






Abstracting Sketches through Simple Primitives

-

Supplementary Material

Stephan Alaniz^{1,2}, Massimiliano Mancini¹, Anjan Dutta³,
Diego Marcos⁴, and Zeynep Akata^{1,2,5}

¹ University of Tübingen, ² MPI for Informatics, ³ University of Surrey,
⁴ Wageningen University, ⁵ MPI for Intelligent Systems

1 Network Architecture Details

For all our networks that take sketches as input, we use the Transformer [7] architecture with GeLU activations [2], 8 self-attention heads and a layer size of 128 for the self-attention layers and 512 for the fully-connected layers. Our network f has 6 Transformer layers and embeds strokes by adding a single embedding token to the input sequence of points. We find that adding positional embeddings to encode the order of points does not improve the performance, so we only use the coordinates as input for each point.

We train a sketch classifier on Quickdraw with the same architecture as our stroke encoder, i.e. we use the 6-layer Transformer architecture. The first three layers embed strokes and the last three layers embed the full sketch by taking the sequence of stroke embeddings as input. First, we pass the sequence of points of individual strokes (or primitives in PMN) through the first three Transformer layers without positional embedding. To integrate stroke relative positions, the global position and scale are mapped linearly and added to the stroke embeddings. We then pass this stroke embedding sequence to the last three Transformer layers. A linear layer maps the final sketch embedding to the class logits before taking the cross entropy loss on the ground truth class label. When partial sketches are evaluated, we mask out unused strokes or sub-strokes at the input to obtain the predicted class logits.

For FG-SBIR, we train a Siamese network [5] on the original training sketches with the same architecture of [4,1], based on the InceptionV3 [6] architecture with an embedding size of 128. Since the network acts on natural images and images of sketches, we render our primitive reconstructions to images when evaluating them on the task. Similarly, partial sketches are rendered and fed to the CNN to obtain the retrieval scores for different budget and when applying DSA and GDSA.

2 Compatibility function

Our proposed PMN model uses a compatibility function ϕ to choose which primitive to match to a human stroke. In theory, the model can also be trained without

| Batch size | with ϕ | without ϕ | factor |
|------------|-------------|----------------|--------|
| 32 | 13.14 ms | 118.82 ms | 9.04× |
| 64 | 14.28 ms | 226.46 ms | 15.85× |
| 128 | 20.16 ms | 449.22 ms | 22.28× |
| 256 | 34.79 ms | 905.66 ms | 26.03× |
| 512 | 65.23 ms | out-of-memory | - |

Table 1. Average time in milliseconds for a forward pass of PMN using the compatibility function (with ϕ) or not using it (without ϕ) on a V100 GPU.

ϕ , where the loss function is applied to each transformed primitive independently, regardless of how well a primitive fits a human stroke. At inference time, without ϕ the distance transform needs to be calculated for each transformed primitive to determine which one best fits the human stroke.

Using ϕ brings two main advantages. The first is reducing the inference time, without requiring the computation of the distance transform, and the second is providing a better definition of the training loss. Firstly, at inference time, we do not require the distance transform to be computed which is the most computationally expensive operation in our pipeline. To quantify this speed-up obtained by using ϕ , we measured the time a forward pass takes on a V100 GPU (in milliseconds) with and without ϕ in Table 1. We see a speed-up of an order of magnitude at small batch sizes of 32 to up to 26 times faster inference time at a batch size of 256. Using a batch size of 512 is not possible without ϕ as the 32GB of memory of the V100 is not sufficient to calculate all required distance transforms. On the other hand, we can use a batch size of up to 16384 when using ϕ (tested at powers of two).

Secondly, without ϕ , the loss cannot reach zero, due to the distance transform between target strokes and their most different primitives (e.g. a circle-like human stroke vs the "line" primitive). With ϕ instead, the loss can become close to zero since only the most compatible primitives will be used to compute the loss. While we found no clear difference between the two strategies in terms of overall performance on downstream tasks, with ϕ the training loss becomes more expressive when comparing varying configurations and it avoids eventual loss spikes caused by matching primitives to strokes of very different shape.

