EVIDENTIAL DEEP LEARNING FOR HIGH-CONFIDENCE SAMPLE SELECTION IN NOISY LABEL LEARNING

Marc Pascual Roig

Thesis supervisors:
Prof. Petia Radeva
Dr. Bhalaji Nagarajan

Master's Thesis (Master's Degree in Artificial Intelligence)
UB - UPC - URV

1. INTRODUCTION
2. THEORETICAL FRAMEWORK
3. PROPOSED MODEL
4. RESULTS AND ANALYSIS
5. CONCLUSIONS

1.1. PROBLEM STATEMENT

Deep learning success relies heavily on **large-scale, labeled** datasets

DNNs tend to memorize[2] noisy labels, leading to overfitting and poor generalization

LNL methods are essential for improving model **robustness** and **reliability**

Clean annotations are difficult to obtain (Web-based data, human biases)

Label noise[1] is common in large datasets (8-38%)

Data augmentation, dropout, weight decay, batch normalization are often **insufficient**

[1]Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. (2021), 'Understanding deep learning (still) requires rethinking generalization', Communications of the ACM 64(3), 107–115.
[2]Xiao, T., Xia, T., Yang, Y., Huang, C. and Wang, X. (2015a), Learning from massive noisy labeled data for image classification, in '2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2691–2699.

1.2. OBJECTIVES

- Thorough **understanding** of the fundamentals of **LNL**.
- Comprehensive study of **Uncertainty Quantification** in DNNs.

- Investigate how **EDL** can be incorporated into the **DivideMix algorithm**.
- **Design** a sample selection-based LNL pipeline.

- **Evaluate** the proposed model.
- **Analyze** the **results** and the **uncertainty estimation** of our model.

1. INTRODUCTION
2. THEORETICAL FRAMEWORK
3. PROPOSED MODEL
4. RESULTS AND ANALYSIS
5. CONCLUSIONS

2.1. NOISE IN DATASETS

Types of noise:

- Feature noise
- Label noise (corrupted class labels)

Main sources of label noise:

- Cognitive bias & ambiguity
- Web-based labeling

2.1. NOISE IN DATASETS

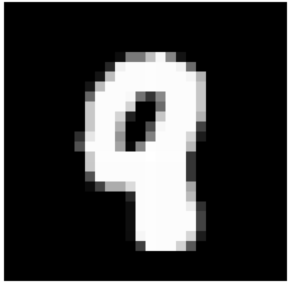
Types of noise:

- ~~Feature noise~~
- Label noise (corrupted class labels)

Main sources of label noise:

- Cognitive bias & ambiguity
- Web-based labeling

Frog → Cat
CIFAR-10



9 → 8
MNIST

Baboon → Siamang
ImageNet

Spider → Thick
ImageNet

2.2. NOISE CONFIGURATION

1. Noise type

Instance-independent noise:

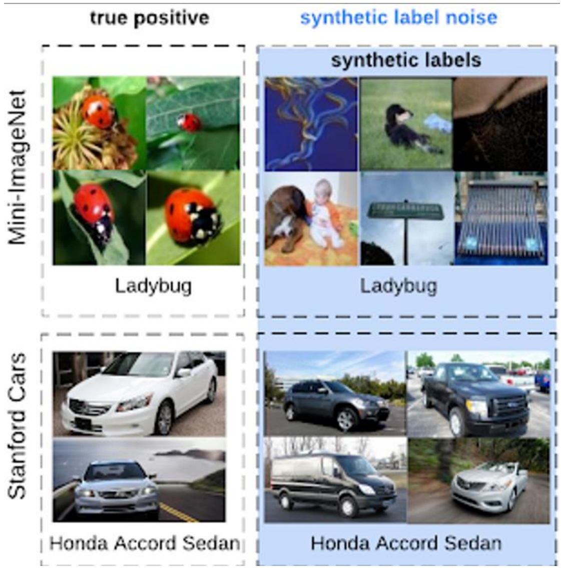
1. Symmetric noise (random)
2. Asymmetric noise (between specific classes)

Instance-dependent noise:

- Mislabeling depends on both features and true label.

2. Noise rate (%)

(20%-80%)



2.3. APPROACHES IN LNL

1. Robust Architecture

- Modifies the network structure to explicitly handle noisy labels.
- Very difficult to model!!!

2. Loss adjustment

- Adjusts the training dynamics.
- Relies on clean data and noise estimation!!!

3. Robust Loss

- Some loss functions are more robust to noise.
- *Example*: Contrastive Loss.

4. Robust regularization

- Prevents overfitting to noisy data and improve generalization.
- *Example*: Mixup interpolation.

Song, H., Kim, M., Park, D., Shin, Y. and Lee, J.-G. (2022), 'Learning from noisy labels with deep neural networks: A survey'. URL: https://arxiv.org/abs/2007.08199

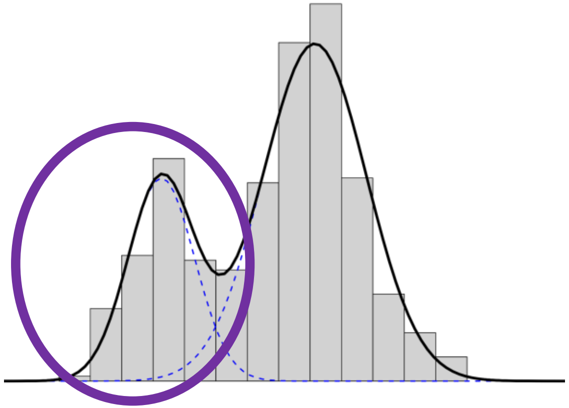
2.4. SAMPLE SELECTION

5. Sample selection

- Identifies and prioritizes high-confidence (likely clean) samples during training to reduce the impact of noisy labels.

Gaussian Mixture Model[1] (GMM)

- *Small-loss trick.*
- Memorization effect[2].



Low-loss samples (likely clean)

[1]Reynolds, D. (2015). Gaussian mixture models. In *Encyclopedia of biometrics* (pp. 827-832). Springer, Boston, MA.
[2]Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. (2021), 'Understanding deep learning (still) requires rethinking generalization', Communications of the ACM 64(3), 107–115.

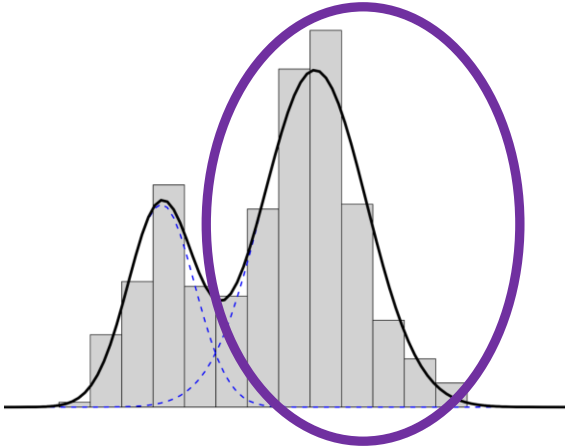
2.4. SAMPLE SELECTION

5. Sample selection

- Identifies and prioritizes high-confidence (likely clean) samples during training to reduce the impact of noisy labels.

