Improving Time-to-Target Performance in Text-Based Person Search

Jiwoong Choi[1] and Hyunjoo Song[1*]

[1*]School of Computer Science and Engineering, Soongsil University, Seoul, South Korea .

*Corresponding author(s). E-mail(s): hsong@ssu.ac.kr;
Contributing authors: contact.jiwoong.choi@vis.ssu.ac.kr;

Abstract

This study investigates a cost-efficient training strategy for text-based person search under the memory and time constraints of single-GPU training. The strategy jointly uses automatic mixed precision (AMP), an exponential moving average (EMA) of model parameters, and gradient checkpointing (GC) to address complementary aspects of training cost: computation time, time to a target retrieval accuracy, and activation memory, respectively. The results show that the appropriate configuration depends on the resource constraint. In the CUHK-PEDES ablation, the complete AMP+EMA+GC configuration reduces process-level peak VRAM by 70.1% relative to the FP32 reference and reaches its best validation checkpoint, 71.06% t2i R@1, in 0.85 hours of training. Thus, although GC incurs recomputation overhead, the acceleration from AMP and the earlier model selection enabled by EMA more than offset this cost relative to the reference. These findings support AMP+EMA+GC as a practical training strategy for memory and time constrained single-GPU training rather than as a modification to the retrieval architecture.

1 Introduction

Training deep learning models imposes substantial computational and memory demands, and access to high-end multi-GPU infrastructure is not always available [4]. This constraint is characteristic of edge computers, where computation is kept close to the data source. In such settings, improving the computational, temporal, and memory efficiency of training is an important systems objective, because it governs whether

a model can be produced and updated at all when high-end computing resources are limited.

In this study, we address this problem in text-based person search under the joint compute-time and GPU-memory constraints of single-GPU training. Because a single commodity GPU provides substantially less VRAM than a multi-GPU data-center configuration, reducing training time alone is insufficient. Rather than modifying the retrieval architecture, we empirically analyze a unified training strategy that combines automatic mixed precision (AMP), an exponential moving average (EMA) of model parameters, and gradient checkpointing (GC). These techniques address complementary aspects of training efficiency: AMP improves computational throughput, EMA reduces the time required to reach a target retrieval accuracy, and GC reduces activation memory. We therefore treat their joint AMP+EMA+GC configuration as the proposed approach.

2 Related Work

2.1 Cost-Efficient and Fast-Converging Training

Training modern vision and language models consumes substantial computation and energy, and the cost of reaching a given accuracy has itself become a reporting concern [25, 32]. When high-end accelerators are not available, the efficiency of the training procedure, often determines whether a model can be produced at all. One line of work on resource-constrained training addresses this by changing *what* is optimized: TinyTrain and ElasticTrainer update only a dynamically selected subset of tensors, and POET combines rematerialization with paging to fit training into a fixed memory budget [12, 17, 26]. Our study instead keeps the model and the learning objective unchanged and adjusts only procedure-level settings, so that any efficiency gain transfers directly to the underlying retrieval model.

This evaluation perspective is established practice. DAWNBench introduced time-to-accuracy, the wall-clock time to reach a fixed validation target, as a primary metric, and reported that it is stable across runs and that models tuned for it generalize normally [7]; MLPerf adopted the same metric for standardized training comparison [21]. Reporting a single efficiency number can be misleading when compute, memory, and time do not move together, which argues for reporting them jointly [8]. We follow this reasoning in Section 3.1 by defining a time-to-target and reporting time and memory costs separately rather than as a single score.

2.2 Text-Based Person Search

Text-based person search (TBPS), also referred to as text-to-image person re-identification, retrieves pedestrian images from a gallery using free-form descriptions. Research in this area has primarily pursued higher retrieval accuracy, particularly Rank-1 accuracy, by progressively refining cross-modal alignment. Representative approaches have moved from global embedding matching to part-level or fine-grained correspondence modeling [6, 19, 31], and more recent methods exploit CLIP-based

representations, implicit relation reasoning, noisy-correspondence learning, and hierarchical concept decomposition [1, 14, 16, 27, 34].

This limitation has motivated cost-efficient alternatives. For example Min et al. freeze the CLIP backbone and train a lightweight cross-modal adapter with only 2 million trainable parameters, corresponding to 1.3% of the model, for text-to-person retrieval [23]. Their results show that reducing the number of updated parameters can preserve competitive retrieval accuracy while substantially lowering adaptation overhead. Such work suggests that practical TBPS systems should not be assessed solely by their final Rank-1 score. Training time, peak GPU memory, and the amount of trainable state are also deployment-relevant criteria. Accordingly, our study shifts the emphasis from adding retrieval-specific architectural complexity to evaluating whether a general training strategy can reach a predefined Rank-1 target under explicit single-GPU time and memory constraints.

3 Proposed Approach

Let $\mathcal{D} = \{(I_i, T_i, y_i)\}_{i=1}^{N}$ denote a person retrieval dataset, where I_i is a pedestrian image, T_i is its textual description, and y_i is the corresponding person identity. Given a text query, the objective is to rank gallery images such that images sharing the query identity are retrieved before non-matching images. We therefore use text-to-image Rank-1 accuracy (t2i R@1) as the primary retrieval metric.

3.1 Problem Setup

Our objective is not limited to maximizing the final retrieval accuracy. Under a single-GPU constraint, we instead seek to minimize the resources required to reach a predefined performance target. For a target accuracy ρ, the time-to-target is defined as

$$T_\rho = \min\{t \mid R_{\text{t2i@1}}(t) \geq \rho\}, \tag{1}$$

where $R_{\text{t2i@1}}(t)$ is the checkpoint-selection t2i R@1 observed at time t.

We fix ρ to a dataset-specific reference rather than an arbitrary round number. The reference is the t2i R@1 attained by fully fine-tuning the CLIP dual encoder on that dataset without any retrieval-specific module; this is precisely the operating point our training strategy is designed to reach at lower cost. We take the reference values from Chen et al., who report full CLIP fine-tuning accuracies of 68.17% on CUHK-PEDES, 61.88% on ICFG-PEDES, and 59.60% on RSTPReid [3]. Accordingly, the initial targets are $\rho = 0.6817$ for CUHK-PEDES, $\rho = 0.6188$ for ICFG-PEDES, and $\rho = 0.5960$ for RSTPReid; Section 4.2 states how ρ is reset once a configuration exceeds it.

All configurations share the same learning framework. We fine-tune the CLIP dual encoder [28] with a standard bidirectional image–text contrastive objective $\mathcal{L}_{\text{global}}$, computed over in-batch similarities in a shared embedding space as the average of the image-to-text and text-to-image InfoNCE terms; following supervised contrastive

learning [15], all in-batch samples sharing a person identity are treated as positives. This formulation is standard in text-based person retrieval, so we keep it fixed throughout and do not treat it as a contribution.

We further adopt two low-overhead accuracy enhancements that are widely used in prior studies: ReID-specific image augmentation (random horizontal flipping, padded random cropping, and random erasing) and identity-level multiple-positive supervision, as in IRRA and RDE [14, 27]. Because they add no network modules, learnable parameters, or inference-time computation, we regard them as inexpensive training practices rather than methodological contributions. Together they raise validation accuracy to approximately 71% at little additional cost and without substantial changes to the model architecture.

