Batch-Incremental Triplet Sampling for Training Triplet Networks Using Bayesian Updating Theorem

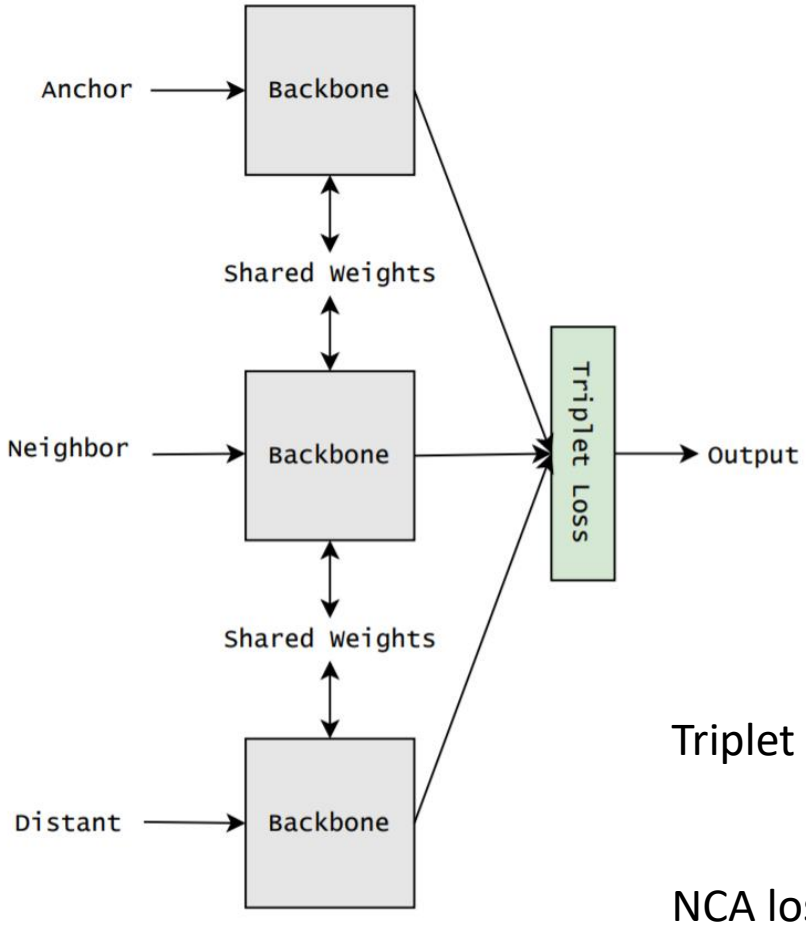
Milad Sikaroudi, Benyamin Ghojogh, Fakhri Karray, Mark Crowley, H.R. Tizhoosh

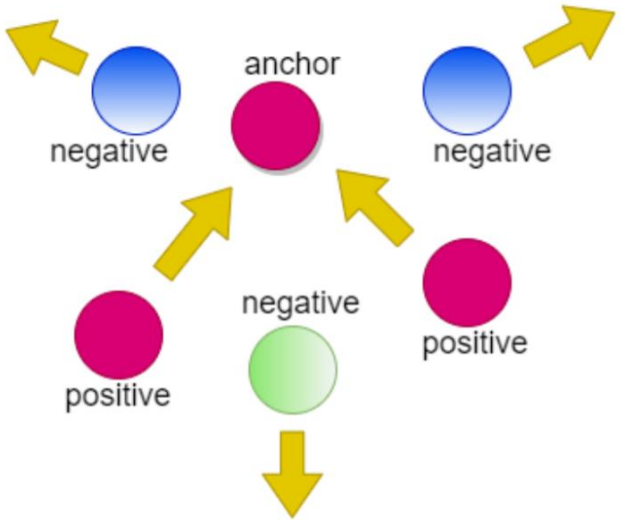
KIMIA Lab, University of Waterloo, ON, Canada
Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada

