Asymmetric Feature Fusion for Image Retrieval

Hui Wu, Min Wang, Wengang Zhou, Zhenbo Lu, Houqiang Li

1CAS Key Laboratory of Technology in GIPAS, University of Science and Technology of China
2Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

wh241300@mail.ustc.edu.cn, {wangmin, luzhenbo}@iai.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn

Abstract

In asymmetric retrieval systems, models with different capacities are deployed on platforms with different computational and storage resources. Despite the great progress, existing approaches still suffer from a dilemma between retrieval efficiency and asymmetric accuracy due to the limited capacity of the lightweight query model. In this work, we propose an Asymmetric Feature Fusion (AFF) paradigm, which advances existing asymmetric retrieval systems by considering the complementarity among different features at the gallery side. Specifically, it first embeds each gallery image into various features, e.g., local features and global features. Then, a dynamic mixer is introduced to aggregate these features into compact embedding for efficient search. On the query side, only a single lightweight model is deployed for feature extraction. The query model and dynamic mixer are jointly trained by sharing a momentum-updated classifier. Notably, the proposed paradigm boosts the accuracy of asymmetric retrieval without introducing any extra overhead to the query side. Exhaustive experiments on various landmark retrieval datasets demonstrate the superiority of our paradigm.

1. Introduction

Image retrieval [17, 30, 34, 40, 49, 53] has been studied for a long time in the literature. Typically, high-performing image retrieval systems deploy a large powerful model to embed both query and gallery images, which is widely known as symmetric image retrieval. However, in some real-world applications, e.g., mobile search, the query side is constrained by resource limitation and thus cannot meet the overhead of deploying a large model. To this end, the paradigm of asymmetric image retrieval is first proposed in HVS [9] and AML [3], which has attracted increasing attention from the community [26, 38, 43, 56, 62]. In such a paradigm, deep representation models with different capacities are first trained to be compatible and then deployed on platforms with different resources to strike a balance between retrieval accuracy and efficiency, as shown in Fig. 1a.

Figure 1. Illustration of (a) previous single-feature asymmetric retrieval pipeline and (b) our asymmetric feature fusion paradigm. Due to limited capacity of the lightweight model, existing pipeline achieves efficiency for the query side at the cost of retrieval accuracy degradation. In contrast, our approach enhances existing asymmetric retrieval pipeline from the perspective of gallery feature fusion. For efficient retrieval, a dynamic mixer is introduced to aggregate multiple gallery features into a compact embedding. Query model and mixer are jointly trained with compatible constraints. Our method realizes high efficiency without sacrificing retrieval accuracy.

For an asymmetric retrieval system, the most crucial thing is to ensure that the embedding spaces of different models are well aligned. To this end, BCT [38] first proposes a backward-compatible learning framework, in which the classifier of the gallery model is inherited to guide the learning of the query model. Recently, various efforts have been devoted to improving cross-model feature compatibility in terms of training objectives [3, 26, 51, 56, 57], model structures [8, 9], etc. Despite the great progress, a dilemma...
is still unresolved, i.e., the accuracy of asymmetric retrieval is still unsatisfactory compared to that of symmetric retrieval, especially in limited-resource and large-scale scenarios, as shown in Fig. 2 (□ vs. ○, . . . , △ vs. +). We argue that such dilemma is due to the low capacity of lightweight query model, which cannot perfectly achieve feature compatibility with the static powerful gallery model.

To alleviate above issue, we introduce a new paradigm named Asymmetric Feature Fusion (AFF). It boosts the accuracy of existing asymmetric retrieval systems by considering the complementarity among different features, as shown in Fig. 1b. On the gallery side, it deploys several large powerful models on the cloud to extract diverse features, e.g., local features, which are suitable for capturing local matches, or global features that are effective for holistic semantic matching. For efficient retrieval, a dynamic mixer is further proposed to aggregate diverse gallery features into compact embedding, which allows efficient vector search [11, 18] to be exploited. As for the query side, queries are embedded with a single lightweight model. It eliminates time-consuming multiple feature extraction and aggregation processes, realizing a solution suitable for resource-constrained platforms. During training, all the gallery models are fixed, while the mixer and query model are trained jointly by a momentum-updated classifier for achieving feature compatibility.