Rather than collapsing heterogeneous costs into a single efficiency score, we compare configurations at an equal retrieval target ρ and report training time and peak GPU memory separately, alongside accuracy. The primary time measure is the cumulative training time to reach ρ; we additionally report end-to-end time, including training and validation, as a wall-clock reference, and peak memory as the maximum GPU memory observed during training. This keeps the accuracy–time–memory trade-off explicit. A configuration is accordingly considered more cost-efficient only when it reaches the same target t2i R@1 at both lower training time and lower peak memory; a memory reduction alone, or an accuracy gain at unreported computational cost, is insufficient.

To reduce these costs jointly, we combine three complementary techniques, none of which changes the global image–text contrastive objective $\mathcal{L}_{\text{global}}$. Automatic mixed precision (AMP) reduces the computation time of an update; an exponential moving average (EMA) $\bar{\theta}$ of the trainable parameters θ, maintained with decay β, is intended to reduce the number of updates needed to attain the target accuracy; and gradient checkpointing (GC) lowers the peak activation memory required for an update. Their joint use is the central training strategy investigated in this study.

Algorithm 1 summarizes the resulting per-mini-batch procedure: the image and text encoders perform a checkpointed forward pass under AMP, the scaled contrastive loss is backpropagated with activation recomputation, the optimizer updates θ, and the EMA state $\bar{\theta}$ is then updated; validation is performed with $\bar{\theta}$, while subsequent training continues from θ. Each component is detailed in Sections 3.2–3.4, with the EMA update given by Eq. (2). AMP introduces precision casting and gradient scaling, EMA introduces a shadow copy of the trainable parameters and a small update cost, and GC introduces additional forward computation; these overheads are included in all reported measurements.

3.2 Automatic Mixed Precision

Automatic mixed precision (AMP) executes eligible operations in lower precision while retaining a full-precision FP32 copy of the master weights and computing numerically sensitive operations in FP32. Dynamic loss scaling is used to reduce the risk of gradient underflow during backpropagation [22]. In this study, AMP primarily targets the computational component of training cost: it reduces the duration of each update and

Algorithm 1 Integrated AMP, EMA, and gradient-checkpointed training

Require: Training data $\mathcal{D}$, trainable parameters θ, EMA decay β

Initialize $\bar{\theta} \leftarrow \theta$
for each training mini-batch $\mathcal{B}$ **do**
    Compute global embeddings using checkpointed encoders under AMP
    Compute and scale the global contrastive loss $\mathcal{L}_{\text{global}}$
    Recompute checkpointed activations and backpropagate the scaled loss
    Attempt to update θ with the optimizer
    if the optimizer update succeeds **then**
        $\bar{\theta} \leftarrow \beta\bar{\theta} + (1-\beta)\theta$
    end if
    if validation is scheduled **then**
        Evaluate t2i R@1 using $\bar{\theta}$
        Record training time, end-to-end training time, and peak memory
    end if
end for

can also reduce tensor memory, without changing the model architecture or training objective. Its effect is evaluated both independently and jointly with EMA and GC.

3.3 EMA-Based Reduction of Time-to-Target Performance

The parameter values obtained at individual optimization steps may fluctuate because of stochastic mini-batch gradients. Such fluctuations can delay the first validation checkpoint that reaches a target retrieval accuracy. We use an exponential moving average (EMA) of the trainable parameters to obtain a temporally smoothed evaluation model. EMA has been shown to provide strong early performance and to reduce optimization noise through weight averaging [24]. The analysis of Morales-Brotons et al. focuses on EMA under SGD-based optimization, whereas our Transformer model is trained with AdamW. This distinction is important because Transformers can exhibit substantial variation in Hessian spectra across parameter blocks, for which a single learning rate, as used by SGD, is less effective than the adaptive coordinate-wise scaling provided by Adam-type optimizers [35]. Thus, we adopt the EMA mechanism from the prior work while evaluating it in the AdamW setting used throughout this study.

Let θ_k be the trainable parameters after the kth successful optimizer update. The EMA parameters $\bar{\theta}_k$ are updated by

$$\bar{\theta}_k = \beta\bar{\theta}_{k-1} + (1-\beta)\theta_k, \qquad 0 \le \beta < 1, \tag{2}$$

where β controls the averaging horizon. The EMA state is initialized from the trainable parameters before optimization and is updated only after a successful optimizer step. This ordering prevents skipped mixed-precision updates from contaminating the averaged model.

Training gradients and optimizer states remain associated with θ; $\bar{\theta}$ is not differentiated. During validation, the trainable parameters are temporarily replaced by $\bar{\theta}$, after which the original parameters are restored. Checkpoint selection is consequently

based on the EMA validation model. EMA does not reduce the duration of an individual epoch. Its intended efficiency gain is instead a reduction in the number of updates and elapsed time required to first reach the target in Eq. (1). We therefore evaluate EMA using time-to-target rather than attributing a per-step acceleration to parameter averaging.

3.4 Memory-Efficient Training via Gradient Checkpointing

Standard backpropagation stores every intermediate activation produced during the forward pass so that it is available when the corresponding gradient is computed. For transformer encoders, this activation storage can dominate GPU memory and restrict the batch size or input resolution. Gradient checkpointing addresses this by storing activations only at a sparse set of checkpoints along the depth of the encoder and discarding the rest; when the backward pass needs an activation that was not stored, the forward computation between the nearest checkpoints is simply repeated to reconstruct it. This is the computation–memory trade-off introduced by Chen et al. [5]: keeping fewer activations lowers memory usage, but requires more forward computation to be redone during the backward pass. We adopt their checkpoint selection and recomputation strategy without modification and apply it to both the visual and textual transformer blocks; doing so preserves the original model and learning objective while reducing activation memory, but increases training time per update because of the added recomputation.

4 Experimental Results and Analysis

4.1 Datasets and Experimental Settings

4.1.1 Datasets

We evaluate our method on CUHK-PEDES [18], ICFG-PEDES [9], and RSTPReid [36]. These datasets associate pedestrian photographs and natural-language captions with person identities, so an identity is generally represented by multiple photographs and, collectively, multiple captions. The following counts were obtained directly from the annotation files used in our experiments. CUHK-PEDES contains 40,206 annotated images of 13,003 identities: 34,054 images of 11,003 identities for training, 3,078 images of 1,000 identities for validation, and 3,074 images of 1,000 identities for testing. All but four identities have between 2 and 15 images. Although the accompanying dataset documentation reports two captions per image and 80,412 captions in total, the released annotation file used here contains 80,440 caption strings because nine image records contain more than two captions. ICFG-PEDES contains 54,522 single-caption annotation records for 4,102 identities, divided into 34,674 training records for 3,102 identities and 19,848 test records for 1,000 identities. Each identity has between 3 and 22 records and therefore multiple photographs and captions; one image path is repeated in the annotation file, giving 54,521 unique image paths. ICFG-PEDES provides only training and test splits and no official validation split, so we adopt a five-fold cross-validation protocol on the training set for model selection on this dataset. RSTPReid contains 20,505 images of 4,101 identities, with exactly five

images per identity and two captions per image (41,010 captions in total). Its training, validation, and test splits contain 3,701, 200, and 200 identities, respectively. RSTPReid does define an official validation split, but that split is anomalous: a detailed audit of the official annotation file reveals severe cross-identity caption duplication within it, with 446 out of the 2,000 validation queries (22.30%) sharing completely identical text descriptions across multiple distinct person identities, arising from 41 shared caption templates spanning up to 15 different identities each. This widespread textual ambiguity undermines the validity of standard rank-based evaluation on the official validation set, as retrieving ground-truth matching visual appearances of other identities sharing the identical text query is penalized as false positives. We provide a comprehensive breakdown and concrete sample records in Appendix A, and, on the basis of that analysis, likewise adopt a five-fold cross-validation protocol on the training set for RSTPReid rather than relying on the contaminated official validation split.

