Sketch3T: Test-Time Training for Zero-Shot SBIR

Aneeshan Sain$^{1,2}$  Ayan Kumar Bhunia$^1$  Vaishnav Potlapalli*  Pinaki Nath Chowdhury$^{1,2}$  Tao Xiang$^{1,2}$  Yi-Zhe Song$^{1,2}$

$^1$SketchX, CVSSP, University of Surrey, United Kingdom.  $^2$iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.

{a.sain, a.bhunia, p.chowdhury, t.xiang, y.song}@surrey.ac.uk

Abstract

Zero-shot sketch-based image retrieval typically asks for a trained model to be applied as is to unseen categories. In this paper, we question to argue that this setup by definition is not compatible with the inherent abstract and subjective nature of sketches – the model might transfer well to new categories, but will not understand sketches existing in different test-time distribution as a result. We thus extend ZS-SBIR asking it to transfer to both categories and sketch distributions. Our key contribution is a test-time training paradigm that can adapt using just one sketch. Since there is no paired photo, we make use of a sketch raster-vector reconstruction module as a self-supervised auxiliary task. To maintain the fidelity of the trained cross-modal joint embedding during test-time update, we design a novel meta-learning based training paradigm to learn a separation between model updates incurred by this auxiliary task from those off the primary objective of discriminative learning. Extensive experiments show our model to outperform state-of-the-arts, thanks to the proposed test-time adaption that not only transfers to new categories but also accommodates to new sketching styles.

1. Introduction

Sketch-based image retrieval (SBIR) is by now a well-established topic in the vision community [14, 16]. Research efforts have mainly focused on addressing the sketch-photo domain gap, incurred by abstraction [33], drawing style [47] and stroke saliency [19]. Despite great strides made, the field remains plagued by the data scarcity problem – sketches are notoriously difficult to collect [4, 6].

Zero-shot SBIR (ZS-SBIR) in particular represents the main body of work behind this push for addressing data scarcity. It specifically examines the scarcity issue from a category transfer perspective, and strives for utilising sketch-photo pairs from seen categories to train a model that could be directly applied on those unseen (see Fig. 1(a)).

In this paper, we question this otherwise commonly accepted setup at definition level. We importantly argue that the very assumption of being able to apply a trained model as is to unseen categories, is by definition incompatible with the inherent subjective nature of sketch data. This largely results in a model that might well understand the semantic category shift, but not acute to changes in sketching style and abstraction level (both being prevalent problems in sketch [47]). Alleviating this problem is particularly crucial for the practical adaption of SBIR, as otherwise retrieval performance will incur a significant drop – a system that understands “my” sketches, might not understand “yours”.

This paper thus extends the conventional definition of ZS-SBIR to embrace this new problem, i.e., a new ZS-SBIR framework that (i) not only transfers knowledge on to unknown categories, (ii) but also adapts to the unique style of new sketches. We implement this by adopting a test-time-training framework that adapts to new categories and new styles at inference time. That is, instead of anticipating the distribution shifts via normal training, we intend to learn...
them at test time. The beauty of our solution lies in that we achieve a higher accuracy without any additional training sketch-photo pairs, but with just a single query sketch, no more than what is required in a typical ZS-SBIR setup (Fig. 1). It follows that this single sketch will first adjust the model to unseen style and category, and then use again as query to retrieve using the updated model, all test time.

Implementing this test-time-training framework despite intuitive is not trivial. There are two major challenges: Firstly, we have access to only the query sketches during inference, without any label or paired photo for supervision. Secondly, this test-time update should not degrade the joint embedding (that conducts retrieval) which has been learned using sketch-pairs. Solution to the first issue requires a task where labels can be obtained freely/synthetically during inference itself. Here we make clever use of the vectorised nature of sketches, and utilise a self-supervised task of sketch-raster to sketch-vector translation [5] to update the feature-extractor at inference. It follows that via this translation operation, the model adapts itself to the new style/category of the test sketch.

The second issue gets tackled at model design. In particular, we consolidate the said sketch self-reconstruction module as an auxiliary task within a meta-learning framework [26]. It follows that the model is meta-learned in a way, such that updates on the auxiliary task only happens in the inner loop, which then prevents it from distorting the joint embedding space whose updates occurs elsewhere in the outer loop via a triplet loss. This training strategy essentially ensures that the trained model now knows how to accommodate the auxiliary task loss without affecting the latent space too adversely, and accordingly defends itself against test-time updates from the sketch self-reconstruction auxiliary task.

More specifically, our framework shares a feature extractor amongst three diverging branches (Fig. 2 (left)): (i) a primary branch learns the cross-modal embedding over a triplet loss [66] using paired sketch-photo information, (ii) an auxiliary sketch branch that focuses on self-modal reconstruction to update and condition the shared feature-extractor towards better sketch-encoding, and (iii) an auxiliary photo branch, where we use photo-to-edgemap translation to condition the photo features. Having this photo branch also presents the option of updating the model on the unseen test-set photo-gallery to yield better photo features for retrieval, however is not compulsory. Note that only the auxiliary sketch branch get updated (and hence the shared feature extractor) at test-time upon a query sketch.