Compared to previous retrieval approaches, the proposed paradigm has two unique advantages. First, it fuses various features on the gallery side, which notably advances the retrieval accuracy of existing asymmetric retrieval systems.

Although the extraction and aggregation processes increase the computational complexity, they are typically performed on resource-rich cloud platforms. In addition, gallery images are embedded offline in advance, whose computational overhead has no influence on the query side. Second, compared with multi-feature fusion methods, our paradigm only deploys a single lightweight model on the query side, which is free of the complex and time-consuming multi-feature extraction and aggregation. Thus, it introduces no extra computational and storage overhead for the query side. Overall, with the proposed asymmetric feature fusion paradigm, our approach achieves high retrieval efficiency and accuracy simultaneously, as shown in Fig. 2 (□ vs. ○, . . . , △ vs. +).

To evaluate our approach, comprehensive experiments are conducted on popular landmark retrieval datasets. The proposed paradigm realizes promising performance improvement for existing asymmetric retrieval systems and leads to the state-of-the-art results across the public benchmarks.

2. Related Work

Feature Representation. In image retrieval, feature representation plays a key role. Hand-crafted local features [2, 24] are widely used in early image retrieval systems [27, 32, 40]. Recently, local features extracted from convolutional neural networks (CNNs) are shown to be more effective [7, 29, 31, 46]. They learn feature detection and representation jointly by attention mechanism [31, 48, 52, 55] or non-maximal suppression [10]. The detected local features are further utilized for geometric verification [32] or aggregated into compact representations by VLAD [16], ASMK [47], etc., for efficient retrieval. Recently, global features such as RMAC [49], GeM [34], DELG [4], DOLG [59], Token [54] and CVNet [21], are typically extracted from CNNs by spatial pooling [34, 41, 42], which demonstrate more effectiveness in holistic semantic matching over local features.

Despite the great progress, existing image retrieval systems usually deploy large powerful models for high retrieval accuracy. However, some real-world applications need to deploy query models on resource-constrained platforms, e.g., mobile phones, which cannot meet the demand of large models for computational and storage resources. To address this issue, our approach focuses on the setting of asymmetric retrieval, where the query side deploys a lightweight model while the gallery side applies a large one.

Feature Compatibility. The paradigm of feature compatibility learning is first proposed by BCT [38]. It enforces the feature of the query model to be close to the corresponding class centroids of the gallery model. Under this paradigm, several efforts [3, 9, 26, 56, 62] have been devoted to improving the feature compatibility across different models. Specifically, AML [3] introduces asymmetric regression loss and contrastive loss to train the query model. CSD [56] takes a step further by constraining the query
model to maintain the nearest neighbor structure in the embedding space of the gallery model. Recently, LCE [26] proposes to align the classifiers of different models with a tight boundary loss. HVS [9] further resorts to neural architecture search technique to search for the optimal compatibility-aware model architecture. FCT [36] stores “side information”, which is later leveraged to transfer the gallery features for other retrieval tasks. Besides, when solving the model regression problem, methods including PCT [58], REG-NAS [8] and RACT [61], also utilize feature compatibility to alleviate “negative flip”.

Differently, to boost existing asymmetric retrieval systems, we introduce a new asymmetric feature fusion paradigm. It enhances the discriminativeness of image features by aggregating diverse features just at the gallery side. Our approach is readily combined with existing methods to achieve better asymmetric retrieval accuracy efficiently.

**Lightweight Network.** The architecture of deep convolutional neural networks [13, 20] has been evolving for many years. As the application complexity increases [35], model size becomes larger, requiring more computational and storage resources. However, besides accuracy, resource overhead is another important consideration in real-world applications. Real-world tasks usually expect to deploy optimal models on the target resource-constrained platforms. The immediate demand motivates a series of work, e.g., MobileNets [14, 37], ShuffleNets [25, 65], GhostNets [12] and EfficientNets [45], for lightweight model design.

In this work, we focus on asymmetric retrieval in resource-constrained scenarios. Various lightweight models mentioned above are utilized as query models on resource-constrained end platforms.

