Decoupled modeling for NL Scoring (DE-NL)

Lantian Li

2020-08-31

General theory of verification decision

- Two-class hypothesis test
  - $\boldsymbol{H_0}$: the speech x is from the claimed speaker.
  - $\boldsymbol{H_1}$: the speech x is from an impostor.

- This is known as the likelihood ratio test.

$$LR = \frac{p(x|H_0)}{p(x|H_1)}$$

From LR to NL

- Normalized likelihood
  - $p(x|H_0)$: denotes as $p_c(x)$, which is a speaker-dependent item.
  - $p(x|H_1)$: denotes as $p(x)$, which is a speaker-independent item.

$$NL(x|c) = \frac{p(x|H_0)}{p(x|H_1)} = \frac{p_c(x)}{p(x)}$$

NL reflects two key elements in open-set verification

- How to determine $p(x|c)$ for an unseen class c ?
  - We need an accurate $p(x|c)$ to describe the within-class variance.

- How to define $p(x)$ for any test data x ?
  - We need a global $p(x)$ to represent the normalization item.

Decoupled modeling for NL scoring

$$NL = \frac{p_c(x)}{p(x)} = \frac{p(x|x_1^c, \dots, x_n^c)}{p(x)} = \frac{\int p(x|u)p(u|\, x_1^c, \dots, x_n^c)\mathrm{d}u}{\int p(x|u)\, p(u)\mathrm{d}u}$$

- Decouple NL to **three** components
  - Enrollment: $p(u|x_1^c, \dots, x_n^c)$ produces the posterior of class mean.
  - Prediction: $p(x|u)$ computes the likelihood of x belonging to class c.
  - Normalization: $p(x)$ computes the likelihood of x from all classes.

How to decouple ?

- Enrollment $p(u|x_1^c, \ldots, x_n^c)$ and Normalization $p(x)$ are relevant to a global generative model, e.g., PLDA.
  - $p_g(u) = N(u; 0, \varepsilon I)$
  - $p_g(x|u) = N(x; u, \sigma I)$
- Predication $p(x|c)$ regards as a local model
  - $p_l(x|u) = N(x; u, \Sigma')$

$$NL = \frac{p_c(x)}{p(x)} = \frac{p(x|x_1^c, \ldots, x_n^c)}{p(x)} = \frac{\int p_l(x|u) p_g(u|\, x_1^c, \ldots, x_n^c) \mathrm{d}u}{\int p_g(x|u)\, p_g(u) \mathrm{d}u}$$

Training process

- Global training
  - ML-PLDA

$$p(\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \propto |\sigma \mathbf{I}|^{-n/2} |\boldsymbol{\epsilon} \mathbf{I}|^{-1/2} |(n/\sigma + 1/\boldsymbol{\epsilon})\mathbf{I}|^{-1/2} \exp\Big\{ -\frac{1}{2\sigma}\Big\{ \sum_i ||\boldsymbol{x}_i||^2 - \frac{n^2 \boldsymbol{\epsilon}}{n\boldsymbol{\epsilon} + \sigma} ||\bar{\boldsymbol{x}}||^2 \Big\}\Big\}, \quad (3)$$

where $|\cdot|$ defined is the absolute value of the determinant of a matrix. Given a training set consisting of K classes, the parameters $\boldsymbol{\epsilon}$ and σ can be optimized by maximizing the likelihood function:

$$\mathcal{L}(\boldsymbol{\epsilon}, \sigma) = \sum_{k=1}^{K} p(\boldsymbol{x}_1^k, ..., \boldsymbol{x}_{n_k}^k),$$

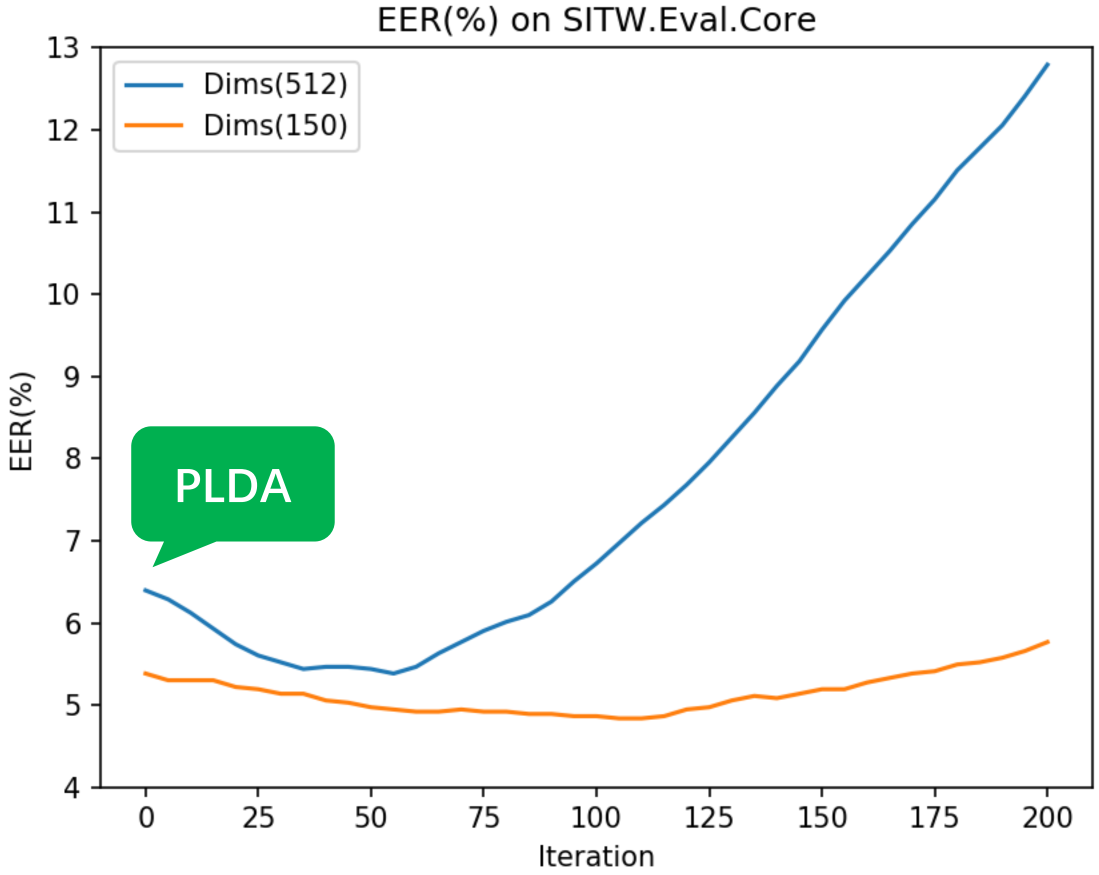
where $\boldsymbol{x}_i^k$ is the i-th sample of the k-th class.