Our contributions are: (a) We offer a fresh extension on the ZS-SBIR paradigm, by proposing a novel test-time training framework that dynamically adapts a trained encoder to new sketches (b) To retrain transferable cross-modal embedding knowledge during inference, we propose a meta-learning framework that integrates primary discriminative learning with auxiliary tasks, such that updates from the latter are constrained towards benefiting the primary objective. (c) Extensive experiments and ablation confirm our method to be superior to existing state-of-the-arts.

2. Related Works

Sketch Based Image Retrieval (SBIR): SBIR involves finding an image corresponding to a given query-sketch. Aiming to retrieve photos of the same category, category-level SBIR [12, 49] began with using handcrafted descriptors [60] like SIFT [31], Gradient Field HOG [22], Histogram of Edge Local Orientations [44] or Learned Key Shapes [45], for constructing local [22] or global [42] joint photo-sketch representations. Shifting to deep-learning, methods [11, 28, 64] usually trained Siamese-like networks to fetch similar photos over a distance-metric in a cross-modal joint embedding space, over ranking losses [12]. Contemporary research include embedding sketch features to binary hash-codes [28, 68] for computational ease. Sketch as a query [10] however, prides in its ability to model fine-grained details. Research thus advanced to fine-grained SBIR [7, 38, 52] beginning with deformable-part models [25]. Aided by new datasets [54, 66], FG-SBIR flourished with the introduction of triplet-ranking models [66], learning a joint sketch-photo manifold. Attention mechanisms along with higher-order losses [54], hybrid generative-discriminative cross-domain image generation [39], textual tags [53] and mixed-modal jigsaw solving based pre-training strategy [40], enhanced it further. While Sain et al. [46] discovered cross-modal hierarchy in sketches, Bhunia et al. [8] employed reinforcement learning in an early retrieval scenario. Although further works have addressed low-resource data via semi-supervised learning [4], or style-diversity in sketches via meta-learning disentanglement [47], training during inference to bridge the train-test data distribution gap, remains unseen in SBIR.

Zero-Shot Learning: To deal with the data scarcity, a separate branch of literature has evolved within the SBIR pipeline that aims to generalise the knowledge learned from seen training classes to unseen testing categories. A zero-shot (ZS) SBIR pipeline was first introduced by Yelamarthi et al. [65], with an aim to minimise sketch-photo domain gap by approximating photo features from given sketches via image-to-image translation, thus aligning sketch-photo features jointly to generalise onto unseen classes. In contrast later works [14, 16] used semantic representation (word2vec) of class labels to learn a joint manifold capable of semantic transfer to unseen categories. While, [16] used adversarial training to align sketch, photo and semantic representation, [14] employed a gradient reversal layer to minimise sketch-photo domain gap. Other works include preserving training knowledge via knowledge distilla-
tion [29] to improve generalisability, and alleviating sketch-image heterogeneity via Kronecker fusion layer with graph convolution [50], thus enhancing semantic relations among data towards a generative hashing scheme for ZS-SBIR.

While earlier ZS-SBIR methods fixed model weights after training on seen classes, we advocate for one that adapts to novel classes during inference. Please note that this ‘adaptation protocol’ must not be confused with that of few-shot learning [51,63] which considers access to a few labelled samples. We however have no access to labelled data from unseen categories under ZS-SBIR setup. To adapt to unseen classes, we thus employ a self-supervised task for sketch and photo branch each, whose loss could be computed using labels that can be obtained freely/synthetically. Additionally, this self-supervised objective should imbibe knowledge of unseen classes via a few gradient update steps within a reasonable remit of edge device deployment.

**Self-supervised Auxiliary Tasks:** Constrained by the absence of labels during inference, our choice of task for test-time training should be a self-supervised one. Self-supervision involves designing pretext tasks that can learn semantic information without human annotations [23], such as image colorization [69], super-resolution [24], frame order recognition [32], solving jigsaw puzzles [36,40], image in-painting [41], relative patch location prediction [15], etc. Importantly, Asano et al. [2] shows self-supervised learning on a single image, can produce low level features that generalise well. However, they use complex tuple selection [32] or patch-sampling strategies [36,40] and relation-operations which leads to complex design issues in batch size, sampling strategies, or data-balancing that need tuning. We thus opt for simple self-modal reconstruction for test-time training. As an auxiliary task during training, it should improve robustness of the primary task [20], like rotation prediction [58] or via entropy minimisation [62]. Similar notion has been used in few-shot learning [55], domain generalization [9], and unsupervised domain adaptation [27,57]. Following suit, we use raster-to-vector decoding and image-to-edgemap translation as auxiliary tasks for sketch and photo branch respectively during training.