Gaussian Mixture Model[1] (GMM)

- *Small-loss trick.*
- Memorization effect[2].



High-loss samples (likely noisy)

[1]Reynolds, D. (2015). Gaussian mixture models. In *Encyclopedia of biometrics* (pp. 827-832). Springer, Boston, MA.
[2]Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. (2021), 'Understanding deep learning (still) requires rethinking generalization', Communications of the ACM 64(3), 107–115.

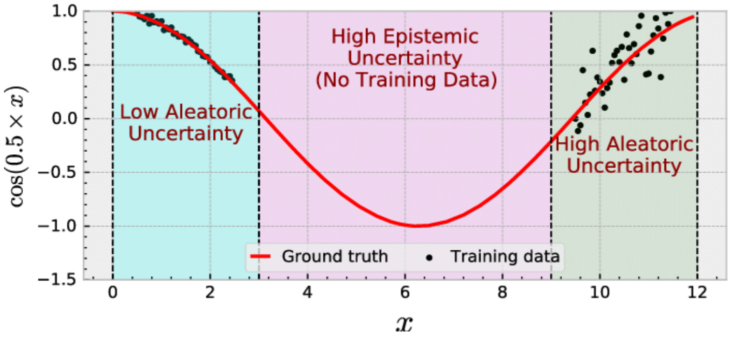
2.5. UNCERTAINTY QUANTIFICATION

Neural netwoks have:

- Poor **uncertainty calibration**.
- **Overconfident predictions** even when incorrect.

- **Aleatoric uncertainty**: fluctuations in the data distribution.
- **Epistemic uncertainty**: limitations of the model.

1. **Reduce** the epistemic uncertainty.
2. **Penalize** overconfident predictions.



Tuna, Ömer & Catak, Ferhat Ozgur & Eskil, Taner. (2022). Uncertainty as a Swiss army knife: new adversarial attack and defense ideas based on epistemic uncertainty. Complex & Intelligent Systems. 9. 10.1007/s40747-022-00701-0.

2.6. APPROACHES

1. Ensembles

- Costly: minimum 5 neural networks.

2. BNN

- Converge to local optimum.

3. Test-time data augmentation

- High-dependent on the type of augmentations.

4. Single deterministic methods (Dirichlet distribution)

- Incorporates model uncertainty estimation implicitly.
- Requires only one single forward pass.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R. and Zhu, X. X. (2022), 'A survey of uncertainty in deep neural networks'. URL: https://arxiv.org/abs/2107.03342

2.7. DIRICHLET DISTRIBUTION

Network outputs are treated as a Dirichlet Distribution:

1. Network output:
$f(x) = [z_1, z_2, \ldots, z\_C]$
(Raw logits from the classification network)

2. Apply non-negative activation:
e_i = Exponential(z_i) → Evidence for class i

3. Add prior weight to form Dirichlet parameters:
$\alpha_i = e_i + W$ → Parameters of a Dirichlet distribution
(W is typically 1 or 10 / C)

4. Compute belief and uncertainty:
S = sum over all α_i
$b_i = e_i / S$ → Belief mass for class i

A **Dirichlet distribution** over categorical probabilities is defined by parameters:
$\boldsymbol{\alpha = [\alpha_1, \alpha_2, \ldots]}$.

Each $\alpha_i>0$ represents the ***evidence*** for class i.

2.8. SUBJECTIVE LOGIC

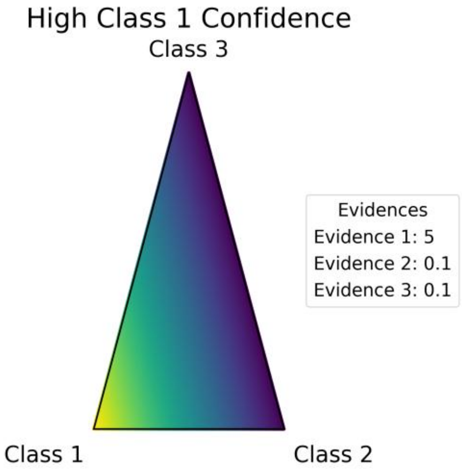
An opinion is a triplet:
$\omega = (b, u, a)$
where:

- **$b = [b_1, b_2, \ldots, b_k]$** is the belief mass over k classes.
- **u (vacuity)** is the **uncertainty mass ($u = C*W/S$).**
- **$a = [a_1, a_2, \ldots, a_k]$** is the base rate (prior).
- **$b_1 + b_2 + \ldots + b_k + u = 1$.**

2.8. SUBJECTIVE LOGIC

An opinion is a triplet:
ω = (b, u, a)
where:

- **b = [b_1, b_2, ..., b_k]** is the belief mass over k classes.
- **u (vacuity)** is the **uncertainty mass (u = C*W/S).**
- **a = [a_1, a_2, ..., a_k]** is the base rate (prior).
- **b_1 + b_2 + ... + b_k + u = 1.**

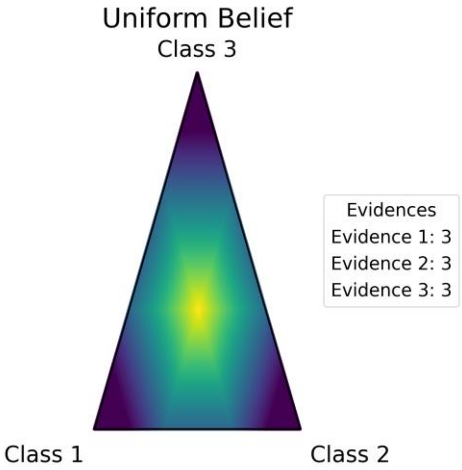


- Low vacuity
- Low dissonance

2.8. SUBJECTIVE LOGIC

An opinion is a triplet:
ω = (b, u, a)
where:

- **b = [b_1, b_2, ..., b_k]** is the belief mass over k classes.
- **u (vacuity)** is the **uncertainty mass (u = C*W/S).**
- **a = [a_1, a_2, ..., a_k]** is the base rate (prior).
- **$b_1 + b_2 + ... + b_k + u = 1$.**

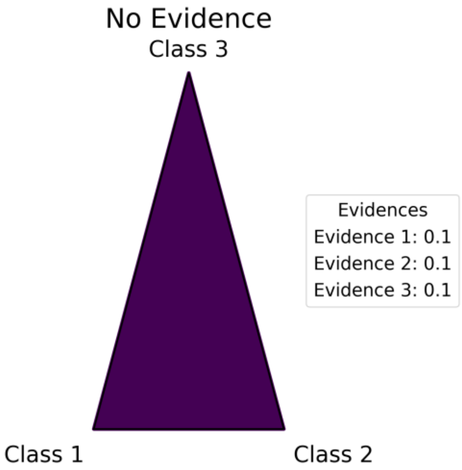


- Low vacuity
- High dissonance

2.8. SUBJECTIVE LOGIC

An opinion is a triplet:
$\omega = (b, u, a)$
where:

- $b = [b_1, b_2, \ldots, b_k]$ is the belief mass over k classes.
- **u (vacuity)** is the **uncertainty mass (u = C*W/S).**
- $a = [a_1, a_2, \ldots, a_k]$ is the base rate (prior).
- $b_1 + b_2 + \ldots + b_k + u = 1.$



- High vacuity
- High dissonance

1. INTRODUCTION
2. THEORETICAL FRAMEWORK
3. **PROPOSED MODEL**
4. RESULTS AND ANALYSIS
5. CONCLUSIONS

3.1. BASELINE: DIVIDEMIX

- Benchmark algorithm.
- Foundation of many SOTA models.
- Flexible.

