Global Proxy-based Hard Mining for Visual Place Recognition

Amar Ali-bey, Brahim Chaib-draa, Philippe Giguère

Department of Computer Science and Software Engineering, Laval University, Canada

MOTIVATION

Learning deep representations for visual place recognition is commonly performed using pairwise or triple loss functions that highly depend on the hardness of the examples sampled at each training iteration. Existing techniques address this by using computationally and memory expensive offline hard mining, which consists of identifying, at each iteration, the hardest samples from the training set. In this paper we introduce a new technique that performs global hard mini-batch sampling based on proxies. To do so, we add a new end-to-end trainable branch to the network, which generates efficient place descriptors (one proxy for each place). These proxy representations are thus used to construct a global index that encompasses the similarities between all places in the dataset, allowing for highly informative mini-batch sampling at each training iteration.

METHODOLOGY

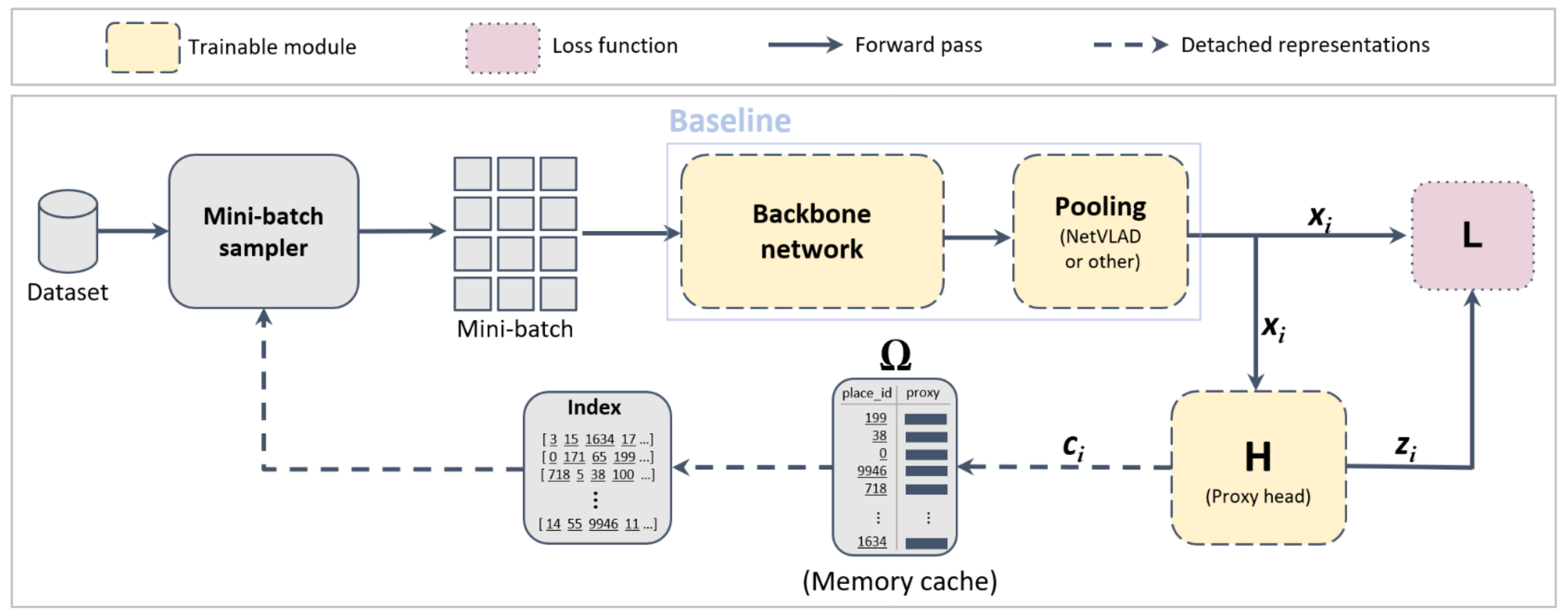


Figure 1: A diagram of our proposed method. We add a new end-to-end trainable branch to the network (proxy head $\mathcal{H}$) that projects highly dimensional vectors $\mathbf{x}_i$ into very compact representations $\mathbf{z}_i$; we use the latter to compute one proxy descriptor $\mathbf{c}_i$ for each place in the mini-batch. We detach each proxy from the computational graph and cache it into a memory bank Ω. Then, at the begining of each epoch, we construct an index upon Ω, in which places are gathered together according to the similarity of their proxies. This index is used to sample mini-batches containing similar places, which yields highly informative pairs or triplets. We call this strategy Global Proxy-based Hard Mining (GPM).

