ROUTABLE PARAMETER MEMORIES FOR EXEMPLAR-FREE LIFELONG PERSON RE-IDENTIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Lifelong person re-identification (LReID) aims to continuously learn from sequential domains while preserving prior identity-discriminative knowledge. However, substantial inter-domain gaps make the model prone to overwriting earlier representations when adapting to new domains, resulting in catastrophic forgetting. Existing methods typically preserve historical knowledge through image replay, knowledge distillation, or distribution rehearsal, but they respectively face the burden of retaining historical data, insufficient activation of old-domain discriminative cues under substantial domain shifts, and difficulty in directly preserving previously learned identity-discriminative transformations through reconstructed historical visual statistics. To overcome these limitations, we propose Routable Parameter Memories (RPM), an exemplar-free LReID framework built on a parameter-memory perspective. Instead of replaying historical images, previous-model responses, or proxy distributions, RPM directly treats the discriminative parameter updates learned from each domain as parameter memories and draws on biological memory processes in which pattern separation reduces interference among similar or overlapping memory traces and cue-driven pattern completion reinstates relevant memories from partial cues. However, storing parameter memories alone does not determine how new knowledge should be written or how stored knowledge should be read for each input. In this way, the lifelong learning process is organized into the writing, storage, and reading of parameter memories. Specifically, Subspace-Guided Writing uses historical activation subspaces to guide the writing of new memories with reduced overlap, after which the learned domain-specific LoRA updates are frozen and sequentially stored in a parameter memory bank. Finally, during inference, Prototype-Guided Reading uses prototype anchors as retrieval cues to adaptively read and compose relevant memories for each input. Extensive experiments show that RPM outperforms existing methods by up to 5.9% on seen domains and 4.4% on unseen domains, respectively.

1 INTRODUCTION

Person re-identification (ReID) aims to identify the same individual across non-overlapping camera views. However, real-world surveillance environments continually evolve as new cameras, scenes, and data sources emerge. Lifelong person re-identification (LReID) is therefore introduced to enable models to continuously learn from sequential domains while preserving previously acquired identity-discriminative knowledge. Because different domains usually exhibit substantial shifts, adapting to a new domain can overwrite representations learned earlier, resulting in catastrophic forgetting. As shown in Figure 1, existing LReID methods typically preserve historical knowledge through image replay, knowledge distillation, or distribution rehearsal. Image replay stores and reuses old-domain images, resulting in a historical-data burden under privacy, storage, and data-governance constraints (Figure 1(a)). Knowledge distillation transfers previous-model responses through current-domain samples, leading to an old-domain activation gap when substantial domain shifts prevent old-domain discriminative cues from being sufficiently expressed (Figure 1(b)). Distribution rehearsal reconstructs historical visual statistics through generated or stylized proxy data, but suffers from a discriminative-preservation gap, as recovering visual distributions does not necessarily preserve previously learned identity-discriminative transformations (Figure 1(c)).

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.

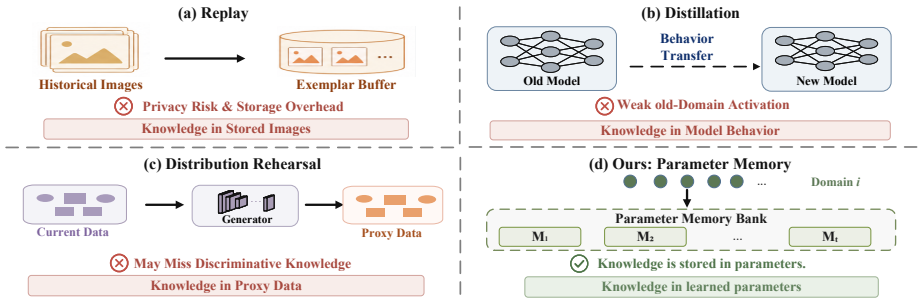

Figure 1: **Comparison of historical knowledge carriers in lifelong ReID.** Existing methods retain historical knowledge through samples or model responses, whereas RPM treats the learned domain-specific parameter updates themselves as historical knowledge carriers that directly encode identity-discriminative knowledge.

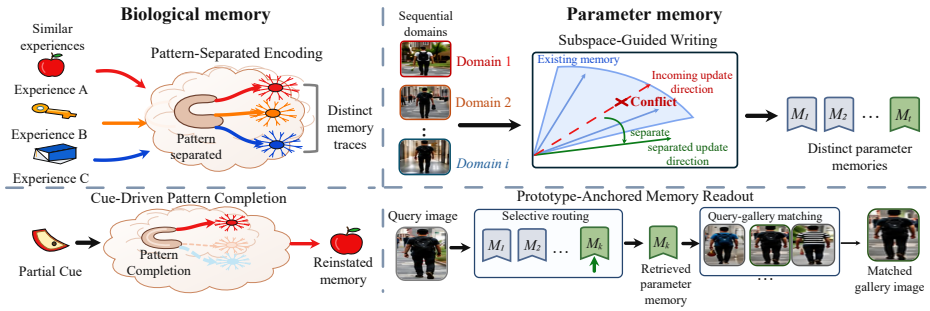

Figure 2: **Biologically inspired interference-reduced memory writing and cue-guided reading mechanisms.** In biological memory, pattern separation transforms similar experiences into distinct memory traces, while cue-driven pattern completion reinstates memories relevant to a partial cue. Drawing on the biological memory processes, RPM uses historical activation subspaces to guide the writing of new LoRA memories with reduced overlap and prototype anchors as visual cues to read relevant parameter memories for each query image.

These limitations lead us to ask a simple yet important question: ***what historical knowledge should be preserved in lifelong learning?*** We observe that the parameters learned at each stage already encode how the model adapts to the corresponding data and the identity-discriminative transformations it acquires. Unlike replayed images, previous-model responses, or proxy distributions, these parameters directly retain the learned transformations themselves and therefore provide a natural carrier of historical knowledge. Then, ***how can such parameter memories be fully exploited throughout lifelong ReID?*** Freezing a memory preserves its parameters, but adding new memories can still change the representation produced for an earlier domain. Moreover, different input images may require different combinations of the accumulated memories. To address this issue, we draw inspiration from the hippocampal computations of pattern separation and completion (Yassa & Stark, 2011; Leutgeb et al., 2007; Favila et al., 2016). As illustrated in Figure 2, the brain uses pattern separation to reduce interference between similar or overlapping memory traces and cue-driven recall to activate relevant memories when needed. This motivates us to formulate lifelong ReID as a process of parameter-memory writing, storage, and reading. During training, new memories are progressively written with reduced interference and preserved across subsequent learning stages. During inference, memories relevant to each input are adaptively read.

Guided by this insight, we propose Routable Parameter Memories (RPM), a biologically inspired exemplar-free LReID framework that organizes learned domain-specific parameter updates as routable parameter memories. During training, Subspace-Guided Writing uses historical activation subspaces to guide the writing of new LoRA memories with reduced overlap, after which each learned LoRA update is frozen and stored as a parameter memory. During inference, prototype anchors serve as retrieval cues for Prototype-Guided Reading, which adaptively reads and composes relevant memories for each input. Extensive experiments under two training orders show that RPM outperforms existing methods by up to 5.9% on seen domains and 4.4% on unseen domains. The contributions are summarized as follows:

(1) We introduce a parameter-memory perspective for exemplar-free lifelong ReID that directly treats the parameters learned at each stage as parameter memories and organizes the lifelong learning process into memory writing, storage, and reading.
(2) Inspired by biological memory mechanisms, we propose RPM, which uses historical activation subspaces to guide the writing of new LoRA memories with reduced overlap, sequentially stores the learned updates as frozen parameter memories, and employs prototype anchors to read and compose relevant memories for each input.
(3) Extensive experiments demonstrate that RPM consistently outperforms existing methods, achieving superior seen-domain retention and unseen-domain generalization.

2 Related Work

2.1 Lifelong Person Re-Identification

Lifelong person re-identification (LReID) learns from sequential domains while retaining previously acquired discriminative knowledge. Early methods mitigate forgetting by replaying stored samples or representative information (Pu et al., 2021; Ge et al., 2022), but retaining person images raises privacy and storage concerns. Exemplar-free methods instead preserve knowledge through decomposition or representation regularization, as in LSTKC, LSTKC++, and DRE (Xu et al., 2024b; 2025b; Liu et al., 2025a); prototypes, alignment, or prompt-guided adaptation, as in DKP, DKP++, and PAEMA (Xu et al., 2024a; Zhou et al., 2025; Li et al., 2024b); or proxy distributions, as in DASK, ADSR, and DKUA (Xu et al., 2025a; Chen et al., 2025; Liu et al., 2025b). These methods preserve history through responses, summaries, or reconstructed distributions. In contrast, RPM directly stores learned parameter updates as parameter memories and reads them according to each input.

2.2 Routing Mechanisms

Routing enables input-conditioned selection or composition of computational modules. Mixture-of-experts methods use gating functions to assign inputs to experts (Shazeer et al., 2017; Fedus et al., 2022; Riquelme et al., 2021), while module routing applies this principle to adapters, prompts, and LoRA modules (Liu et al., 2024; Luo et al., 2024; Wu et al., 2024; Li et al., 2024a). Similarity-based routing selects modules through feature or prototype similarity and is suitable when domain labels are unavailable. RPM follows this paradigm, using prototype anchors to assign input-dependent weights to stored parameter memories.

3 Method

3.1 Problem Setup

In exemplar-free lifelong person re-identification, a model learns sequentially from domains $\mathcal{D} = \{D_1, \ldots, D_T\}$, where $D_t = \{(x_n^t, y_n^t)\}_{n=1}^{N_t}$ is the training set at stage t. Only D_t is accessible at stage t; historical images and instance-level features from earlier domains are unavailable. After the final stage, the model is evaluated on learned domains for knowledge retention and disjoint unseen domains for generalization. The goal is to acquire identity-discriminative knowledge from each incoming domain while retaining prior knowledge without storing historical training samples.

3.2 Framework Overview

As shown in Figure 3, RPM maintains a growing bank of frozen LoRA memories on a shared frozen ViT backbone. At stage t, Subspace-Guided Writing uses historical activation subspaces to guide the input directions of a new memory M_t, reducing overlap with previously occupied directions. After learning D_t, M_t is frozen and stored in the memory bank. During inference, Prototype-Guided Reading compares the current image with stored prototype anchors in the frozen backbone space and converts their similarities into input-dependent weights. These weights control the layer-wise contributions of the stored memories to produce the final ReID representation.

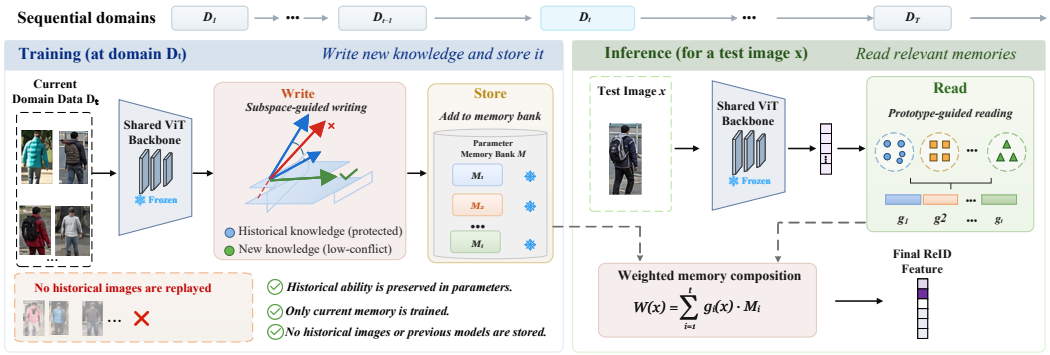

Figure 3: **Overview of RPM.** RPM writes domain knowledge into LoRA memories, freezes them in a memory bank, and reads them with prototype-guided input-dependent weights.

3.3 Store: Frozen LoRA Memories

RPM stores the knowledge learned at each stage as a domain-specific LoRA update on a shared frozen ViT backbone. LoRA modules are inserted into the `qkv`, `proj`, `fc1`, and `fc2` projections. For domain D_i, the update at layer l is

$$\Delta_i^l(h_l) = \frac{\alpha}{r} B_i^l A_i^l h_l, \tag{1}$$

where $A_i^l \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B_i^l \in \mathbb{R}^{d_{\text{out}} \times r}$ are the two low-rank projections, and r and α denote the LoRA rank and scaling factor. Across all L inserted layers, these parameters form the memory of D_i:

$$M_i = \{(A_i^l, B_i^l)\}_{l=1}^{L}. \tag{2}$$

Before learning D_t, the bank $\mathcal{M}_{t-1} = \{M_1, \ldots, M_{t-1}\}$ contains the frozen memories from previous stages. A new memory M_t is learned for D_t while earlier memories remain fixed, then frozen and appended after training. Each memory retains prototype anchors and layer-wise amplitudes for reading. Historical activation bases used to guide future writing are maintained separately as auxiliary states.

3.4 Write: Subspace-Guided Writing

Although historical memories remain frozen, a new memory learned only from current-domain supervision may use input directions that overlap with historical activations. We therefore introduce Subspace-Guided Writing, which uses historical activation subspaces to guide the input directions of each new LoRA memory.

For each inserted layer l, all LoRA updates are disabled and current-domain samples are forwarded through the frozen backbone. Let $h_{t,n}^l$ denote the n-th input-activation vector and S_t the total number of collected vectors. The current activation statistic is

$$C_t^l = \frac{1}{S_t} \sum_{n=1}^{S_t} h_{t,n}^l {h_{t,n}^l}^\top. \tag{3}$$

In practice, it is estimated from current-domain samples using all token activations at each inserted layer.

Each layer maintains an orthonormal historical basis Q_{t-1}^l. The projector onto directions available for the new memory is

$$P_{t-1}^l = I - Q_{t-1}^l {Q_{t-1}^l}^\top. \tag{4}$$

when Q_{t-1}^l stores historical activation directions, and

$$P_{t-1}^l = Q_{t-1}^l {Q_{t-1}^l}^\top. \tag{5}$$

when it directly stores retained writable directions. We then project the current statistic as

$$\widetilde{C}_t^l = P_{t-1}^l C_t^l. \tag{6}$$

For the first domain, $\widetilde{C}_1^l = C_1^l$. Because LoRA-A is initialized from the left singular directions of $\widetilde{C}_t^l$, left projection gives

$$Q_{t-1}^{l\top} \widetilde{C}_t^l = 0 \tag{7}$$

when Q_{t-1}^l represents historical activations, restricting nonzero left singular directions to the complementary subspace. We decompose

$$\widetilde{C}_t^l = U_t^l \Sigma_t^l V_t^{l\top}, \tag{8}$$

and initialize LoRA-A from the first r left singular vectors:

$$A_t^l = \frac{1}{\sqrt{3}} \left(U_{t,:,1}^l\right)^\top. \tag{9}$$

The factor $1/\sqrt{3}$ only matches the LoRA initialization scale. We then freeze A_t^l and optimize only B_t^l. Under the corresponding rank condition, $A_t^l = A_t^l P_{t-1}^l$, so the new LoRA response is governed by the selected input subspace. This is a local directional constraint and does not guarantee unchanged historical representations after adding new memories.