3.1. BASELINE: DIVIDEMIX

1. WarmUp (few epochs):

Regular training with noisy labels.

Learns general patterns.

Li, J., Socher, R. and Hoi, S. C. H. (2020), 'Dividemix: Learning with noisy labels as semi supervised learning'. URL: https://arxiv.org/abs/2002.07394

3.1. BASELINE: DIVIDEMIX

2. Semi-supervised stage
Co-Teaching (avoid confirmation bias)

1. WarmUp (few epochs):

Regular training with noisy labels.

Learns general patterns.

3.1. BASELINE: DIVIDEMIX

2. Semi-supervised stage
Co-Teaching (avoid confirmation bias)

1. WarmUp (few epochs):

Regular training with noisy labels.

Learns general patterns.

2.1. Sample modeling:

Algorithm: GMM.

Distribution: Loss (small-loss hypothesis).

Li, J., Socher, R. and Hoi, S. C. H. (2020), 'Dividemix: Learning with noisy labels as semi supervised learning'. URL: https://arxiv.org/abs/2002.07394

3.1. BASELINE: DIVIDEMIX

2. Semi-supervised stage
Co-Teaching (avoid confirmation bias)

1. WarmUp (few epochs):

Regular training with noisy labels.

Learns general patterns.

2.1. Sample modeling:

Algorithm: GMM.

Distribution: Loss (small-loss hypothesis).

2.2. Label Co-Guessing and Co-Refinement:

Noisy samples: average predictions.

Clean samples: refinement of labels.

Li, J., Socher, R. and Hoi, S. C. H. (2020), 'Dividemix: Learning with noisy labels as semi supervised learning'. URL: https://arxiv.org/abs/2002.07394

3.1. BASELINE: DIVIDEMIX

2. Semi-supervised stage
Co-Teaching (avoid confirmation bias)

1. WarmUp (few epochs):

Regular training with noisy labels.

Learns general patterns.

2.1. Sample modeling:

Algorithm: GMM.

Distribution: Loss (small-loss hypothesis).

2.2. Label Co-Guessing and Co-Refinement:

Noisy samples: average predictions.

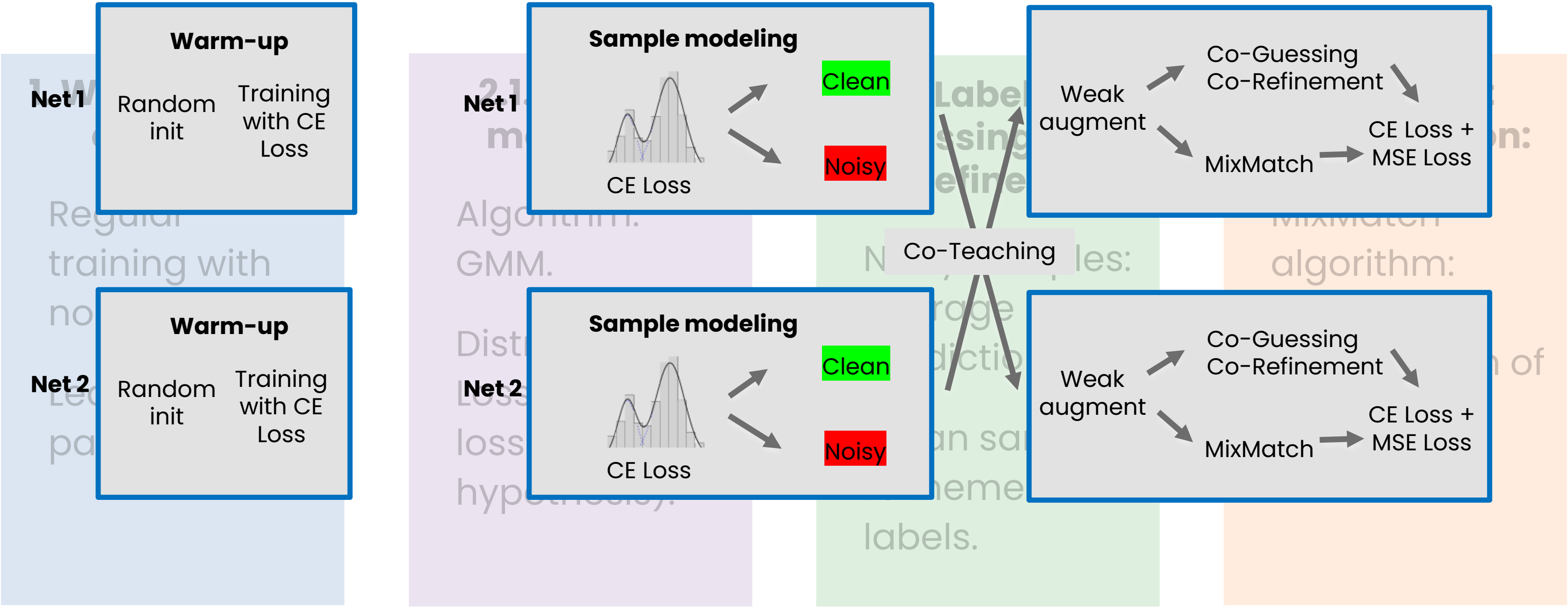
Clean samples: refinement of labels.

2.3. Robust regularization:

MixMatch algorithm: linear interpolation of two images.