4.1.2 Implementation Details

All experiments were implemented in PyTorch using OpenCLIP and executed on a single RTX 5070 Ti. The image encoder was a 12-layer ViT-B/16 with a width of 768 and a patch size of 16×16, while the text encoder was CLIP's 12-layer Transformer with a width of 512, 8 attention heads, and a context length of 77 tokens. Both encoders were initialized from OpenAI's pretrained CLIP weights and fine-tuned end to end. Input images were resized to 384×128 pixels and normalized using the CLIP channel statistics. Training augmentation comprised random horizontal flipping with probability 0.5, 10-pixel padding followed by random cropping, and random erasing with an area ratio sampled from $[0.02, 0.4]$. Validation used only resizing and normalization. All available captions were used, and samples sharing the same person identity within a mini-batch were treated as positives. We used shuffled mini-batches rather than identity-balanced sampling. The model was trained for at most 60 epochs with a batch size of 64, no gradient accumulation and four data-loading workers.

As discussed in Section 3.3, the prior study that motivated our use of EMA investigated deep learning models optimized with SGD [24]. However, because our study uses the Transformer-based CLIP model, we selected AdamW in accordance with evidence that SGD can underperform AdamW on Transformer architectures, including ViT-base, even after careful learning-rate tuning [35].

Unless AMP was enabled, training and validation used FP32. The AMP variants kept model parameters in FP32, performed eligible operations under FP16 autocasting, and used dynamic gradient scaling [22]. For EMA variants, the moving average of all trainable parameters was updated after each successful optimizer step with decay $\beta = 0.999$ and was used for validation and checkpoint selection. Gradient-checkpointed variants enabled activation checkpointing in both OpenCLIP encoders. These options did not otherwise alter the data, objective, optimizer, or learning-rate schedule.

4.2 Results

We evaluate the individual and combined effects of automatic mixed precision (AMP), exponential moving average (EMA), and gradient checkpointing (GC) on CUHK-PEDES. All configurations use the common training protocol described in Section 4.1.2; only the three efficiency options are varied. We separately analyze their effects on validation convergence and time to the selected validation checkpoint.

All analyses use the time-to-target metric of Eq. (1). When a configuration attains t2i R@1 above the initial target ρ of Section 3.1, we reset ρ to that attained best-checkpoint accuracy and report time-to-target against the higher level; on CUHK-PEDES this places the operating target near 71%.

4.2.1 Accuracy and Convergence

Table 1 separates checkpoint quality from its resource cost. EMA is the main contributor to early checkpoint quality: every EMA-enabled run reaches its best result by epoch 15, whereas all non-EMA runs require more than 50 epochs. The FP32 EMA+GC configuration achieves the highest validation t2i R@1 of 71.31% at epoch 15. AMP+EMA and AMP+EMA+GC both achieve 71.06% at epoch 10. AMP alone improves the best validation t2i R@1 from 70.01% to 70.46%, but does not produce the early convergence observed with EMA.

Table 1 Validation accuracy, convergence, and peak VRAM of the training configurations. VRAM is the maximum process-level GPU memory observed in W&B system telemetry during each run.

AMP	EMA	GC	Best R@1 (%)	Best Epoch	VRAM (GiB)
			70.01	57	14.15
		✓	70.44	53	4.41
	✓		71.05	9	14.73
	✓	✓	71.31	15	5.00
✓			70.46	59	8.91
✓		✓	70.46	59	3.63
✓	✓		71.06	10	9.51
✓	✓	✓	71.06	10	4.23

4.2.2 Time Cost to the Best Checkpoint

Table 2 reports end-to-end training time and training time to the best validation checkpoint. AMP+EMA is the fastest configuration, reaching its best checkpoint in 46 minutes end to end and 0.63 hours of training. Relative to the reference configuration, it reduces training time by 92.6%. AMP alone reduces end-to-end time from 9 hours to 3 hours 55 minutes, demonstrating substantial per-update acceleration even though its best checkpoint remains late in training. GC introduces the expected recomputation overhead. For the proposed AMP+EMA+GC configuration, this overhead increases training time from 0.63 to 0.85 hours relative to AMP+EMA, but the

complete configuration still reduces training time by 90.0% relative to the 8.54-hour FP32 reference. The AMP and EMA gains therefore more than offset the time lost to GC while retaining GC's substantial activation-memory reduction.

Table 2 Time cost to the best validation checkpoint. End-to-end time includes training and validation, whereas training time excludes validation.

AMP	EMA	GC	End-to-end training time	Training time (h)
			9 h 00 min	8.54
		✓	10 h 24 min	9.91
	✓		1 h 36 min	1.36
	✓	✓	3 h 07 min	2.81
✓			3 h 55 min	3.62
✓		✓	4 h 47 min	4.47
✓	✓		46 min	0.63
✓	✓	✓	1 h 01 min	0.85

4.2.3 Practical Configuration Selection

These findings indicate that no configuration dominates all resource dimensions. AMP+EMA is preferable when training time is the only constraint, FP32+EMA+GC provides the highest validation accuracy, and AMP+GC minimizes peak memory. Rajagopal and Bouganis describe edge devices as having "limited compute resources and memory budget" during on-device training [29]. We therefore select the full AMP+EMA+GC configuration as the proposed approach. It reaches the same best R@1 and epoch as AMP+EMA while reducing process-level peak VRAM from 9.51 to 4.23 GiB. Its 0.22-hour training-time overhead relative to AMP+EMA is outweighed by a 7.69-hour reduction relative to FP32 training, showing that AMP and EMA can absorb the recomputation cost of GC under a constrained memory budget.

Furthermore, Table 3 reproduces the CUHK-PEDES results compiled by Min et al. [23], including their CLIP full fine-tuning baseline, and adds the complete AMP+EMA+GC configuration from this study.

4.2.4 Comparison with Reproduced Reference

We additionally reproduced IRRA [14] and RDE [27], the two comparison methods that fit on the same GPU under the available implementations. Table 4 reports measurements from these reproductions. All runs used a batch size of 64. Training time is the cumulative wall-clock time spent on training, excluding validation, and peak memory is the maximum value recorded during training.

Table 3 Comparison with published methods on CUHK-PEDES. The result for this study is obtained by selecting the complete AMP+EMA+GC checkpoint on the official validation split and evaluating it once on the official test split. Published test-set results, including the CLIP full fine-tuning baseline, are reproduced from Table 1 of Min et al. [23]. A dash denotes an unreported metric or year.

Method	Year	R@1 (%)	R@5 (%)	R@10 (%)	mAP (%)
DSSL [36]	2021	59.98	80.41	87.56	–
SAF [19]	2022	64.13	82.62	88.40	58.61
TIPCB [6]	2022	64.26	83.19	89.10	–
AXM-Net [10]	2022	64.44	80.52	86.77	58.73
LGUR [30]	2022	65.25	83.12	89.00	–
IVT [31]	2022	65.59	83.11	89.21	–
Han et al. [11]	2021	64.08	81.73	88.19	60.08
CFine [34]	2022	69.57	85.93	91.15	–
CSKT [20]	2024	69.70	86.92	91.80	62.74
PGA [13]	2025	69.44	87.03	92.88	62.83
Min et al. [23]	2026	70.93	87.81	92.91	64.02
CLIP (full fine-tuning) [23]	–	68.17	86.47	91.47	61.12
Ours (AMP+EMA+GC)	2026	70.11	87.90	92.72	63.93

Table 4 Reproduction results on CUHK-PEDES. The proposed approach uses the validation split for checkpoint selection, whereas the reproduced IRRA and RDE runs evaluate the test split at each epoch. R@1 is the official test-split t2i R@1 for every method; the reported training-time values are nonetheless indicative rather than strictly split-matched.