IEEE ICPR 2020 Conference

January 2021

Triplet Network and Loss Functions





Triplet loss:

$$\sum_{i=1}^{c}\sum_{j=1,\,j\neq i}^{c}\sum_{a=1}^{w}\sum_{p=1,\,p\neq a}^{w}\sum_{n=1}^{w}\left[m+\mathcal{D}(y_a^i,y_p^i)-\mathcal{D}(y_a^i,y_n^j)\right]_+$$

NCA loss:

$$-\sum_{i=1}^{c}\sum_{a=1}^{w}\ln\left(\frac{\exp(-\mathcal{D}(y_a^i,y_p^i))}{\sum_{j=1,\,j\neq i}^{c}\sum_{n=1}^{w}\exp(-\mathcal{D}(y_a^i,y_n^j))}\right)$$

Related Work

- Triplet Mining methods:
  - Batch all
  - Batch semi-hard
  - Batch hard
  - Easy positive
  - Negative sampling

Our Contribution

- Triplet sampling methods (e.g., negative sampling) sample from existing embedded points
- Our method samples stochastically from distribution of embedded data
- Our method updates the distribution of embedded data dynamically by batch-incremental triplet sampling

Bayesian Updating Theorem

$$\mathbb{P}(\theta|X) = \frac{\mathbb{P}(X|\theta)\,\mathbb{P}(\theta)}{\mathbb{P}(X)} \implies \mathbb{P}(\theta|X) \propto \mathbb{P}(X|\theta)\,\mathbb{P}(\theta)$$

$$\mathbb{P}(\theta) \mapsto \mathbb{P}(\theta|X)$$

posterior distribution $\mathbb{P}(\theta|X)$

prior distribution $\mathbb{P}(\theta)$

Multivariate Normal distribution: $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) := \frac{1}{\sqrt{(2\pi)^d\,|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$

$$\boldsymbol{\Sigma} \sim \mathcal{W}_d^{-1}(\boldsymbol{\Sigma}'^{-1}, n')$$

$$\boldsymbol{\mu}|\boldsymbol{\Sigma} \sim \mathcal{N}(\boldsymbol{\mu}', (1/n')\boldsymbol{\Sigma})$$

Normal Inverse Wishart distribution:

$$\mathbb{P}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathrm{NIW}(\boldsymbol{\mu}', \nu_1', \boldsymbol{\Sigma}', \nu_2') := \frac{|\boldsymbol{\Sigma}'|^{\nu_2'/2}|\boldsymbol{\Sigma}|^{-((\nu_2'+d)/2+1)}}{2^{(\nu_2' d)/2}\Gamma_d(\frac{\nu_2'}{2})(\frac{2\pi}{\nu_1'})^{d/2}} \times \exp\left(-\frac{1}{2}\mathbf{tr}(\boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-1}) - \frac{\nu_1'}{2}(\boldsymbol{\mu}-\boldsymbol{\mu}')^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}')\right)$$

Bayesian Updating Theorem

Normal Inverse Wishart distribution:

$$\mathbb{P}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathrm{NIW}(\boldsymbol{\mu}', \nu_1', \boldsymbol{\Sigma}', \nu_2') := \frac{|\boldsymbol{\Sigma}'|^{\nu_2'/2}|\boldsymbol{\Sigma}|^{-((\nu_2'+d)/2+1)}}{2^{(\nu_2' d)/2}\Gamma_d(\frac{\nu_2'}{2})(\frac{2\pi}{\nu_1'})^{d/2}} \times \exp\left(-\frac{1}{2}\mathbf{tr}(\boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-1}) - \frac{\nu_1'}{2}(\boldsymbol{\mu}-\boldsymbol{\mu}')^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}')\right)$$

$$\mathbb{P}(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \boldsymbol{x}^0, \boldsymbol{\mu}', \nu_1', \boldsymbol{\Sigma}', \nu_2') = \mathrm{NIW}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma} \,\middle|\, \boldsymbol{\eta}, \nu_1' + n_0, \boldsymbol{\Upsilon}, \nu_2' + n_0\right)$$

$$\mathbb{R}^d \ni \boldsymbol{\eta} := \frac{\nu_1'\boldsymbol{\mu}' + n_0\boldsymbol{\mu}^0}{\nu_1' + n_0}$$