- Local training
  - MLLR $x' = Mx$

$$\begin{aligned}
\mathcal{L}(\mathrm{M}) &= \prod_k^K \prod_{i=1}^{n_k} \int p_l(\boldsymbol{x}_i^k | \boldsymbol{\mu}, \tilde{\boldsymbol{\Sigma}}) p_g(\boldsymbol{\mu} | \boldsymbol{x}_1^k, ..., \boldsymbol{x}_{n_k}^k) \mathrm{d}\boldsymbol{\mu} \\
&= \prod_k^K \prod_{i=1}^{n_k} \int p_g(\mathrm{M}\boldsymbol{x}_i^k | \boldsymbol{\mu}, \sigma \mathbf{I}) p_g(\boldsymbol{\mu} | \boldsymbol{x}_1^k, ..., \boldsymbol{x}_{n_k}^k) \mathrm{d}\boldsymbol{\mu} \\
&= \prod_k^K \prod_{i=1}^{n_k} \mathcal{N}(\mathrm{M}\boldsymbol{x}_i^k; \frac{n_k \epsilon}{n_k \boldsymbol{\epsilon} + \sigma} \bar{\boldsymbol{x}}_k, \mathbf{I}(\sigma + \frac{\epsilon\sigma}{n_k \boldsymbol{\epsilon} + \sigma}))
\end{aligned}$$

Any numerical optimizer can be employed to optimize the above objective function, for instance stochastic gradient descend (SGD).