**Meta Learning:** This aims to extract transferable knowledge from a series of related tasks, to help adapt to unseen tasks with a few training samples [18,61]. Broadly speaking these algorithms fall in three groups. **Metric-based methods** [51,59] strive to create a metric space where learning is efficient with just a few samples. **Memory network based approaches** [37] attain knowledge across tasks, to generalise on the unseen task. **Optimization-based techniques** [18,35,56] optimises a model, such that it can adapt to any test data quickly. Specifically, we use the popular model-agnostic meta-learning (MAML) algorithm [18] (enhanced to MAML++ [1]), due its compatibility with any model trained via gradient descent, and diverse application range with several variants [1,35,43,48]. Besides using it to condition our model in a test-time scenario during training, we modify it to meta-train learnable stroke-specific weights for reconstruction, like learning rates in MetaSGD [26].

### 3. Background Study

**Baseline SBIR:** Sketch Based Image Retrieval aims at retrieving an image pertaining to a sketch query. For categorical SBIR [12], the image is retrieved from a gallery having images of different classes, and ideally belongs to the same category as that of the sketch. Formally, our model learns an embedding function, $f_{\theta}(\cdot) : \mathbb{R}^{H\times W \times 3} \rightarrow \mathbb{R}^d$, mapping a rasterised sketch or photo $I$ to a $d$-dimensional feature. Given a gallery of $G = \{C_i\}_{i=1}^M$ categories, having $N_i$ photos each, our core SBIR model obtains a list of photo ($p$) features $G = \{f_{\theta}(p^C)\}_{j=1}^{N_i} | \{M\}$. Thereafter, pairwise distances are calculated and corresponding images are retrieved over a precision metric [14]. A state-of-the-art CNN ($f_{\theta}(\cdot)$) extracts features of query sketch ($S$), matching photo ($P^+$) and an unmatched one ($P^-$) which are trained on a triplet loss objective [66], where minimising the loss signifies bringing the sketch-feature ($f_S$) closer to the positive photo-feature ($f_{P^+}$) while distancing it from the negative ($f_{P^-}$) one in the joint embedding space.

$$\mathcal{L}_{\text{Tri}}^9 = \max\{0, m + \delta(f_S, f_{P^+}) - \delta(f_S, f_{P^-})\}$$

where, $\delta(a,b) = \|a - b\|^2$, is a distance metric and $m$ is a margin hyperparameter, obtained empirically.

**Test-time Training:** During inference, given a query-sketch ($S^T$), the trained feature extractor ($\theta_e$) is updated based on a proxy-task to adapt to this specific test-sample. This task must be self-supervised to be free of label-cost. Features then extracted by the updated model ($\hat{\theta}_e$) are used to calculate pairwise distances for retrieval. Constrained by the unavailability of labels during inference, self-modal reconstruction is a common task-choice. More importantly, this task is used during training as well, as an auxiliary task to improve the model’s primary objective. Consequently, we have three sets of parameters: the shared feature encoder ($\theta_e$), the exclusive primary-task parameters ($\theta_p$) and auxiliary-task parameters ($\theta_a$). During test-time training, the common feature extractor is updated using the auxiliary task loss ($\mathcal{L}_{\text{aux}}$) to perform primary task on $S^T$ as,

$$\min_{\theta_e} \mathcal{L}_{\text{aux}}(S^T; \theta_e, \theta_a) , \ f_{S^T} = f_{\hat{\theta}_e, \theta_p}(S^T)$$

After operating on $S^T$, $\hat{\theta}_e$ is discarded as standard practice, and feature extractor is re-initialised with $\theta_e$ for a fresh adaptation on the next test sample.

### 4. Methodology

**Overview:** We aim to devise a SBIR framework that learns to alleviate test-train distribution gap by aligning a trained model to the test-data distribution, thus achieving
better retrieval accuracy. To this end, we design a SBIR model which is trained in a meta-learning framework, augmented via auxiliary training and enhanced (for the first time) via a test-time training paradigm. First, a feature extractor (Sec. 3(i)) encodes a query-sketch (S), its matched photo (P⁺), and an unmatched one (P⁻) to obtain features f_S, f_P⁺ and f_P⁻ all ∈ ℜ^d respectively using Fθ_e(·). Thereafter the model is trained in two branches (Fig. 2). While the primary branch (θ_p) instills cross-modal discriminative knowledge via triplet loss objective on those three features (Eq. 1), the auxiliary branch (θ_a) is trained on a self-modal reconstruction loss to improve primary task. Accordingly, we perform raster-to-vector decoding for sketch, and photo-to-edgemap translation for photo, to obtain reconstruction loss. Furthermore we associate learnable weights to every sketch-stroke which are meta-learned along with other modules in a meta-learning framework, to imbibe the knowledge of relative importance of strokes during reconstruction towards a better retrieval accuracy. For every test sample during inference, the feature-extractor is first initialised with trained parameters (θ_e). Following Sec. 3(ii), it is updated via reconstruction loss to adapt to the test distribution. Features extracted by the updated model are used for retrieval.

4.1. Model Architecture

Our pipeline starts with a feature-extractor (θ_e), which bifurcates into a primary branch (θ_p) focused at cross-modal discriminative learning, and an auxiliary branch (θ_a) for self-reconstruction task. The feature extractor (shared between two branches) first encodes a photo or sketch-image into a d-dimensional feature, Fθ_e(·) : ℜ^{H×W×C} → ℜ^d which is then used accordingly in either branch.