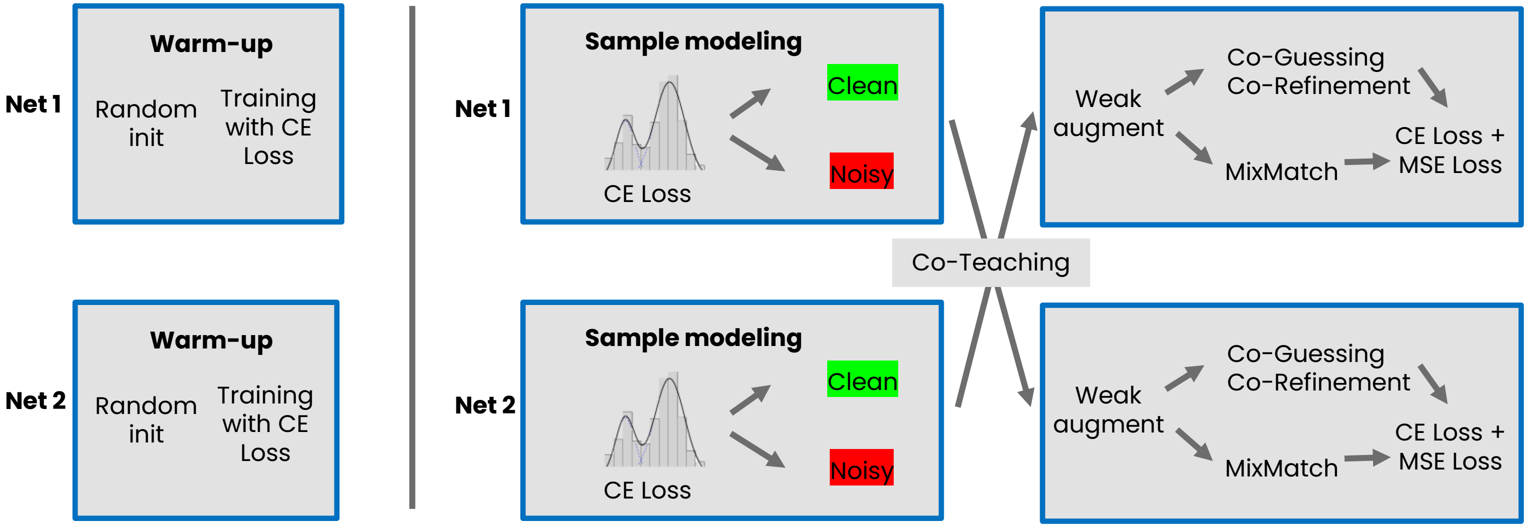
Li, J., Socher, R. and Hoi, S. C. H. (2020), 'Dividemix: Learning with noisy labels as semi supervised learning'. URL: https://arxiv.org/abs/2002.07394

3.1. BASELINE: DIVIDEMIX



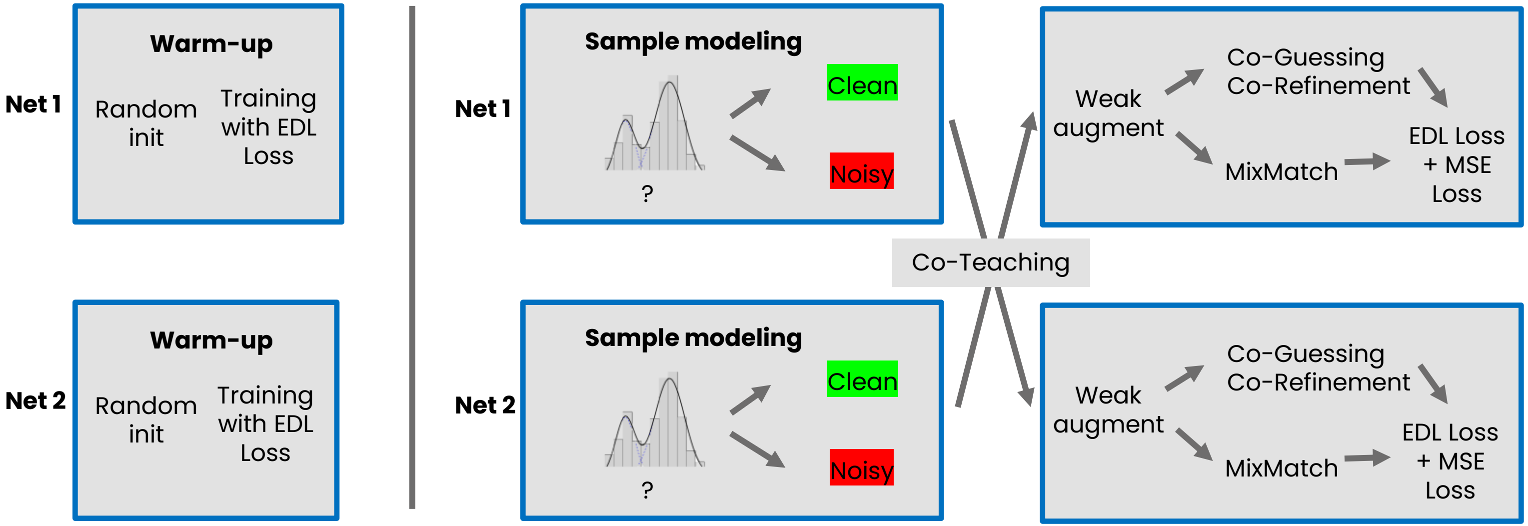
3.2. EDL

Motivation: Uncertainty Unawareness of DNNs / Overconfident predictions.



3.2. EDL

Motivation: Uncertainty Unawareness of DNNs / Overconfident predictions.



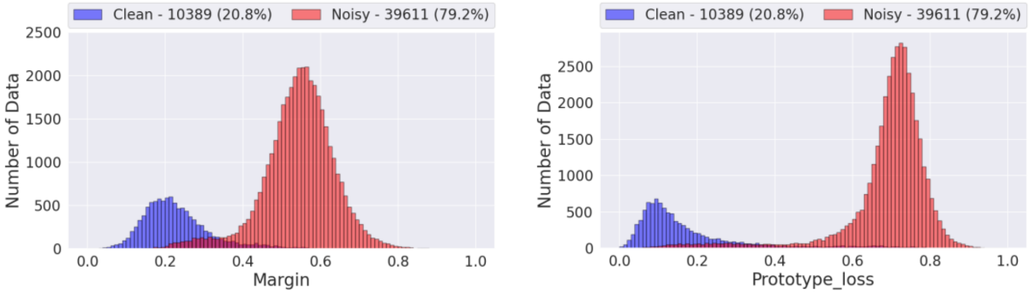
3.3. SAMPLE SELECTION

Motivation:

- Sample selection does not include **feature information**.
- **Easy classes** → low loss.
- **Difficult classes** → high loss.

1. 2D GMM:

- Prototype Loss distribution (feature information)
- Margin distribution



2. Class-Balance in the clean set.

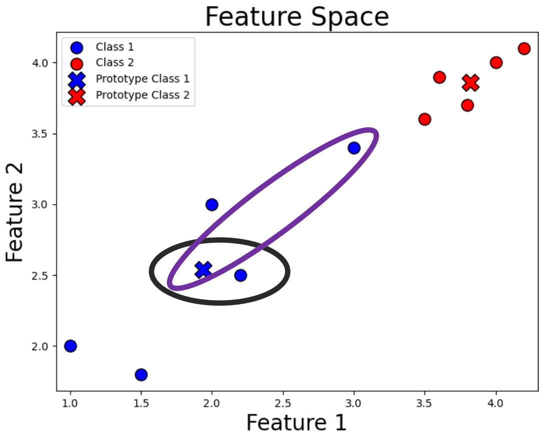
3.3. SAMPLE SELECTION

1. Margins:

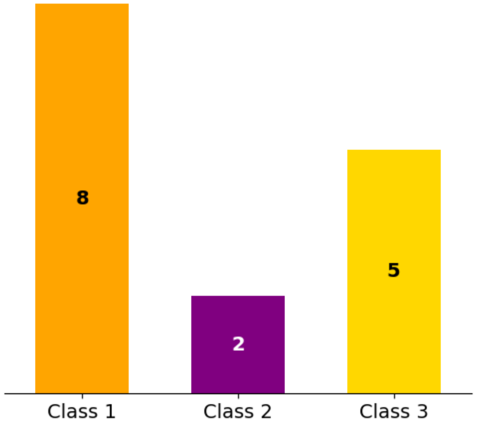
Evidence 1	Evidence 2	Evidence 3	Evidence 4	Evidence 5
2	5	10	12	1