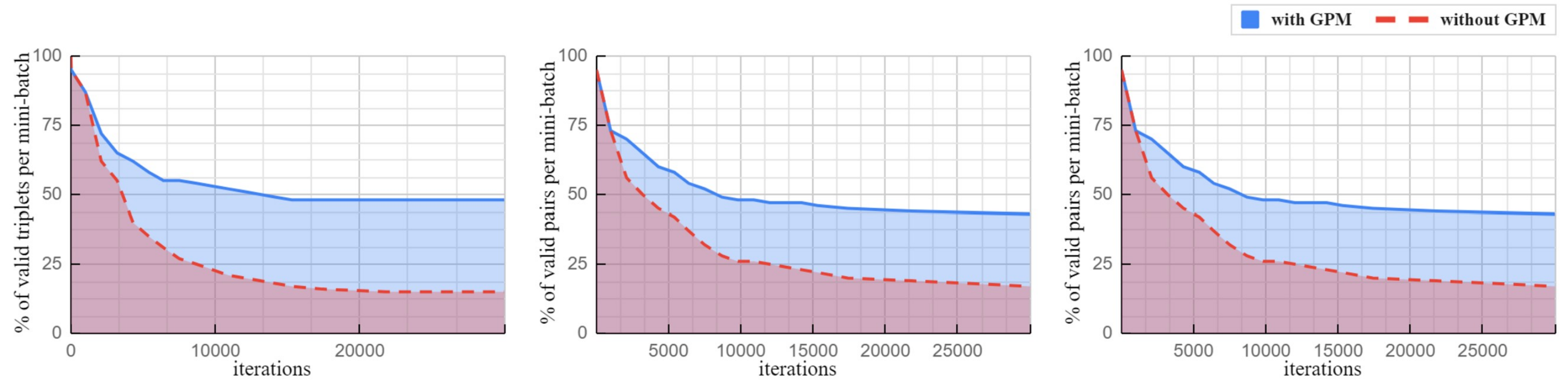
```
Algorithm 1: Index based mini-batch sampling
input  : Ω: the memory bank comprising proxies representing all places in the
         dataset
         M: the number of places per mini-batch.
output: L: a list of tuples, where each tuple contains M identities of places that
         need to be sampled in the same mini-batch.
$\mathcal{S} \leftarrow k$-NN$(k = M)$   ▷ Initialize a k-NN module $\mathcal{S}$ with k equal to M the number of
  places per mini-batch.
$\mathcal{S}$.add(Ω)                                ▷ Add the contents of Ω to $\mathcal{S}$ as references.
while $\mathcal{S} \neq \emptyset$ do
    Randomly pick a place $c_i$ from $\mathcal{S}$
    $\mathbf{T} \leftarrow \mathcal{S}$.search($c_i$)          ▷ Search $\mathcal{S}$ for the M-most similar places to $c_i$.
    $\mathcal{L} \leftarrow \mathcal{L} \cup \mathbf{T}$                              ▷ Append the M identities to $\mathcal{L}$.
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathbf{T}$                       ▷ Remove from $\mathcal{S}$ all places present in T.
```

Figure 2: At the end of each epoch, we use k-NN to build an index upon Ω, in which places are gathered together according to the similarity of their proxies (similar places need to appear in the same mini-batch)

QUANTITATIVE RESULTS

Loss function	Hard mining		Pitts250k-test			MSLS-val			SPED			Nordland		
	OHM	GPM	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Triplet			77.0	90.0	93.6	67.7	79.2	82.4	53.7	69.5	75.8	8.4	16.3	20.6
		✓	81.3	91.9	94.9	71.8	82.0	86.3	57.3	71.8	77.8	11.8	20.3	25.9
	✓		87.5	95.4	96.9	74.0	85.1	87.7	62.4	78.6	83.2	10.1	17.9	22.6
	✓	✓	**90.0**	**96.4**	**97.6**	**77.6**	**88.0**	**90.4**	**71.3**	**83.7**	**87.3**	**20.2**	**33.2**	**38.8**
Contrastive			83.0	93.0	95.2	72.7	82.8	85.8	53.7	67.2	74.8	8.0	13.8	17.3
		✓	88.8	95.2	96.8	79.0	85.8	88.5	67.7	79.2	83.4	20.8	33.9	41.5
	✓		84.5	94.0	95.9	74.6	84.7	87.8	63.4	76.9	82.5	14.6	25.2	31.2
	✓	✓	**90.4**	**96.4**	**97.6**	**79.3**	**88.5**	**90.7**	**73.5**	**85.5**	**88.9**	**31.4**	**46.4**	**53.5**
Multi-Similarity			84.0	93.3	95.5	72.7	82.7	86.5	50.7	65.1	71.5	9.4	17.9	21.7
		✓	89.4	96.0	97.3	81.2	89.1	90.9	64.6	76.4	80.6	18.0	30.1	36.0
	✓		89.5	96.3	97.6	77.4	87.2	90.1	74.6	86.8	89.9	29.1	43.3	50.2
	✓	✓	**91.5**	**97.2**	**98.1**	**82.0**	**90.4**	**91.4**	**79.4**	**90.6**	**93.2**	**38.5**	**53.9**	**60.7**