Method	R@1(%)	Training time (h)	VRAM(GiB)
IRRA [14]	73.41	4.83	12.11
IRRA (AMP+EMA+GC) [14]	73.77	3.32	7.10
RDE [27]	75.05	4.79	8.51
RDE (AMP+EMA+GC) [27]	75.81	6.41	3.83
Ours (AMP+EMA+GC)	70.11	0.85	3.47

Figure 1 compares the proposed AMP+EMA+GC approach with the original IRRA and RDE reproductions over the first two cumulative training hours. At the 71% target, the proposed approach is 3.07× faster than IRRA and 2.19× faster than RDE. These results support the time-to-target contribution of the complete approach, but Table 4 also exposes the accuracy–efficiency trade-off: IRRA and RDE ultimately exceed our official test-split R@1 by 3.30 and 4.94 percentage points, respectively.

The reproduced IRRA and RDE implementations were also trained with AMP, EMA, and GC enabled jointly; the solid IRRA and RDE curves in Figure 1 show these enhanced reproductions against the complete proposed approach. The proposed AMP+EMA+GC configuration reaches 71% t2i R@1 in 0.85 hours while reducing process-level peak VRAM to 4.23 GiB; it remains 1.47× and 1.22× faster than the enhanced IRRA and RDE runs, respectively. For IRRA, the combined configuration reduces T_{71} from 2.61 to 1.25 hours and peak allocated memory from 12.11 to 7.10 GiB, while increasing the best t2i R@1 from 73.41% to 73.77%. For RDE, it reduces T_{71} from

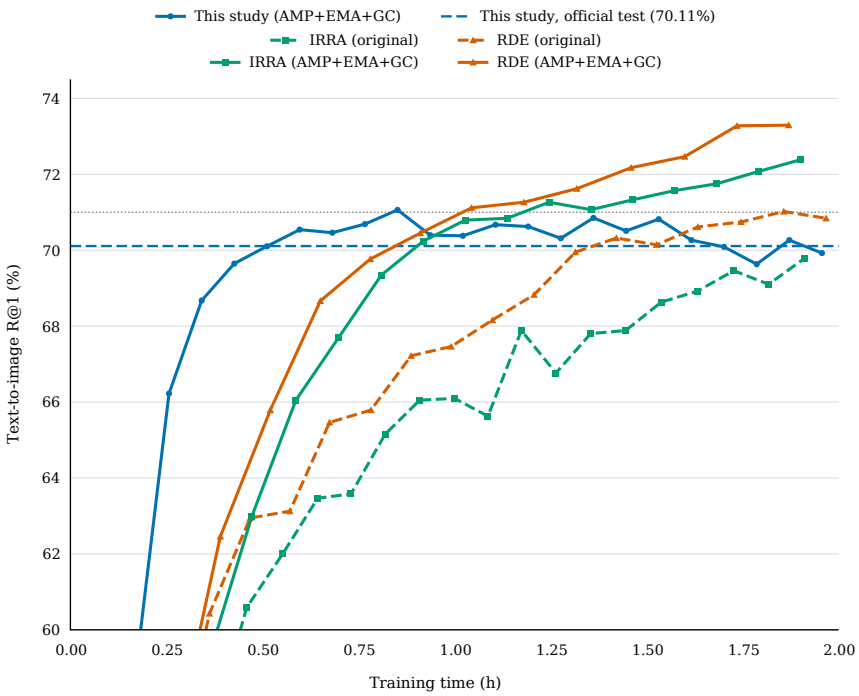

Fig. 1 Text-to-image R@1 on CUHK-PEDES over the first two cumulative training hours. The proposed AMP+EMA+GC approach is compared with the IRRA and RDE reproductions in their original form (dashed) and retrained with AMP, EMA, and GC enabled jointly (solid). All runs use a batch size of 64. The proposed approach uses the validation split, whereas IRRA and RDE use the test split. The dotted horizontal line marks 71% t2i R@1. The dashed blue horizontal line marks the proposed approach's 70.11% t2i R@1 on the official test split (Table 3).

1.86 to 1.04 hours and peak allocated memory from 8.51 to 3.83 GiB, while increasing the best t2i R@1 from 75.05% to 75.81%. These are combined AMP+EMA+GC comparisons; separate single-factor ablations were not run for the reproduced IRRA and RDE implementations.

The memory results clarify why GC is integral to the proposed memory-constrained approach. AMP+EMA alone uses 9.51 GiB of process-level peak VRAM. Adding GC reduces this value to 4.23 GiB, a 55.5% reduction. The resulting AMP+EMA+GC training time of 0.85 hours is slightly longer than the 0.63 hours required by AMP+EMA, but remains substantially shorter than either reproduced reference method. Thus, AMP and EMA offset the recomputation overhead needed to obtain GC's memory reduction, which is the relevant trade-off when VRAM is constrained.

4.3 Results on ICFG-PEDES and RSTPReid

The CUHK-PEDES experiments above provide the primary ablation because that dataset supplies distinct training, validation, and test splits. For ICFG-PEDES and RSTPReid, we instead use a fixed five-fold protocol to assess whether the complete AMP+EMA+GC method transfers beyond the primary dataset. Neither RSTPReid

offers a usable official validation split for model selection, for the reasons detailed in Section 4.1.1: ICFG-PEDES provides no official validation split, and the official RSTPReid validation split is contaminated by cross-identity caption duplication.

For each secondary dataset, we collect the unique identities from the official training split, sort them, and shuffle them once. The shuffled identity list is assigned to five folds by strided indexing. Each run trains on four folds and validates on the remaining fold, so no identity or image path is shared between its training and validation subsets. The official test split is excluded from training, validation, early stopping, and checkpoint selection. Validation uses every image and caption in the held-out fold and ranks each text query against the complete held-out image gallery.

Each fold run selects the checkpoint with the highest validation text-to-image R@1, using mAP only to break an R@1. We report the mean and sample standard deviation across the five selected checkpoints. To examine subsequent generalization, we take the fold model with the highest validation R@1 and evaluate it once on the full official test split. Because that model has seen only four of the five training folds, its official-test result characterizes a selected fold model trained on 80% of the official training identities.

Table 5 Five-fold validation of the complete AMP+EMA+GC method on the secondary datasets. Val R@1, VRAM, and training time are the mean ± sample standard deviation across five identity-disjoint folds. VRAM is the maximum process-level GPU memory observed in W&B system telemetry during each fold run, and training time is the cumulative training GPU-hours to the best validation checkpoint, excluding validation. The Test R@1 column takes the fold model with the highest validation R@1 and evaluates it once on the official test split.

Dataset	Val R@1 (%)	Test R@1 (%)	VRAM (GiB)	Training time (h)
ICFG-PEDES	71.70±0.77	59.55	4.94±0.01	0.57±0.19
RSTPReid	53.97±0.53	53.35	4.95±0.01	0.95±0.27

We reproduced IRRA [14] and RDE [27] on ICFG-PEDES and RSTPReid under the same single-GPU budget and a batch size of 64, both in their original form and with AMP, EMA, and GC enabled jointly. Table 6 reports the resulting training time and peak memory. As on CUHK-PEDES, the combined configuration lowers peak allocated memory substantially, from 12.09 to about 7.0 GiB for IRRA and from about 8.5 to 3.82 GiB for RDE on both datasets, while the recomputation overhead of GC lengthens training time by roughly 0.7 to 1.0 hours. Relative to these references, the complete AMP+EMA+GC method of this study trains in 0.57 hours on ICFG-PEDES and 0.95 hours on RSTPReid at a process-level peak VRAM near 4.9 GiB (Table 5); its identity-disjoint fold checkpoints reach lower R@1, consistent with the accuracy–efficiency trade-off already observed on the primary dataset.