**Feature Fusion.** Feature fusion has been widely studied in computer vision, e.g., detection [22, 60], multimedia retrieval [15, 39], etc. As for image retrieval, it is broadly divided into three levels. The first is feature level, where features of different modalities [5], scales [42, 59], etc., are effectively fused into a single feature. The second is indexing level, where multiple features are jointly indexed [64, 66], or multiple visual vocabularies are fused together [68]. The last is ranking level. Given several ranking lists returned by different retrieval algorithms, graph-based [63], or context-based [67] methods fuse them into the final ranking list.

However, all these methods require to extract multiple features on the query side. It inevitably increases the computational and storage complexity, which is hardly affordable for resource-constrained platforms. In contrast, we introduce a new asymmetric feature fusion paradigm, in which only a single lightweight model is deployed to embed queries and the gallery set is processed offline by various large powerful models on the cloud platforms. The proposed paradigm boosts the accuracy of asymmetric retrieval without adding any extra overhead to the query side.

### 3. Preliminary on Asymmetric Retrieval

Asymmetric image retrieval aims to deploy models of different sizes on different platforms to realize search efficiency while preserving retrieval accuracy. Given a query set \( Q \) and a gallery set \( G \), query model \( \phi_q : x \to \mathbb{R}^D \) and gallery model \( \phi_g : x \to \mathbb{R}^D \) are deployed to embed them into \( L_2 \)-normalized features, respectively. Then, the cosine similarities or Euclidean distances between query and gallery features are calculated to measure the similarities between images. Usually, an asymmetric retrieval system is expected to achieve similar accuracy as that of a symmetric retrieval system, i.e., \( \mathcal{M}(\phi_q(Q), \phi_g(G)) \approx \mathcal{M}(\phi_g(Q), \phi_g(G)) \), where \( \mathcal{M}(\cdot, \cdot) \) denotes the evaluation metric for retrieval, e.g., mAP or Recall@K.

Despite the promising performance achieved by existing asymmetric retrieval methods, we still observe notable retrieval accuracy degradation when compared to deploying large powerful models on both query and gallery sides (\( \mathcal{O}, \triangle, \cdots, \triangle \text{ vs. } + \) in Fig. 2), i.e., \( \mathcal{M}(\phi_q(Q), \phi_g(G)) > \mathcal{M}(\phi_g(Q), \phi_g(G)) \). This is due to the limited capacity of lightweight models, which cannot perfectly achieve feature compatibility with large powerful models.

In this work, we alleviate the dilemma from the perspective of feature fusion and a new asymmetric feature fusion paradigm is introduced. Specifically, various large powerful models are deployed on the gallery side to extract features, which are further aggregated into compact embedding with a mixer. As for the query side, only a lightweight model is deployed, which is jointly trained with the mixer for feature compatibility. The proposed paradigm improves the accuracy of asymmetric retrieval systems without introducing any overhead to the resource-constrained query side.

### 4. Asymmetric Feature Fusion

#### 4.1. Overview

As shown in Fig. 3, our AFF consists of multiple global feature models \( \{\phi^i_g : x \to \mathbb{R}^{D_i}\}_{i=1}^K \) and local feature models \( \{\phi^i_l : x \to \mathbb{R}^{N_i \times d_i}\}_{i=1}^M \) on the gallery side and a lightweight model \( \phi_q : x \to \mathbb{R}^d \) on the query side. Let \( \mathcal{T} \) denote a training dataset. On the gallery side, each image \( x \in \mathcal{T} \) is first embedded into multiple global features \( G = \{g^i \in \mathbb{R}^{D_i}\}_{i=1}^K \) and several sets of local features \( L = \{l^i \in \mathbb{R}^{N_i \times d_i}\}_{i=1}^M \), respectively:

\[
g^i = \phi^i_g(x), \quad i = 1, 2, \ldots, K, \quad (1)
\]
\[
l^i = \phi^i_l(x), \quad i = 1, 2, \ldots, M. \quad (2)
\]

Typically, each local feature is associated with a coordinate tuple and a scale factor, indicating the location and image scale from which it is extracted. Our method ignores these information. All the global and local features...
are mapped to the same dimension of $d$ by the corresponding fully-connected layers:

$$f_g^i = g^i W_g^i \in \mathbb{R}^{d_i}, \quad i = 1, 2, \ldots, K,$$

(3)

$$f_l^i = l^i W_l^i \in \mathbb{R}^{n_i \times d_i}, \quad i = 1, 2, \ldots, M.$$  