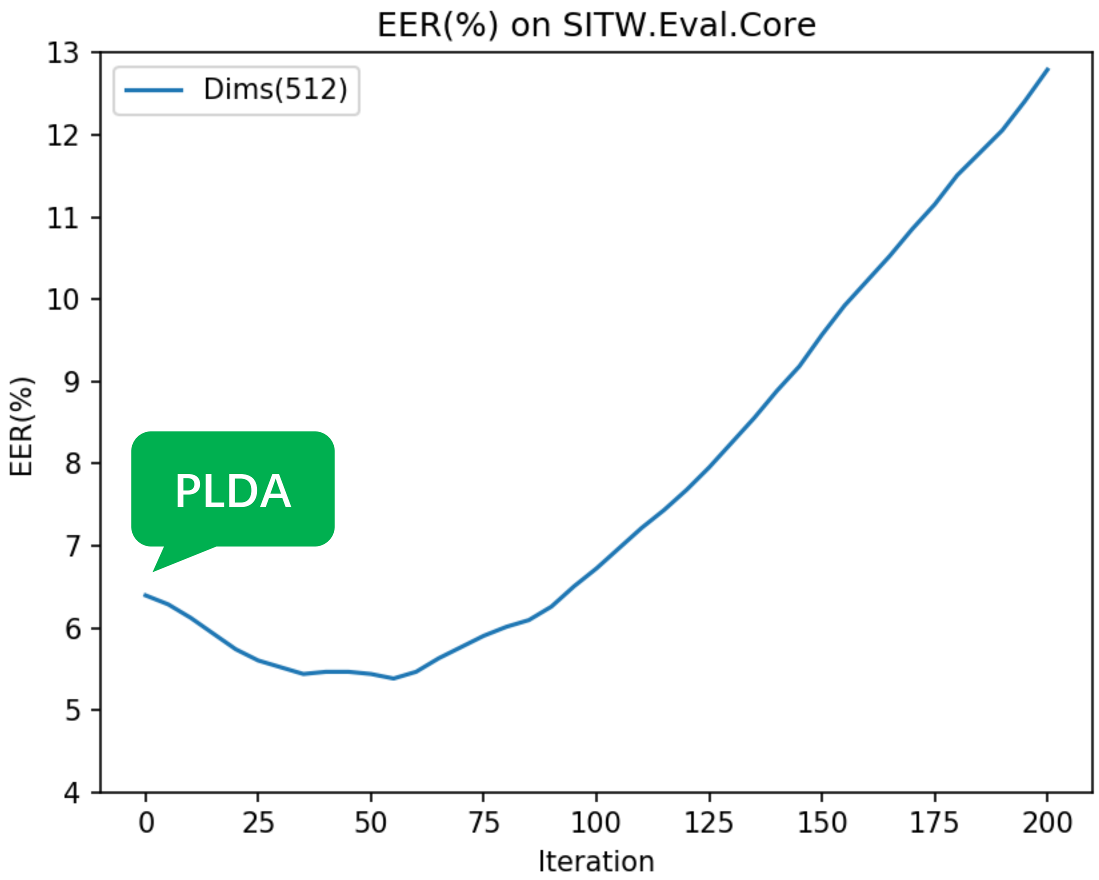
Basic EER results

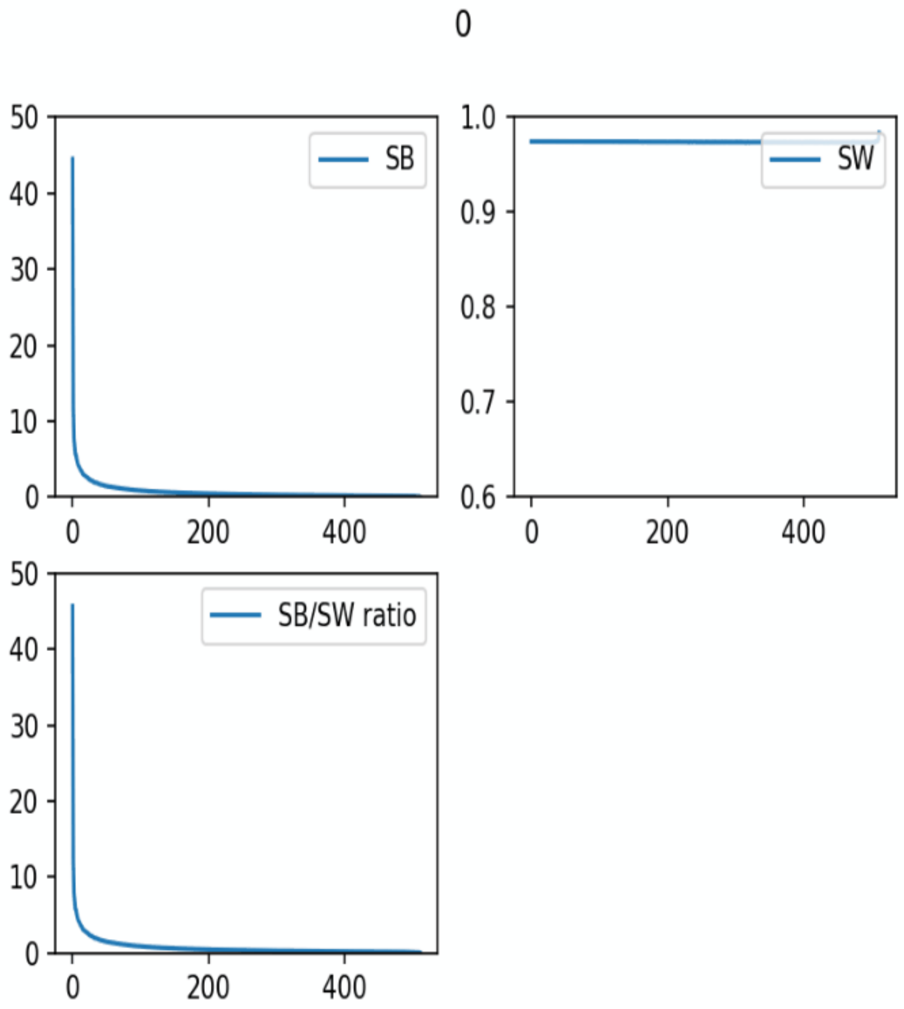


EER(%) results on SITW.Eval.Core

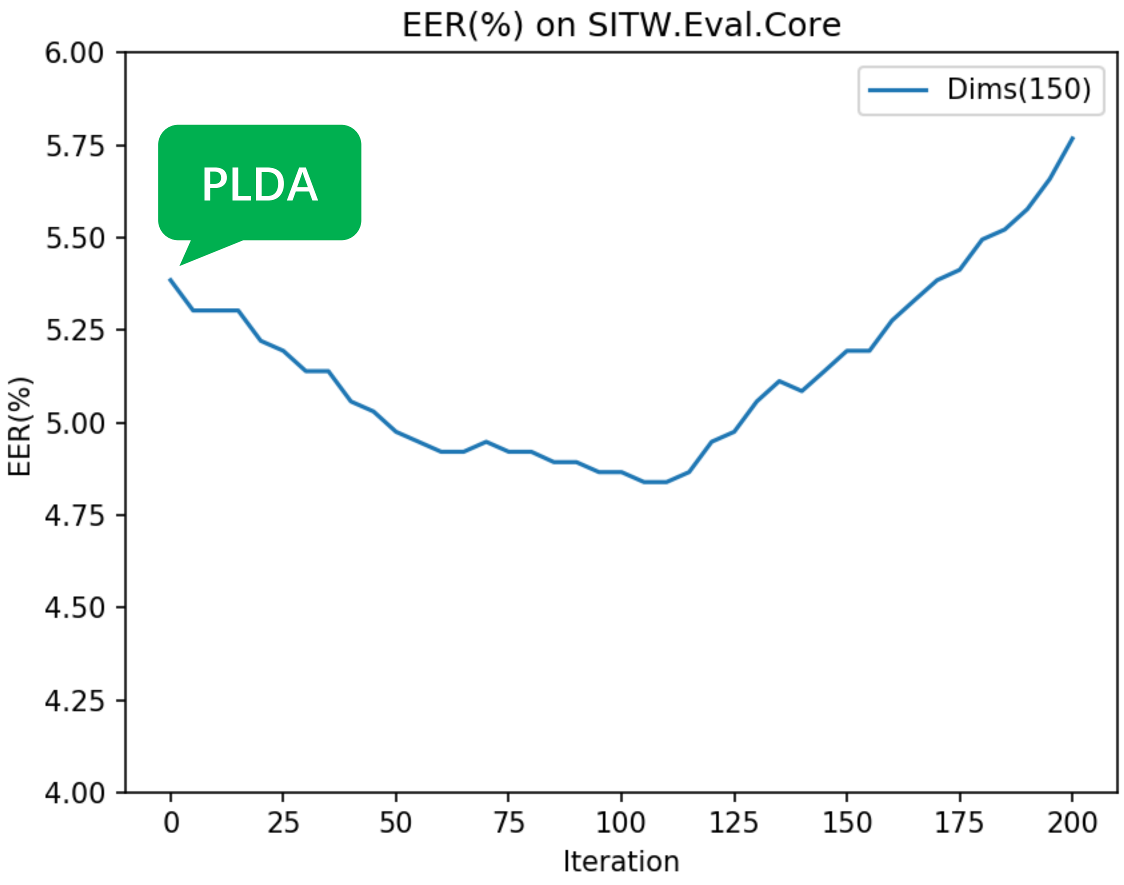
x-vector	PLDA	DE-NL