Table 6 Reproduced IRRA and RDE on the secondary datasets (batch size 64), original and with AMP+EMA+GC. Reference R@1 is the best per-epoch test-split t2i R@1; for Ours it is the best fold model evaluated once on the official test split, with training time and VRAM as the five-fold mean ± standard deviation (Table 5).

Method	R@1 (%)	Training time (h)	VRAM (GiB)
ICFG-PEDES			
IRRA [14]	63.70	2.72	12.09
IRRA (AMP+EMA+GC) [14]	63.66	3.42	7.02
RDE [27]	67.70	3.62	8.51
RDE (AMP+EMA+GC) [27]	67.54	4.30	3.82
Ours (AMP+EMA+GC)	59.55	0.57±0.19	4.94±0.01
RSTPReid			
IRRA [14]	60.05	2.79	12.09
IRRA (AMP+EMA+GC) [14]	64.35	3.72	7.01
RDE [27]	65.05	3.60	8.53
RDE (AMP+EMA+GC) [27]	64.00	4.59	3.82
Ours (AMP+EMA+GC)	53.35	0.95±0.27	4.95±0.01

For broader context, Tables 7 and 8 reproduce the ICFG-PEDES and RSTPReid results compiled by Min et al. [23], including their CLIP full fine-tuning baseline, and add the official-test R@1 of the complete AMP+EMA+GC fold model from this study. As on CUHK-PEDES, the CLIP full fine-tuning baseline trails the parameter-efficient method of Min et al. [23] on both secondary datasets.

Table 7 Comparison with published methods on ICFG-PEDES. Published test-set results, including the CLIP full fine-tuning baseline, are reproduced from Table 2 of Min et al. [23]. The result for this study is the best-validation AMP+EMA+GC fold model evaluated once on the official test split (Table 5); it selects checkpoints on identity-disjoint folds, and all reported metrics for this study are text-to-image. A dash denotes an unreported metric or year.

Method	Year	R@1 (%)	R@5 (%)	R@10 (%)	mAP (%)
SAF [19]	2022	54.86	72.13	79.13	32.76
TIPCB [6]	2022	54.96	74.72	81.89	–
LGUR [30]	2022	59.02	75.32	81.56	–
IVT [31]	2022	56.04	73.60	80.22	–
CFine [34]	2022	60.83	76.55	86.42	–
CSKT [20]	2024	58.90	77.31	83.56	33.87
PGA [13]	2025	58.10	76.95	84.06	32.58
Min et al. [23]	2026	61.88	79.39	85.15	36.67
CLIP (full fine-tuning) [23]	–	56.74	75.72	82.26	31.84
Ours (AMP+EMA+GC)	2026	59.55	77.28	83.20	36.75

Table 8 Comparison with published methods on RSTPReid. Published test-set results, including the CLIP full fine-tuning baseline, are reproduced from Table 3 of Min et al. [23]. The result for this study is the best-validation AMP+EMA+GC fold model evaluated once on the official test split (Table 5); it selects checkpoints on identity-disjoint folds, and all reported metrics for this study are text-to-image. A dash denotes an unreported metric or year.

Method	Year	R@1 (%)	R@5 (%)	R@10 (%)	mAP (%)
DSSL [36]	2021	32.43	55.08	63.19	–
SAF [19]	2022	44.05	67.30	76.25	36.81
IVT [31]	2022	46.70	70.00	78.80	–
CFine [34]	2022	50.55	72.50	81.60	–
CSKT [20]	2024	57.75	81.30	88.35	46.43
PGA [13]	2025	55.37	81.42	88.60	45.93
Min et al. [23]	2026	59.60	81.60	87.50	47.93
CLIP (full fine-tuning) [23]	–	54.05	80.70	88.00	43.41
Ours (AMP+EMA+GC)	2026	53.35	77.45	84.75	44.66

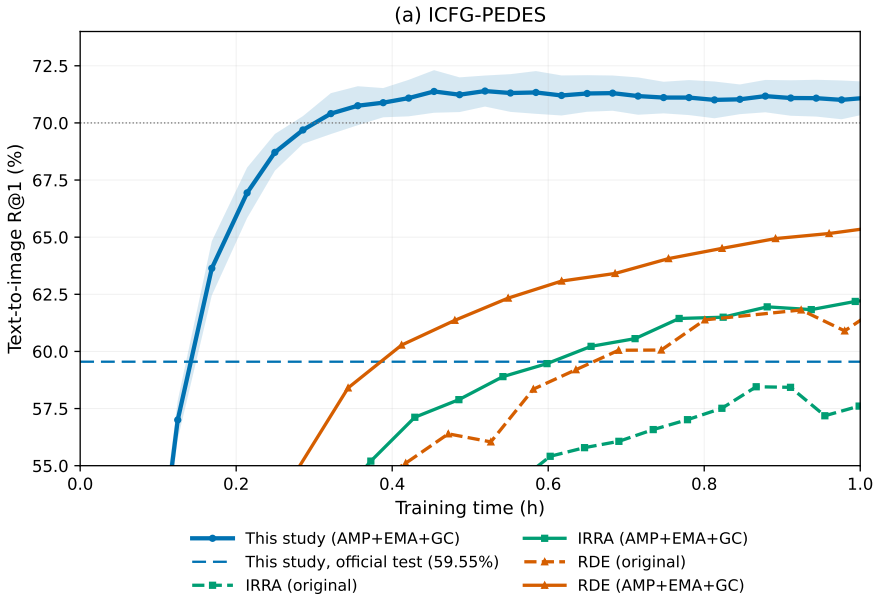

Fig. 2 Text-to-image R@1 against cumulative training hours on ICFG-PEDES, over the first training hour.

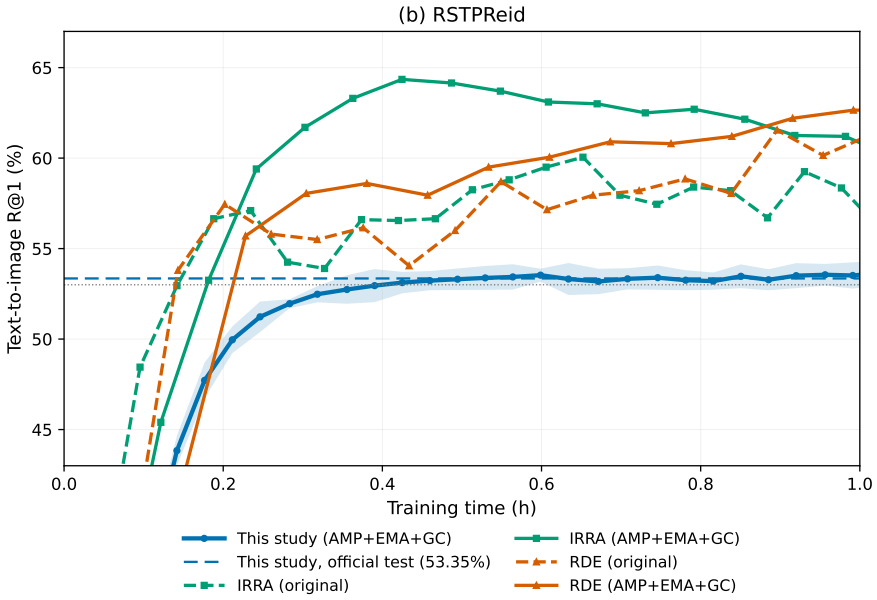

Fig. 3 Text-to-image R@1 against cumulative training hours on RSTPReid, over the first training hour.