Table 1: Ablation. We study the performance gain of three loss functions. For each loss, we train 4 NetVLAD [1] networks. 2 of which are baselines (one with Online Hard Mining (OHM) and one without), and the other 2 are to compare the performance gain introduced by our method (GPM). Training is performed on GSV-Cities dataset [2].



(a) Triplet loss (b) Contrastive loss (c) Multi-Similarity loss

Figure 3: Percentage of valid triplets/pairs per mini-batch during the training. GPM constructs highly informative batches, which keeps the number of valid pairs/triplets higher during all the training phase.

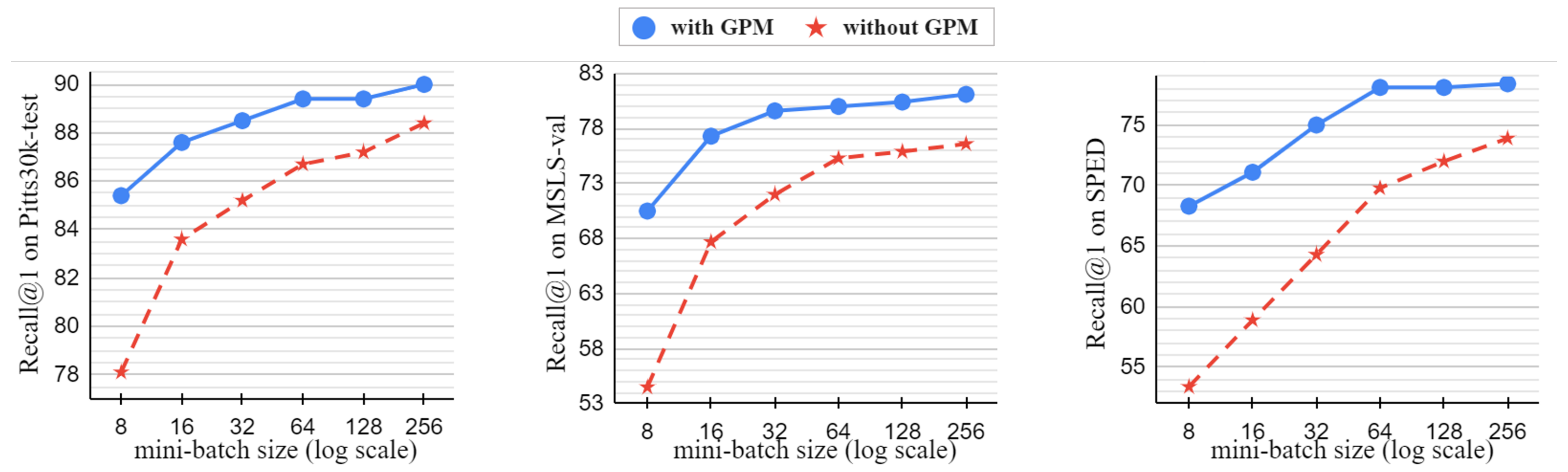
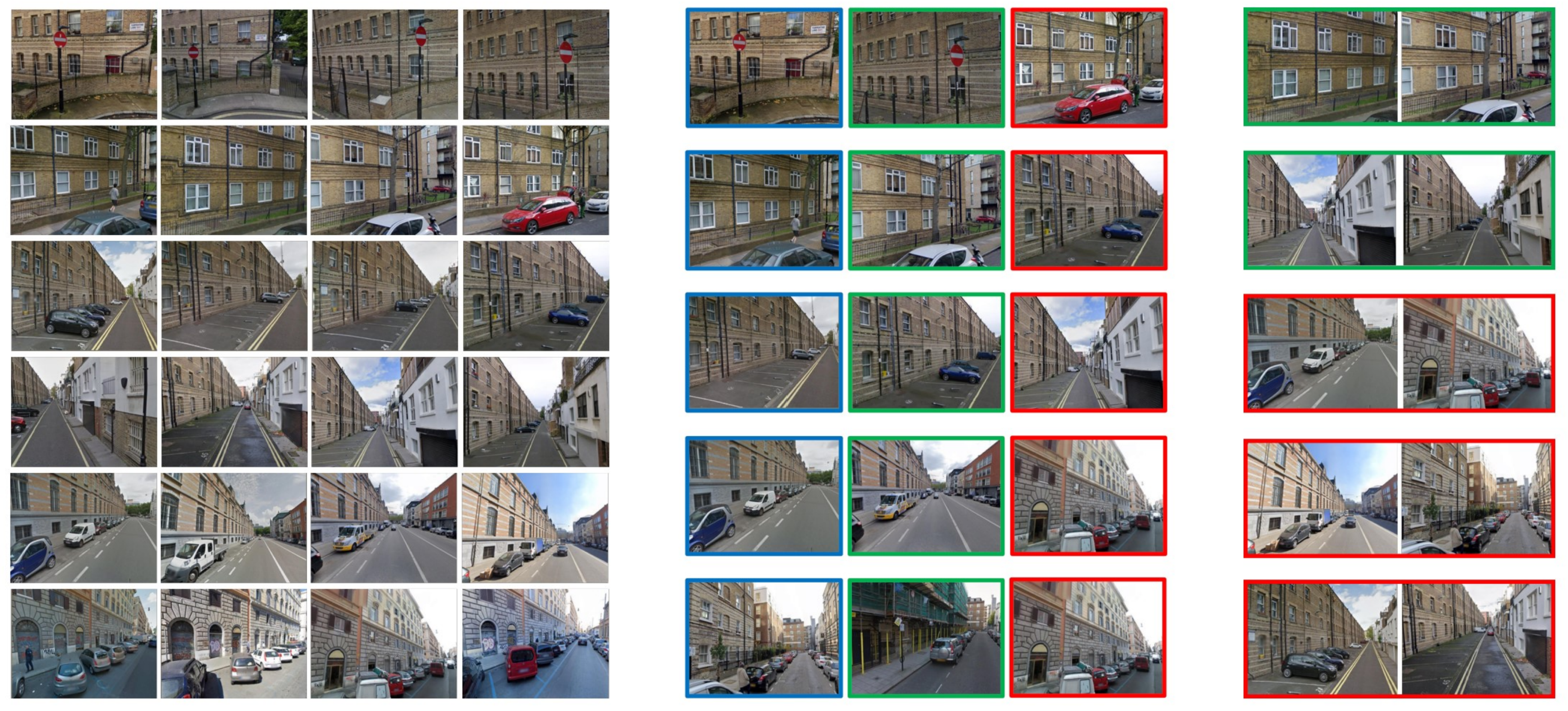