512	6.397%	5.385%
150	5.385%	4.839%

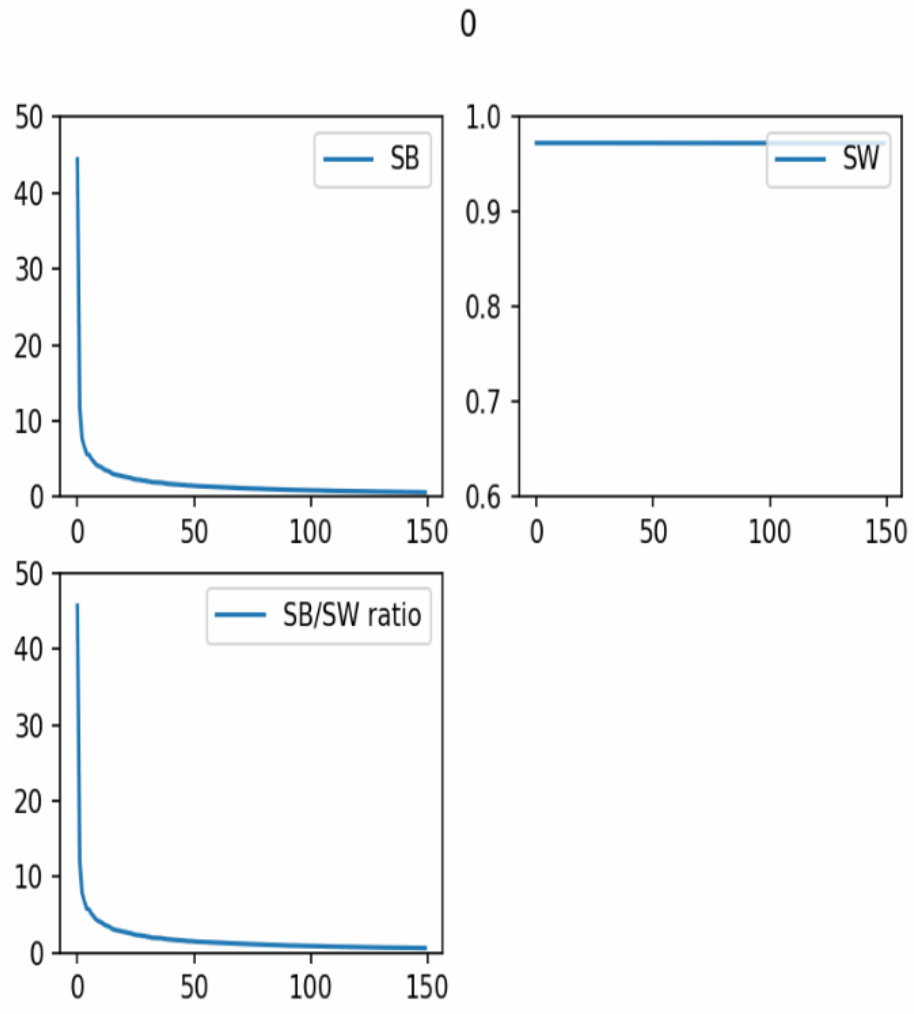
Change of Statistics (512)





Change of Statistics (150)





We need more thinking

- Observations
  - DE-NL outperforms the standard PLDA.
  - The curve of within-speaker variance does not match the PLDA assumption.

- Questions
  - How to explain the change of within-speaker variance ?
  - How to determine the optimal iteration ?