(4)

After that, various gallery features are stacked together to form a feature sequence:

$$F = [f_g^1; \ldots; f_g^K; f_l^1; \ldots; f_l^M] \in \mathbb{R}^{N \times d},$$  

(5)

where $N = K + \sum_{i=1}^M n_i$ is the total number of the gallery features. To reduce the storage overhead of the gallery side and improve search efficiency, a mixer $\phi_{mix}: \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^d$ (Sec. 4.2) is further introduced to transform $F$ into a compact embedding $g_{mix} = \phi_{mix}(F) \in \mathbb{R}^d$. On the query side, the same training image $x$ is embedded into $q$ by the lightweight query model: $q = \phi_q(x) \in \mathbb{R}^d$.

During training, the well-trained gallery models are kept frozen. Only the dynamic mixer $\phi_{mix}$ and the query model $\phi_q$ are jointly trained for feature compatibility. The final objective function (Sec. 4.3) consists of two losses:

$$\arg \min_{\phi_{mix}, \phi_q} \frac{1}{|T|} \sum_{x \in T} (\ell_{disc}(\phi_{mix}, x) + \ell_{comp}(\phi_q, x)),$$  

(6)

where $\ell_{disc}(\phi_{mix}; x)$ ensures the discrimination of the aggregated feature $g_{mix}$, and $\ell_{comp}(\phi_q; x)$ is designed to align query feature $q$ and aggregated feature $g_{mix}$ in the same latent space so that they are mutually compatible.

### 4.2. Dynamic Mixer

Given an image encoded by various features $F$, feature fusion aims to combine those features for better retrieval accuracy. A simple way is to concatenate various features and perform dimension reduction, which is implemented by several fully-connected layers (Fig. 4a). However, it leads to an over-parameterized mixer when the number of features is large, which may cause overfitting.

In this work, attention mechanism [50] is adopted to aggregate various features (Fig. 4b). A learnable fusion token $f_{fusion} \in \mathbb{R}^d$ is first added to the top of $F$ to form the input:

$$F_{input} = [f_{fusion}; F] \in \mathbb{R}^{(N+1) \times d}.$$  

(7)

Then, $F_{input}$ is iteratively processed $C$ times by a trans-
former layer, which is formulated as:
\[
\tilde{Z}^{i+1} = \text{LN}(Z^i + \text{MHSA}(Z^i)),
\]
\[
Z^{i+1} = \text{LN}((\tilde{Z}^{i+1} + \text{MLP}(\tilde{Z}^{i+1})))
\]
\[
\text{MLP}(\tilde{Z}^{i+1}) = \text{GeLU}(\tilde{Z}^{i+1}W_1)W_2,
\]
t = 0, \ldots, C - 1,
where \( Z^0 = F_{\text{input}} \); MHSA is the Multi-Head Self-Attention \([50]\); MLP is a two-layer perceptron with parameter matrices \( W_1 \in \mathbb{R}^{d \times d^*} \) and \( W_2 \in \mathbb{R}^{d^* \times d} \), and an intermediate dimension \( d_e = 2 \times d \); LN is the layer normalization \([1]\). The final output fusion token \( \tilde{f}_{\text{fusion}} = Z^C_{0,:} - 1 \)

4.3. Training Objective Functions

To ensure the superiority of our asymmetric feature fusion paradigm, there are two requirements needed to be guaranteed. First, the aggregated feature \( g^\text{mix} \) is expected to be more discriminative than any single gallery feature. To this end, following the state-of-the-art metric learning methods \([4,21,59]\) in image retrieval, classification is adopted as the aggregated feature \( g^\text{mix} \) for the gallery side.

\[
\ell_{\text{disc}}(\phi_{\text{mix}}, x) = -\log\frac{e^{s \cdot \cos(\theta^1_y + m)}}{\sum_{j \neq y} e^{s \cdot \cos(\theta^1_j)}} + \sum_{j \neq y} e^{s \cdot \cos(\theta^1_j)},
\]

where \( y \) is the label of the training image \( x \), \( s \) is a scale factor, \( m \) is the margin, and \( \theta^1_y = \arccos\left(\frac{\omega^\text{mix}}{\|\omega^\text{mix}\|}, g^\text{mix}\right) \) is the angle between the \( y \)-th \( L_2 \)-normalized prototype of the classifier \( \psi^\text{mix} \) and the feature \( g^\text{mix} \).