Primary branch : In addition to the backbone feature extractor, this branch lowers the feature dimension of extracted feature to d_p using a linear layer, Hθ_p(·) : ℜ^d → ℜ^{d_p} for better learning. Geared towards instilling a discriminative knowledge, the model is trained on a cross-modal triplet objective following Eq. 1, as:

\[ L_{tri}^{θ_e,θ_p} = \max \{0, m + \delta(f_S^{d_p}, f_P^+) - \delta(f_S^{d_p}, f_P^-)\} \]  

Auxiliary branch : Owing to the supervision-less test-time training paradigm, we needed to choose an auxiliary task which (a) is self-supervised, so that it can be performed free of label-cost during inference and (b) can complement the primary task in a way such that the extra features learned provide broader interpretation of input data [30]. We thus opt for a self-modal reconstruction task for either modality. In both cases, the latent feature is first reduced to a lower d_a dimension. For sketch, as vector coordinates are available, we perform sketch raster-to-vector decoding.

Sketch Vectorization: In vector-format one can use five-element vector \( v_t = (x_t, y_t, q_{t1}, q_{t2}, q_{t3}) \in ℜ^{5×5} \) to represent pen states for stroke-level modelling; \( T \) being the sequence length. Essentially, \((x_t, y_t)\) denotes the absolute coordinate value in a normalised \( H \times W \) canvas, while the last three represent binary one-hot vectors [19] of three respective pen-states: pen touching the paper, pen lifted, and end of drawing. Starting with a \( d_a \)-dimensional sketch feature \( f_S^{d_a} \), a linear-embedding layer obtains the initial hidden state \( h_t|_{t=0} \) of the decoder RNN (θ_S) as:

\[ h_0 = W_h f_S^{d_a} + b_h. \]  

It is then updated as:

\[ h_t = \text{RNN}(h_{t-1}; [f_S^{d_a}, \psi_{t-1}]), \]  

where \( \psi_{t-1} \) is the last predicted point and [·] signifies concatenation. A fully-connected layer then predicts five-element vectors at every time step: \( \psi_t = W_p h_t + b_p \), where \( \psi_t = (x_t, y_t, q_{t1}, q_{t2}, q_{t3}) \in ℜ^{2+3} \) - first two logits for coordinates, last three for pen-states. Using (\( \hat{x}_t, \hat{y}_t, \hat{q}_{t1}, \hat{q}_{t2}, \hat{q}_{t3} \)) as ground-truth at \( t^{th} \) step, mean-square error [47] \( L_{(t)}^{(MSE)} = ||\hat{x}_t - x_t||_2^2 + ||\hat{y}_t - y_t||_2^2 \) and categorical cross-entropy losses [4] \( L_{(t)}^{(CE)} = - \sum_{i=1}^{3} \hat{q}_{ti} \log \left( \frac{\exp(q_{ti})}{\sum_{j=1}^{3} \exp(q_{tj})} \right) \) are used to...
train the absolute coordinate and pen state prediction (softmax normalised) respectively, as:

\[
    \mathcal{L}_{\text{rec}}^S(\psi_t; \theta_e, \theta_a^p) = \frac{1}{T} \sum_{t=1}^{T} \left( \mathcal{L}_{\text{MSE}}^{(t)} + \mathcal{L}_{\text{CE}}^{(t)} \right)
\]

**Photo-to-Edgemap Translation**: Edgemap holding a lower domain gap with a sketch (both contain only structural information) than a photo, enables this task to align structural information than a photo, enabling this task to align gradients in favour of a better sketch representation, thus augmenting primary objective better than direct photo-to-photo translation. An edgemap corresponding to the match-

\[
    \mathcal{E} = \text{edge}(P^+) \in \mathbb{R}^{H \times W \times 3}
\]

\[
    \text{edge}() \text{ is a function that extracts an edgemap from a photo using 2D filters on the grey-scaled input image.}
\]

\[
    \mathcal{E} = \text{Dec}(f_{p+}).
\]

We thus have our reconstruction loss as:

\[
    \mathcal{L}_{\text{rec}}^P(\theta_e, \theta_a^p) = \| \mathcal{E} - \mathcal{E} \|_2
\]

For notational brevity, at times we use \( \theta_a = \{\theta_a^p, \theta_a^e \} \).

4.2. Meta-Learning Auxiliary Reconstruction

**Overview**: Decreasing the test-train data distribution gap especially for sketches which hold unconstrained diversity [47] is quite non-trivial a task. Aiming to alleviate it using test-time training alone would be an ambitious goal, if not insufficient. We thus take to the meta-learning training paradigm [21] where, the goal is to learn good initialization parameters representing an across-task shared knowledge among related tasks, such that it can quickly adapt to any novel task, with a few gradient update iterations. This simulates a test-time training paradigm in the training itself, which thus conditions the encoder to adapt better during inference. We modify a popular optimization-based meta-learning algorithm is model-agnostic meta-learning (MAML) [18] to suit our purpose.