Margin = 12 -10 = 2

2. Prototype Loss:

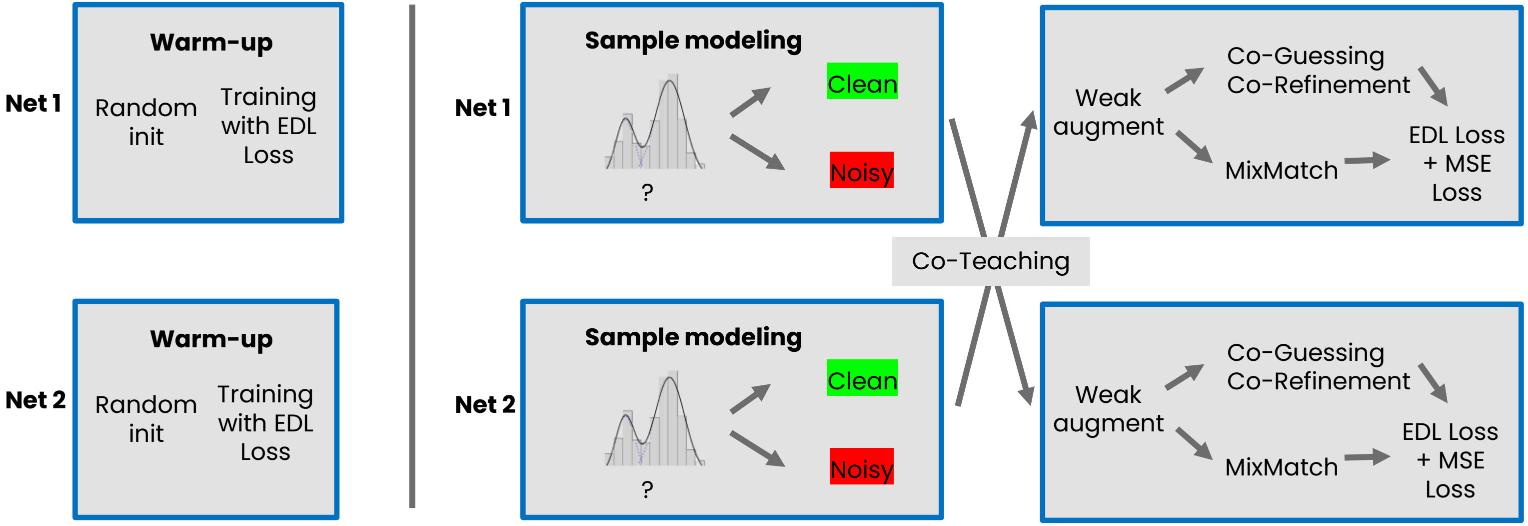


3. Class-Balance Refinement:

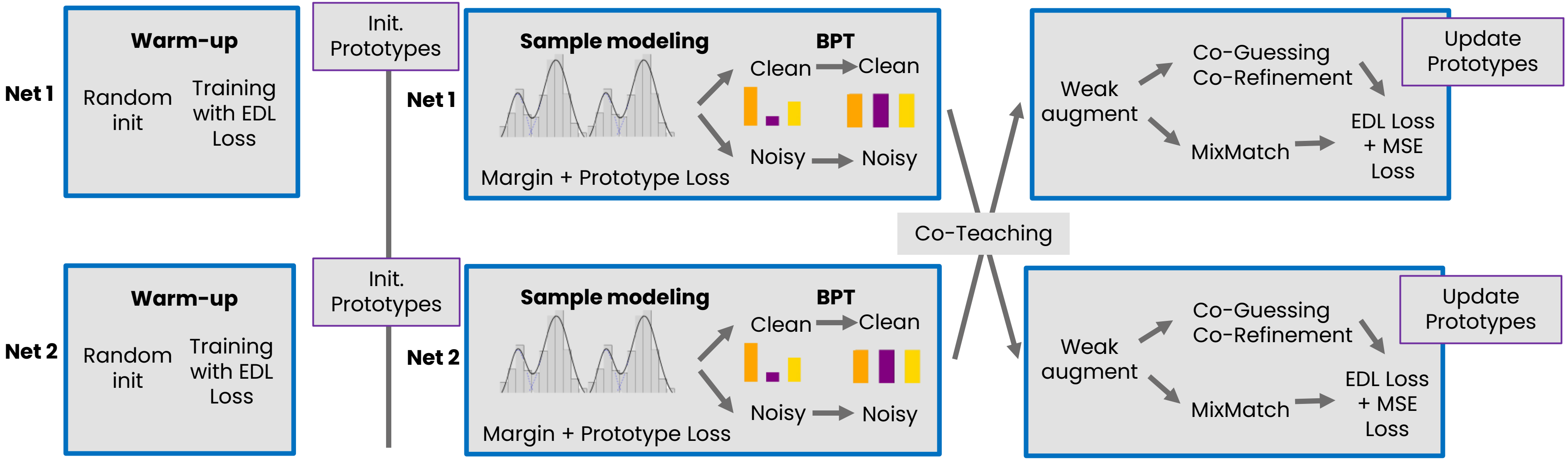


Zhang, Q., Jin, G., Zhu, Y., Wei, H. and Chen, Q. (2024), 'Bpt-plr: A balanced partitioning andtraining framework with pseudo-label relaxed contrastive loss for noisy label learn ing', Entropy 26(7).

3.3. SAMPLE SELECTION

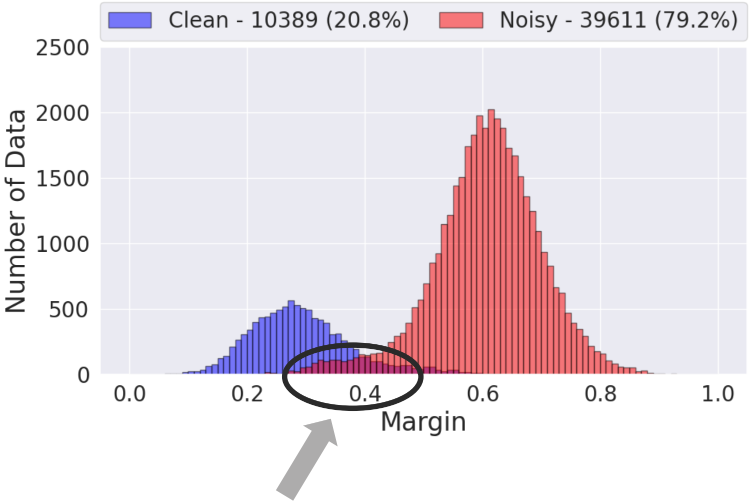


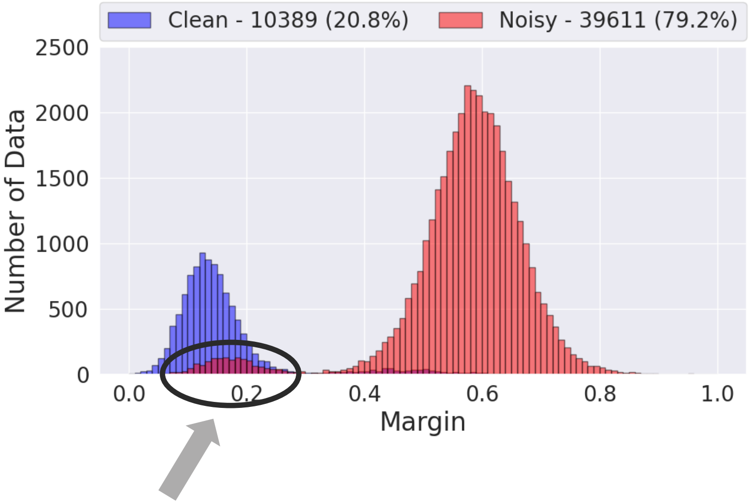
3.3. SAMPLE SELECTION



3.4. CONTRASTIVE LEARNING

Motivation: WarmUp introduces Noise memorization / Memorization of noisy samples increases.