3 Affine Transformation

The affine transformation applied on primitives to reconstruct human strokes differs when computing the loss and when recreating a whole sketch. During training, human strokes and primitives are normalized to the range $[-1, 1]$ while retaining their aspect ratio by subtracting the mean of its points μ and then dividing by the size of the longest side w . The function $h(z_p^h, z_s^h)$ predicts the transformation T_s^p to align p with s on this normalized scale. When reconstructing full sketches, primitives are first transformed by T_s^p followed by

| Transformation | Budget | | |
|-----------------------|--------------|--------------|--------------|
| | 10% | 20% | 30% |
| rotate | 44.05 | 64.35 | 73.12 |
| scale | 57.39 | 69.59 | 73.92 |
| shear | 57.38 | 74.52 | 80.93 |
| scale, rotate | 64.75 | 80.08 | 85.46 |
| shear, rotate | 64.69 | 81.74 | 87.86 |
| shear, scale | 65.99 | 82.12 | 87.69 |
| shear, scale, rotate | 67.11 | 83.73 | 88.86 |
| rotate, scale, rotate | 67.08 | 83.69 | 89.15 |

Table 2. Classification accuracy on Quickdraw at budgets of 10%, 20% and 30% for different types of transformations learned by h .

denormalizing based on the mean and size of the human stroke to obtain the transformed primitive $pT_s^p * w_s + \mu_s$. In practice, we combine the scaling factor w_s and the translations μ_s into the transformations matrix T_s^p before applying it to p as it also assures that we always use at most six floating point values (maximum number of parameters of a 2D transformation matrix) for our fixed budget communication messages.

The affine transformation predicted by h can be defined in several different ways. We do not allow arbitrary affine transformations in order to retain similarity with the original shape. For instance, if the scaling factors are not controlled, any shape can be collapsed into a line by applying a small factor on one of the axis. Therefore, we restrict the scaling transformation to be a proportional scale where one axis is scaled by a value between 0.05 and 1 while the other is fixed (at 1). Since scaling alone does not provide enough flexibility to fit primitives to strokes, we experiment with combining scaling with rotation and shear transformations. As reported in Table 2, the composite transformation of rotate-scale-rotate works best and is chosen for all of our experiments. Notably, these transformations are applied in order, but except for rotate-scale-rotate, changing the order of the transformations, does not have a significant impact on the performance.

4 Additional Quickdraw Results

All budget levels. Figure 1 shows the performance of all evaluated methods at different budget levels between 0% and 100%. It illustrates the difference between selection-based and shape-based methods. While selection-based methods steadily increase in classification accuracy as the budget increases, shape-based methods have a more steep increase in the beginning and flatten off afterwards, making them favorable in low-budget regimes. As PMN performs lossy compression of the sketches, it requires at most a budget of 70% to abstract the whole sketch. The intersection of PMN with GDSA is at around 55% budget.

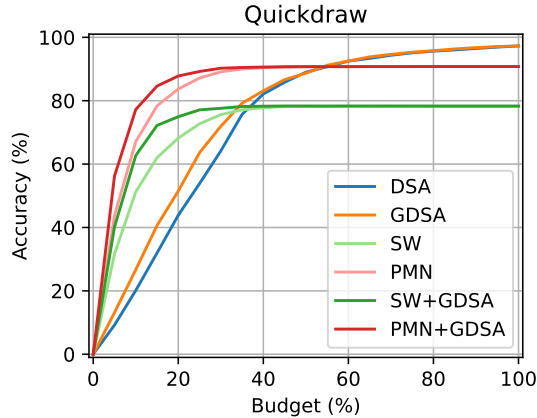


Fig. 1. Classification accuracy on Quickdraw at varying budgets between 0% and 100% evaluated with a classifier trained on the original human-drawn sketches.

Quickdraw-345. In the main paper, we follow [4] and we use Quickdraw with 9 classes. Here, we also train and evaluate our PMN model and the compared methods on the Quickdraw dataset with 345 classes. Table 3 shows the results. The trend is consistent with Quickdraw-9, with PMN+GDSA performing the best, and shape-based abstraction outperforming selection-based strategies at low budgets.

| Abstraction method | | Budget (%) | | | |
|--------------------|----------|--------------|--------------|--------------|--------------|
| Type | Name | 10 | 20 | 30 | 100 |
| Selection | DSA [4] | 1.18 | 2.78 | 7.22 | 70.12 |
| | GDSA [3] | 1.39 | 3.45 | 9.04 | |
| Shape | SW [8] | 5.56 | 13.51 | 18.95 | 23.17 |
| | PMN | 11.45 | 25.50 | 33.33 | 38.55 |
| Selection | SW+GDSA | 5.92 | 14.82 | 20.12 | 23.17 |
| +Shape | PMN+GDSA | 13.43 | 27.80 | 34.87 | 38.55 |

Table 3. Classification accuracy on the Quickdraw345 dataset at budgets of 10%, 20% and 30% evaluated with a classifier trained on the original human-drawn sketches.