$$\mathbb{R}^{d\times d} \ni \boldsymbol{\Upsilon} := \nu_2'\boldsymbol{\Sigma}' + n_0\boldsymbol{\Sigma}^0 + \frac{\nu_1' n_0}{\nu_1' + n_0}(\boldsymbol{\mu}^0 - \boldsymbol{\mu}')(\boldsymbol{\mu}^0 - \boldsymbol{\mu}')^\top$$

$$\mathbb{P}(\boldsymbol{\mu} \,|\, \boldsymbol{x}^0) = t_{\nu_2'+n_0-d+1}\left(\boldsymbol{\eta}, \frac{\boldsymbol{\Upsilon}}{(\nu_1' + n_0)(\nu_2' + n_0 - d + 1)}\right),$$

$$\mathbb{P}(\boldsymbol{\Sigma} \,|\, \boldsymbol{x}^0) = \mathcal{W}_d^{-1}(\boldsymbol{\Upsilon}^{-1}, \nu_2' + n_0),$$

Bayesian Updating In Our Method

$$\mathbb{P}(\boldsymbol{\mu} \mid \boldsymbol{x}^0) = t_{\nu_2'+n_0-d+1}\left(\boldsymbol{\eta}, \frac{\boldsymbol{\Upsilon}}{(\nu_1'+n_0)(\nu_2'+n_0-d+1)}\right),$$

$$\mathbb{P}(\boldsymbol{\Sigma} \mid \boldsymbol{x}^0) = \mathcal{W}_d^{-1}(\boldsymbol{\Upsilon}^{-1}, \nu_2'+n_0),$$

$$\boldsymbol{\mu}^{0,j} \leftarrow \mathbb{E}(\boldsymbol{\mu}^j \mid \boldsymbol{x}^{0,j}) = \boldsymbol{\eta}^j \overset{(12)}{=} \frac{n'\boldsymbol{\mu}'^j + n_0\boldsymbol{\mu}^{0,j}}{n'+n_0},$$

$$\boldsymbol{\Sigma}^{0,j} \leftarrow \mathbb{E}(\boldsymbol{\Sigma}^j \mid \boldsymbol{x}^{0,j}) \overset{(6)}{=} \frac{\boldsymbol{\Upsilon}^{-1}}{n'+n_0-d-1}, \forall\, n'+n_0 > d+1,$$

Algorithm (in every incoming batch)

$\{\boldsymbol{x}_i\}_{i=1}^{b} \leftarrow$ Feed $\{\boldsymbol{z}_i\}_{i=1}^{b}$ to the triplet network

for *class* j *from* 1 *to* c **do**

  if *it is first mini-batch* **then**

    $\boldsymbol{\mu}^{0,j} := (1/n') \sum_{i=1}^{n'} \boldsymbol{x}_i'^{j}$

    $\boldsymbol{\Sigma}^{0,j} :=$

    $(1/n') \sum_{i=1}^{n'} (\boldsymbol{x}_i'^{j} - \boldsymbol{\mu}^{0,j})(\boldsymbol{x}_i'^{j} - \boldsymbol{\mu}^{0,j})^\top$

  else

    $\boldsymbol{\mu}'^{j} := (1/n') \sum_{i=1}^{n'} \boldsymbol{x}_i'^{j}$

    $\boldsymbol{\mu}^{0,j} := (n'\boldsymbol{\mu}'^{j} + n_0 \boldsymbol{\mu}^{0,j})/(n' + n_0)$

    if $n' + n_0 > d + 1$ **then**

      $\boldsymbol{\Upsilon} := n'\boldsymbol{\Sigma}'^{j} + n_0 \boldsymbol{\Sigma}^{0,j} +$

      $\frac{n' n_0}{n' + n_0} (\boldsymbol{\mu}^{0,j} - \boldsymbol{\mu}'^{j})(\boldsymbol{\mu}^{0,j} - \boldsymbol{\mu}'^{j})^\top$