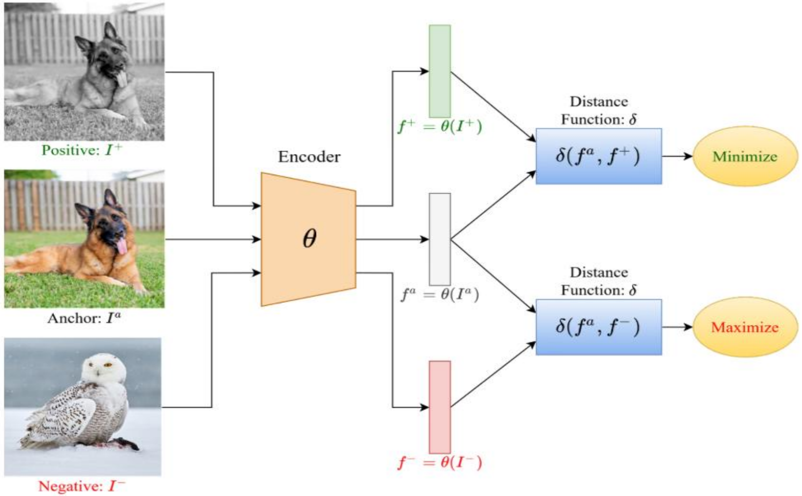




3.4. CONTRASTIVE LEARNING

Robust Training Loss: Self-supervised Contrastive Learning (PLR Loss[1]).

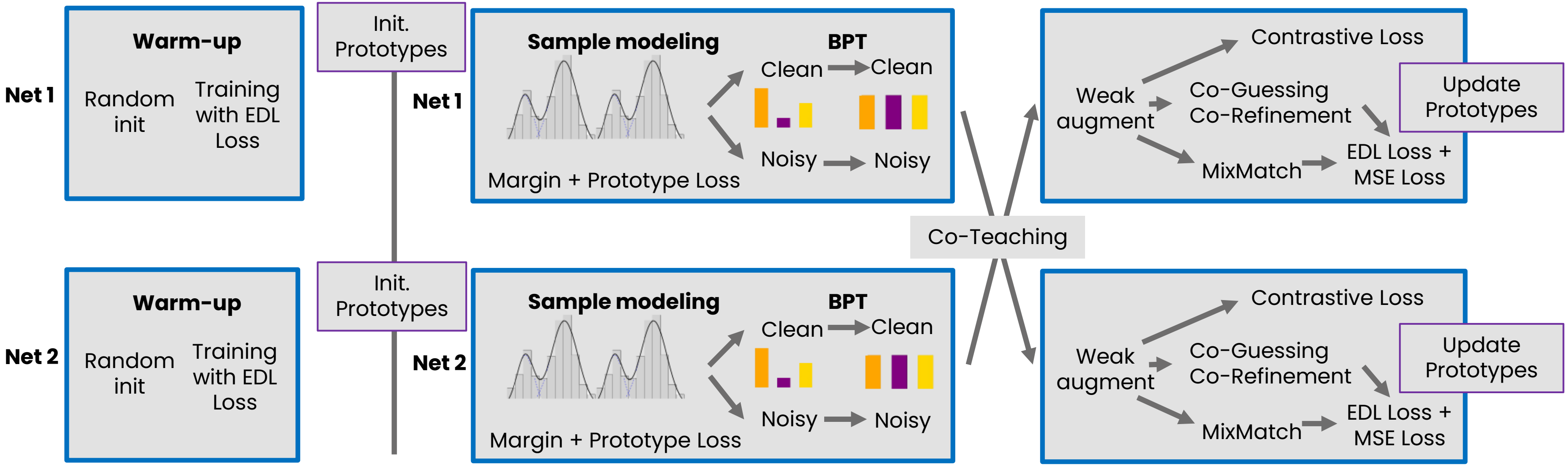
- **Positive examples** are pulled together.
- **Negative examples** are pushed apart.



PLR Loss pipeline[2] .

[1]Zhang, Q., Jin, G., Zhu, Y., Wei, H. and Chen, Q. (2024), 'Bpt-plr: A balanced partitioning and training framework with pseudo-label relaxed contrastive loss for noisy label learning', Entropy 26(7).
[2]https://www.v7labs.com/blog/contrastive-learning-guide

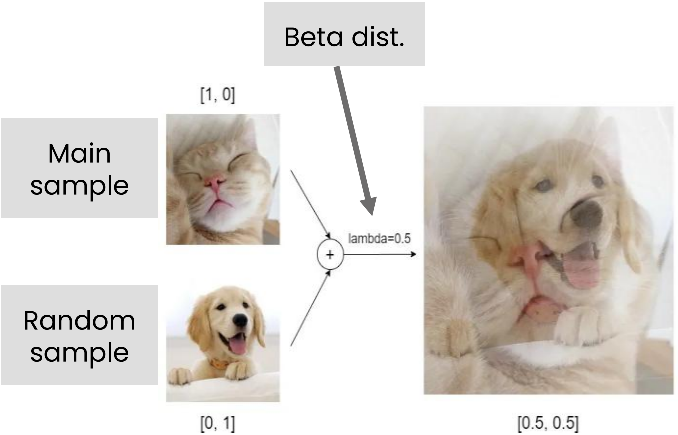
3.4. CONTRASTIVE LEARNING



3.5. VACUITY MIXMATCH

Motivation: Reduce epistemic uncertainty.

Standard MixMatch[1]: A Beta distribution weights the MixUp augmentations.



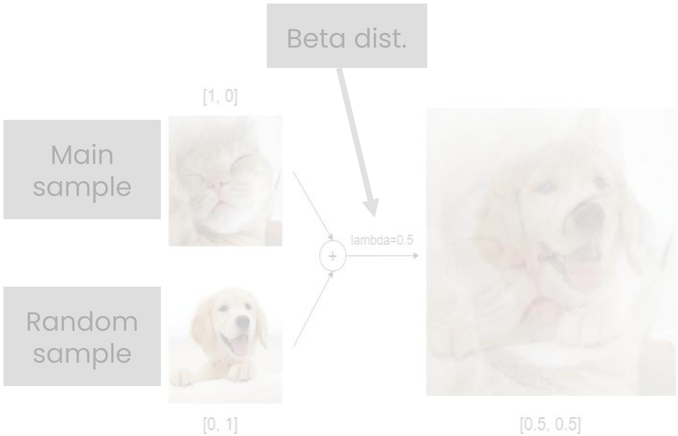
[1]Boost any Image classifier - Mixup Augmentation | Medium
[2]Nagarajan, B., Marques, R., Aguilar, E. and Radeva, P. (2024), 'Bayesian dividemix++ for enhanced learning with noisy labels', Neural Networks 172, 106122.