5 Additional FG-SBIR Results

In the main paper we report the results of sketch-based image retrieval on top-10 accuracy, following previous works [4]. For completeness, Table 4 shows the results for top-1 retrieval accuracy. With this metric, we observe the same trend in all datasets, with PMN+GDSA achieving the best results at low budgets.

Additionally, apart from the *Selection*-based and *Shape*-based abstraction methods discussed in the main paper, *On-the-Fly Fine-Grained Sketch Based*

Image Retrieval (OTF) [1] proposes a *Finetuning*-based approach that can be employed specifically for the FG-SBIR. Such a method does not learn to abstract, but finetunes the embedding network with partial sketches, optimizing the FG-SBIR ranking to better retrieve their respective images.

Results for OTF are added to Table 4. Since shape-based abstraction is orthogonal to finetuning, we also evaluate the combination of SW/PMN with OTF for FG-SBIR. OTF generally performs well when there is a shift in the data distribution fed through the sketch embedding network. For instance, at 10% budget on ShoeV2 and ChairV2, OTF outperforms GDSA as finetuning the embedding works better than selecting more relevant parts of the sketch. This gap closes as we increase the budget, and at 30% GDSA already performs better than OTF on both datasets. Similarly, OTF boosts SW more than our PMN when combined, as the sketches reproduced by SW less accurately resemble the original sketches while the reconstructions of our model stay closer to the original data distribution.

| Type | Name | ShoeV2, Budget (%) | | | | ChairV2, Budget (%) | | | |
|----------------------|----------|--------------------|-------------|-------------|-------------|---------------------|-------------|-------------|-------------|
| | | 10 | 20 | 30 | 100 | 10 | 20 | 30 | 100 |
| Finetuning | OTF | 2.4 | 3.5 | 5.1 | 36.5 | 3.3 | 5.6 | 9.0 | 53.6 |
| Selection | DSA[4] | 1.4 | 2.4 | 4.1 | 36.5 | 2.5 | 6.2 | 10.2 | 53.6 |
| | GDSA [3] | 1.9 | 3.5 | 6.5 | | 2.8 | 7.4 | 12.1 | |
| Shape | SW | 3.3 | 6.2 | 8.0 | 9.1 | 9.0 | 14.2 | 16.7 | 17.9 |
| | PMN | 6.8 | 16.1 | 18.2 | 20.0 | 16.4 | 31.9 | 35.9 | 37.5 |
| Shape +Finetuning | SW+OTF | 4.8 | 8.4 | 10.7 | 11.9 | 9.9 | 17.0 | 18.9 | 20.3 |
| | PMN+OTF | 9.2 | 17.1 | 18.9 | 20.7 | 16.7 | 31.9 | 35.2 | 38.0 |
| Shape +Selection | SW+GDSA | 4.2 | 7.2 | 8.6 | 9.1 | 9.3 | 14.9 | 17.0 | 17.9 |
| | PMN+GDSA | 9.6 | 17.4 | 19.2 | 20.0 | 20.7 | 33.8 | 36.8 | 37.5 |

Table 4. Top-1 accuracy for FG-SBIR on ShoeV2 and ChairV2.

References

1. Bhunia, A.K., Yang, Y., Hospedales, T.M., Xiang, T., Song, Y.: Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In: CVPR (2020)
2. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
3. Muhammad, U.R., Yang, Y., Hospedales, T.M., Xiang, T., Song, Y.: Goal-driven sequential data abstraction. In: ICCV (2019)
4. Muhammad, U.R., Yang, Y., Song, Y., Xiang, T., Hospedales, T.M.: Learning deep sketch abstraction. In: CVPR (2018)
5. Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: ICCV (2017)

6. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. NeurIPS (2017)
8. Xiao, C., Wang, C., Zhang, L., Zhang, L.: Sketch-based image retrieval via shape words. In: ACM ICMR (2015)