POISE maintains a smaller identity space with competitive all-seen and old-category performance. Together with the matched-capacity comparison, these results show that POISE improves output-space organization through compatibility-aware allocation rather than unrestricted capacity growth.

For the ViTPose readout, the number of persistent identity rows directly determines the size of the expandable output layer. POISE therefore provides a balance between identity capacity and continual performance by expanding the output space only when compatibility-based reuse is insufficient.

D.5.3 Prototype Encoder Sensitivity

POISE uses frozen visual representations only to construct routing prototypes, while the pose estimator and continual optimization remain unchanged. We therefore investigate whether persistent identity organization is sensitive to the prototype encoder. Table 31 compares different pretrained visual representations under the same backbone and continual-learning configuration.

The results show that POISE remains effective with different prototype encoders, suggesting that the proposed identity organization mechanism is not tied to a specific visual representation. DINOv3 achieves slightly better performance than MAE and CLIP, indicating that stronger visual features can provide more reliable compatibility evidence for cross-category identity assignment. However, the relatively small performance variation across encoders suggests that the main improvements originate from compatibility-guided output organization rather than from the visual encoder alone. These results further support the modular design of POISE, where prototype construction provides routing evidence while continual learning strategies independently optimize the prediction functions.

E Limitations and Discussion

POISE establishes a persistent output organization mechanism for CCIKD and demonstrates its compatibility with diverse continual-learning strategies. This section discusses the assumptions, challenges, and future directions associated with this framework.

E.1 Prototype-Based Identity Compatibility and Output Growth

POISE estimates cross-category identity compatibility from visual and geometric prototypes extracted from training observations. This compatibility provides an evidence-based routing signal rather than human-defined semantic correspondence. Therefore, the quality of identity organization

depends on the available visual and geometric evidence, and improving compatibility estimation under diverse appearance and observation conditions remains an important direction.

The routing safeguards, including compatibility thresholds, margins, member support, and side consistency, improve the reliability of identity reuse, but routing decisions remain dependent on estimated compatibility evidence. The append-only identity space prioritizes stable output correspondence, introducing a trade-off between route persistence and output growth. Future work could investigate adaptive identity migration or pruning mechanisms under explicit resource constraints.

E.2 Toward More General Continual-Learning Strategies for CCIKD

POISE focuses on organizing persistent output identities and provides a unified interface for existing continual-learning strategies. The experiments demonstrate that compatibility-guided output organization complements diverse preservation mechanisms, including parameter regularization, distillation, and parameter-efficient learning.

A promising direction is to further investigate continual-learning strategies that are jointly designed with persistent output organization. Since CCIKD couples heterogeneous output evolution with knowledge preservation, future methods could explicitly model the interaction among identity allocation, representation adaptation, and output expansion to achieve more effective long-term learning.

E.3 Task and Evaluation Scope

The current evaluation follows the top-down CCIKD setting, where each instance is provided with a bounding box and a known category label. This setting allows us to isolate the challenges of continual keypoint learning and persistent output organization from errors introduced by upstream detection, category recognition, and instance association. As a result, the reported improvements specifically measure the ability to organize evolving keypoint outputs under controlled category-aware inference. Extending POISE toward more open-world scenarios, including joint category and keypoint discovery, end-to-end detection, and continual learning with unknown incoming categories, remains an important direction for future research.

The current CCIKD formulation assumes that each arriving category is accompanied by a predefined keypoint schema and is evaluated under predefined category arrival protocols. Under this setting, POISE focuses on organizing persistent output identities across heterogeneous schemas rather than discovering the schemas themselves. A broader open-world continual keypoint detection setting would require jointly addressing category discovery, schema evolution, landmark definition, and persistent output organization. In addition, evaluations with broader category streams, adaptive arrival trajectories, long-tail distributions, and larger-scale independent trials would further characterize the robustness and generality of CCIKD methods.