Second, query feature \( q \) and aggregated feature \( g^\text{mix} \) should be compatible with each other. One may share the same classifier between \( \phi^\text{mix} \) and \( \phi_q \), which has shown effectiveness in previous methods \([9,26,38]\). However, our approach expects to train \( \phi^\text{mix} \) and \( \phi_q \) jointly. Simply sharing classifier couples the training of networks with different capabilities, which may damage the discriminative capability of the aggregated embedding \( g^\text{mix} \). Besides, the classifier parameters \( \omega^\text{mix} \) evolve rapidly, which cannot provide a stable target to the query model. To this end, we decouple the training processes of \( \phi^\text{mix} \) and \( \phi_q \), while ensuring feature compatibility through a momentum update mechanism. ArcFace loss is still adopted for training query model \( \phi_q \):

\[
\ell_{\text{comp}}(\phi_q, x) = -\log\frac{e^{s \cdot \cos(\theta^2_y + m)}}{e^{s \cdot \cos(\theta^2_y + m)} + \sum_{j \neq y} e^{s \cdot \cos(\theta^2_j)}},
\]

where \( \theta^2_y = \arccos\left(\frac{\omega^\text{q}}{\|\omega^\text{q}\|}, q\right) \) is the angle between the \( y \)-th \( L_2 \)-normalized prototype of the classifier \( \psi^\text{q} \) and the feature \( q \). Differently, the parameter \( \omega^\text{q} \) is not updated through back-propagation, but a moving-averaged version of \( \omega^\text{mix} \):

\[
\omega^\text{q} \leftarrow \alpha \omega^\text{q} + (1 - \alpha) \omega^\text{mix},
\]

where \( \alpha \in [0, 1) \) is a momentum coefficient. Only the parameters \( \omega^\text{mix} \) are updated by back-propagation. This momentum update in Eq. (11) decouples the training of \( \phi^\text{mix} \) and \( \phi_q \) while making \( \omega^\text{q} \) evolve more smoothly than \( \omega^\text{mix} \).

5. Experiments

5.1. Experimental Setup

Evaluation Datasets and Metrics. We evaluate the proposed framework on three landmark retrieval datasets, including GLDv2-Test \([53]\), Revisited Oxford (ROx) \([33]\). GLDv2-Test contains 761, 757 gallery images, and 390/750 images as public/private query sets, respectively. The evaluation metric is mAP@100. As for ROx and RP, there are 70 queries for both of them, with 4, 993 and 5, 007 gallery images, respectively. mAP on the Medium and Hard settings are reported. Large-scale results are reported with the \( \text{R} \text{M} \) \([33]\) dataset added (+ 1M).

Gallery and Query models. Four global features including DELG \([4]\), Token \([54]\), DOLG \([59]\), CVNet \([21]\) and two local features HOW \([48]\), *DELG \([4]\) are adopted as gallery features. As for query model, we only keep the feature extractor of lightweight models, e.g., ShuffleNets \([25]\) and MobileNets \([37]\), with a GeM pooling \([34]\) layer and a whitening layer added at the end.

Training Details. GLDv2 \([53]\) is adopted for training, which consists of 1, 580, 470 images with 81, 311 classes. All the gallery features are extracted offline for training efficiency. During training, a 512 × 512-pixel region is cropped from each randomly resized training image, followed by random color jittering. We jointly train the mixer and query model for 20 epochs on four NVIDIA RTX 3090 GPUs with a batch size of 128. SGD is adopted as the optimizer with a weight decay of 0.01 and an initial learning rate of 0.001, which linearly decays to 0 when the desired number of steps is reached. \( C \) in Eq. (8) is set to 4. \( d \) and \( d_e \) in Eq. (8) are both set to 2, 048. Margin \( m \) and scale \( s \) in Eq. (9) and Eq. (10) are set as 0.3 and 32.0, respectively.