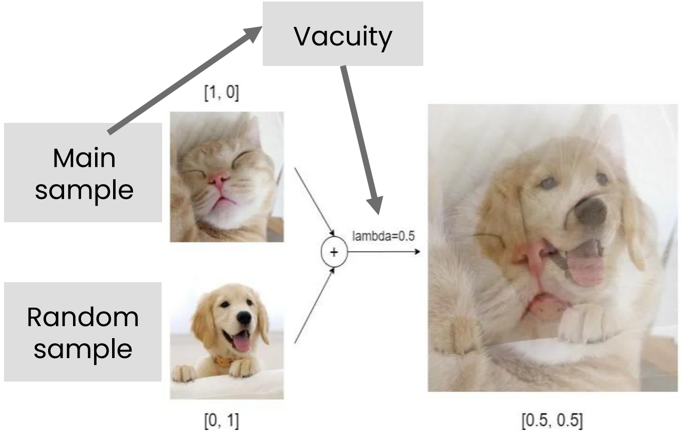
3.5. VACUITY MIXMATCH

Motivation: Reduce epistemic uncertainty.

Standard MixMatch[1]: A Beta distribution weights the MixUp augmentations.

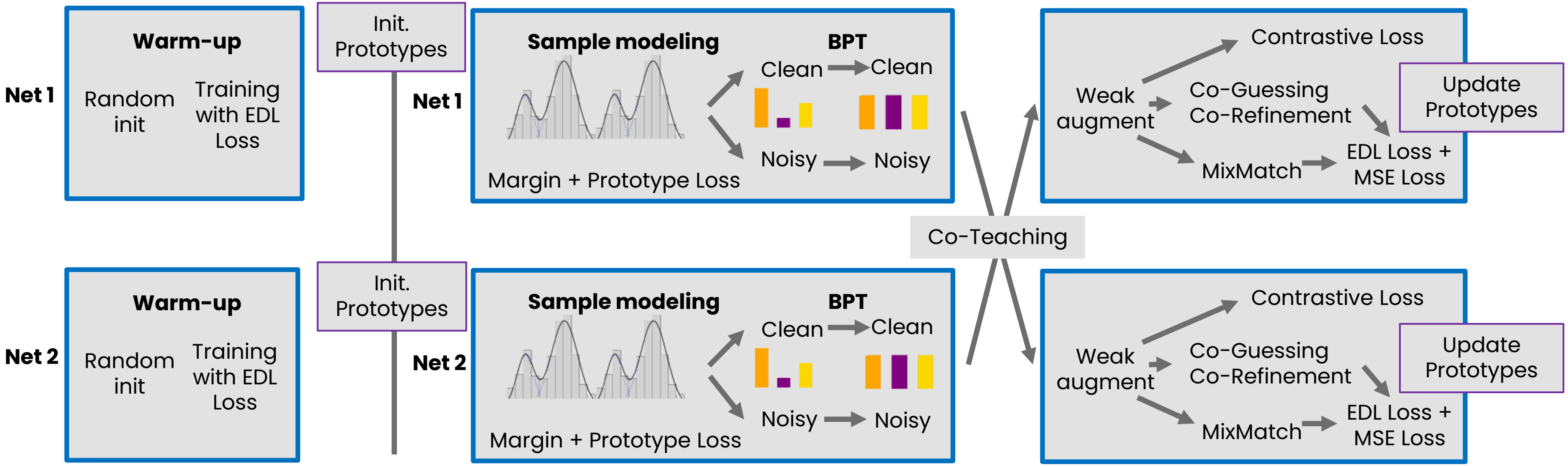
Vacuity MixMatch[2]: Network uncertainty weights the MixUp augmentations.



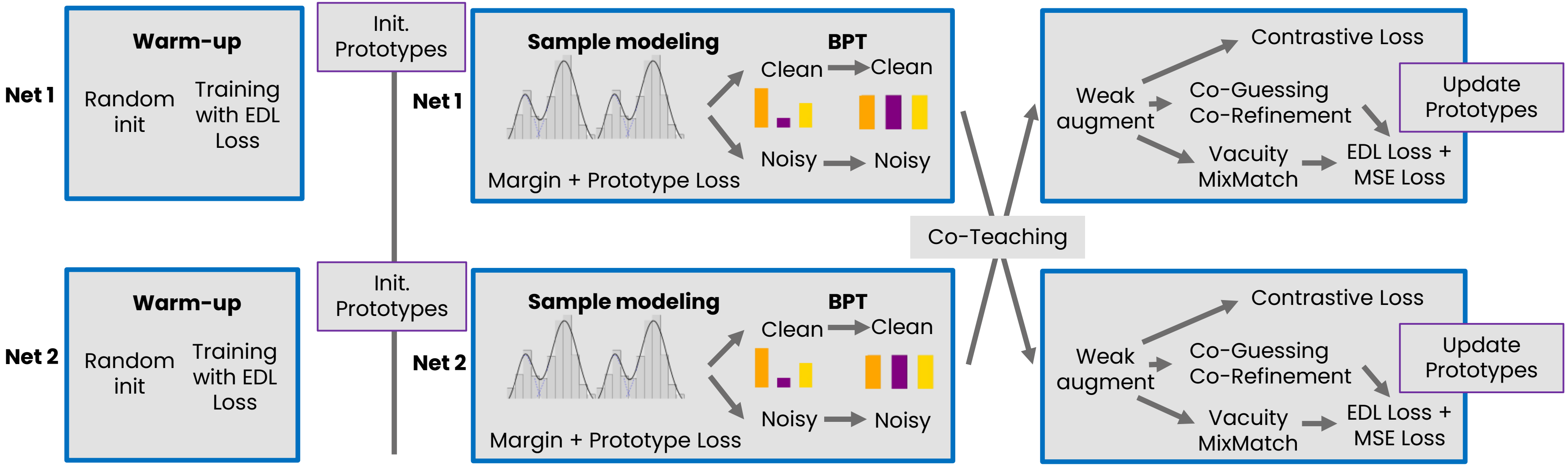


[1] Boost any Image classifier - Mixup Augmentation | Medium
[2] Nagarajan, B., Marques, R., Aguilar, E. and Radeva, P. (2024), 'Bayesian dividemix++ for enhanced learning with noisy labels', Neural Networks 172, 106122.