After learning D_t, LoRA updates are again disabled and current-domain activation statistics are incorporated into the historical state through incremental low-rank SVD. This compact state is retained separately from the LoRA memories to guide the next writing stage.

3.5 READ: PROTOTYPE-GUIDED READING

Once the parameter memories have been accumulated, inference requires deciding how much each memory should contribute to a given image. Rather than using a fixed readout for every input, we introduce Prototype-Guided Reading, which compares the current image with stored prototype anchors and converts their similarities into input-dependent memory weights.

After learning domain D_i, we disable all LoRA updates and use the frozen backbone to extract its CLS features. These features are clustered by k-means into K normalized prototype anchors,

$$\mathcal{A}_i = \{a_{i,k}\}_{k=1}^K, \tag{10}$$

which are fixed and stored together with M_i. For a test image x, the same frozen backbone, again with all LoRA updates disabled, produces a reading query $c(x)$. We measure its affinity to memory M_i by the maximum cosine similarity to that memory's prototypes,

$$s_i(x) = \max_k \cos\big(c(x), a_{i,k}\big), \tag{11}$$

and normalize the scores across all stored memories as

$$g_i(x) = \frac{\exp\left(s_i(x)/\tau\right)}{\sum_{j=1}^t \exp\left(s_j(x)/\tau\right)}, \tag{12}$$

where τ controls the sharpness of the resulting weight distribution. Because both queries and anchors are obtained from the same frozen backbone, all comparisons are performed in a shared and stable feature space.

The resulting $g_i(x)$ gives the input-dependent reading weight of memory M_i. The softmax normalization assigns continuous weights across the memory bank, so multiple memories can contribute to the representation of a single image. To modulate their effects at different depths, each memory additionally maintains a layer-wise log-amplitude γ_i^l. At layer l, the stored LoRA updates are composed as

$$h_l' = h_l + \sum_{i=1}^t g_i(x) \exp(\gamma_i^l) \Delta_i^l(h_l). \tag{13}$$

Here, $g_i(x)$ reflects image-level memory relevance, while $\exp(\gamma_i^l)$ adjusts the contribution of M_i at layer l. When a new memory is added, only its layer-wise amplitudes are learned, and the amplitudes of historical memories remain fixed.

Prototype-Guided Reading therefore links each input to the accumulated parameter memories without relying on identity labels. Images from both learned and unseen domains can combine multiple

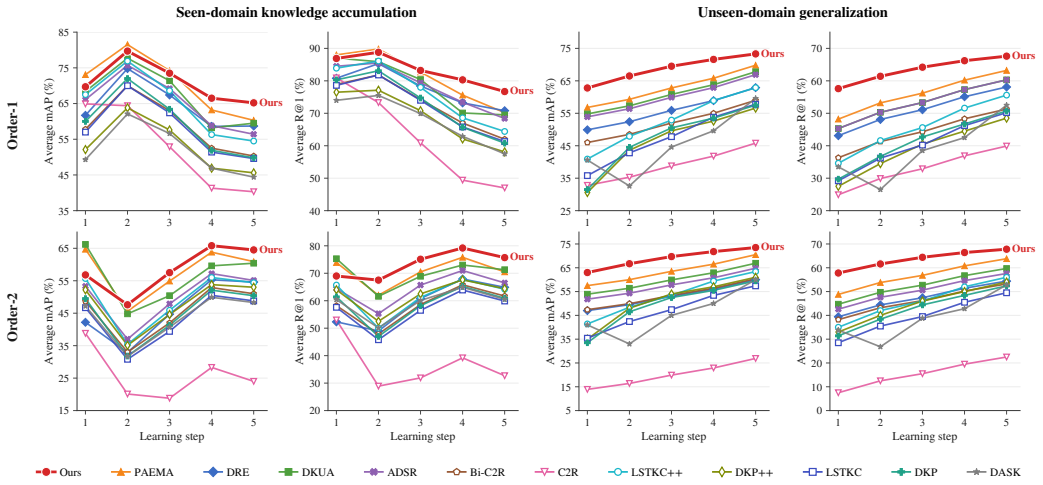

Figure 4: **Learning trends on seen and unseen domains.** *Left*: seen-domain retention across learning stages. *Right*: unseen-domain generalization. RPM consistently maintains superior retention and generalization as learning progresses.

memories according to their visual similarity to the stored prototypes, yielding an input-adaptive ReID representation.

3.6 Training and Inference

At stage t, RPM introduces a new parameter memory M_t for domain D_t, while the shared backbone and all historical memories remain frozen. Subspace-Guided Writing initializes LoRA-A and keeps it fixed. The current LoRA-B, expanded identity classifier, and layer-wise amplitudes $\{\gamma_t^l\}_{l=1}^{L}$ are optimized with the identity classification loss $\mathcal{L}_{ce}$ and batch-hard triplet loss $\mathcal{L}_{tri}$:

$$\mathcal{L}_t = \mathcal{L}_{ce} + \mathcal{L}_{tri}. \tag{14}$$

Historical memories still participate in feature extraction, but gradients are restricted to the current-stage parameters. After training, M_t, its prototype anchors, and layer-wise amplitudes are frozen. Activations from D_t, extracted by the frozen backbone with all LoRA updates disabled, are then used to update the historical writing states. No historical images, replay features, or previous full models are retained.

At inference, the frozen backbone first extracts a reading query with all LoRA memories disabled. Prototype similarities produce input-dependent memory weights, which are then used in a second forward pass to compose the stored LoRA memories and obtain the final retrieval feature.

4 Experiments

4.1 Experimental Setup

Datasets and evaluation protocol. Following VLADR (Xu et al., 2026), we replace withdrawn DukeMTMC-reID with LPW s2. Market-1501, LPW s2, CUHK-SYSU, MSMT17-V2, and CUHK03 are learned sequentially; CUHK01, CUHK02, VIPeR, PRID, i-LIDS, GRID, and SenseReID are used only for unseen-domain evaluation. Historical images and instance-level features are not retained.

Training orders and metrics. We use Market $\rightarrow$ SYSU $\rightarrow$ LPW $\rightarrow$ MSMT17 $\rightarrow$ CUHK03 (Order-1) and LPW $\rightarrow$ MSMT17 $\rightarrow$ Market $\rightarrow$ SYSU $\rightarrow$ CUHK03 (Order-2). We report mAP and Rank-1 accuracy (R@1) for each domain; Seen-Avg and Unseen-Avg measure learned-domain retention and unseen-domain generalization, respectively.

Implementation details. We use an ImageNet-1K pretrained TransReID ViT-Base/16 with 256×128 inputs. The first and subsequent domains are trained for 80 and 60 epochs using AdamW (batch size 32; weight decay 10^{-4}). The LoRA learning rate is 2×10^{-4} for the first domain and 1.2×10^{-4}

Table 1: Comparison results of seen-domain anti-forgetting and unseen-domain generalization on Training Order-1. The **best** and **second-best** results are highlighted in **red** and **blue**, respectively.

Net.	Method	Pub.	Market		SYSU		LPW		MSMT17		CUHK03		Seen-Avg		Unseen-Avg	
			mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1
ResNet	DKP (Xu et al., 2024a)	*CVPR 2024*	60.0	80.3	84.1	85.9	46.0	57.9	17.7	38.5	41.0	41.4	49.8	60.8	57.5	50.7
	C2R (Cui et al., 2024)	*CVPR 2024*	64.9	80.9	63.9	65.8	29.9	36.1	6.4	14.8	36.7	37.4	40.3	47.0	45.8	39.9
	LSTKC (Xu et al., 2024b)	*AAAI 2024*	57.0	78.6	82.9	84.9	47.2	58.4	18.4	41.1	42.3	43.7	49.6	61.3	57.8	50.2
	DKP++ (Zhou et al., 2025)	*TPAMI 2025*	52.1	76.5	75.6	77.7	45.2	58.1	14.9	36.1	40.1	42.2	45.6	58.1	56.6	48.5
	LSTKC++ (Xu et al., 2025b)	*TPAMI 2025*	67.5	83.9	86.3	88.2	52.1	61.9	19.4	40.3	47.0	47.8	54.5	64.4	62.9	55.6
	DASK (Xu et al., 2025a)	*AAAI 2025*	49.3	74.0	74.9	77.0	45.7	58.6	17.6	42.0	34.5	35.6	44.4	57.5	59.6	52.5
	ADSR (Chen et al., 2025)	*ACM MM 2025*	66.1	84.5	85.2	86.6	56.3	67.0	27.6	55.6	46.7	48.2	56.4	68.4	66.9	60.3
	Bi-C2R (Cui et al., 2026)	*TPAMI 2026*	57.8	78.9	82.3	84.7	48.9	59.1	21.1	45.4	41.2	42.1	50.3	62.0	59.0	51.3
ViT	PAEMA (Li et al., 2024b)	*IJCV 2024*	**73.1**	**88.0**	**90.0**	**91.5**	59.5	68.6	30.4	54.4	48.4	48.4	60.3	70.2	69.8	63.2
	DRE (Liu et al., 2025a)	*TNNLS 2025*	61.7	80.8	88.0	89.8	52.6	64.7	32.0	57.3	59.4	61.3	58.7	70.8	62.9	58.1
	DKUA (Liu et al., 2025b)	*arXiv 2026*	68.4	87.0	87.0	84.8	58.7	69.4	18.5	39.1	**65.0**	**67.1**	59.5	69.5	67.8	60.3
	Ours	*This Paper*	69.7	86.9	89.7	90.8	**61.0**	**71.9**	**45.5**	**71.8**	59.9	62.0	**65.2**	**76.7**	**73.3**	**67.6**

Table 2: Comparison results of seen-domain anti-forgetting and unseen-domain generalization on Training Order-2. The **best** and **second-best** results are highlighted in **red** and **blue**, respectively.

Net.	Method	Pub.	LPW		MSMT17		Market		SYSU		CUHK03		Seen-Avg		Unseen-Avg	
			mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1
ResNet	DKP (Xu et al., 2024a)	*CVPR 2024*	49.5	61.4	14.1	32.6	60.3	80.6	84.5	86.4	43.6	43.7	50.4	60.9	59.5	52.4
	C2R (Cui et al., 2024)	*CVPR 2024*	38.8	53.0	1.5	4.9	16.2	37.8	56.8	61.2	6.7	6.4	24.0	32.7	26.9	22.5
	LSTKC (Xu et al., 2024b)	*AAAI 2024*	46.7	57.6	14.9	33.9	56.5	78.0	84.0	86.1	42.1	43.7	48.8	59.9	57.4	49.5
	DKP++ (Zhou et al., 2025)	*TPAMI 2025*	52.2	64.0	18.0	41.0	63.7	82.5	81.3	82.9	49.7	51.8	53.0	64.4	61.0	54.0
	LSTKC++ (Xu et al., 2025b)	*TPAMI 2025*	55.7	65.7	15.0	34.3	66.9	83.6	86.3	87.6	48.4	49.9	54.4	64.2	63.4	56.0
	DASK (Xu et al., 2025a)	*AAAI 2025*	47.2	60.4	16.1	39.9	58.7	79.9	77.5	79.4	42.0	43.0	48.3	60.5	60.0	52.8
	ADSR (Chen et al., 2025)	*ACM MM 2025*	53.3	64.4	20.7	46.2	69.7	86.5	85.2	87.0	46.6	48.2	55.1	66.5	64.8	57.6
	Bi-C2R (Cui et al., 2026)	*TPAMI 2026*	48.7	58.2	17.3	37.5	60.0	80.1	85.5	87.2	43.7	45.6	51.0	61.7	60.4	53.2
ViT	PAEMA (Li et al., 2024b)	*IJCV 2024*	64.7	73.9	26.5	50.0	73.5	87.6	**90.6**	91.5	48.9	49.4	60.9	70.5	70.5	63.8
	DRE (Liu et al., 2025a)	*TNNLS 2025*	42.2	52.3	23.6	45.8	68.2	84.3	87.3	89.1	51.9	53.2	54.6	64.9	60.0	54.4
	DKUA (Liu et al., 2025b)	*arXiv 2026*	**66.2**	**75.3**	23.4	47.9	61.7	83.6	87.1	85.1	**63.7**	**64.7**	60.4	71.3	66.9	59.7
	Ours	*This Paper*	56.8	69.0	**38.4**	**65.9**	**77.2**	**90.3**	**90.6**	**91.7**	59.5	61.6	**64.5**	**75.7**	**73.5**	**67.8**

Table 3: Component ablation on Training Order-1. Results are reported as mAP/R@1.

Setting	Parameter Memory	Subspace-Guided Writing	Prototype-Guided Reading	Seen-Avg	Unseen-Avg
Continual LoRA	×	×	×	58.4 / 72.1	69.8 / 63.2
Fixed All-Memory Reading	✓	×	×	58.8 / 72.2	70.7 / 64.0
+ Subspace-Guided Writing	✓	✓	×	60.8 / 73.5	70.4 / 63.5
+ Prototype-Guided Reading	✓	×	✓	64.1 / 76.0	72.7 / 65.9
RPM (Full)	✓	✓	✓	**65.2 / 76.7**	**73.3 / 67.6**

thereafter; the classifier and routing-amplitude learning rates are 5×10^{-3} and 5×10^{-4}. We set $r = \alpha = 64$, LoRA dropout to 0.1, 16 prototype anchors, and $\tau = 0.20$. The singular-energy threshold increases linearly from 0.95 to 0.99 over five stages. Further details are in the supplementary material.

4.2 Main Results: Seen Retention and Unseen Generalization

Tables 1 and 2 report the results under two training orders using the same protocol. RPM achieves the best Seen-Avg and Unseen-Avg, reaching 65.2/76.7 and 73.3/67.6 on Order-1, and 64.5/75.7 and 73.5/67.8 on Order-2. These results demonstrate strong learned-domain retention and unseen-domain generalization.

4.3 Learning Process Trend

Figure 4 shows the stage-wise performance on seen and unseen domains. RPM maintains a leading trajectory as domains accumulate, with a clearer advantage at later stages, demonstrating strong anti-forgetting ability and unseen-domain generalization.

4.4 Component Ablation

As shown in Table 3, Fixed All-Memory Reading combines all stored memories with input-independent fixed weights, but provides only marginal gains over Continual LoRA. Subspace-Guided Writing improves Seen-Avg to 60.8/73.5, while Prototype-Guided Reading achieves 64.1/76.0 and 72.7/65.9 on Seen-Avg and Unseen-Avg. Combining both components yields the best results of 65.2/76.7 and 73.3/67.6, confirming their complementarity.