Figure 4: Impact of the mini-batch size. The horizontal axis shows M the number of places in the mini-batch. GPM is effective for a wide range of mini-batch sizes, with more impact when smaller mini-batches are used for training. This is of great importance when training hardware resources are limited.

	Baseline (no GPM)	Global Proxy-based Hard Mining (GPM)						Global hard mining without proxy
Dimensionality	0	32	64	128	256	512	1024	32768
Training time (hours)	1.93	1.93	1.93	1.93	1.94	2.05	2.1	4.83
GPU memory (GB)	10.4	+0.002	+0.002	+0.002	+0.03	+0.06	+0.14	+0.0
Cache size (GB)	0.0	+0.008	+0.016	+0.032	+0.064	+0.128	+0.256	+8.0
Recall@1 (%)	86.6	89.1	89	89.3	89.4	89	89.2	88.7

Table 2: Memory and computation cost of different dimensions of the proxy head compared against the baseline (without GPM). We also compare against global mining without a proxy head, where the memory bank is filled with the highly dimensional NetVLAD representations (rightmost column).

QUALITATIVE RESULTS



(a) A mini-batch sampled with GPM (b) Valid triplets (c) Valid pairs

Figure 5: (a) An example of a mini-batch containing 6 places sampled from a dataset of 65k places using GPM. Each place is depicted by 4 images (a row). This highlights the ability of our technique to construct mini-batches containing similar places, which in turn increases the presence of hard pairs and triplets. (b) A subset of hard triplets generated from the mini-batch, each row consists of a triplet with the blue as anchor, green as the positive and red as the hard negative. (c) A subset of positive (green) and negative (red) pairs. All triplets and pairs have been mined in an online fashion from the mini-batch sampled by GPM.

CONCLUSION FUTURE WORK

In this work, we proposed a novel technique that:

- Allows to sample highly informative mini-batches at each training iteration.
- Requires practically no additional computation time and negligible memory footprint.
- Significantly improves performance of existing techniques.

Future work can focus on:

- New architecture of the proxy head.
- Different ways of building the global index.

CONTACT INFORMATION

LinkedIn: https://linkedin.com/in/amaralibey
GitHub: https://github.com/amaralibey
Email: amar.ali-bey.1@ulaval.ca

REFERENCES

[1] Arandjelovic *et al.* NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.

[2] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022.