**Task Sampling**: In a meta-learning framework [21], a model is trained from various related labelled tasks. To sample a task \( T_i \sim p(T) \) here, we first select a random category \( C_i \) out of \( M \) categories. Out of all sketch-photo pairs in \( C_i, N_i \) and \( r_i \) pairs are randomly chosen for meta-training (\( D_i^{\text{trn}} \)) and meta-validation(\( D_i^{\text{val}} \)) respectively. Training here consists of two nested loops. The inner loop update is performed over \( D_i^{\text{trn}} \) with an aim to minimise the loss in the outer loop over \( D_i^{\text{val}} \). Within every set, hard negatives are chosen from rest \( M-1 \) categories ensuring completely dissimilar instances.

**Meta-learning stroke-weights**: Furthermore, as sketch raster-to-vector decoding is a sequential problem, \( \mathcal{L}_{\text{rec}}^S \) involves a summation operation (Eq. 4) over the stroke sequence, thus treating every stroke-specific loss equally. Arguably, this task-specific adaptation for sequential reconstruction could be boosted if weight values for each stroke-specific loss are learned, such that the model adapts better with respect to those strokes holding higher semantic significance. Intuitively, our model thus learns an across-task knowledge where given a sketch, properties of certain strokes could be closer to the encoded knowledge of MAML’s initialisation parameter [3], to enhance easier retrieval once reconstructed. On the contrary, considerable anomalies could exist among certain strokes that are redundant or distracting during retrieval using average knowledge encapsulated inside MAML’s initialisation parameter. Consequently, during outer-loop adaptation, the model update should prioritise optimising with respect to those particular strokes whose semantic importance is more inclined towards the unknown regarding the model’s initialization. We thus intend to learn stroke-specific weights for stroke-wise reconstruction loss instead of averaging over all strokes.

**Meta-Optimisation**: Regarding factors influencing such weights, literature shows that gradients used for adaptation in inner loop hold knowledge [3] related to disagreement (i.e. this information needs further learning or assimilation during adaptation) against model’s initialization parameters. Computing gradients for all model-parameters, being quite cumbersome, we calculate gradient of \( t^{th} \) stroke-specific reconstruction loss with respect to final decoding step (parameter \( \phi \)) as \( \nabla_\phi \mathcal{L}_{\text{rec}}^S(\theta_e, \theta_a^p) \). It is then concatenated with gradients of triplet loss (Eq. 2) which deals with the full sketch representation with respect to \( \phi \) (both gradient matrices being flattened) as \( \mathcal{J}_t = \text{concat}(\nabla_\phi \mathcal{L}_{\text{rec}}^S(\theta_e, \theta_a^p), \nabla_\phi \mathcal{L}_{\text{trn}}(\theta_e, \theta_p)) \). We posit that gradient of the triplet objective and stroke-specific reconstruction losses guides towards determining how to weigh different stroke-specific losses. We thus pass this \( \mathcal{J}_t \) via a network \( g_T \) predicting a scalar weight value for \( t^{th} \)-stroke-specific loss as \( \eta_t = g_T(\mathcal{J}_t) \). Here, \( g_T \) is designed as a 3-layer MLP network having parameters \( \eta_t \) followed by a sigmoid to generate weights. Eq. 4 thus becomes:

\[
    \mathcal{L}_{\text{rec}}^S(\psi_t; \theta_e, \theta_a^p) = \frac{1}{T} \sum_{t=1}^{T} \eta_t \cdot \left( \mathcal{L}_{\text{MSE}}^{(t)} + \mathcal{L}_{\text{CE}}^{(t)} \right)
\]

Summing up, we have our inner loop loss and update as,

\[
    \mathcal{L}_{\text{trn}}(\theta_e, \theta_p, \theta_a) = \lambda_{\text{Tri}} \mathcal{L}_{\text{Tri}} + \lambda_{\text{rec}} \mathcal{L}_{\text{rec}}^{S} + \mathcal{L}_{\text{rec}}^{P},
\]

\[
    (\theta_e', \theta_p') \leftarrow (\theta_e, \theta_p) - \alpha \nabla_\Theta \mathcal{L}_{\text{trn}}(\Theta; \mathcal{D}_{\text{trn}})
\]

where, \( \Theta = \{ \theta_e, \theta_p, \theta_a \}, \alpha \) is the learnable inner loop learning rate and \( \lambda_{\text{Tri}}, \lambda_{\text{rec}} \) are hyper-parameters determined empirically. With updated model parameters, the primary objective is computed as the loss over validation set (\( D_{\text{val}} \)) as \( \mathcal{L}_{\text{val}} = \mathcal{L}_{\text{trn}}(\theta_e', \theta_p', \mathcal{D}_{\text{val}}) \) which updates all model parameters. As \( \mathcal{L}_{\text{trn}}(\theta_e', \theta_p') \) depends on \( \theta_e', \theta_p' \) and \( \theta_a \) via inner-loop update (Eq. 7), a higher order gradient needs to be calculated for outer loop optimisation with learning rate \( \beta \) as:
\[(\Theta, \eta, \alpha) \leftarrow (\Theta, \eta, \alpha) - \beta \nabla_{\theta_e, \theta_p, \eta, \alpha} \sum_{T_i} \mathcal{L}_{val}(\theta_e^{(i)}, \theta_p^{(i)})\]

(8)

The model updates by averaging gradients over a meta-batchsize of \(B\) sampled tasks.