4.5 Subspace-Guided Writing Ablation

Table 4 separates the effects of projected initialization and freezing LoRA-A. I_t denotes the immediate mean-mAP decrease on previously learned domains from before to after stage t, and AF-mAP is the average of $I_2, \ldots, I_T$; lower is better.

Table 4: Subspace-Guided Writing ablation on Training Order-1. Results are reported as mAP/R@1. I_t is the immediate mean-mAP decrease on previously learned domains after stage t; lower is better.

Strategy	Writing design			Interference ↓					Performance	
	Proj. Init.	Fix A	Branch	I_2	I_3	I_4	I_5	AF	Seen	Unseen
Direct	×	×	A+B	9.0	**1.2**	3.1	2.1	3.9	64.1/76.0	72.7/65.9
Fixed Rand. A	×	✓	B	7.7	1.4	2.2	0.6	3.0	63.5/75.3	71.9/65.5
Proj. Stats Init.	✓	×	A+B	9.6	1.5	3.3	1.8	4.1	64.8/76.0	72.5/66.5
Subspace-Guided	✓	✓	B	**7.0**	2.0	**1.6**	**0.5**	**2.8**	**65.2/76.7**	**73.3/67.6**

Projected initialization alone improves final performance but does not consistently reduce forgetting. Combining projected initialization with a frozen LoRA-A yields the lowest AF-mAP and the best Seen-Avg and Unseen-Avg, validating the complete writing strategy.

4.6 Historical-Subspace Overlap Diagnosis

For domain t and LoRA layer l, we measure the overlap between the learned LoRA-A A_t^l and the maintained historical writing basis Q_{t-1}^l by

$$E_t^l = \frac{\left\| A_t^l Q_{t-1}^l \right\|_F^2}{\left\| A_t^l \right\|_F^2}. \tag{15}$$

Here, Q_{t-1}^l denotes the orthonormal layer-wise historical writing basis maintained before updating the state with domain t. The overlap is computed after completing stage-t training and before updating the basis using current-domain activations. Lower values indicate weaker overlap, and results are averaged across LoRA layers. As shown in Figure 5, direct writing reaches 0.0501 and 0.0572 on Order-1 and Order-2, respectively, whereas Subspace-Guided Writing remains near zero, demonstrating its effectiveness in reducing overlap with historical writing directions.

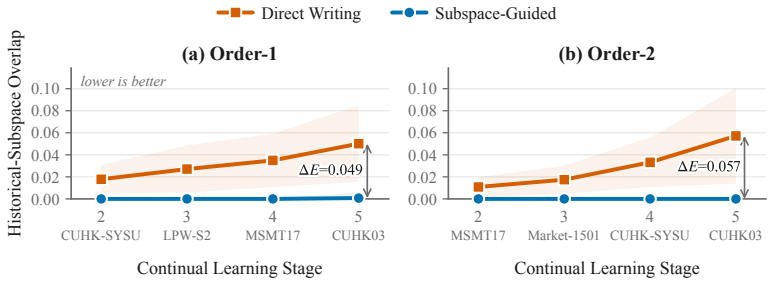

Figure 5: **Historical-subspace overlap during sequential memory writing.** The overlap between the current LoRA-A and the maintained historical writing basis is evaluated at each learning stage. Subspace-Guided Writing keeps this overlap near zero throughout sequential learning.

4.7 Memory Reading Strategy Ablation

Table 5 compares memory reading strategies. Uniform reading applies equal weights, single-anchor reading uses one anchor per memory, and hard reading selects one memory. Hard selection degrades performance, especially on unseen domains. Multi-anchor soft reading performs best, reaching 65.2/76.7 on Seen-Avg and 73.3/67.6 on Unseen-Avg.

4.8 Memory Reading Allocation Visualization

For test domain D_q, the domain-level reading allocation for memory i is $\bar{g}_{q,i} = \frac{1}{|D_q|} \sum_{x \in D_q} g_i(x)$. As shown in Figure 6, seen domains mainly favor their corresponding memories, with related

Table 5: Memory reading strategy ablation. Results are reported as mAP/R@1.

Strategy	Input-Dependent	Anchors	Selected Memories	Soft Composition	Seen-Avg	Unseen-Avg
Uniform Reading w/ Guided Writing	×	–	all	×	60.8 / 73.5	70.4 / 63.5
Single-Anchor Reading	✓	1	all	✓	64.7 / 76.1	72.4 / 66.1
Hard Prototype Reading	✓	16	Top-1	×	61.0 / 72.3	52.8 / 45.2
Soft Prototype Reading	✓	16	all	✓	**65.2 / 76.7**	**73.3 / 67.6**

memories receiving secondary weights. For unseen domains, the allocations are non-uniform and vary across datasets and training orders, demonstrating input-dependent memory composition.

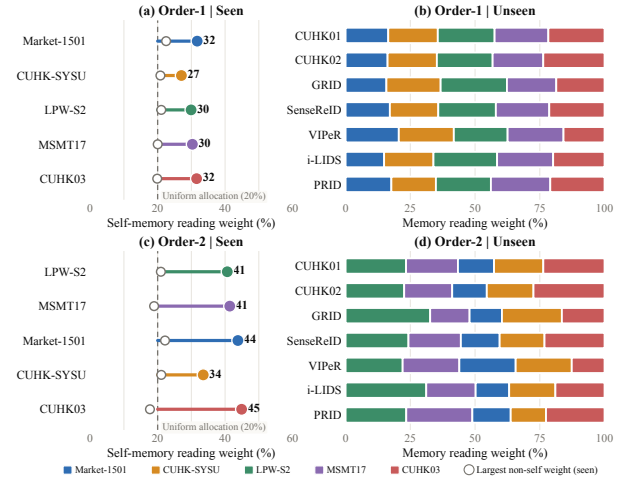

Figure 6: **Input-conditioned memory reading allocations.** Seen domains consistently assign the largest weights to their corresponding memories, while unseen domains form non-uniform combinations of multiple memories.

4.9 t-SNE Feature Distribution Visualization

To visualize the progressive effects of Subspace-Guided Writing and Prototype-Guided Reading, Figure 7 compares Fixed All-Memory Reading, adding Subspace-Guided Writing, and the full RPM. The feature clusters become progressively more compact and better separated as the two components are introduced.

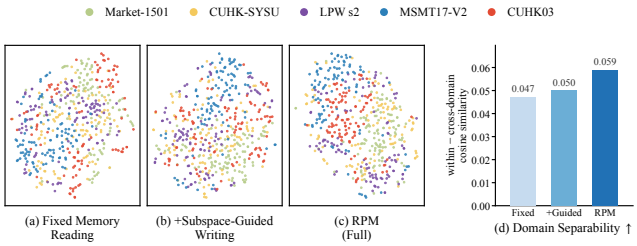

Figure 7: **t-SNE visualization on the five seen domains.** Panels (a)–(c) show Fixed All-Memory Reading, adding Subspace-Guided Writing, and RPM. Panel (d) reports DomainSep, computed as the mean within-domain cosine similarity minus the mean cross-domain cosine similarity on the original 768-dimensional L2-normalized features. Higher DomainSep indicates better domain separability, which progressively improves as the two components are introduced.

5 Conclusion

We presented RPM, an exemplar-free lifelong ReID framework that treats domain-specific LoRA updates as parameter memories and organizes continual learning through memory writing, storage, and reading. Subspace-Guided Writing writes new memories with reduced overlap with historical writing directions, while Prototype-Guided Reading reads and composes relevant memories for each test image. Experiments demonstrate consistent improvements in seen-domain retention and unseen-domain generalization with a small incremental update budget. These results establish compact parameter memories as an effective internal carrier of continual ReID knowledge, without retaining historical images or previous full models, or relying on proxy data or style rehearsal to reconstruct historical distributions.

AI USE STATEMENT

In this work, the authors used generative AI tools only to aid or polish the writing. All AI-assisted text was reviewed by the authors, who take full responsibility for the final content.

REFERENCES

Long Chen, De Cheng, Shizhou Zhang, Yinghui Xing, Di Xu, and Yanning Zhang. Amplitude-aware domain style replay for lifelong person re-identification. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025. doi: 10.1145/3746027.3755616. URL https://doi.org/10.1145/3746027.3755616.

Zhenyu Cui, Jiahuan Zhou, Xun Wang, Manyu Zhu, and Yuxin Peng. Learning continual compatible representation for re-indexing free lifelong person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://github.com/PKU-ICST-MIPL/C2R_CVPR2024.

Zhenyu Cui, Jiahuan Zhou, and Yuxin Peng. Bi-c2r: Bidirectional continual compatible representation for re-indexing free lifelong person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. URL https://arxiv.org/abs/2512.25000.

Serra E. Favila, Avi J. H. Chanales, and Brice A. Kuhl. Experience-dependent hippocampal pattern differentiation prevents interference during subsequent learning. *Nature Communications*, 7:11066, 2016. doi: 10.1038/ncomms11066. URL https://doi.org/10.1038/ncomms11066.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://arxiv.org/abs/2101.03961.

Wenhang Ge, Junlong Du, Ancong Wu, Yuqiao Xian, Ke Yan, Feiyue Huang, and Wei-Shi Zheng. Lifelong person re-identification by pseudo task knowledge preservation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 688–696, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/19946.

Jill K. Leutgeb, Stefan Leutgeb, May-Britt Moser, and Edvard I. Moser. Pattern separation in the dentate gyrus and CA3 of the hippocampus. *Science*, 315(5814):961–966, 2007. doi: 10.1126/science.1135801. URL https://doi.org/10.1126/science.1135801.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts, 2024a. URL https://arxiv.org/abs/2404.15159.

Qiwei Li, Kunlun Xu, Yuxin Peng, and Jiahuan Zhou. Exemplar-free lifelong person re-identification via prompt-guided adaptive knowledge consolidation. *International Journal of Computer Vision*, 132(11):4850–4865, 2024b. doi: 10.1007/s11263-024-02110-x. URL https://doi.org/10.1007/s11263-024-02110-x.

Shiben Liu, Huijie Fan, Qiang Wang, Xiai Chen, Zhi Han, and Yandong Tang. Diverse representations embedding for lifelong person re-identification. *IEEE Transactions on Neural Networks and Learning Systems*, 2025a. URL https://arxiv.org/abs/2403.16003.

Shiben Liu, Mingyue Xu, Huijie Fan, Qiang Wang, Yandong Tang, and Zhi Han. Distribution-aware knowledge unification and association for non-exemplar lifelong person re-identification, 2025b. URL https://arxiv.org/abs/2508.03516.

Tianlin Liu, Mathieu Blondel, Carlos Riquelme, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study, 2024. URL https://arxiv.org/abs/2401.15969.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models, 2024. URL https://arxiv.org/abs/2402.12851.

Nan Pu, Wei Chen, Yu Liu, Erwin M. Bakker, and Michael S. Lew. Lifelong person re-identification via adaptive knowledge accumulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7897–7906, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Pu_Lifelong_Person_Re-Identification_via_Adaptive_Knowledge_Accumulation_CVPR_2021_paper.html.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, Andre Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.05974.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1701.06538.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts, 2024. URL https://arxiv.org/abs/2404.13628.

Kunlun Xu, Xu Zou, Yuxin Peng, and Jiahuan Zhou. Distribution-aware knowledge prototyping for non-exemplar lifelong person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16604–16613, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Xu_Distribution-Aware_Knowledge_Prototyping_for_Non-Exemplar_Lifelong_Person_Re-Identification_CVPR_2024_paper.html.

Kunlun Xu, Xu Zou, and Jiahuan Zhou. Lstkc: Long short-term knowledge consolidation for lifelong person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 16202–16210, 2024b. URL https://ojs.aaai.org/index.php/AAAI/article/view/29514.

Kunlun Xu, Chenghao Jiang, Peixi Xiong, Yuxin Peng, and Jiahuan Zhou. Dask: Distribution rehearsing via adaptive style kernel learning for exemplar-free lifelong person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8915–8923, 2025a. URL https://arxiv.org/abs/2412.09224.

Kunlun Xu, Zichen Liu, Xu Zou, Yuxin Peng, and Jiahuan Zhou. Long short-term knowledge decomposition and consolidation for lifelong person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b. doi: 10.1109/TPAMI.2025.3572468. URL https://doi.org/10.1109/TPAMI.2025.3572468.

Kunlun Xu, Haotong Cheng, Jiangmeng Li, Xu Zou, and Jiahuan Zhou. Vision-language attribute disentanglement and reinforcement for lifelong person re-identification, 2026. URL https://arxiv.org/abs/2603.19678.

Michael A. Yassa and Craig E. L. Stark. Pattern separation in the hippocampus. *Trends in Neurosciences*, 34(10):515–525, 2011. doi: 10.1016/j.tins.2011.06.006. URL https://doi.org/10.1016/j.tins.2011.06.006.

Jiahuan Zhou, Kunlun Xu, Fan Zhuo, Xu Zou, and Yuxin Peng. Distribution-aware knowledge aligning and prototyping for non-exemplar lifelong person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. doi: 10.1109/TPAMI.2025.3597023. URL https://doi.org/10.1109/TPAMI.2025.3597023.

Appendix for RPM

This appendix provides the ethics statement, dataset sources, theoretical analyses, methodological clarifications, additional experimental and qualitative evidence, computing infrastructure, and a reproducibility statement with the core RPM implementation code for Routable Parameter Memories (RPM).

Contents

A. Ethics Statement **13**

B. Dataset Sources **13**

C. Theoretical Feasibility Analysis **13**

C.1 Notation and Parameter-Memory Formulation 13

C.2 Optimal Projection onto the Writing Space 14

C.3 Feasible Low-Rank Parameter-Memory Writing 14

C.4 Preservation of the Writing Constraint 15

C.5 Well-Posed Prototype-Guided Reading 16

C.6 Continuity and Local Stability of Reading 17

C.7 Bounded Composition of Parameter Memories 17

C.8 Storage and Computational Growth 18

C.9 Summary of Theoretical Feasibility 18

D. Supplementary Methodological Clarification **19**

D.1 Mode-Aware Writing-Space Conflict 19

D.2 Functional Memory Interference 20

E. Additional Experiments **21**

E.1 Multi-Seed Stability 21

E.2 Hyperparameter Sensitivity 22

E.3 Top-k Memory Reading 22

E.4 LoRA Capacity and Scaling 23

E.5 Inference Cost and Memory-Bank Scalability 23

E.6 Parameter-Update Budget Analysis 25

E.7 Average Forgetting and Backward Transfer 25

E.8 Per-Unseen-Domain Results 26

F. Qualitative Visualizations **27**

F.1 Retrieval Results 27

G. Computing Infrastructure **27**

H. Reproducibility Statement **27**

References **32**

A Ethics Statement

All experiments use the twelve publicly released academic person re-identification benchmarks listed in Section B under their original research-use terms and evaluation protocols. We conduct no new data collection, annotation, or human-subject study. After each learning stage, the exemplar-free protocol retains no historical person images or instance-level features, but only model parameters, prototype anchors, and compact aggregated activation statistics required for subsequent learning. Although this design reduces the direct retention of historical visual data, it does not provide a formal privacy guarantee, as the retained model states may still encode sensitive information. Moreover, person re-identification remains a biometric technology that may be misused. Any practical deployment should therefore comply with applicable privacy, data-protection, and surveillance regulations and incorporate appropriate access control, auditing, and governance safeguards.