Analysis of local model $p_l(x|u)$

$$\mathcal{L}(m) = -\sum_{k}^{K}\sum_{i}^{n_k}(m x_i^k - \frac{n_k\epsilon}{n_k\epsilon+\sigma}\bar{x}_k)^2.$$

Then let the gradient to be zero:

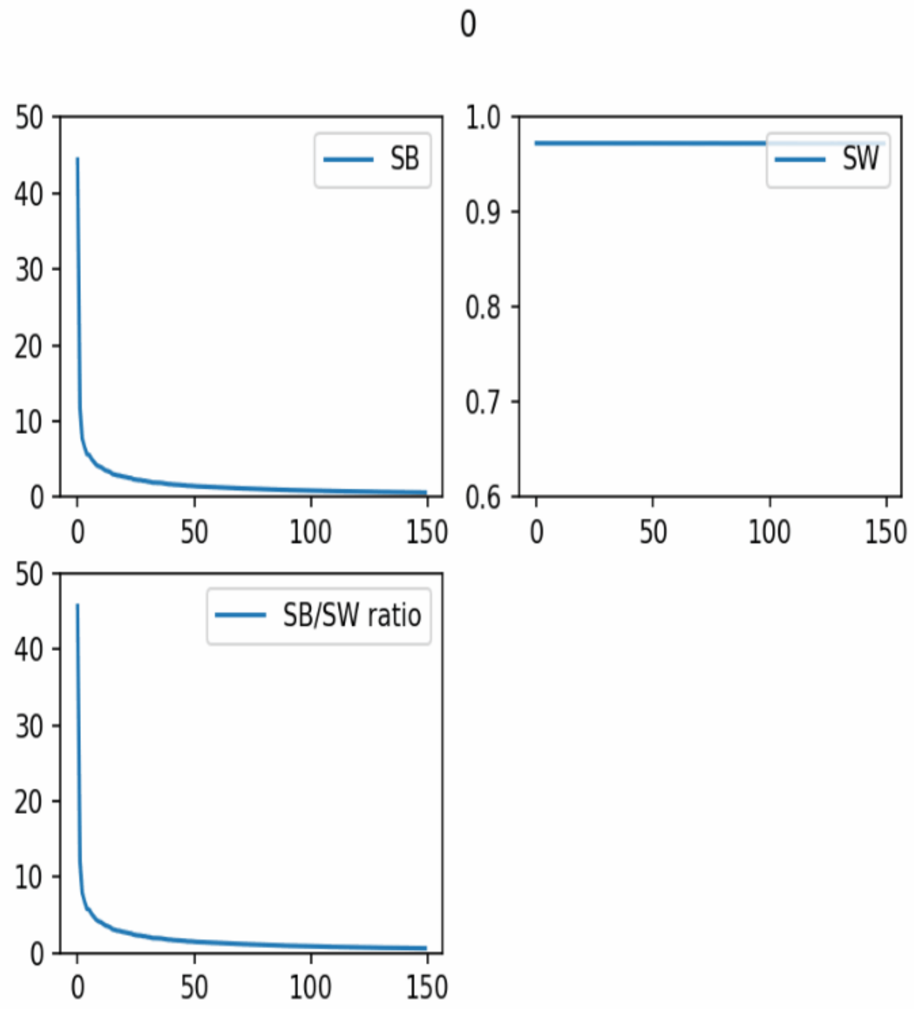
$$\frac{\partial \mathcal{L}(m)}{\partial m} = -\sum_{k}^{K}\sum_{i}^{n_k} 2(m x_i^k - \frac{n_k\epsilon}{n_k\epsilon+\sigma}\bar{x}_k)x_i^k = 0.$$

A simple arrangement shows the follows:

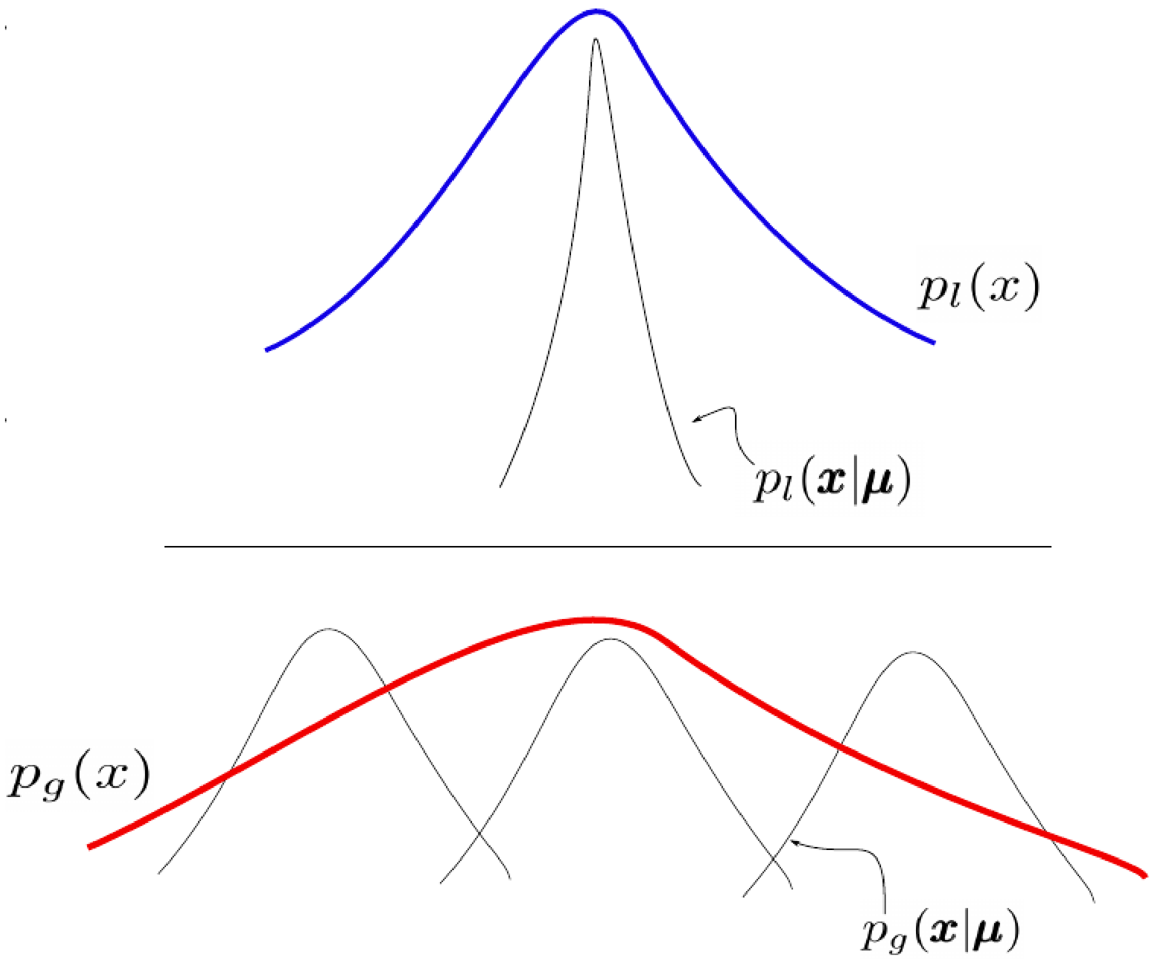
$$m^* = \frac{\sum_k^K \frac{n_k^2\epsilon}{n_k\epsilon+\sigma}\bar{x}_k^2}{\sum_k^K\sum_i^{n_k}(x_i^k)^2}.$$