5.2. Ablation Study

Variants of the mixer. In Tab. 1, we compare our proposed mixer against simple baseline (MLP), LAFF \([15]\), and Or-thF. \([59]\). Our transformer-based mixer iteratively extracts useful information from various gallery features via a fusion token, which achieves the best retrieval accuracy.

Impact of different gallery features. In Tab. 2, we investigate how our method responds when different gallery features are added. Results show that each feature is beneficial for the task and features have different effects when
Mixer dynamically extracts informative features from direct feature ensemble, performance degradation in this setup. To summarize, our mixer reduces performance drops from \(71\) to \(52\). This further leads to a huge performance enhancement.

**Gaussian White Noise**

is almost unaffected. We further consider an extreme case, e.g., \(m\) or \(n\), for noise features. First, three feature extractors are deployed to embed query images. Mixer is adopted as gallery model, and CVNet [21]. RGeM [34], VGeM [34] and RMAC [34] are added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\).

In Tab. 3, we investigate the robustness of the proposed framework to noisy features. First, three feature extractors with unsatisfactory performance including RGeM [34], VGeM [34], and RMAC [49], are added to the gallery model zoo. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\).

In Tab. 3, we investigate the robustness of the proposed framework to noisy features. First, three feature extractors with unsatisfactory performance including RGeM [34], VGeM [34], and RMAC [49], are added to the gallery model zoo. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\).

In Tab. 4, we compare our method with traditional ASMK [47]-based local feature aggregation approach and contrast it with other existing methods. Our method achieves superior performance, especially in terms of mAP. This demonstrates the effectiveness of our proposed mixer for image retrieval and retrieval is performed in different gallery sets. Our method dynamically aggregates diverse features to enhance the representation of gallery images.

**Robustness.** When it comes to feature fusion, there is usually no idea whether a specific feature is beneficial for the current task or not in real-world applications. Feature fusion inevitably introduces some noisy features.

In Tab. 3, we investigate the robustness of the proposed framework to noisy features. First, three feature extractors with unsatisfactory performance including RGeM [34], VGeM [34], and RMAC [49], are added to the gallery model zoo. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\).

In Tab. 4, we compare our method with traditional ASMK [47]-based local feature aggregation approach and contrast it with other existing methods. Our method achieves superior performance, especially in terms of mAP. This demonstrates the effectiveness of our proposed mixer for image retrieval and retrieval is performed in different gallery sets. Our method dynamically aggregates diverse features to enhance the representation of gallery images.

**Robustness.** When it comes to feature fusion, there is usually no idea whether a specific feature is beneficial for the current task or not in real-world applications. Feature fusion inevitably introduces some noisy features.

In Tab. 3, we investigate the robustness of the proposed framework to noisy features. First, three feature extractors with unsatisfactory performance including RGeM [34], VGeM [34], and RMAC [49], are added to the gallery model zoo. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\). Our method is almost unaffected. We further consider an extreme case, in which **Gaussian White Noise** is directly added to the gallery model. These noisy features cause significant performance degradation for direct feature ensemble, e.g., mAP on \(\text{ROx} + 1\text{M}\) drops from \(75.30\) to \(71.52\).
proaches [48, 52]. Even under the asymmetric setting, it achieves better accuracy with less overhead for the query side. In Tab. 5b, our method also achieves higher accuracy than re-ranking [32, 44] methods under the asymmetric setting. Note that the re-ranking process introduces significant retrieval latency, while the proposed paradigm only needs efficient vector search. To summarize, our method aggregates diverse features on the gallery side, which greatly reduces the overhead for the query side and online retrieval latency while improving retrieval accuracy.

**Analysis of the training strategy.** Our AFF adopts the momentum update mechanism to jointly train the mixer and query model. Here, we expect to answer two questions:

(1) **Whether momentum update is necessary or not?** Tab. 6 shows the accuracy of the asymmetric retrieval with different momentum values \(\alpha\) in Eq. (11). The accuracy increases gradually as \(\alpha\) increases. It performs well when \(\alpha\) is in 0.99 \(\sim\) 0.999, showing that a slowly progressing classifier \(\psi_q\) is beneficial. The importance of the momentum update is also confirmed by the results in Tab. 8. When our AFF is combined with different approaches, coupling the training process of mixer and query model leads to significant performance degradation, e.g., the mAP on ROFx + 1M drops from 69.66 to 48.31 for “Mixer + CSD [56]”. All these results support our motivation of decoupling the training of the mixer and query model.