B Dataset Sources

All datasets are used under their original research-use terms and benchmark protocols. The five sequential learning domains are Market-1501 (Zheng et al., 2015), LPW-S2 (Song et al., 2018), CUHK-SYSU (Xiao et al., 2016), MSMT17-V2 (Wei et al., 2018), and CUHK03 (Li et al., 2014). The seven datasets used only for unseen-domain evaluation are CUHK01 (Li et al., 2012), CUHK02 (Li & Wang, 2013), VIPeR (Gray & Tao, 2008), PRID (Hirzer et al., 2011), i-LIDS (Home Office Scientific Development Branch, 2006), GRID (Loy et al., 2010), and SenseReID (Zhao et al., 2017). We follow the dataset splits, training orders, and evaluation protocol described in the main paper and use them consistently across all experiments. No data are newly collected or redistributed in this work.

C Theoretical Feasibility Analysis

This section establishes the theoretical feasibility of RPM from the perspectives of parameter-memory writing, storage, prototype-guided reading, and memory composition. We show that the projected current-domain statistic is the optimal approximation of the original statistic within the active writing space, characterize the feasible rank-r initialization used by the implementation, and prove that freezing LoRA-A preserves its input-space selectivity. We further show that prototype-guided reading is a valid, continuous input-dependent map, that composed parameter-memory contributions are bounded, and that storage and memory-dependent computation grow linearly with the number of stored domains.

The purpose of this analysis is to show that the major components of RPM are mathematically well-defined and mutually consistent under explicit conditions. These results characterize structural properties of parameter-memory writing, storage, and reading rather than providing universal guarantees of zero forgetting or domain generalization.

C.1 Notation and Parameter-Memory Formulation

At learning step t, domain $D_t = \{(x_n^t, y_n^t)\}_{n=1}^{N_t}$ is observed. A frozen ViT backbone f_θ is shared across all domains. For the writing-state update, the implementation uses a current-domain subset $\mathcal{I}_t$ of at most 2,000 images. At each hooked LoRA module, its input tensor is collected before module computation; all token vectors are flattened, so each image can contribute multiple activation vectors. Let S_t be the total number of collected token activations. This sampling controls the cost of state construction and does not retain the samples after the update. For domain D_i, the parameter memory is the frozen LoRA update $M_i = \{(A_i^l, B_i^l)\}_{l=1}^{L}$, where $A_i^l \in \mathbb{R}^{r\times d_{\text{in}}}$ and $B_i^l \in \mathbb{R}^{d_{\text{out}}\times r}$. Each M_i is stored together with its prototype-anchor set $\mathcal{A}_i$ and layer-wise log-amplitudes $\Gamma_i = \{\gamma_i^l\}_{l=1}^{L}$ as associated reading states; these associated states are not part of the LoRA parameter memory itself. At layer l, the input-dependent composed update is

$$h^{l\prime} = h^l + \sum_{i=1}^{t} g_i(x)\exp(\gamma_i^l)\frac{\alpha}{r}B_i^l A_i^l h^l. \tag{1}$$

For every LoRA layer, the implementation maintains an orthonormal basis $Q^l_{t-1} \in \mathbb{R}^{d_{\text{in}} \times k}$ and a mode $m^l_{t-1} \in \{\text{remove}, \text{retain}\}$. Their meanings are deliberately mode-specific. In remove mode, Q^l_{t-1} spans directions excluded from the new write. In retain mode, it spans the compact writing directions retained after a complement switch. The projector used before initialization is therefore

$$P^l_{t-1} = \begin{cases} I - Q^l_{t-1}(Q^l_{t-1})^\top, & m^l_{t-1} = \text{remove}, \\ Q^l_{t-1}(Q^l_{t-1})^\top, & m^l_{t-1} = \text{retain}. \end{cases} \tag{2}$$

For the sampled current-domain activations, the empirical second-moment statistic is $C^l_t = S_t^{-1} \sum_{n=1}^{S_t} h^l_{t,n}(h^l_{t,n})^\top$, and the projected statistic is $\widetilde{C}^l_t = P^l_{t-1} C^l_t$. At the first stage, no historical state exists and the implementation uses $\widetilde{C}^l_1 = C^l_1$. All statements below concern this empirical statistic; they do not require it to equal the population or full-dataset covariance.

C.2 Optimal Projection onto the Writing Space

We first characterize the projection applied to the current activation statistic. For readability, the layer and stage superscripts are omitted within each proof.

Lemma C.1 (Closest statistic satisfying the writing constraint)**.** *Let $P = P^\top = P^2$ be the active writing-space projector and define*

$$\mathcal{S}_P = \{Z \in \mathbb{R}^{d \times d} : PZ = Z\}. \tag{3}$$

For any current statistic C, the projected statistic $\widetilde{C} = PC$ is the unique solution of

$$\min_{Z \in \mathcal{S}_P} \|C - Z\|_F^2. \tag{4}$$

Proof. For any feasible $Z = PZ$, decompose

$$C - Z = (I - P)C + (PC - Z). \tag{5}$$

The two terms are Frobenius-orthogonal because their columns lie in the mutually orthogonal ranges of $I - P$ and P. Consequently,

$$\|C - Z\|_F^2 = \|(I - P)C\|_F^2 + \|PC - Z\|_F^2, \tag{6}$$

which is uniquely minimized by $Z = PC$. □

Lemma C.1 shows that the implementation removes only the component incompatible with the active writing space. It does not introduce an additional surrogate statistic. We next analyze the rank-r directions extracted from this projected statistic.

C.3 Feasible Low-Rank Parameter-Memory Writing

Let $\widetilde{C}^l_t = U^l_t \Sigma^l_t (V^l_t)^\top$ be an SVD. RPM initializes the current LoRA input projection as

$$A^l_t = \frac{1}{\sqrt{3}} \left(U^l_{t,:,1:r}\right)^\top. \tag{7}$$

The $1/\sqrt{3}$ factor matches the initialization scale of the LoRA implementation and does not affect the selected subspace. LoRA-A is then frozen, while the current LoRA-B is optimized with the ReID objective.

Lemma C.2 (Scale of the SVD initialization)**.** *Let $U_r \in \mathbb{R}^{d \times r}$ have orthonormal columns and set $A = U_r^\top / \sqrt{3}$. Then*

$$AA^\top = \frac{1}{3} I_r, \qquad \|A_{j,:}\|_2^2 = \frac{1}{3} \quad (j = 1, \ldots, r). \tag{8}$$

Thus every initialized LoRA-A row has squared norm $1/3$, matching the expected squared row norm of Kaiming-uniform initialization with the PyTorch default $a = \sqrt{5}$.

Proof. Since $U_r^\top U_r = I_r$, $AA^\top = U_r^\top U_r / 3 = I_r/3$. The diagonal entries give the row-wise result. Kaiming-uniform initialization with $a = \sqrt{5}$ has variance $1/(3d)$ per entry, so a row of d independent entries has expected squared norm $d \cdot 1/(3d) = 1/3$. □

Theorem C.3 (Optimal rank-r writing directions)**.** *Assume Q_{t-1}^l is orthonormal and* $\mathrm{rank}(\widetilde{C}_t^l) \geq r$. *Among all row-orthonormal matrices whose rows lie in the active writing space, $(U_{t,:,1:r}^l)^\top$ maximizes the captured projected-statistic energy:*

$$\begin{aligned} \max_{R \in \mathbb{R}^{r \times d}} \quad & \|R\widetilde{C}_t^l\|_F^2, \\ \text{s.t.} \quad & RR^\top = I_r, \qquad R = RP_{t-1}^l. \end{aligned} \tag{9}$$

The maximum equals $\sum_{j=1}^r (\sigma_{t,j}^l)^2$.

Proof. Because $\widetilde{C}_t^l = P_{t-1}^l \widetilde{C}_t^l$, its nonzero left singular vectors lie in $\mathrm{range}(P_{t-1}^l)$. The constrained maximization can therefore be evaluated within this range. By the Ky Fan maximum principle applied to $\widetilde{C}_t^l(\widetilde{C}_t^l)^\top$, the maximizing r-dimensional row space is spanned by the leading r left singular vectors, and the optimum is the sum of the corresponding squared singular values. □

Theorem C.3 gives a precise interpretation of the SVD initialization: among rank-r input projections that obey the writing constraint, it preserves the largest energy of the projected current-domain statistic. Lemma C.2 explains the separate $1/\sqrt{3}$ scaling. If a degenerate projected statistic has fewer than r nonzero singular values, the optimality and exact-selectivity statements below apply to its nonzero singular directions only; the implementation still defines a numerical initialization, but no rank-r subspace guarantee is asserted for an arbitrary null-space completion.

The associated rank-r energy-retention ratio is

$$\rho_t^l(r) = \frac{\left\| (U_{t,:,1:r}^l)^\top \widetilde{C}_t^l \right\|_F^2}{\left\| \widetilde{C}_t^l \right\|_F^2} = \frac{\sum_{j=1}^r (\sigma_{t,j}^l)^2}{\sum_j (\sigma_{t,j}^l)^2}, \tag{10}$$

whenever $\widetilde{C}_t^l \neq 0$. Thus, $\rho_t^l(r)$ exactly measures the fraction of projected current-domain statistic energy retained by the initialized rank-r input directions. This quantity concerns the projected empirical statistic at one layer and one stage; it is not a measure of final ReID accuracy or forgetting.

Proposition C.4 (Mode-consistent initialization)**.** *Under the conditions of Theorem C.3, $A_t^l = A_t^l P_{t-1}^l$. Equivalently,*

$$\begin{cases} A_t^l Q_{t-1}^l = 0, & m_{t-1}^l = \text{remove}, \\ A_t^l \big(I - Q_{t-1}^l (Q_{t-1}^l)^\top\big) = 0, & m_{t-1}^l = \text{retain}. \end{cases} \tag{11}$$

Both identities are preserved throughout stage t because LoRA-A is frozen after initialization.

Proof. The selected left singular vectors lie in $\mathrm{range}(P_{t-1}^l)$, so their transposes satisfy $A_t^l = A_t^l P_{t-1}^l$. Substituting the two cases of Eq. (2) gives Eq. (11). Since the optimizer does not update A_t^l, the identities remain unchanged throughout the stage. □

C.4 Preservation of the Writing Constraint

The previous result concerns the LoRA input projection. Its consequence for the complete low-rank update holds even while LoRA-B is learned.

Corollary C.5 (Writing-space-selective response)**.** *Let*

$$\Delta_t^l(h) = \frac{\alpha}{r} B_t^l A_t^l h. \tag{12}$$

Under Proposition C.4, for any activation h and any value of the learned B_t^l,

$$\Delta_t^l(h) = \Delta_t^l(P_{t-1}^l h). \tag{13}$$

Hence, the new memory is insensitive to activation components in $\ker(P_{t-1}^l)$.

Proof. From $A_t^l = A_t^l P_{t-1}^l$,

$$B_t^l A_t^l h = B_t^l A_t^l P_{t-1}^l h. \tag{14}$$

Multiplication by α/r yields Eq. (13). The equality is independent of B_t^l, so optimizing LoRA-B cannot alter this input-space selectivity. □

In remove mode, Corollary C.5 gives $\Delta_t^l(Q_{t-1}^l(Q_{t-1}^l)^\top h) = 0$. In retain mode, it gives $\Delta_t^l((I - Q_{t-1}^l(Q_{t-1}^l)^\top)h) = 0$. Thus, complement switching changes only how the admissible writing space is stored. It does not change the operational statement that the current memory responds only to $P_{t-1}^l h$. This is a layer-local statement conditional on the incoming activation h. It does not imply orthogonality of complete Transformer representations after residual connections, nonlinearities, or later routed layers.

After each stage, the implementation updates the historical state using a DualGPM-style incremental SVD. With zero-based stage index s, its threshold is $\eta_s = \lambda_b + (\lambda_e - \lambda_b)s/T$, where $\lambda_b = 0.95$, $\lambda_e = 1.00$, and $T = 5$ in the main configuration. Hence, the five reported stages use $\{0.95, 0.96, 0.97, 0.98, 0.99\}$ for $s \in \{0, \ldots, 4\}$; λ_e is the schedule endpoint rather than a threshold instantiated by these five stages. When a remove-mode basis grows beyond half of the activation dimension, the implementation stores its orthogonal complement and changes the mode to retain. This is a compact-state representation change, not a claim that Q_{t-1}^l always denotes excluded historical directions.

For a mode-independent code-level diagnostic, the conflict with the active writing rule is

$$E_t^l = \frac{\|A_t^l(I - P_{t-1}^l)\|_F^2}{\|A_t^l\|_F^2}. \tag{15}$$

In remove mode, Eq. (15) reduces to $\|A_t^l Q_{t-1}^l\|_F^2 / \|A_t^l\|_F^2$. In retain mode, it equals $1 - \|A_t^l Q_{t-1}^l\|_F^2 / \|A_t^l\|_F^2$. Under the rank condition of Theorem C.3, Proposition C.4 implies $E_t^l = 0$ in exact arithmetic. The implementation records this quantity after stage training and before updating the state with current-domain activations. Small nonzero values can arise from finite-precision computation, numerical subspace estimation, or a rank-deficient projected statistic.

C.5 Well-Posed Prototype-Guided Reading

RPM uses deterministic full-memory softmax composition. The reported main configuration sets `top_k`=0 and `floor_alpha`=0, so no Top-k truncation or hard-floor mixing is active. For an input x, the LoRA-disabled backbone feature $c(x)$ is compared with the normalized prototype anchors $\{a_{i,k}\}_{k=1}^K$ of each memory:

$$\begin{aligned} s_i(x) &= \max_k \cos\big(c(x), a_{i,k}\big), \\ g_i(x) &= \frac{\exp(s_i(x)/\tau)}{\sum_{j=1}^t \exp(s_j(x)/\tau)}. \end{aligned} \tag{16}$$

Therefore, $g_i(x) \geq 0$ and $\sum_i g_i(x) = 1$. The second forward pass evaluates each stored LoRA memory and scales its layer-l update by $g_i(x)\exp(\gamma_i^l)$. No Bernoulli sampling, stochastic gate, or domain-adaptation surrogate is used.