      $\boldsymbol{\Sigma}^{0,j} := \boldsymbol{\Upsilon}^{-1}/(n' + n_0 - d - 1)$

    else

      $\boldsymbol{\Sigma}^{0,j} := (1/n') \sum_{i=1}^{n'} (\boldsymbol{x}_i'^{j} -$

      $\boldsymbol{\mu}'^{j})(\boldsymbol{x}_i'^{j} - \boldsymbol{\mu}'^{j})^\top$

for *instance* i *from* 1 *to* b **do**

  anchor $\leftarrow \boldsymbol{x}_i$

  for *class* j *from* 1 *to* c **do**

    if $j = y_i$ **then**

      Sample $(c-1)$ positive instances

      $\sim \mathcal{N}(\boldsymbol{\mu}^{0,j}, \boldsymbol{\Sigma}^{0,j})$

    else

      Sample a negative instance

      $\sim \mathcal{N}(\boldsymbol{\mu}^{0,j}, \boldsymbol{\Sigma}^{0,j})$

Minimize the *triplet*/NCA loss with the $(b \times (c-1))$ triplets.

BUT & BUNCA

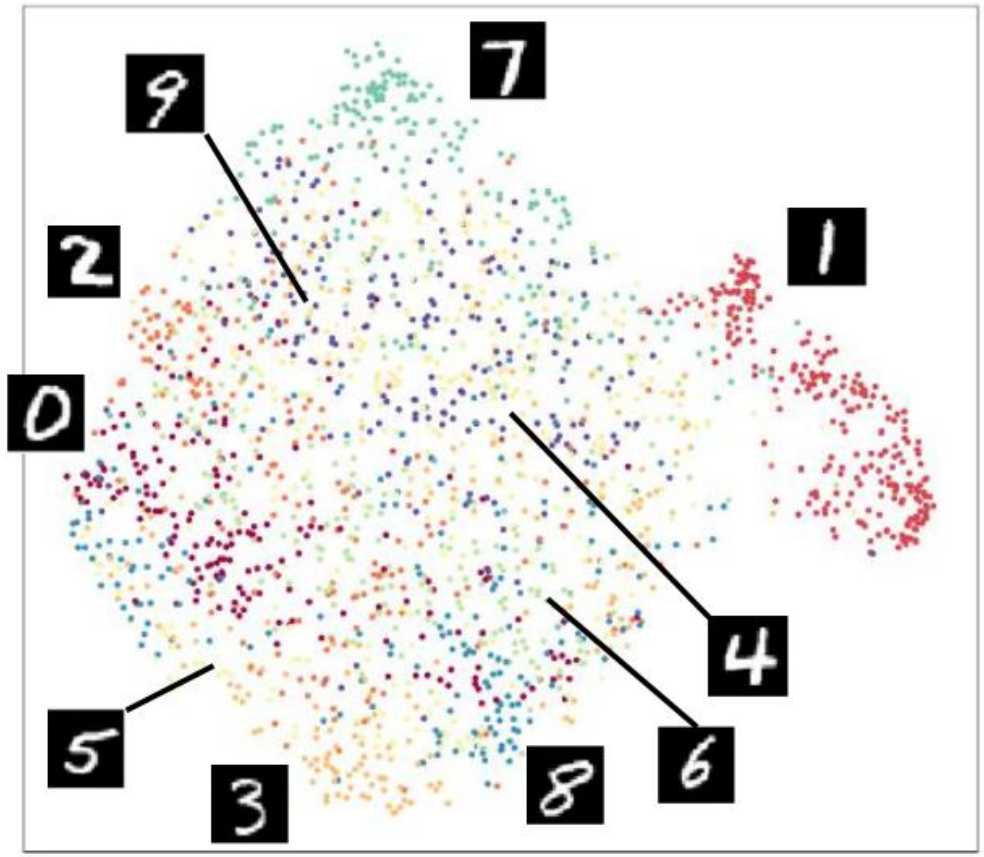
- In every batch: b anchors, (c-1) positives, (c-1) negatives
- Bayesian Updating with Triplet loss (BUT)

$$\text{minimize} \sum_{i=1}^{b} \sum_{k=1}^{c-1} \sum_{\ell=1}^{c-1} \left[m + \|\boldsymbol{x}_i - \boldsymbol{x}_k\|_2^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_\ell\|_2^2 \right]_+$$