(2) **Is joint training more effective than two-stage training?** Two-stage training means that the mixer is first trained, followed by the compatibility training of the query model. It avoids the difficulties arising in joint training but leads to a more complex and time-consuming training process. Besides, due to the limited capacity of the query model, directly aligning it to a static powerful model limits the feature compatible training, which is confirmed by previous methods [19, 28]. Experiment results are shown in Tab. 7. When
our AFF is combined with various existing methods, joint training yields a consistent performance boost.

**Generalizability analysis.** In fact, our AFF is feasible to be combined with various existing methods, which leads to different training loss $\ell_{\text{comp}}(\phi_q, x)$. For example, when combined with REG [3], we maintain a momentum-updated version $\bar{\phi}_{\text{mix}}^\phi$ of mixer $\phi_{\text{mix}}$, whose output feature is denoted as $\bar{g}^\phi$. Then, $\ell_{\text{comp}}(\phi_q, x)$ is formulated as $\ell_{\text{comp}}(\phi_q, x) = \|q - \bar{g}^\phi\|_2^2$. More details about the combinations with existing methods, e.g., CSD [56] and LCE [26], are provided in the supplementary materials. As shown in Tab. 8, our method enhances all existing asymmetric retrieval systems, leading to a promising accuracy improvement without adding any overhead to the query side. All the results demonstrate the generalizability of our method.

### 5.3. Comparison to the state-of-the-art methods

In Tab. 9, our method is compared with the state-of-the-art asymmetric methods (SOTA). Previous asymmetric retrieval methods, adopting a single feature on the gallery side, still suffer severe performance degradation especially in the large-scale case. Our method performs feature fusion at the gallery side, which significantly improves the accuracy of asymmetric retrieval, e.g., the mAP on R$\text{Oxf}$ + 1M increases from 64.42 to 70.47. Besides, compared to direct feature ensemble, our method introduces no extra computation and storage overhead to the query side, which is friendly for various resource-constrained platforms, e.g., the latency of feature extraction on the query side is comparable to the previous asymmetric retrieval method, e.g., 16.5 ms for ours vs. 16.1 ms for previous SOTA.

### 6. Conclusion

In this work, we introduce a new asymmetric feature fusion paradigm to enhance the accuracy of existing asymmetric systems. It alleviates the dilemma of trade-off between retrieval efficiency and accuracy, caused by the limited capacity of lightweight models. The proposed paradigm deploys various models at the gallery side to extract features, which are further aggregated into compact embedding with a dynamic mixer for efficient retrieval. As for the query side, only a single lightweight model is deployed for feature extraction. The query model and the mixer are jointly trained to achieve feature compatibility. With the proposed framework, the accuracy of asymmetric retrieval is significantly boosted without introducing any additional computational and storage overhead to the query side. Experiments on several datasets demonstrate the wide applicability and excellent effectiveness of our paradigm.

**Acknowledgements.** This work was supported in part by the National Natural Science Foundation of China under Contract 62102128 and 62021001, and in part by the Fundamental Research Funds for the Central Universities under contract WK3490000007. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution and the Supercomputing Center of the USTC.

---

**Table 9. Comparison to the state-of-the-art methods.** $^d$: feature dimension is $d$; $^\downarrow$: lower is better; $^\uparrow$: re-evaluate official public weights; $^\dag$: re-implementation; Ext.: latency of feature extraction for query side; Mem.: average memory footprint for R1M dataset; MobV2: MobileNetv2; R101: ResNet101; Mixer: aggregate various features into a compact vector with $\phi_{\text{mix}}$; Ensemble: feature concatenation.
References


[31] Yuxin Song, Ruolin Zhu, Min Yang, and Dongliang He. Gate or not to aggregate: Selective match kernels for image retrieval. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2754–2763, 2022. 2