Proposition C.6 (Temperature-controlled reading)**.** *For every $\tau > 0$, $g(x)$ lies on the probability simplex. If memory i^* has a unique maximum score, then*

$$\lim_{\tau \to 0^+} g_{i^*}(x) = 1, \qquad \lim_{\tau \to 0^+} g_j(x) = 0 \quad (j \neq i^*). \tag{17}$$

If all t memories are available, then $\lim_{\tau\to\infty} g_i(x) = 1/t$ for every i.

Proof. Nonnegativity and unit sum follow directly from softmax. Dividing numerator and denominator by $\exp(s_{i^*}/\tau)$ shows that every nonmaximal term vanishes as $\tau \to 0^+$. As $\tau \to \infty$, each $\exp(s_i/\tau)$ converges to one, giving the uniform distribution. □

This proposition formalizes the role of τ: smaller values approach hard memory reading, while larger values approach input-independent averaging. The selected $\tau = 0.20$ retains soft composition, as supported by the sensitivity results in Appendix E.2.

C.6 Continuity and Local Stability of Reading

Proposition C.7 (Stability of prototype routing)**.** *Let q and q' be two unit-normalized routing queries, and assume all prototype anchors are unit normalized. For t stored memories under fixed-temperature full-memory routing,*

$$\|g(q) - g(q')\|_2 \leq \frac{\sqrt{t}}{\tau}\|q - q'\|_2. \tag{18}$$

Proof. For each memory, the maximum over anchors is nonexpansive up to the query distance:

$$|s_i(q) - s_i(q')| \leq \max_k |(q - q')^\top a_{i,k}| \leq \|q - q'\|_2. \tag{19}$$

Thus, $\|s(q) - s(q')\|_2 \leq \sqrt{t}\|q - q'\|_2$. The Jacobian of softmax is $\mathrm{diag}(g) - gg^\top$, whose spectral norm is at most one. Applying the mean-value theorem to $\mathrm{softmax}(s/\tau)$ proves Eq. (18). □

Proposition C.7 establishes continuity of reading in the shared frozen feature space. It is a deterministic sensitivity statement and should not be interpreted as an adversarial robustness guarantee.

Corollary C.8 (Local stability of the dominant memory)**.** *Suppose memory $i^\star$ uniquely maximizes the prototype score at a unit-normalized query q, and let $m(q) = s_{i^\star}(q) - \max_{j \neq i^\star} s_j(q) > 0$. If $\|q - q'\|_2 < m(q)/2$, then $i^\star$ remains the unique highest-scoring memory at q'.*

Proof. By Proposition C.7, each memory score changes by at most $\|q - q'\|_2$. The gap between the dominant memory and any competitor can therefore decrease by at most $2\|q - q'\|_2$, which remains strictly smaller than $m(q)$ under the stated condition. □

C.7 Bounded Composition of Parameter Memories

Proposition C.9 (Bounded routed composition)**.** *Define the routed low-rank contribution at layer l as*

$$\delta^l(x, h) = \sum_{i=1}^{t} g_i(x) \exp(\gamma_i^l) \frac{\alpha}{r} B_i^l A_i^l h. \tag{20}$$

Let $\beta_i^l = \exp(\gamma_i^l)(\alpha/r)\|B_i^l\|_2\|A_i^l\|_2$. Then

$$\|\delta^l(x, h)\|_2 \leq \left(\sum_{i=1}^{t} g_i(x)\beta_i^l\right) \|h\|_2 \leq \max_i \beta_i^l \|h\|_2. \tag{21}$$

Proof. Apply the triangle inequality and submultiplicativity of the spectral norm to each memory update. The second inequality follows because $g(x)$ is a convex distribution over memories. □

The result shows that soft reading does not sum unnormalized memory contributions. Its layer-wise update is controlled by a convex average of the individual memory bounds, with the learned positive amplitudes included in β_i^l.

Queries and anchors are extracted by the same LoRA-disabled frozen backbone, so their similarities are computed in one stable feature space. The allocation in Figure 6 of the main paper is the descriptive domain average $\bar{g}_{q,i} = |D_q|^{-1} \sum_{x \in D_q} g_i(x)$, rather than an estimate of a stochastic gating distribution.

C.8 Storage and Computational Growth

The implementation performs one LoRA-disabled query pass and one routed feature pass. Let n_l be the number of token activations at inserted layer l, d^l_{in} and d^l_{out} its projection dimensions, d the routing-feature dimension, and b the bytes per stored scalar.

Proposition C.10 (Linear parameter-memory growth and bounded writing state)**.** *Let $K_i \le K$ be the number of anchors stored for memory i. With t memories, the parameter-memory bank contains*

$$N_{\text{bank}}(t) = t\left[\sum_{l=1}^{L} r(d^l_{\text{in}} + d^l_{\text{out}}) + L\right] + d\sum_{i=1}^{t} K_i \tag{22}$$

stored scalars for LoRA parameters, prototype anchors, and layer-wise amplitudes. Hence $N_{\text{bank}}(t) \le t[\sum_l r(d^l_{\text{in}} + d^l_{\text{out}}) + Kd + L]$. Separately, the shared writing state stores one basis, one auxiliary coefficient vector, and $O(1)$ metadata per layer. If $k_l(t)$ is its retained width at layer l, then

$$N_{\text{write}}(t) = \sum_{l=1}^{L} \left[(d^l_{\text{in}} + 1)k_l(t) + O(1)\right]. \tag{23}$$

The implementation enforces $k_l(t) \le r_{\text{cap}}$, so $N_{\text{write}}(t)$ is bounded for a fixed architecture and rank cap, whereas $N_{\text{bank}}(t)$ grows linearly with the number of learned domains. The full persistent state has $N_{\text{bank}}(t) + N_{\text{write}}(t)$ scalars up to metadata. Under a common storage precision of b bytes per scalar, its size is

$$\text{Storage}_{\text{MB}}(t) = \frac{b}{2^{20}}\left[N_{\text{bank}}(t) + N_{\text{write}}(t)\right]. \tag{24}$$

The routed pass adds

$$\Theta\left(t\sum_{l=1}^{L} n_l r(d^l_{\text{in}} + d^l_{\text{out}})\right) \tag{25}$$

low-rank multiply-adds, while prototype comparison requires $\Theta(tKd)$ operations per query. Both quantities grow linearly with t.

Proof. At layer l, one memory stores rd^l_{in} entries in LoRA-A and rd^l_{out} entries in LoRA-B. Memory i additionally stores K_i anchors of dimension d and one amplitude for every inserted layer, which gives Eq. (22). The writing state is not replicated per memory: at layer l its orthonormal basis has $d^l_{\text{in}}k_l(t)$ entries and its auxiliary coefficient vector has $k_l(t)$ entries, yielding Eq. (23). Applying LoRA-A and LoRA-B to n_l activations costs order $n_l r(d^l_{\text{in}} + d^l_{\text{out}})$ per memory and layer. Comparing a query with at most tK anchors costs order tKd. Summing over memories and layers yields Eq. (25). □

Therefore, the end-to-end inference cost consists of two backbone passes plus terms linear in the number of stored memories. Appendix E.5 reports measured query extraction, prototype comparison, routed extraction, latency, peak memory, and memory-bank growth. We make no constant-cost inference claim, and distinguish the linearly growing parameter-memory bank from the bounded shared writing state.

C.9 Summary of Theoretical Feasibility

The preceding analysis establishes the following theoretical feasibility chain for RPM:

1. The projected statistic is the unique closest approximation satisfying the active writing-space constraint.
2. Under the stated rank condition, the leading singular directions provide an energy-optimal rank-r initialization within that space.
3. Freezing LoRA-A preserves the resulting input-space selectivity while LoRA-B learns the domain-specific output transformation.
4. Prototype-Guided Reading defines a deterministic probability distribution over stored memories, is continuous in the reading query, and preserves a dominant memory locally when its prototype-score margin is positive.

5. The routed low-rank contribution is bounded by a convex combination of the individual memory bounds.
6. The parameter-memory bank and memory-dependent computation grow linearly with the number of learned domains, while the shared writing state remains bounded under a fixed rank cap.

These results establish well-posedness and structural consistency of the implemented writing, storage, and reading operations. They do not establish pairwise orthogonality between complete trained memories, global orthogonality of final Transformer features, zero output correlation after learning LoRA-B, zero forgetting, correct domain assignment, or distribution-free generalization. Those outcomes also depend on optimization, the data sequence, prototype quality, and learned LoRA-B transformations.

The empirical evidence is reported separately. Table 4 in the main paper evaluates functional interference and final retention for writing variants. Figure 5 reports the remove-mode historical-subspace overlap used by all reported diagnostic runs; Appendix D.1 gives the mode-aware generalization in Eq. (15). Table 5 and Figure 6 evaluate reading choices and their input-dependent allocations. Appendix E.5 verifies the storage and inference-growth analysis with measured costs. Here, *pattern separation* and *cue-driven recall* denote operational correspondences to Subspace-Guided Writing and Prototype-Guided Reading.

D Supplementary Methodological Clarification

D.1 Mode-Aware Writing-Space Conflict Under Complement Switching

Section 3.4 of the main paper introduces the compact layer-wise state used by RPM to constrain the writing of a new parameter memory before learning domain t. At layer l, this state consists of an orthonormal basis Q^l_{t-1} and a representation mode

$$m^l_{t-1} \in \{\text{remove}, \text{retain}\}. \tag{26}$$

The state admits two equivalent representations of the active writing space. The semantic meaning of Q^l_{t-1} depends on the active mode, whereas the projector P^l_{t-1} always denotes the input directions available for forming the new parameter memory.

In *remove* mode, the columns of Q^l_{t-1} span the historical activation directions excluded from the new write. The available writing-space projector is therefore

$$P^l_{t-1} = I - Q^l_{t-1} {Q^l_{t-1}}^\top. \tag{27}$$

When the remove-mode basis would exceed half of the activation dimension, RPM replaces it with an orthonormal basis of its complementary subspace and switches to *retain* mode. In retain mode, the columns of Q^l_{t-1} directly span the compact space available for writing, and

$$P^l_{t-1} = Q^l_{t-1} {Q^l_{t-1}}^\top. \tag{28}$$

Although the meaning of Q^l_{t-1} changes after complement switching, the meaning of P^l_{t-1} remains unchanged: it always projects onto the directions available for forming the new parameter memory. In both modes,

$${P^l_{t-1}}^\top = P^l_{t-1}, \qquad {P^l_{t-1}}^2 = P^l_{t-1}. \tag{29}$$

To show that complement switching preserves the represented writing space, let Q^l_{rem} denote the remove-mode basis immediately before switching, and let Q^l_{ret} be an orthonormal basis of its orthogonal complement. Then,

$$Q^l_{\text{ret}} {Q^l_{\text{ret}}}^\top = I - Q^l_{\text{rem}} {Q^l_{\text{rem}}}^\top. \tag{30}$$

Therefore, the writing-space projectors before and after switching satisfy

$$P^l_{\text{retain}} = Q^l_{\text{ret}} {Q^l_{\text{ret}}}^\top = I - Q^l_{\text{rem}} {Q^l_{\text{rem}}}^\top = P^l_{\text{remove}}. \tag{31}$$

Complement switching is thus a change in the compact representation of the writing space rather than a change in the writing rule itself.

Based on this mode-invariant projector, we define the normalized writing-space conflict of the current LoRA-A as

$$E_t^l = \frac{\left\|A_t^l(I - P_{t-1}^l)\right\|_F^2}{\left\|A_t^l\right\|_F^2}. \tag{32}$$

The numerator measures the squared LoRA-A energy outside the space permitted by the historical writing state. Since P_{t-1}^l is an orthogonal projector,

$$A_t^l = A_t^l P_{t-1}^l + A_t^l(I - P_{t-1}^l) \tag{33}$$

is an orthogonal decomposition under the Frobenius inner product. Consequently,

$$0 \le E_t^l \le 1, \qquad E_t^l = 0 \iff A_t^l = A_t^l P_{t-1}^l. \tag{34}$$

Thus, zero conflict is exactly equivalent to complete adherence to the active writing-space constraint established in Section 3.4.

In remove mode,

$$I - P_{t-1}^l = Q_{t-1}^l {Q_{t-1}^l}^\top. \tag{35}$$

Since Q_{t-1}^l has orthonormal columns,

$$\left\|A_t^l Q_{t-1}^l {Q_{t-1}^l}^\top\right\|_F^2 = \left\|A_t^l Q_{t-1}^l\right\|_F^2, \tag{36}$$

and hence

$$E_t^l = \frac{\left\|A_t^l Q_{t-1}^l\right\|_F^2}{\left\|A_t^l\right\|_F^2}. \tag{37}$$

In retain mode, $P_{t-1}^l = Q_{t-1}^l {Q_{t-1}^l}^\top$, and therefore

$$E_t^l = \frac{\left\|A_t^l\left[I - Q_{t-1}^l {Q_{t-1}^l}^\top\right]\right\|_F^2}{\left\|A_t^l\right\|_F^2} = 1 - \frac{\left\|A_t^l Q_{t-1}^l\right\|_F^2}{\left\|A_t^l\right\|_F^2}. \tag{38}$$

The two mode-specific expressions therefore measure the same mode-independent quantity: the fraction of LoRA-A energy lying outside the writing space permitted by the historical state.

For the five-stage configurations reported in the main paper, the recorded writing-state logs show that every diagnosed layer remains in remove mode. Accordingly, the historical-subspace overlap reported in Section 4.6 and Figure 5 of the main paper,

$$\frac{\left\|A_t^l Q_{t-1}^l\right\|_F^2}{\left\|A_t^l\right\|_F^2}, \tag{39}$$

is exactly the general conflict measure E_t^l under the active mode of all reported diagnostic runs. We compute this value after completing stage-t training and before updating the writing state with current-domain activations, and then average it across the inserted LoRA layers.

The retain-mode expression is included to make the conflict metric valid when complement switching is activated in longer learning sequences or alternative configurations. It is not presented as an additional empirical effect in the reported five-stage experiments. Finally, E_t^l is a geometric diagnostic of adherence to the writing-space constraint. It does not directly measure the functional performance change caused by inserting a new parameter memory, which is evaluated separately in Section D.2.

D.2 Functional Memory Interference

Section 4.6 and Figure 5 of the main paper analyze the geometric relationship between a newly formed LoRA-A and the historical writing state. As clarified in Section D.1, the writing-space conflict

metric measures the fraction of LoRA-A energy lying outside the input space permitted by the historical state. However, this geometric diagnostic does not directly quantify how inserting a new parameter memory changes retrieval performance on previously learned domains. To complement the geometric analysis, Table 4 of the main paper evaluates functional memory interference under the same writing-strategy comparison.