4.3. Test-time Training for SBIR

Once trained, it is now important to align the trained model parameters to the test-data distribution before using them to encode test-sketches for retrieval. First, before test-time training starts, the model is adapted to the test-set photo-distribution using only the photo-to-edgemap auxiliary branch over a few \((\tau_p)\) gradient steps to update the feature extractor to \(\theta_P^T\). The trained feature extractor \((\theta_e)\) encodes test-photo \(P^T\) to \(f_{P^T} = \mathcal{F}_{\theta_P^T}(S^T)\) and uses it to update itself via auxiliary reconstruction task-loss \(\mathcal{L}_{rec}^P(P^T; \theta_e, \theta_P^T)\) (Eq. 5).

\[\theta_e^T \leftarrow \theta_e - \alpha^T \nabla_{\theta_e} \mathcal{L}_{rec}^P(D_{val}^{P^T})\]

(9)

This aligns the model parameters to the test-set photo distribution for retrieval. Please note that this step is optional and that one can directly use \(\theta_e\) instead, before starting test-time training. Now the photo-updated trained feature extractor \((\theta_e^P)^T\) encodes a test-set query-sketch \((S^T)\), to \(f_{S^T} = \mathcal{F}_{\theta_e^P}(S^T)\). The auxiliary sketch-vectoriser obtains reconstruction loss \(\mathcal{L}_{rec}^S(\psi_{\theta_e^P}^T; \theta_P^T, \theta_e^T)\) (Eq. 4), where \(\psi_{\theta_e^P}^T\) is the vector-representation of \(S^T\). \(\mathcal{L}_{rec}^S\) updates the feature extractor over \(\tau_s\) steps, using which corresponding test-sketch feature \((f_S^T)\) is extracted for retrieval,

\[\theta_e^T \leftarrow \theta_e^P - \alpha^T \nabla_{\theta_e} \mathcal{L}_{rec}^S(\theta_e^P, \theta_e, D_{val}^S)
\]

\[f_S^T = \mathcal{F}_{\theta_e^P}(S^T)\]

(10)

where, \(\alpha^T\) is the learning rate. Once evaluated, the feature-extractor is re-initialised with the photo-adapted model parameters \((\theta_e^P)\), or directly \(\theta_e\), if choosing to skip photo-adaptation, for the next test-sample.

5. Experiments

Datasets: For category-level SBIR, we use: (i) Sketchy [49] (extended) – contains 75k sketches across 125 categories with about 73k images [28] in total. Following [65] we split it as 21 testing classes disjoint from rest 104 training classes which are separated as 73 : 31 for meta-train : meta-test to avoid photo overlap between Sketchy [49] and ImageNet [13] datasets. (ii) TU-Berlin Extension [17] – contains 250 object categories with 80 free-hand sketches per category. Photo part is extended using 204,489 natural images of the same categories from [67]. Following [14] we keep 30 random classes for testing, while 220 training classes are split randomly as 150 for meta-train and 70 for meta-test. Category-level SBIR is evaluated similar to [28] using mean average precision (mAP@all) and precision considering top 200 (P@200) retrievals.

Implementation Details: A VGG-16 network pre-trained on ImageNet is used as the shared feature extractor with final output dimension \(d = 512\). The primary branch linear layer projects it to \(d_p = 64\) for triplet objectives. For auxiliary branch, the photo branch reduces to \(d_p = 128\) before feeding to a decoder consisting of a series of stride-2 convolutions, with BatchNormRelu activation on every convolutional layer except the output that has tanh for activation. For sketch-decoding a GRU decoder of hidden state size 128 is used. Furthermore, we use Adam optimiser in both inner and outer loops with learning rates \(\alpha = 0.0005\) (initial) and \(\beta = 0.0001\) respectively during meta-learning with single-step gradient update. During test-time adaptation learning-rate is empirically set at 0.0001 for both photo and sketch, with \(\tau_s = \tau_p = 4\) gradient steps. Hyper-parameters \(\lambda_{tri}, \lambda_{rec}\) are empirically set to 0.7 and 0.3 respectively. We use a meta-batch size of 32 and set margin \(m\) to 0.3.