4.3.1 Discussion of ICFG-PEDES Results

The discrepancy on ICFG-PEDES is best understood as a difference in evaluation difficulty rather than as a 13-point loss caused by training on only four of the five folds. ICFG-PEDES was introduced by Ding et al. [9] and was constructed from images gathered from MSMT17 [33]. In our five-fold protocol, each held-out fold is an identity-disjoint subset of the official training split. Because identities are shuffled before fold assignment, the fold holdout empirically retains nearly the same camera mixture as the training split: its camera-distribution total-variation (TV) distance from the training split is only 0.023. By contrast, the official test gallery contains 19,848 images from 1,000 identities, whereas a fold gallery contains 6,923 images from 621 identities. Thus, the official test gallery is 2.87× larger and contains 1.61× as many identities. Its camera distribution is also substantially farther from the training distribution, with a TV distance of 0.312.

The camera labels in this analysis, such as `c01`, `c05`, and `c14`, are source-camera identifiers encoded in the MSMT17-style image filenames; they are not identity labels or semantic scene categories. The MSMT17 paper describes the source data as being recorded by a 15-camera network deployed in indoor and outdoor scenes over an extended period with complex lighting variation [33]. Consequently, a camera ID is useful here as a proxy for camera-specific imaging conditions, including viewpoint, background, distance, and illumination. All 15 camera IDs occur in both the official training and test splits, so this is not an unseen-camera or camera-disjoint shift. Rather, it is a camera-composition shift: cameras that are frequent during training

become less prevalent in the official test split, whereas cameras that are rare during training receive substantially greater weight.

Table 9 Selected camera proportions in the ICFG-PEDES official training and official test splits. The Δ column is the official-test proportion minus the training proportion, in percentage points.

Camera ID	Train Split (%)	Test Split (%)	Δ (pp)
c01	23.35	9.92	−13.43
c05	17.56	9.46	−8.10
c14	17.23	9.24	−7.99
c04	0.76	6.84	+6.08
c06	2.09	7.95	+5.85
c09	0.89	6.15	+5.26

For example, 23.35% of the training images are associated with c01, but only 9.92% of the official-test images are. Conversely, c04 is rare in training (0.76%) but accounts for 6.84% of the official test split; the same reweighting is visible for c06 and c09. The larger gallery and identity set therefore introduce more distractors, while the changed camera mixture alters the relative frequency of the camera-specific conditions encountered at test time. The epoch-wise curves provide complementary evidence: the fold-validation and official-test curves have similar shapes, peak at the same epoch (epoch 16), and are approximately parallel with the fold-validation curve shifted upward. This pattern is consistent with an evaluation-set difficulty offset, although it does not show that camera composition alone accounts for the entire performance gap.

4.3.2 Discussion of RSTPReid Results

The performance gains observed on CUHK-PEDES and in ICFG-PEDES fold validation do not extend consistently to RSTPReid, where both validation and test results remain comparatively modest. Annotation quality is one possible contributing factor. Qin et al. [27] report inaccurate or mismatched image–text pairs in the original RSTPReid dataset. Although both ICFG-PEDES and RSTPReid are derived from MSMT17, Scale-TBPS [2] reports markedly different retention rates under the same model-based reliability filter: approximately 90% for ICFG-PEDES, compared with only 60.3% for RSTPReid. For the remaining 39.7% of RSTPReid samples, none of the three pretrained models retrieves the ground-truth image within the top 25 candidates. This difference raises concerns about correspondence quality in RSTPReid, although the filtering rate is not a verified annotation-error rate. We therefore consider annotation quality a plausible contributor to the comparatively low validation and test performance, rather than an established cause.

5 Future Work

The results confirm that combining AMP, EMA, and GC to reach a target performance level within a short training time is a meaningful approach for edge environments.

The natural next step is therefore to investigate architectural changes that raise R@1 further within this same training regime. Applying AMP, EMA, and GC to the reproduced IRRA and RDE baselines also accelerated their early performance gains, but both methods still required training for the full number of epochs reported in their original papers to reach their final published performance. Moreover, applying AMP, EMA, and GC to the reproduced IRRA and RDE implementations could actually increase the total training time relative to their original form. This indicates that an architectural study optimized specifically for the approach proposed here is needed, and that we will pursue architectural directions that are well suited to the approach identified in this study.

6 Conclusions

This study examined the joint use of AMP, EMA, and GC as a cost-efficient training strategy for text-based person search under a single GPU. The three techniques address different aspects of training efficiency: AMP accelerates computation, EMA shortens the time required to obtain a strong checkpoint, and GC reduces activation memory through recomputation. The techniques operate at the training-procedure level and do not require complex additional neural-network modules or modifications to the retrieval architecture. The ablation results show that their benefits are complementary but involve different trade-offs. The full AMP+EMA+GC configuration reduces process-level peak VRAM from 14.15 to 4.23 GiB and reaches its best validation checkpoint in 0.85 hours, compared with 8.54 hours for the FP32 reference. Although GC adds recomputation overhead, AMP accelerates individual updates and EMA enables earlier model selection; together, these gains more than compensate for the time lost to GC.

The secondary-dataset experiments show that validation scores must be interpreted together with the evaluation protocol. Five-fold validation gives 71.70±0.77% t2i R@1 on ICFG-PEDES and 53.97±0.53% on RSTPReid. When the strongest fold models are evaluated once on the official test splits, R@1 is 59.55% and 53.35%, respectively. The 13.02-point ICFG-PEDES gap is associated with a 2.87× larger test gallery and a larger change in camera composition, whereas CUHK-PEDES and RSTPReid show gaps of only 0.95 and 1.53 points. The fold-trained official-test results establish the generalization behavior of the selected checkpoints, but they are not full-training benchmark submissions.

Accordingly, the contribution of this work is not AMP or EMA alone, but the systematic evaluation and joint use of AMP, EMA, and GC within a common training framework. We identify AMP+EMA+GC as the proposed configuration for memory- and time-constrained single-GPU training because it addresses limited VRAM without surrendering the training-time reductions from AMP and EMA. This evidence supports a joint memory–time design criterion rather than selecting a configuration solely by its fastest checkpoint or lowest standalone memory measurement.

Appendix A Caption Duplication in RSTPReid

As discussed in Section 4.1.1, our audit of the official annotation file of RSTPReid [36] revealed a widespread cross-identity caption duplication issue in its official validation split. In text-based person search, retrieval evaluation relies on the premise that a natural-language text query describes the distinctive visual appearance of a single person identity (ID). When identical caption strings are assigned to multiple distinct identities, the evaluation becomes ambiguous: the model cannot uniquely distinguish between different identities from identical text, and retrieving true matching appearances for other identities sharing the same description is penalized as false positives under standard Rank-K and mAP metrics.