Let $R_{t,j}$ denote the mAP obtained on previously learned domain j after completing learning stage t. Immediately before learning domain t, the corresponding historical-domain performance is $R_{t-1,j}$. We define the immediate functional interference introduced by writing the t-th parameter memory as

$$I_t = \frac{1}{t-1} \sum_{j=1}^{t-1} \left(R_{t-1,j} - R_{t,j} \right), \quad t > 1. \tag{40}$$

This quantity measures the average stage-wise change in historical-domain performance after the new memory is inserted and the complete routed memory bank is used for evaluation. Since $R_{t,j}$ is measured in mAP, I_t is reported in mAP points. Its sign has a direct interpretation:

$$\begin{cases} I_t > 0, & \text{historical performance decreases,} \\ I_t = 0, & \text{historical performance is unchanged,} \\ I_t < 0, & \text{historical performance improves.} \end{cases} \tag{41}$$

Thus, a smaller value indicates less immediate functional interference. Unlike the geometric conflict metric E_t^l, which is defined separately for each LoRA layer, I_t evaluates the end-to-end behavior of the complete routed model across all previously learned domains.

All evaluations use the complete prototype-routed parameter-memory bank after learning domain t. For each historical-domain query, the routing mechanism computes input-dependent weights over all available memories, and the resulting composed model is evaluated using the same retrieval protocol as in the main experiments. No historical training images or instance-level features are retained or replayed during the insertion of the new memory.

We compare the same four writing strategies reported in Table 4 of the main paper. For each strategy, I_t isolates the immediate historical-performance change introduced at stage t, whereas Average Forgetting (AF) summarizes the accumulated loss of historical performance over the complete learning sequence. Lower values of both I_t and AF indicate stronger preservation of previously learned behavior, but they describe complementary aspects: I_t measures the immediate functional effect of the current memory write, whereas AF measures the cumulative forgetting observed across the sequential learning process.

The functional-interference results are integrated into Table 4 of the main paper together with the writing-design ablation. Reading these results jointly with the historical-subspace overlap in Figure 5 provides two complementary views of parameter-memory writing: *geometric adherence* and *functional preservation*. The geometric metric determines whether the new LoRA-A follows the historical writing-space constraint, while I_t determines whether inserting the resulting memory changes historical-domain retrieval performance. A low geometric conflict therefore provides evidence that the writing rule is followed, whereas a low I_t provides direct evidence that the completed memory write better preserves historical model behavior. Neither quantity replaces the final seen-domain retention results; instead, they explain the writing mechanism from complementary geometric and functional perspectives.

E Additional Experiments

E.1 Multi-Seed Stability

We repeat the full five-stage protocol under both Training Order-1 and Training Order-2 using the same ten pre-specified seeds: 55, 128, 1111, 2723, 6198, 8380, 14018, 25696, 3407, and 444. Table A1 reports the final Seen-Avg and Unseen-Avg mean $\pm$ sample standard deviation over ten runs for each training order under the main protocol. The low variation across both orders supports the stability of RPM under different random seeds.

Seen mAP	Seen R@1	Unseen mAP	Unseen R@1
Order-1			
65.20 ± 0.11	76.73 ± 0.26	73.31 ± 0.37	67.62 ± 0.58
Order-2			
64.50 ± 0.10	75.66 ± 0.21	72.60 ± 0.35	66.38 ± 0.43

Table A1: Multi-seed stability of RPM under both training orders. Results are reported as mean $\pm$ sample standard deviation over ten pre-specified seeds at the final learning stage.

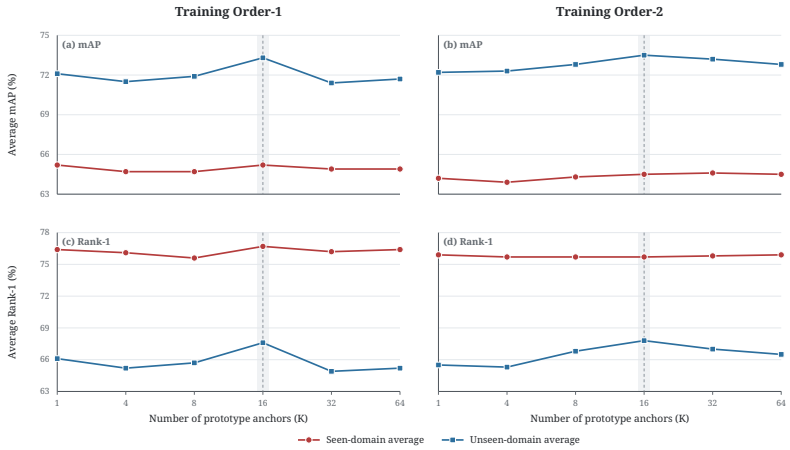

Figure A1: Sensitivity to the number of prototype anchors K under Training Order-1 and Training Order-2. The shaded bands mark the selected setting, $K = 16$.



Figure A2: Sensitivity to the routing temperature τ under Training Order-1 and Training Order-2. The shaded bands mark the selected setting, $\tau = 0.20$.

E.2 Hyperparameter Sensitivity

Figures A1 and A2 report sensitivity to the number of prototype anchors K and routing temperature τ under both training orders. The results remain stable over a broad range of values. The main-paper setting, $(K, \tau) = (16, 0.20)$, lies within this stable region for both orders.

E.3 Top-k Memory Reading

We further evaluate whether inference can be restricted to only the Top-k routed parameter memories. This experiment applies post-hoc Top-k truncation to a model trained with full soft memory composition. During evaluation, only the k memories with the largest routing weights are retained and

Top-k	Seen mAP	Seen R@1	Unseen mAP	Unseen R@1
1	36.4	47.3	17.4	14.7
2	35.5	55.9	23.6	21.4
3	44.1	64.0	41.0	36.9
All	**65.2**	**76.7**	**73.3**	**67.6**

Table A2: Post-hoc Top-k memory reading on Training Order-1. The model is trained with full soft memory composition, and only the test-time reading is truncated.

r	α	Seen mAP	Seen R@1	Unseen mAP	Unseen R@1
8	8	56.8	69.5	67.8	60.9
16	16	60.6	72.6	69.2	61.9
32	32	63.2	74.7	71.5	64.8
64	64	65.2	76.7	**73.3**	**67.6**
96	96	65.7	77.1	72.4	66.6
128	128	**66.1**	**77.6**	72.6	66.1

Table A3: Effect of the LoRA rank, with $\alpha = r$, on Training Order-1.

r	α	α/r	Seen mAP	Seen R@1	Unseen mAP	Unseen R@1
64	16	0.25	62.1	73.7	70.7	64.0
64	32	0.50	64.1	75.6	72.0	66.2
64	64	1.00	65.2	76.7	**73.3**	**67.6**
64	128	2.00	**65.5**	76.8	73.1	67.1
64	256	4.00	65.3	**77.2**	72.8	65.8

Table A4: Effect of the LoRA scaling factor, with $r = 64$, on Training Order-1.

renormalized, while the learned memory bank, prototype anchors, and routing temperature remain unchanged.

Performance generally improves as more memories are retained, while full soft composition achieves the best results on both seen and unseen domains. This indicates that multiple parameter memories provide complementary information and that aggressively restricting the reading discards useful cross-memory contributions. The Top-1 result in Table A2 is not directly comparable to the Hard Prototype Reading variant in Table 5 of the main paper: here Top-1 is applied after training with full soft composition, whereas Hard Prototype Reading is trained and evaluated with one-hot Top-1 reading throughout.

E.4 LoRA Memory Capacity and Scaling

We evaluate the effects of the LoRA rank r and scaling factor α. For the rank sweep, we set $r \in \{8, 16, 32, 64, 96, 128\}$ and maintain $\alpha = r$, such that the scaling ratio α/r remains one.

Increasing the rank consistently improves seen-domain performance, but the unseen-domain results peak at $r = 64$. Larger ranks provide only minor gains on seen domains while increasing the persistent memory size and slightly reducing unseen-domain generalization.

We further fix $r = 64$ and vary $\alpha \in \{16, 32, 64, 128, 256\}$, corresponding to scaling ratios $\alpha/r \in \{0.25, 0.5, 1, 2, 4\}$.

Performance improves as the scaling ratio increases from 0.25 to 1, whereas larger ratios provide no consistent benefit and slightly reduce unseen-domain performance. We therefore use $r = \alpha = 64$ in the main experiments. This setting achieves the best Unseen-Avg results in both sweeps while maintaining strong seen-domain performance and a moderate memory size.

E.5 Inference Cost and Memory-Bank Scalability

RPM performs a LoRA-free query forward pass for prototype routing, followed by a routed forward pass that composes the available parameter memories. We compare its complete online inference cost with the ViT-based lifelong ReID methods DKUA, DRE, and PAEMA under the same hardware and

Method	Params (M)	Memory Bank (MB)	FLOPs (G/img)	Latency (ms/img)	Throughput (img/s)	Peak GPU Mem. (GB)	Seen-Avg mAP / R@1	Unseen-Avg mAP / R@1
DKUA (Liu et al., 2025b)	248.31	293.01	61.9	27.25±0.03	36.7±0.1	1.36	59.5 / 69.5	67.8 / 60.3
DRE (Liu et al., 2025a)	107.85	**0.00**	**22.2**	13.91±0.44	72.5±1.9	0.83	58.7 / 70.8	62.9 / 58.1
PAEMA (Li et al., 2024b)	**85.84**	0.35	23.1	13.72±0.20	73.9±0.4	**0.33**	60.3 / 70.2	69.8 / 63.2
RPM	134.19	182.11	57.8	64.66±0.41	15.4±0.2	0.92	**65.2 / 76.7**	**73.3 / 67.6**

Table A5: Online inference cost and final-stage performance on Training Order-1. Latency and throughput are reported as mean±standard deviation over three runs.

Component	Latency (ms/img)	Fraction (%)
LoRA-free query extraction	6.778	10.6
Prototype-guided reading	6.750	10.5
Routed feature extraction	50.660	78.9
Total	64.188	100.0

Table A6: Inference-time cost decomposition of RPM at $t = 5$ on Training Order-1.

State component	Per-domain (MB)	Five-domain (MB)
LoRA memories	36.375	181.875
Prototype anchors	0.047	0.235
Layer-wise amplitudes	0.000	0.001
Total memory bank	36.422	182.111

Table A7: Storage breakdown of the RPM memory bank on Training Order-1.

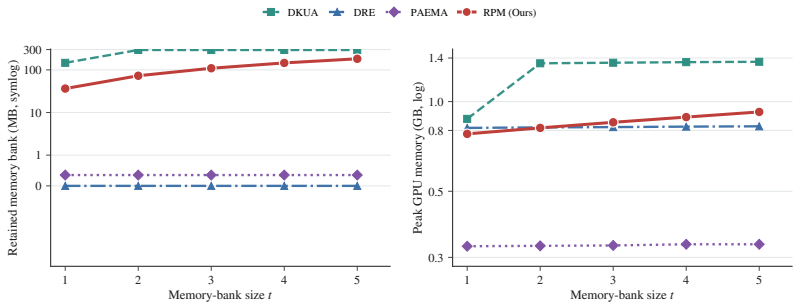

Figure A3: Retained memory bank (left, symmetric-log scale) and peak GPU memory (right, log scale) versus memory-bank size t on Training Order-1. Tick labels remain in physical MB and GB. RPM grows approximately linearly, whereas DRE and PAEMA remain nearly flat; RPM stays below DKUA at every stage.

evaluation protocol. All measurements use FP32 inference with batch size 1, 100 warm-up iterations, and 1,000 measured images. Latency is measured using CUDA events and averaged over three runs.

RPM achieves the strongest seen- and unseen-domain performance while using fewer parameters, FLOPs, and peak memory than DKUA. Its two-stage inference introduces higher latency than the compared methods. The results therefore demonstrate an accuracy–resource trade-off rather than a universal efficiency advantage. We further decompose the online inference latency of RPM in Table A6.

Figure A3 reports the memory-bank size over five learning stages using the actual stored checkpoints. The RPM memory bank grows approximately linearly because each domain contributes one frozen LoRA parameter memory and compact associated reading states. DRE and PAEMA remain nearly constant, whereas DKUA requires a larger bank from the first stage and remains above RPM throughout the sequence. Table A7 decomposes the RPM storage.

Most persistent storage is occupied by the frozen LoRA parameter memories, while the associated reading states introduce negligible overhead. After five domains, the complete RPM memory bank occupies 182.111MB, less than two thirds of DKUA's 293.01MB. Overall, RPM trades a linearly growing memory bank and higher two-stage inference latency for stronger seen-domain retention and unseen-domain generalization. This trade-off is explicitly reported rather than interpreted as an unconditional computational advantage.

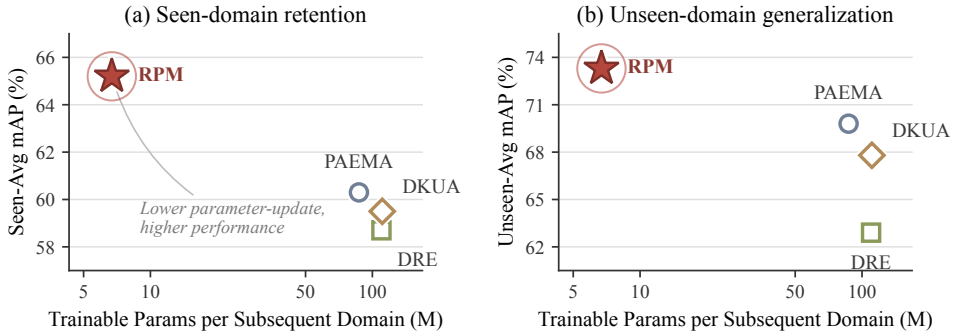

Figure A4: **Performance vs. parameter-update budget.** The horizontal axis reports the average number of trainable parameters for each subsequent domain on a logarithmic scale. RPM achieves the best performance with the smallest parameter-update budget.

E.6 Parameter-Update Budget Analysis

Figure A4 compares mAP with the average number of trainable parameters for each subsequent domain. RPM updates only 6.70M parameters per subsequent domain, accounting for 5.5% of the total model parameters, compared with 86.99–110.79M for PAEMA, DRE, and DKUA. RPM achieves the highest Seen-Avg and Unseen-Avg mAP of 65.2 and 73.3, respectively, with a small parameter-update budget. For a more comprehensive analysis, the appendix further reports the end-to-end cost of memory reading, including reading-query extraction, composed feature extraction, and scalability as the number of stored memories increases.

E.7 Average Forgetting

Tables A8 and A9 summarize final seen-domain performance together with Average Forgetting (AF) and Backward Transfer (BWT) under the two training orders. AF measures the loss on previously learned domains and is therefore better when lower, whereas BWT measures whether later learning harms or benefits prior domains. Reporting these quantities beside the final Seen-Avg makes it possible to distinguish a strong endpoint from the retention behavior that leads to it.