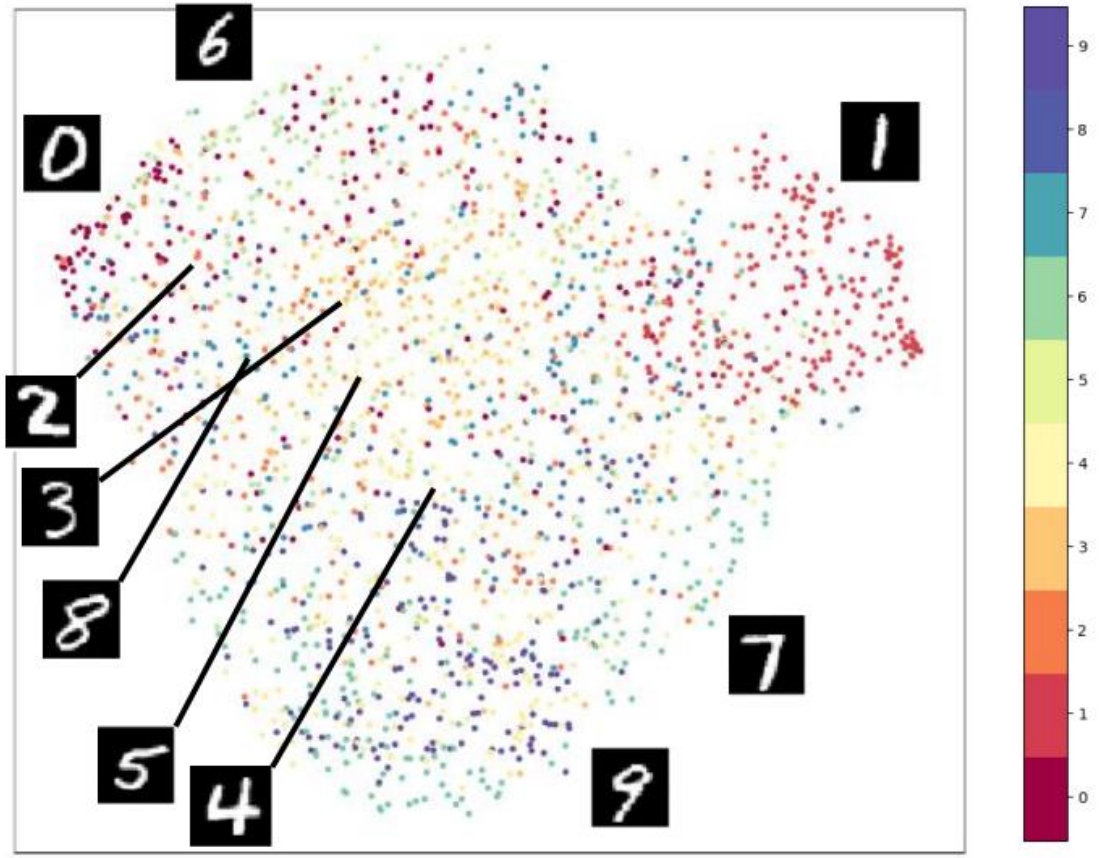
- Bayesian Updating with NCA loss (BUNCA)

$$\text{minimize} \ -\sum_{i=1}^{b} \sum_{k=1}^{c-1} \ln\left(\frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_k\|_2^2)}{\sum_{\ell=1}^{c-1} \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_\ell\|_2^2)} \right).$$