5.1. Competitors

We design several baselines aligned to our motivation from different perspectives to evaluate our framework. (i) State-of-the-art ZS SBIR methods (SOTA): ZS-Cross [65] aligns cross-modal sketch-photo features jointly to generalise onto unseen classes, approximating photo features from given sketches via image-to-image translation. While ZS-CGAN [16] uses semantic representation (word2vec) of class labels to learn a joint manifold capable of semantic transfer to unseen categories in an adversarial paradigm, ZS-GRL [14] combines similar semantic information of class labels with visual sketch information and trains over a gradient-reversal layer to reduce sketch-photo domain gap. ZS-SAKE [29] employs knowledge distilled paradigm using teacher signal from an ImageNet pre-trained CNN model and constrained by semantic information from category-labels to retrieve in a Zero-shot setting. (ii) Test-time training baselines (TTT): Following [58] we design a baseline following our pipeline, TTT-Rotation with a triplet-loss primary objective and the auxiliary task of rotation angle classification on both sketch-image and photos, without meta-learning. Similarly TTT-Affine follows [34] in using affine transformations on input images as auxiliary task for Tes-time-adaptation. (iii) Meta-Learning Baselines (Meta): Meta-SN-ZS simply employs vanilla MAML [18] on top of a simple Siamese-network following [66], trained via triplet loss in both inner and outer loops, in a zero-shot retrieval framework. It adapts using inner loop updates across retrieval tasks over categories in SBIR and over instances in FG-SBIR frameworks. Meta-Aux-ZS is identical to Meta-SN-ZS except that it adapts using both the auxiliary task of self-modal image reconstruction (for both sketch and photo branch) and triplet objective to minimise only triplet loss in the outer-loop. No test-time training is involved in either one.
5.2. Result Analysis and Discussion

Table 1 shows that methods employing Test-time training mostly surpass Zero-shot SBIR methods. Among them, our method consistently outperforms the other state-of-the arts in retrieval accuracy. ZS-Cross [65] with its simplistic cross-modal training paradigm is quickly surpassed over by ZS-CCGAN [16] (by 0.116 mAP@all on Sketchy), as the latter is aided with a cycle consistency loss in an adversarial training paradigm in addition to the guidance from word2vec embeddings of categories -- providing much better generalisability for the unseen classes. Although superior, it fails to outperform ZS-GRL [14] due to the latter’s usage of the gradient-reversal layer that specifically aims to create a domain-agnostic embedding in addition to semantic class labels towards improving accuracy. However in all these methods catastrophic forgetting is a major issue which unavoidably impacts their performance. ZS-SAKE [29] specifically focuses on knowledge preservation to reduce this effect, with the help of a knowledge-distillation paradigm, that aims to preserves the knowledge from pre-trained ImageNet [13] weights while training on the new dataset. The superior result (0.178 mAP@all more than ZS-GRL) demonstrates that original domain knowledge preserved by ZS-SAKE is not only maintaining its ability to be adapted back to the original domain but also helping the model to be more generalizable to the unseen target domain.

Coming to the test-time adaptation paradigm, we report the result of two state-of-the-art paradigms naively implemented towards our retrieval objective on the two datasets. TTT-Rotation [58] performs rotation-angle classification as an auxiliary task with the primary task being cross-modal triplet loss objective to adjust to test-data distribution during inference. The problem of catastrophic forgetting is alleviated to an extent due to the shift in focus from learning a domain-invariant mapping to evolving the latent space to adjust the test-distribution. Naturally we see a relative rise in accuracy of 0.94 mAP@all against ZS-GRL. TTT-Affine [34] having learnable affine transform, enables itself to align the trained parameters towards the test-distribution to a greater extent than TTT-Rotation, thus faring slightly better (0.004) than that in accuracy. Introducing meta-learning in a zero-shot paradigm on top of basic Siamese network trained on Triplet loss (Meta-SN-ZS) improves existing results over the cross-modal ZS experiment (ZS-Cross) by 0.172 mAP@all on Sketchy [49]. This is because meta-learning conditions the model to retain and use knowledge acquired across a set of relative tasks to adapt and generalise onto new tasks in a simulated testing scenario. Attaching an auxiliary-task branch to the primary and training it in the inner loop with the primary objective, further improves result (by 0.035) in the Zero-Shot setting proving its potential in this area. Our method combines the best of these worlds to use auxiliary reconstruction task, in a meta-learning training paradigm, aided with test-time adaptation for optimal accuracy. Additionally it meta-learns the stroke-specific weights for reconstruction, towards better enhancing the primary discriminative objective, thus outperforming the existing methods.

5.3. Ablation Study

We perform a detailed ablative study different architectural choice fro various perspectives in Table 2.

[i] Is meta-learning important: To judge its contribution we design an experiment training without the meta-learning paradigm in the ZS - setup. The model is trained using two losses (primary and auxiliary) and the auxiliary task updates the model during test-time training. Results (Type-II in Table 2) show a stark decrease (by 0.088 mAP@all) against the proposed method, showing how firmly it maintains the discriminative knowledge while training itself, that is otherwise distorted during test-time training. Furthermore using meta-learning avails the option of meta-learning stroke-weights which contributes further.

[ii] Significance of learnable $\eta$: To show the efficacy of the learnable stroke-specific weight for reconstruction loss, we remove $\eta$ simplifying the sketch-reconstruction loss (Eq. 6) to MSE and cross-entropy loss (Eq. 4). Doing so (Type-III) results in decrease from the proposed method signifying learning relative stroke-importance towards reconstruction is beneficial. Furthermore verifying the dependency of $g_\eta$ on gradients from primary objective, we train a model with $g_\eta$ initialised with a random tensor of fitting dimensions. Without the guidance of supporting the primary objective (discriminative learning), the weights are learned sub-optimally leading to a slight drop by 0.014 mAP@all.