According to the linear Gaussian assumption, the mean $\bar{x}_k$ follows Gaussian $N(0,\epsilon+\frac{\sigma}{n_k})$, x_i^k follows Gaussian $N(0,\epsilon+\sigma)$, the expectation of $(x_i^k)^2$ is $(\sigma+\epsilon)$. If we assume $n_k = n$ for all the classes, we have:

$$m^* = \frac{n^2\epsilon}{n\epsilon+\sigma}\frac{K(\epsilon+\frac{\sigma}{n})}{nK(\epsilon+\sigma)} = \frac{\epsilon}{\epsilon+\sigma}.$$



$p_l(x|u)$ vs. $p_g(x|u)$

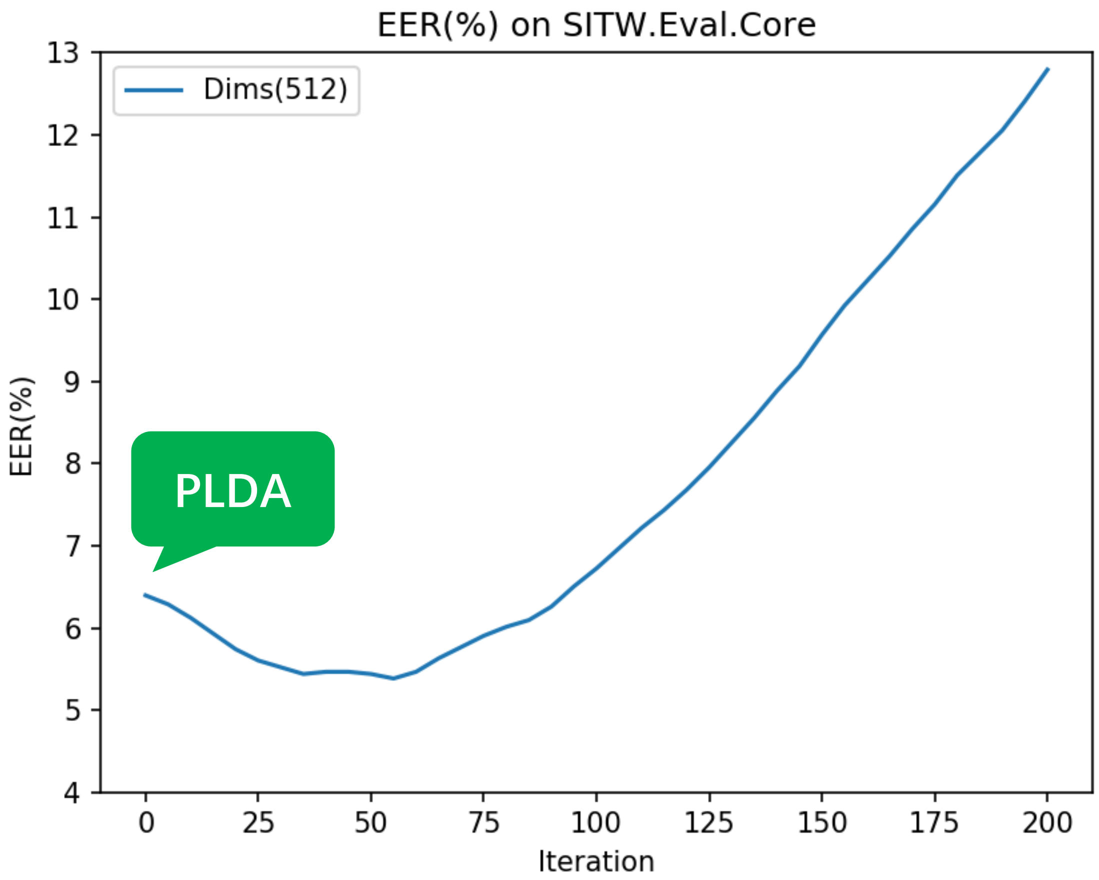


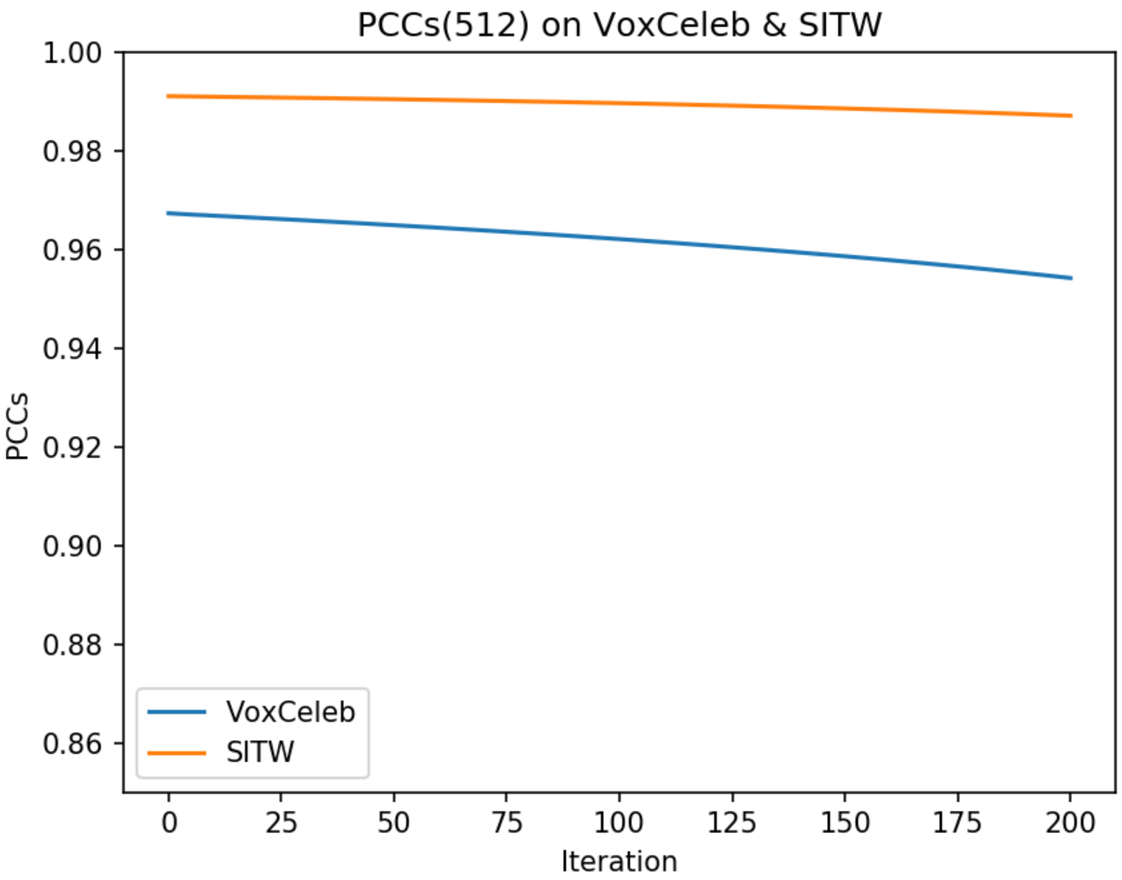
- Good thing
  - More accurate local model $p_l(x|u)$
- Potential problem
  - incorrect normalization item