Experiments on MNIST

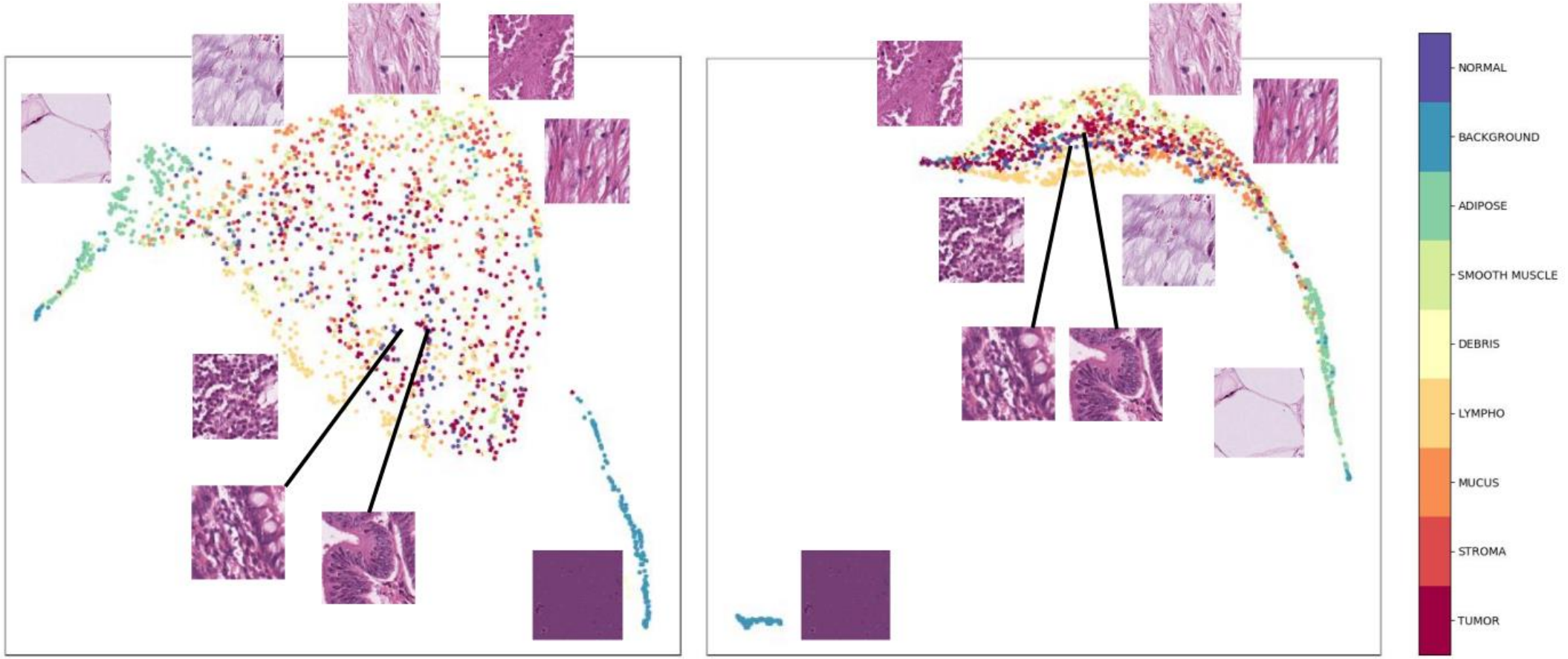


BUT



BUNCA

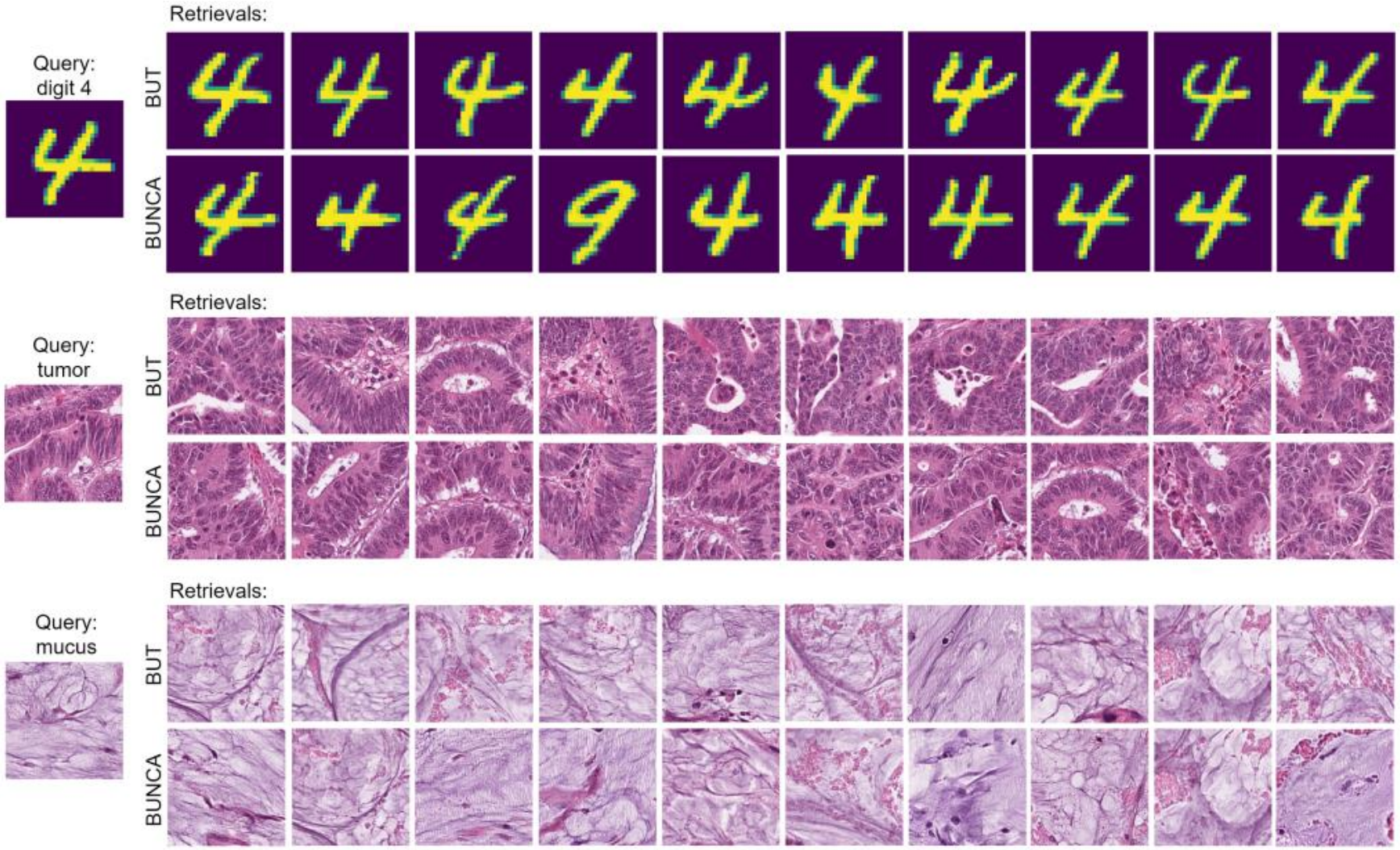
Experiments on CRC



BUT

BUNCA

Query-Retrieval



Numerical Results

MNIST dataset:

	R@1	R@4	R@8	R@16
BA [10]	79.31	93.53	96.55	98.21
BSH [2]	78.95	92.61	96.09	98.17
BH [11]	85.75	95.31	97.43	98.63
EP [12]	73.34	90.09	95.08	97.68
DWS [9]	76.44	91.35	95.72	97.68
NCA [7]	85.40	95.48	97.46	98.76
proxy-NCA [8]	83.71	94.69	97.31	98.55
BUT	88.03	96.25	98.15	99.09
BUNCA	78.67	92.44	95.77	98.02

CRC dataset:

	R@1	R@4	R@8	R@16
BA [10]	38.54	66.76	80.64	89.97
BSH [2]	30.85	60.39	77.73	90.33
BH [11]	79.09	92.60	96.00	97.95
EP [12]	69.94	87.88	93.20	96.38
DWS [9]	76.06	91.31	95.34	97.58
NCA [7]	77.87	92.25	95.92	98.01
proxy-NCA [8]	78.85	92.24	95.80	97.78
BUT	79.14	92.32	95.60	97.65
BUNCA	78.67	92.28	95.64	97.71

Thank you