[iii] Choice of auxiliary task: One of the most significant aspect of this test-time training paradigm is choosing the auxiliary task – not only should it be free of label-cost but it must be well suited to capture the test-time distribution over a few gradient updates so as to align model parameters to the test dataset. Without it the model performs quite poorly (Type-I). Exploring other alternatives we thus design a few experiments (Type V-VII) results of which are shown in Table 2. Type IV (Img2Img) employs image-to-
image translation, decoding the encoded feature via a stride-2 convolutional decoder with BatchNormRelu activation as the auxiliary task, for both branches, i.e sketch-image to sketch-image and photo-to-photo (not edgemap like ours). Type V performs image-to-image translation on the photo-branch, keeping sketch branch same as ours for the auxiliary task. While Type VI chooses rotation-angle classification as the auxiliary task on both sketch and image branch following [58], Type VII employs affine transformation on the photo and sketch-image in either branch, following the auxiliary-task approach of [34], but keeps the rest of the training paradigm like meta-learning identical to ours. In context of SBIR, we observe that sketch raster-to-vector translation holds significance as Type V performs better than Type IV. Furthermore, our method’s superiority over Type V confirms photo-to-edgemap translation to be a better suitable auxiliary task in context of sketches. While types VI and VII both morph the photo and sketch as images, apparently the classification objective alone isn’t sufficiently strong as reconstruction to align model parameters adequately to the test distribution. We also compare efficiency of sketch in terms of vector format – absolute coordinates (ours) vs. offset-coordinate (Type X) [19]. Turns out the former is better for decoding. Comparing sketch decoders between GRU (ours), LSTM (Type VIII) and Transformer (TYPE IX), showed GRU as optimum empirically.

**[iv] Further insights:** Qualitative results on Sketchy [49] are shown in Fig. 3. Fig. 4 shows that, during training one single adaptation step is found to be optimal with the highest performance gain. Diminishing results on higher updates contradicting [18], might be due to detrimental concentration of inner loop on irrelevant sketch details, thus forgetting learned generic prior knowledge. During inference however model parameters find four gradient update steps to be optimal for aligning to the test distribution. More steps induce confusion, leading to a drop in accuracy. Furthermore, an ablative study (Fig. 5) showed optimal feature dimension for primary and auxiliary objectives to be 64 and 128 respectively, almost retaining performance with higher ones. Also, evaluating our model without the optional one-time update (§4.3) on test-set gallery photos, we obtain a slight drop in results to 0.560 mAP@all in Sketchy. Compared to 8.8 ms of ZS-Cross, ours takes 19 ms more per query, due to the additional test-time training involved.

### Table 2. Ablative studies (accuracy on Sketchy).

<table>
<thead>
<tr>
<th>Type</th>
<th>Primary</th>
<th>Auxiliary</th>
<th>Meta</th>
<th>TTT</th>
<th>mAP@all</th>
<th>P@200</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.368</td>
<td>0.452</td>
</tr>
<tr>
<td>II</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.487</td>
<td>0.576</td>
</tr>
<tr>
<td>III</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.561</td>
<td>0.610</td>
</tr>
<tr>
<td>IV</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.528</td>
<td>0.601</td>
</tr>
<tr>
<td>V</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.546</td>
<td>0.605</td>
</tr>
<tr>
<td>VI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.511</td>
<td>0.596</td>
</tr>
<tr>
<td>VII</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.524</td>
<td>0.597</td>
</tr>
<tr>
<td>VIII</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.568</td>
<td>0.619</td>
</tr>
<tr>
<td>IX</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.562</td>
<td>0.615</td>
</tr>
<tr>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.570</td>
<td>0.622</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.575</td>
<td>0.624</td>
</tr>
</tbody>
</table>

**6. Conclusion**

In this paper we extended the definition of ZS-SBIR, asking it to extend not just to novel categories, but also to new style of query sketches. We achieve this by proposing a test-time training paradigm that adapts the trained model using just one sketch. Firstly, we show that sketch raster-to-vector translation on query-sketch alone is reliable to bridge the train-test gap as an auxiliary task. Secondly, we propose a novel meta-learning paradigm to ensure test-time updates from this auxiliary task would not be adversely affecting the joint embedding that is used to conduct retrieval. Extensive experiments with ablative studies show our method to surpass other state-of-the-arts.
References

[22] Rui Hu and John Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. CVIU, 2013. 2
[33] Umar Riaz Muhammad, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Goal-driven sequential data abstraction. In ICCV, 2019. 1
[34] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambach, Evgeny Levinkov, Thomas Brox, and Jan Hendrik Metzen. Test-time adaptation to distribution shift by confidence maximization and input transformation. arXiv preprint arXiv:2106.14999, 2021. 6, 7, 8
[56] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In ICCV, 2017. 2
[66] Lan Yang, Aneeshan Sain, Linpeng Li, Yonggang Qi, Honggang Zhang, and Yi-Zhe Song. S²net: Graph representational network for sketch recognition. In ICME, 2020. 2
[68] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In CVPR, 2016. 2, 3, 6