Table A1 summarizes cross-identity caption duplication across all three splits of RSTPReid. In the training split, 555 unique captions are shared across multiple identities, affecting 2,916 queries (7.88%). In the official test split, only 4 captions are shared across multiple identities, affecting 10 queries (0.50%). In sharp contrast, the official validation split is disproportionately contaminated: out of 2,000 validation queries (across 200 identities and 1,000 images), 446 queries (22.30%) share identical captions across different identities. These 446 queries stem from 41 unique caption strings that were reused across contiguous identity blocks (e.g., IDs 3829–3843, IDs 3797–3806, and IDs 3819–3828).

Table A1 Cross-identity caption duplication across RSTPReid splits. An affected query is a caption whose exact text is assigned to two or more distinct person identities within the split.

Split	Images	Unique IDs	Total Queries	Shared Captions	Affected Queries (%)
Train	18,505	3,701	37,010	555	2,916 (7.88%)
Validation	1,000	200	2,000	41	446 (22.30%)
Test	1,000	200	2,000	4	10 (0.50%)

Table A2 provides representative concrete samples extracted directly from the official annotation file (`data_captions.json`), illustrating the exact caption strings, the number of distinct person identities sharing them, the total query occurrences, and example image file paths. For instance, the caption *"The man is wearing a blue and grey overcoat and a pair of black trousers and there is something in his hand."* is duplicated across 15 distinct identities (IDs 3829–3843) across 48 query instances. This high degree of label ambiguity validates our choice to employ five-fold cross-validation on the training set rather than relying on the official validation split for model selection.

Table A2 Representative samples of cross-identity duplicate captions in the RSTPReid validation split, extracted directly from `data_captions.json`.

Identical Caption Text	Shared IDs	Queries	Distinct ID Range	Sample Image Files
"The man is wearing a blue and grey overcoat and a pair of black trousers and there is something in his hand."	15	48	IDs 3829–3843	`3829_c11_0017.jpg`, `3830_c11_0021.jpg`, `3836_c11_0014.jpg`
"The man is wearing a deep black overcoat and a pair of black trousers and he is wearing a pair of glasses."	10	45	IDs 3797–3806	`3797_c1_0000.jpg`, `3798_c7_0002.jpg`, `3801_c7_0011.jpg`
"The man is wearing a grey sports shirt and he is wearing a pair of black trousers and he is catching a black bag."	10	33	IDs 3819–3828	`3819_c1_0006.jpg`, `3820_c14_0013.jpg`, `3823_c13_0007.jpg`
"The woman is wearing a long pink overcoat and she is catching a red backpack and she is wearing a pair of black stocks."	9	44	IDs 3784–3792	`3784_c14_0005.jpg`, `3785_c14_0004.jpg`, `3788_c14_0001.jpg`
"The man is wearing a deep black overcoat and he is wearing a pair of black trousers and there is something in his hand."	5	19	IDs 3851–3855	`3851_c6_0013.jpg`, `3852_c11_0026.jpg`, `3853_c6_0015.jpg`
"The woman is wearing a deep black overcoat and she is wearing a pair of deep black stocks and she is wearing a pair of sneakers."	5	17	IDs 3869–3873	`3869_c15_0017.jpg`, `3870_c11_0018.jpg`, `3872_c14_0016.jpg`

Figure A1 illustrates visual image samples corresponding to three representative duplicate caption groups from the official validation split. As demonstrated in the figure, pedestrians with completely different visual appearances, genders, and clothing items (e.g., blue coats, dark jackets, red parkas, and women with handbags) are assigned identical natural-language descriptions. During evaluation, querying any of these descriptions will penalize the retrieval of the other distinct identities as false positives despite the identical description, explaining the artificial performance degradation on the official validation split.

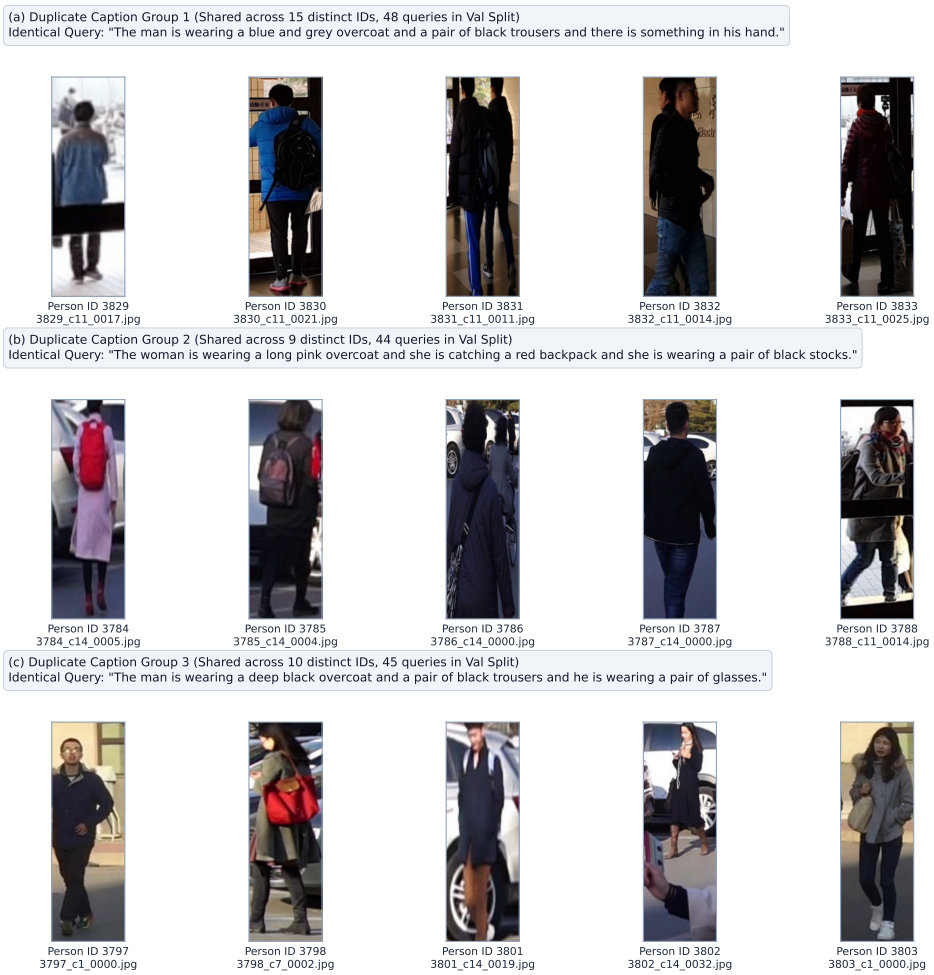

Fig. A1 Visual image samples from the RSTPReid validation split sharing identical text captions across different person identities. In each group, pedestrians possessing distinct visual appearances and person IDs are annotated with the exact same query text.