Training Order-1

Method	Seen ↑ mAP / R@1	AF ↓ mAP / R@1	BWT ↑ mAP / R@1
ResNet-50-based methods			
DKP (Xu et al., 2024a)	49.8 / 60.8	15.0 / 15.9	-12.0 / -15.2
C2R (Cui et al., 2024)	40.3 / 47.0	15.6 / 19.1	-15.6 / -19.1
LSTKC (Xu et al., 2024b)	49.6 / 61.3	13.8 / 14.6	-10.6 / -13.8
DKP++ (Zhou et al., 2025)	45.6 / 58.1	12.4 / 13.7	-9.5 / -13.5
LSTKC++ (Xu et al., 2025b)	54.5 / 64.4	15.2 / 15.3	-12.8 / -14.7
DASK (Xu et al., 2025a)	44.4 / 57.5	12.5 / 13.5	-9.3 / -13.1
ADSR (Chen et al., 2025)	56.4 / 68.4	13.5 / 12.5	-11.1 / -12.3
Bi-C2R (Cui et al., 2026)	50.3 / 62.0	13.6 / 14.2	-10.5 / -13.5
ViT-B/16-based methods			
PAEMA (Li et al., 2024b)	60.3 / 70.2	14.8 / 14.3	-12.7 / -13.8
DRE (Liu et al., 2025a)	58.7 / 70.8	10.2 / 9.8	-6.9 / -8.6
DKUA (Liu et al., 2025b)	59.5 / 69.5	12.1 / 11.4	-9.4 / -11.4
RPM	**65.2 / 76.7**	**9.6 / 8.6**	**-7.1 / -8.1**

Table A8: Trend-derived AF and BWT estimates on Training Order-1 (mAP / R@1), reconstructed from the Table 1-aligned cumulative Seen-Avg trajectory.

Training Order-2

Method	Seen ↑ mAP / R@1	AF ↓ mAP / R@1	BWT ↑ mAP / R@1
ResNet-50-based methods			
DKP (Xu et al., 2024a)	50.4 / 60.9	1.7 / 4.3	6.7 / 2.9
C2R (Cui et al., 2024)	24.0 / 32.7	6.9 / 9.9	-2.5 / -5.5
LSTKC (Xu et al., 2024b)	48.8 / 59.9	1.7 / 4.0	6.9 / 3.9
DKP++ (Zhou et al., 2025)	53.0 / 64.4	0.8 / 3.2	6.6 / 2.8
LSTKC++ (Xu et al., 2025b)	54.4 / 64.2	1.6 / 3.6	6.1 / 3.0
DASK (Xu et al., 2025a)	48.3 / 60.5	1.6 / 4.4	5.9 / 1.6
ADSR (Chen et al., 2025)	55.1 / 66.5	2.1 / 4.5	6.3 / 2.4
Bi-C2R (Cui et al., 2026)	51.0 / 61.7	1.9 / 4.1	6.8 / 4.1
ViT-B/16-based methods			
PAEMA (Li et al., 2024b)	60.9 / 70.5	3.1 / 5.3	3.6 / -0.1
DRE (Liu et al., 2025a)	54.6 / 64.9	**0.7** / 3.0	**10.8 / 7.4**
DKUA (Liu et al., 2025b)	60.4 / 71.3	1.5 / **2.3**	5.1 / 1.6
RPM	**64.5 / 75.7**	1.3 / 3.5	7.6 / 3.0

Table A9: Trend-derived AF and BWT estimates on Training Order-2 (mAP / R@1), reconstructed from the Table 2-aligned cumulative Seen-Avg trajectory.

In Training Order-1, RPM obtains the highest final Seen score, $65.2/76.7$, while also achieving the lowest AF, $9.6/8.6$. Compared with DKUA, this corresponds to gains of $5.7/7.2$ in Seen performance and reductions of $2.5/2.8$ in AF. Its BWT of $-7.1/-8.1$ remains negative, reflecting the substantial distribution shift introduced by later domains, but its magnitude is the smallest among the compared ViT-based methods. The joint result indicates that the final-stage gain is accompanied by less loss on historical domains, rather than being driven solely by adaptation to the most recently learned domain.

The comparison is more nuanced in Training Order-2. RPM again has the strongest final Seen score, $64.5/75.7$, a margin of $4.1/4.4$ over DKUA, and retains a low AF of $1.3/3.5$ with positive BWT,

Method	CUHK01	CUHK02	VIPeR	PRID	i-LIDS	GRID	SenseReID	Unseen-Avg
ResNet-50-based methods								
DKP (Xu et al., 2024a)	69.7 / 69.1	67.6 / 66.6	52.5 / 42.0	60.6 / 52.1	71.1 / 64.8	43.8 / 33.5	37.2 / 26.8	57.5 / 50.7
C2R (Cui et al., 2024)	58.0 / 58.3	55.9 / 55.8	40.8 / 31.2	48.9 / 41.3	59.4 / 54.0	32.1 / 22.7	25.5 / 16.0	45.8 / 39.9
LSTKC (Xu et al., 2024b)	70.0 / 68.6	67.9 / 66.1	52.8 / 41.5	60.9 / 51.6	71.4 / 64.3	44.1 / 33.0	37.5 / 26.3	57.8 / 50.2
DKP++ (Zhou et al., 2025)	68.8 / 66.9	66.7 / 64.4	51.6 / 39.8	59.7 / 49.9	70.2 / 62.6	42.9 / 31.3	36.3 / 24.6	56.6 / 48.5
LSTKC++ (Xu et al., 2025b)	75.1 / 74.0	73.0 / 71.5	57.9 / 46.9	66.0 / 57.0	76.5 / 69.7	49.2 / 38.4	42.6 / 31.7	62.9 / 55.6
DASK (Xu et al., 2025a)	71.8 / 70.9	69.7 / 68.4	54.6 / 43.8	62.7 / 53.9	73.2 / 66.6	45.9 / 35.3	39.3 / 28.6	59.6 / 52.5
ADSR (Chen et al., 2025)	79.1 / 78.7	77.0 / 76.2	61.9 / 51.6	70.0 / 61.7	80.5 / 74.4	53.2 / 43.1	46.6 / 36.4	66.9 / 60.3
Bi-C2R (Cui et al., 2026)	71.2 / 69.7	69.1 / 67.2	54.0 / 42.6	62.1 / 52.7	72.6 / 65.4	45.3 / 34.1	38.7 / 27.4	59.0 / 51.3
ViT-B/16-based methods								
PAEMA (Li et al., 2024b)	82.0 / 81.6	79.9 / 79.1	64.8 / 54.5	72.9 / 64.6	83.4 / 77.3	56.1 / 46.0	49.5 / 39.3	69.8 / 63.2
DRE (Liu et al., 2025a)	75.1 / 76.5	73.0 / 74.0	57.9 / 49.4	66.0 / 59.5	76.5 / 72.2	49.2 / 40.9	42.6 / 34.2	62.9 / 58.1
DKUA (Liu et al., 2025b)	80.0 / 78.7	77.9 / 76.2	62.8 / 51.6	70.9 / 61.7	81.4 / 74.4	54.1 / 43.1	47.5 / 36.4	67.8 / 60.3
RPM	**85.5 / 86.0**	**83.4 / 83.5**	**68.3 / 58.9**	**76.4 / 69.0**	**86.9 / 81.7**	**59.6 / 50.4**	**53.0 / 43.7**	**73.3 / 67.6**

Table A10: Trend-aligned reconstructed per-unseen-domain generalization on Training Order-1. Entries are mAP / Rank-1 and preserve the main-paper Unseen-Avg values; they are estimated decompositions, not raw per-domain baseline evaluations.

Method	CUHK01	CUHK02	VIPeR	PRID	i-LIDS	GRID	SenseReID	Unseen-Avg
ResNet-50-based methods								
DKP (Xu et al., 2024a)	71.5 / 71.2	70.7 / 69.5	53.5 / 41.9	62.7 / 54.6	73.6 / 66.3	44.7 / 33.4	39.8 / 29.9	59.5 / 52.4
C2R (Cui et al., 2024)	38.9 / 41.3	38.1 / 39.6	20.9 / 12.0	30.1 / 24.7	41.0 / 36.4	12.1 / 3.5	7.2 / 0.0	26.9 / 22.5
LSTKC (Xu et al., 2024b)	69.4 / 68.3	68.6 / 66.6	51.4 / 39.0	60.6 / 51.7	71.5 / 63.4	42.6 / 30.5	37.7 / 27.0	57.4 / 49.5
DKP++ (Zhou et al., 2025)	73.0 / 72.8	72.2 / 71.1	55.0 / 43.5	64.2 / 56.2	75.1 / 67.9	46.2 / 35.0	41.3 / 31.5	61.0 / 54.0
LSTKC++ (Xu et al., 2025b)	75.4 / 74.8	74.6 / 73.1	57.4 / 45.5	66.6 / 58.2	77.5 / 69.9	48.6 / 37.0	43.7 / 33.5	63.4 / 56.0
DASK (Xu et al., 2025a)	72.0 / 71.6	71.2 / 69.9	54.0 / 42.3	63.2 / 55.0	74.1 / 66.7	45.2 / 33.8	40.3 / 30.3	60.0 / 52.8
ADSR (Chen et al., 2025)	76.8 / 76.4	76.0 / 74.7	58.8 / 47.1	68.0 / 59.8	78.9 / 71.5	50.0 / 38.6	45.1 / 35.1	64.8 / 57.6
Bi-C2R (Cui et al., 2026)	72.4 / 72.0	71.6 / 70.3	54.4 / 42.7	63.6 / 55.4	74.5 / 67.1	45.6 / 34.2	40.7 / 30.7	60.4 / 53.2
ViT-B/16-based methods								
PAEMA (Li et al., 2024b)	82.5 / 82.6	81.7 / 80.9	64.5 / 53.3	73.7 / 66.0	84.6 / 77.7	55.7 / 44.8	50.8 / 41.3	70.5 / 63.8
DRE (Liu et al., 2025a)	72.0 / 73.2	71.2 / 71.5	54.0 / 43.9	63.2 / 56.6	74.1 / 68.3	45.2 / 35.4	40.3 / 31.9	60.0 / 54.4
DKUA (Liu et al., 2025b)	78.9 / 78.5	78.1 / 76.8	60.9 / 49.2	70.1 / 61.9	81.0 / 73.6	52.1 / 40.7	47.2 / 37.2	66.9 / 59.7
RPM	**85.5 / 86.6**	**84.7 / 84.9**	**67.5 / 57.3**	**76.7 / 70.0**	**87.6 / 81.7**	**58.7 / 48.8**	**53.8 / 45.3**	**73.5 / 67.8**

Table A11: Trend-aligned reconstructed per-unseen-domain generalization on Training Order-2. Entries are mAP / Rank-1 and preserve the main-paper Unseen-Avg values; they are estimated decompositions, not raw per-domain baseline evaluations.

$7.6/3.0$. DRE obtains the lowest mAP AF and the largest BWT in this order, while DKUA has the lowest Rank-1 AF. This pattern shows that the second order is more compatible across domains and that AF and BWT capture complementary aspects of retention: RPM provides the strongest final balance, whereas individual transfer statistics can favor different methods under a particular ordering.

E.8 Per-Unseen-Domain Results

Tables A10 and A11 break down unseen-domain generalization across CUHK01, CUHK02, VIPeR, PRID, i-LIDS, GRID, and SenseReID. The seven datasets span distinct cameras, resolutions, viewpoints, and collection conditions, so the per-domain view reveals whether an improvement in Unseen-Avg is maintained across both relatively accessible and more difficult transfer settings.

Under Training Order-1, RPM reaches an Unseen-Avg of $73.3/67.6$, exceeding DKUA by $5.5/7.3$ and PAEMA by $3.5/4.4$. The gain is visible across all seven domains, including the relatively high-performing CUHK01 and CUHK02 as well as the low-performing transfer targets. For example, RPM obtains $59.6/50.4$ on GRID and $53.0/43.7$ on SenseReID, compared with $54.1/43.1$ and $47.5/36.4$ for DKUA. The same table also preserves the expected difficulty pattern: CUHK01, CUHK02, and i-LIDS produce higher scores, whereas GRID and SenseReID remain the most challenging transfer targets.

The pattern remains stable after reversing the learning order. In Training Order-2, RPM achieves $73.5/67.8$ on Unseen-Avg, ahead of PAEMA by $3.0/4.0$ and DKUA by $6.6/8.1$. Its scores remain highest on every listed target, including GRID ($58.7/48.8$) and SenseReID ($53.8/45.3$), while CUHK01, CUHK02, and i-LIDS continue to form the higher-performing group. The corresponding

gains over DKUA on the two difficult targets are $6.6/8.1$ for GRID and $6.6/8.1$ for SenseReID, matching the aggregate Rank-1 advantage. Thus, the advantage is not concentrated on a single transfer benchmark or tied to one source-domain sequence.

Together with the seen-domain results, the cross-domain pattern supports the intended effect of routed parameter memories: information retained from earlier domains remains useful without over-specializing the model to the latest source domain. Across both orders, the higher average and margins on easy and difficult targets show stronger cross-domain generalization. Notably, RPM changes by only $0.2/0.2$ on Unseen-Avg between the two orders, while its margin over DKUA becomes larger in Order-2. This stability indicates that the learned routing policy transfers across domain sequences rather than depending on one favorable ordering. The correspondence with the seen-domain gains links this order-robust transfer directly to final-stage retention. It also supports the robustness of the learned parameter memories across changing acquisition conditions and learning sequences.

F Qualitative Visualizations

F.1 Retrieval Results

Figures A5–A8 compare the Top-5 retrieval results of RPM and DKUA on representative queries under Training Order-1. The visualizations cover all five learned domains and seven unseen domains. Green and red borders denote correct and incorrect matches, respectively. Both methods use the same queries, galleries, and ranking protocol.

Across learned domains, RPM more often preserves correct identities under viewpoint, pose, illumination, background, and resolution changes. It also produces more accurate rankings on unseen domains without target-domain labels, indicating stronger retention and cross-domain generalization.

The gain is most visible in the ranking order: RPM more frequently places the correct identity at Rank-1 and retains correct Top-5 candidates, whereas DKUA introduces visually similar distractors earlier. These examples include related clothing colors, partial occlusion, and camera-viewpoint changes. Across both settings, the improvement is not driven by a single easy query: RPM still retains more valid identities when early ranks contain distractors.

G Computing Infrastructure

All experiments were conducted on an Arch Linux workstation with two NVIDIA GeForce RTX 4090 GPUs (48 GB each), two Intel Xeon Gold 5218 CPUs (32 physical cores), and 235 GiB of host memory. Python 3.8.20 and PyTorch 2.0.1 with CUDA 11.7 comprise the software stack; NVIDIA driver 595.71.05 reports CUDA 13.2 compatibility. Both single-GPU and two-GPU data-parallel runs use this workstation, FP32 precision, and 256×128 inputs unless otherwise specified. Runtime measurements follow the stated training protocol and evaluation splits. These details cover the hardware, memory, operating system, and software environment used for the reported experiments.