  $p_l(x)$ and $p_g(x)$

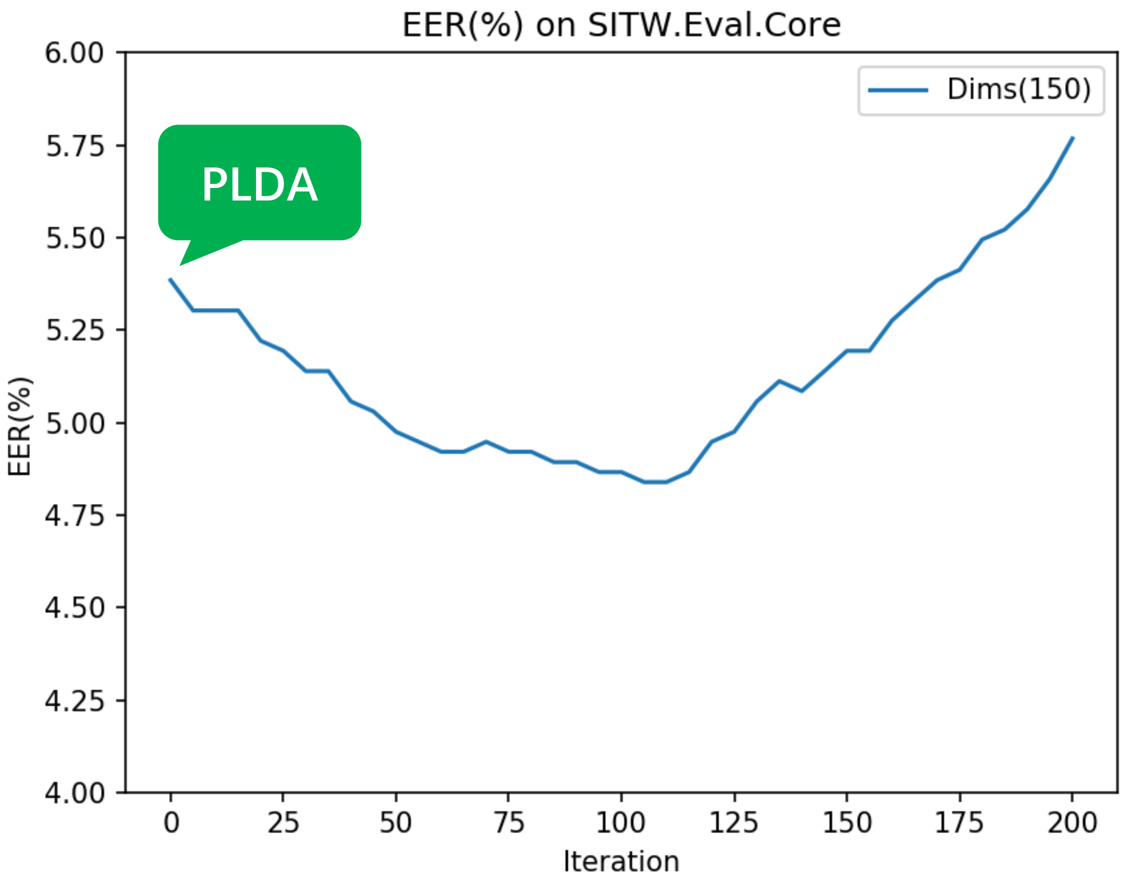
Correlation $\left\{\log p_g(x), \ \log \sum_c p_l(x)\right\}$

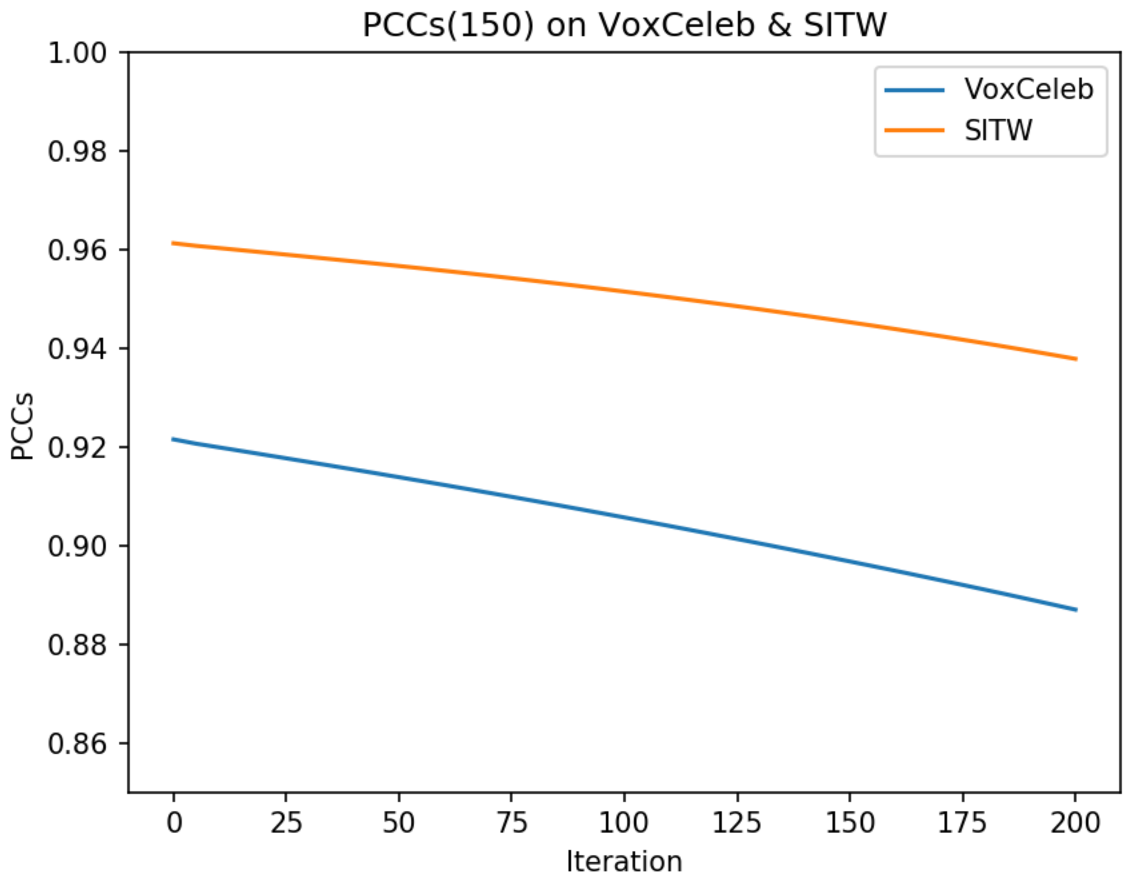
Correlation (512) with iterative training





Correlation (150) with iterative training





Conclusions

- This decoupled NL is flexible and shows good performance.
- We may add a regularization to balance the ideal normalization $\sum_c p_l(x)$ and practical normalization $p_g(x)$.
- More analysis on DE-NL with LN.