References

[1] Cao M, Bai Y, Zeng Z, et al (2024) An empirical study of CLIP for text-based person search. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 465–473, https://doi.org/10.1609/aaai.v38i1.27801, URL https://ojs.aaai.org/index.php/AAAI/article/view/27801

[2] Chatterjee N, Garg S, Subramanyam AV, et al (2026) Unified multi-dataset training for TBPS. URL https://arxiv.org/abs/2601.14978, arXiv:2601.14978

[3] Chen F, He J, Liu Y, et al (2026) Text-based person search via fine-grained cross-modal semantic alignment. Signal Processing: Image Communication 142:117478. https://doi.org/10.1016/j.image.2026.117478

[4] Chen J, Ran X (2019) Deep learning with edge computing: A review. Proceedings of the IEEE 107(8):1655–1674. https://doi.org/10.1109/JPROC.2019.2921977

[5] Chen T, Xu B, Zhang C, et al (2016) Training deep nets with sublinear memory cost. URL https://arxiv.org/abs/1604.06174, arXiv:1604.06174

[6] Chen Y, Zhang G, Lu Y, et al (2022) TIPCB: A simple but effective part-based convolutional baseline for text-based person search. Neurocomputing 494:171–181

[7] Coleman C, Kang D, Narayanan D, et al (2019) Analysis of DAWNBench, a time-to-accuracy machine learning performance benchmark. ACM SIGOPS Operating Systems Review 53(1):14–25. https://doi.org/10.1145/3352020.3352024, URL https://arxiv.org/abs/1806.01427

[8] Dehghani M, Arnab A, Beyer L, et al (2022) The efficiency misnomer. In: International Conference on Learning Representations, URL https://arxiv.org/abs/2110.12894

[9] Ding Z, Ding C, Shao Z, et al (2021) Semantically self-aligned network for text-to-image part-aware person re-identification. URL https://arxiv.org/abs/2107.12666, arXiv:2107.12666

[10] Farooq A, Awais M, Kittler J, et al (2022) AXM-Net: Implicit cross-modal feature alignment for person re-identification. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 4477–4485

[11] Han X, He S, Zhang L, et al (2021) Text-based person search with limited data. In: British Machine Vision Conference, pp 1–13

[12] Huang K, Yang B, Gao W (2023) ElasticTrainer: Speeding up on-device training with runtime elastic tensor selection. In: Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services, pp 440–453, https://doi.org/10.1145/3581791.3596852

[13] Huang Y, Zhang C, Li Z, et al (2025) Prototypical graph alignment for text-based person search. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp 1–5, https://doi.org/10.1109/ICASSP49660.2025.10890808

[14] Jiang D, Ye M (2023) Cross-modal implicit relation reasoning and aligning for text-to-image person retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2787–2797, URL https://openaccess.thecvf.com/content/CVPR2023/

html/Jiang_Cross-Modal_Implicit_Relation_Reasoning_and_Aligning_for_Text-to-Image_Person_Retrieval_CVPR_2023_paper.html

[15] Khosla P, Teterwak P, Wang C, et al (2020) Supervised contrastive learning. In: Advances in Neural Information Processing Systems, pp 18661–18673, URL https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html

[16] Kim G, Eom C (2026) DiCo: Disentangled concept representation for text-to-image person re-identification. Neurocomputing 675:132885. https://doi.org/10.1016/j.neucom.2026.132885, URL https://doi.org/10.1016/j.neucom.2026.132885

[17] Kwon YD, Li R, Venieris S, et al (2024) TinyTrain: Resource-aware task-adaptive sparse training of DNNs at the data-scarce edge. In: Proceedings of the 41st International Conference on Machine Learning, pp 25812–25843, URL https://proceedings.mlr.press/v235/kwon24c.html

[18] Li S, Xiao T, Li H, et al (2017) Person search with natural language description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1970–1979, https://doi.org/10.1109/CVPR.2017.551, URL https://arxiv.org/abs/1702.05729

[19] Li S, Cao M, Zhang M (2022) Learning semantic-aligned feature representation for text-based person search. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp 2724–2728

[20] Liu Y, Li Y, Liu Z, et al (2024) CLIP-based synergistic knowledge transfer for text-based person retrieval. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp 7935–7939

[21] Mattson P, Cheng C, Diamos G, et al (2020) MLPerf training benchmark. In: Proceedings of Machine Learning and Systems, pp 336–349, URL https://proceedings.mlsys.org/paper_files/paper/2020/hash/02522a2b2726fb0a03bb19f2d8d9524d-Abstract.html

[22] Micikevicius P, Narang S, Alben J, et al (2018) Mixed precision training. In: International Conference on Learning Representations (ICLR), URL https://arxiv.org/abs/1710.03740, arXiv:1710.03740

[23] Min H, Zhan G, Fan C, et al (2026) Parameter-efficient transfer for CLIP-based text-to-person retrieval. Signal Processing: Image Communication 147:117598. https://doi.org/10.1016/j.image.2026.117598

[24] Morales-Brotons D, Vogels T, Hendrikx H (2024) Exponential moving average of weights in deep learning: Dynamics and benefits. Transactions on Machine Learning Research URL https://arxiv.org/abs/2411.18704, arXiv:2411.18704 [cs.LG]

[25] Morrison J, Na C, Fernandez J, et al (2025) Holistically evaluating the environmental impact of creating language models. In: Yue Y, Garg A, Peng N, et al (eds) International Conference on Learning Representations, pp 73928–73943, URL https://proceedings.iclr.cc/paper_files/paper/2025/file/b76105b94a8c1286c860ad885ec588f1-Paper-Conference.pdf

[26] Patil SG, Jain P, Dutta P, et al (2022) POET: Training neural networks on tiny devices with integrated rematerialization and paging. In: Proceedings of the 39th International Conference on Machine Learning, pp 17573–17583, URL https://proceedings.mlr.press/v162/patil22b.html

[27] Qin Y, Chen Y, Peng D, et al (2024) Noisy-correspondence learning for text-to-image person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 27197–27206, https://doi.org/10.1109/CVPR52733.2024.02568, URL https://openaccess.thecvf.com/content/CVPR2024/html/Qin_Noisy-Correspondence_Learning_for_Text-to-Image_Person_Re-identification_CVPR_2024_paper.html

[28] Radford A, Kim JW, Hallacy C, et al (2021) Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning, pp 8748–8763, URL https://proceedings.mlr.press/v139/radford21a.html

[29] Rajagopal A, Bouganis CS (2021) perf4sight: A toolflow to model CNN training performance on edge GPUs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp 963–971, URL https://openaccess.thecvf.com/content/ICCV2021W/ERCVAD/papers/Rajagopal_perf4sight_A_Toolflow_To_Model_CNN_Training_Performance_on_Edge_ICCVW_2021_paper.pdf

[30] Shao Z, Zhang X, Fang M, et al (2022) Learning granularity-unified representations for text-to-image person re-identification. In: Proceedings of the 30th ACM International Conference on Multimedia, pp 5566–5574

[31] Shu X, Wen W, Wu H, et al (2022) See finer, see more: Implicit modality alignment for text-based person retrieval. In: European Conference on Computer Vision. Springer, pp 624–641

[32] Strubell E, Ganesh A, McCallum A (2019) Energy and policy considerations for deep learning in NLP. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp 3645–3650, https://doi.org/10.18653/v1/P19-1355, URL https://aclanthology.org/P19-1355/

[33] Wei L, Zhang S, Gao W, et al (2018) 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, https://doi.org/10.1109/CVPR.

2018.00016, URL https://openaccess.thecvf.com/content_cvpr_2018/html/Wei_Person_Transfer_GAN_CVPR_2018_paper.html

[34] Yan S, Dong N, Zhang L, et al (2023) CLIP-driven fine-grained text-image person re-identification. IEEE Transactions on Image Processing 32:6032–6046

[35] Zhang Y, Chen C, Ding T, et al (2024) Why transformers need Adam: A hessian perspective. In: Advances in Neural Information Processing Systems, pp 131786–131823, https://doi.org/10.52202/079017-4190, URL https://doi.org/10.52202/079017-4190

[36] Zhu A, Wang Z, Li Y, et al (2021) DSSL: Deep surroundings-person separation learning for text-based person retrieval. In: Proceedings of the 29th ACM International Conference on Multimedia, pp 209–217, https://doi.org/10.1145/3474085.3475369, URL https://arxiv.org/abs/2109.05534