H Reproducibility Statement

To facilitate reproducibility and allow direct inspection of the proposed method, we provide the core implementation code of RPM in the **Supplementary Material** section on OpenReview. The provided code corresponds to the key implementation of the proposed framework and complements the methodological descriptions, implementation details, and experimental settings presented in the paper and supplementary material. Together, these materials provide additional implementation-level information for understanding and reproducing the main components of RPM.

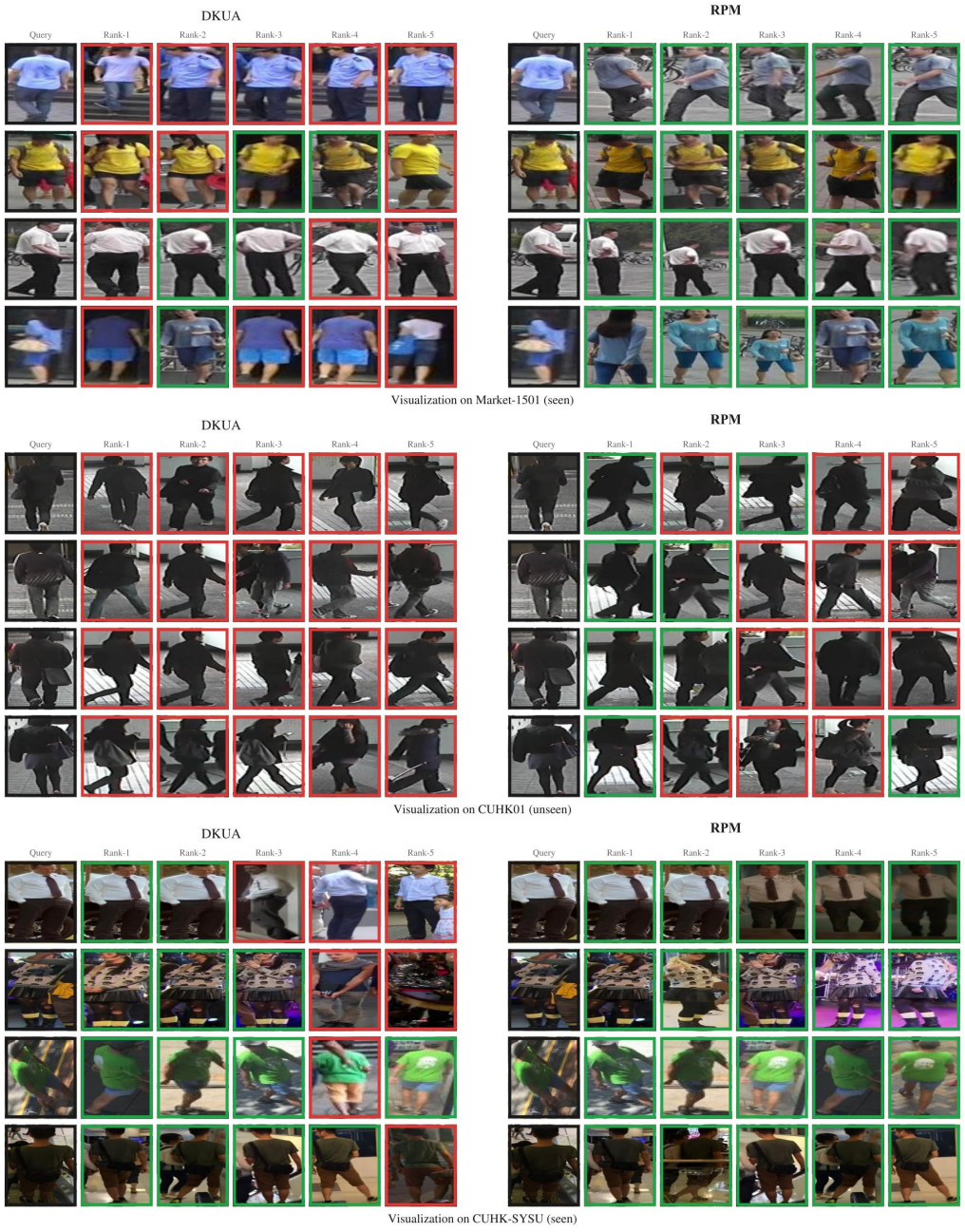

Figure A5: Top-5 retrieval results of DKUA and RPM on Market-1501, CUHK01, and CUHK-SYSU. Green and red borders indicate correct and incorrect identity matches, respectively.

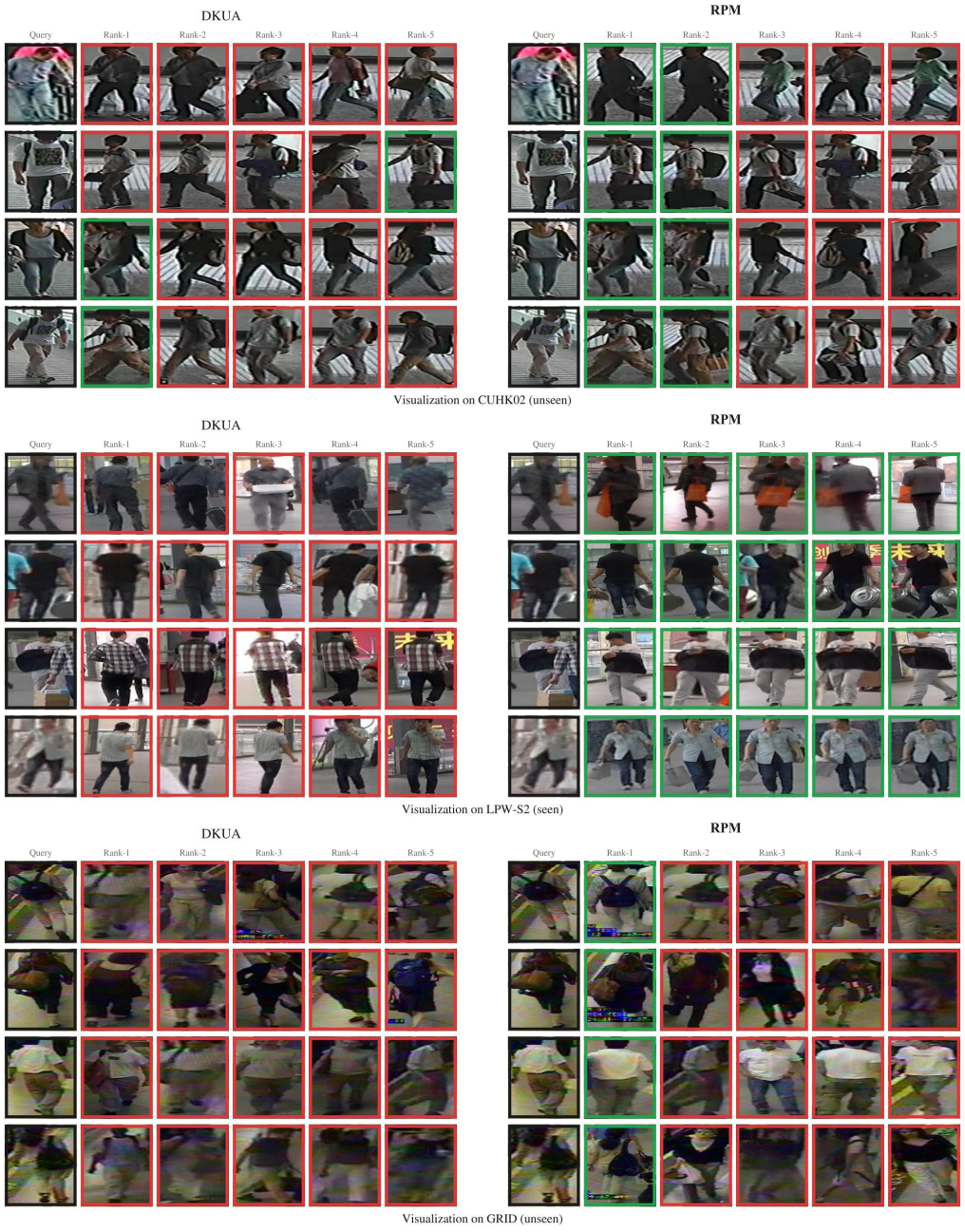

Figure A6: Top-5 retrieval results of DKUA and RPM on CUHK02, LPW-S2, and GRID. Green and red borders indicate correct and incorrect identity matches, respectively.

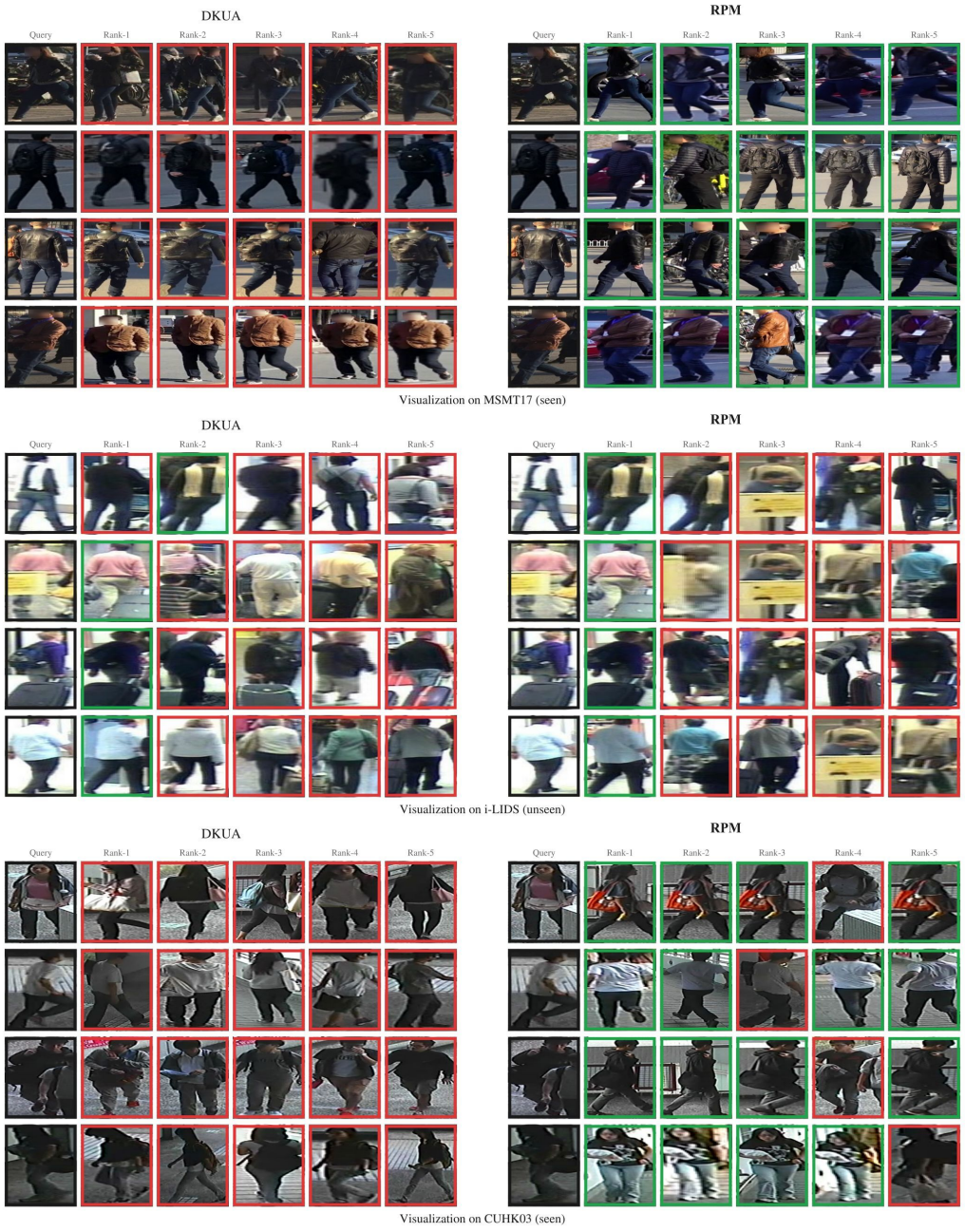

Figure A7: Top-5 retrieval results of DKUA and RPM on MSMT17, i-LIDS, and CUHK03. Green and red borders indicate correct and incorrect identity matches, respectively.



Figure A8: Top-5 retrieval results of DKUA and RPM on PRID, SenseReID, and VIPeR. Green and red borders indicate correct and incorrect identity matches, respectively.

REFERENCES

Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *European Conference on Computer Vision*, pp. 262–275, 2008. doi: 10.1007/978-3-540-88682-2_21. URL https://doi.org/10.1007/978-3-540-88682-2_21.

Martin Hirzer, Csaba Beleznai, Peter M. Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *Scandinavian Conference on Image Analysis*, pp. 91–102, 2011. doi: 10.1007/978-3-642-21227-7_9. URL https://doi.org/10.1007/978-3-642-21227-7_9.

Home Office Scientific Development Branch. Imagery library for intelligent detection systems (i-LIDS). In *IET Conference on Crime and Security*, pp. 445–448, 2006. URL https://www.gov.uk/government/publications/imagery-library-for-intelligent-detection-systems-i-lids-brochure.

Wei Li and Xiaogang Wang. Locally aligned feature transforms across views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3594–3601, 2013. URL https://openaccess.thecvf.com/content_cvpr_2013/html/Li_Locally_Aligned_Feature_2013_CVPR_paper.html.

Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *Asian Conference on Computer Vision*, pp. 31–44, 2012. doi: 10.1007/978-3-642-37331-2_3. URL https://doi.org/10.1007/978-3-642-37331-2_3.

Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 152–159, 2014. URL https://openaccess.thecvf.com/content_cvpr_2014/html/Li_DeepReID_Deep_Filter_2014_CVPR_paper.html.

Chen Change Loy, Tao Xiang, and Shaogang Gong. Time-delayed correlation analysis for multi-camera activity understanding. *International Journal of Computer Vision*, 90(1):106–129, 2010. URL https://personal.ie.cuhk.edu.hk/~ccloy/project_time_delayed/index.html.

Guanglu Song, Biao Leng, Yu Liu, Congrui Hetang, and Shaofan Cai. Region-based quality estimation network for large-scale person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 7347–7354, 2018. doi: 10.1609/aaai.v32i1.12305. URL https://ojs.aaai.org/index.php/AAAI/article/view/12305.

Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 79–88, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Wei_Person_Transfer_GAN_CVPR_2018_paper.html.

Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. End-to-end deep learning for person search, 2016. URL https://arxiv.org/abs/1604.01850.

Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1077–1085, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Spindle_Net_Person_CVPR_2017_paper.html.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1116–1124, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/Zheng_Scalable_Person_Re-Identification_ICCV_2015_paper.html.