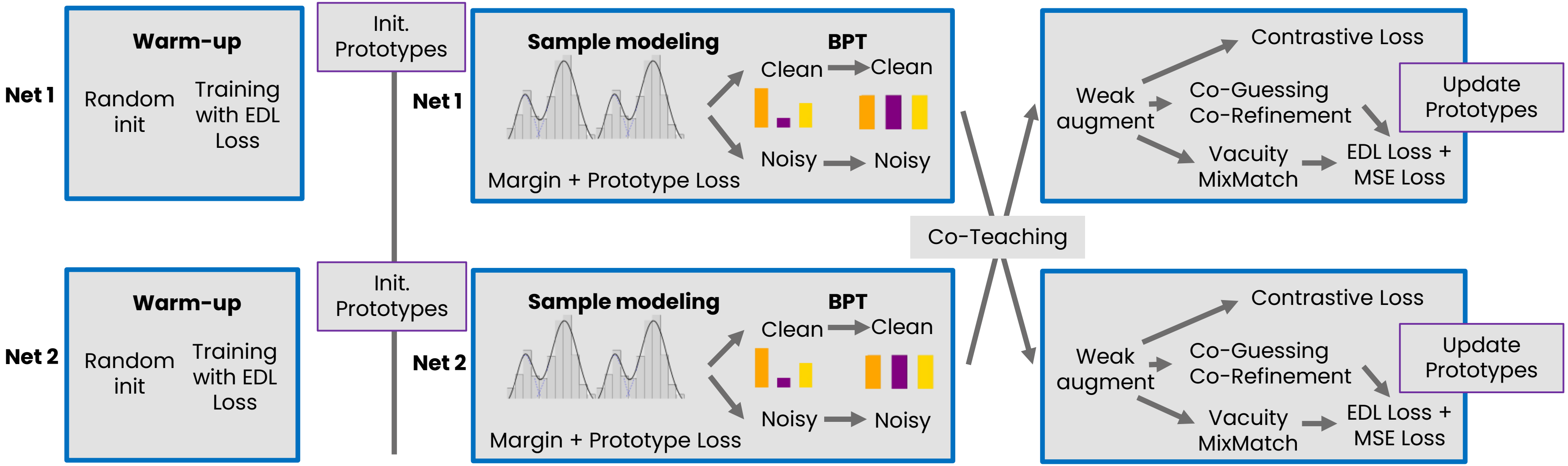
3.5. VACUITY MIXMATCH



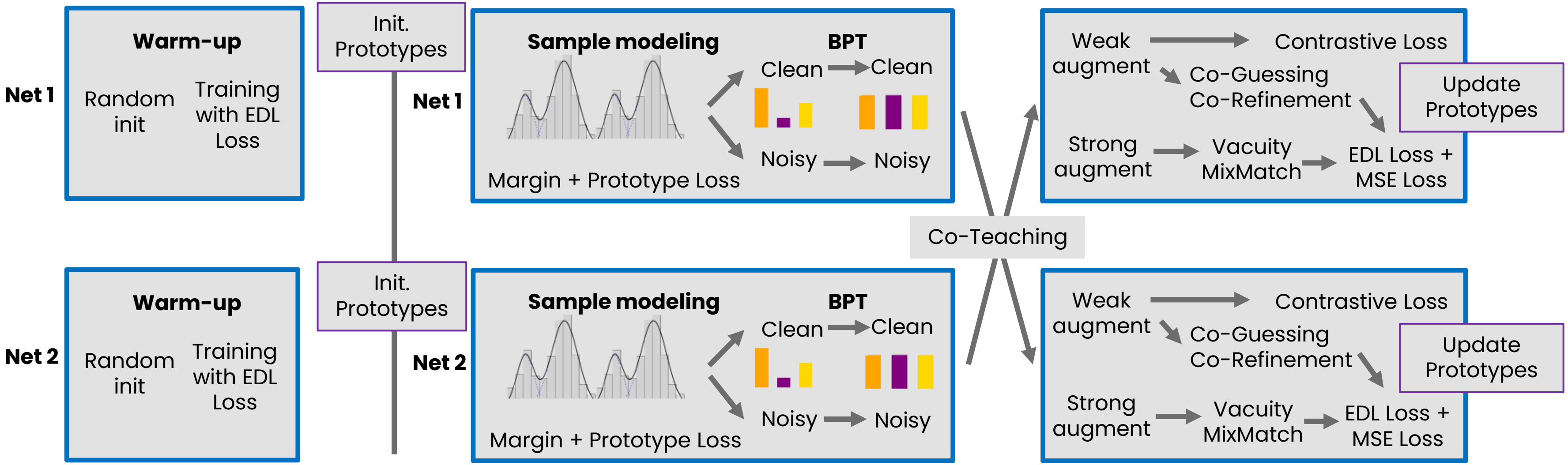
3.5. VACUITY MIXMATCH



3.6. STRONG AUGMENTATIONS

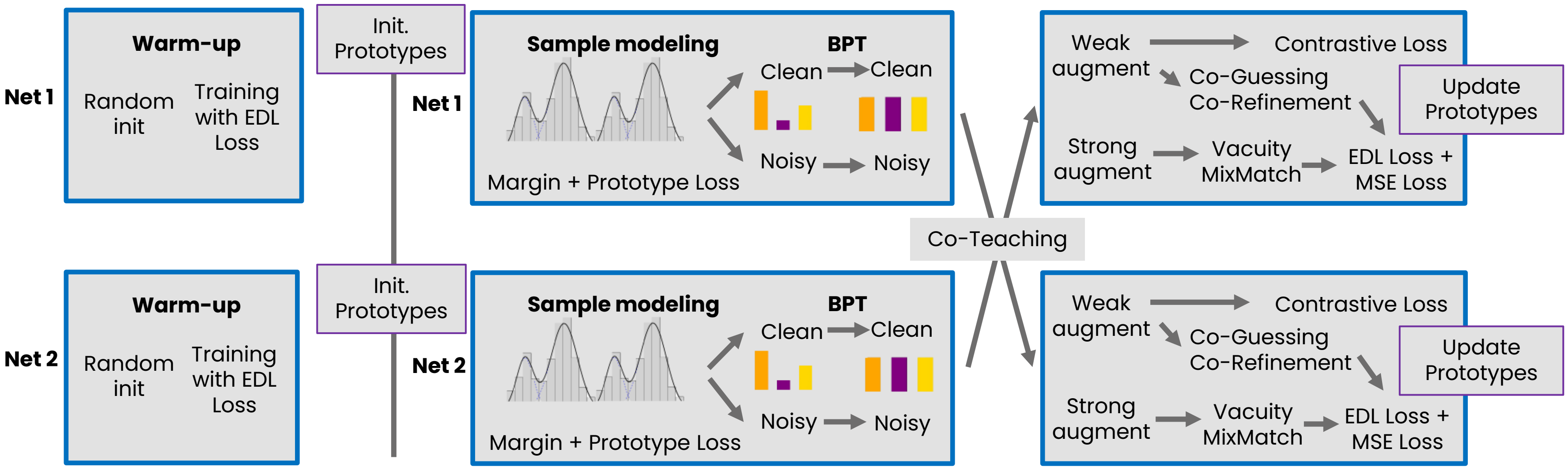


3.6. STRONG AUGMENTATIONS



3.7. PRETRAINED WEIGHTS

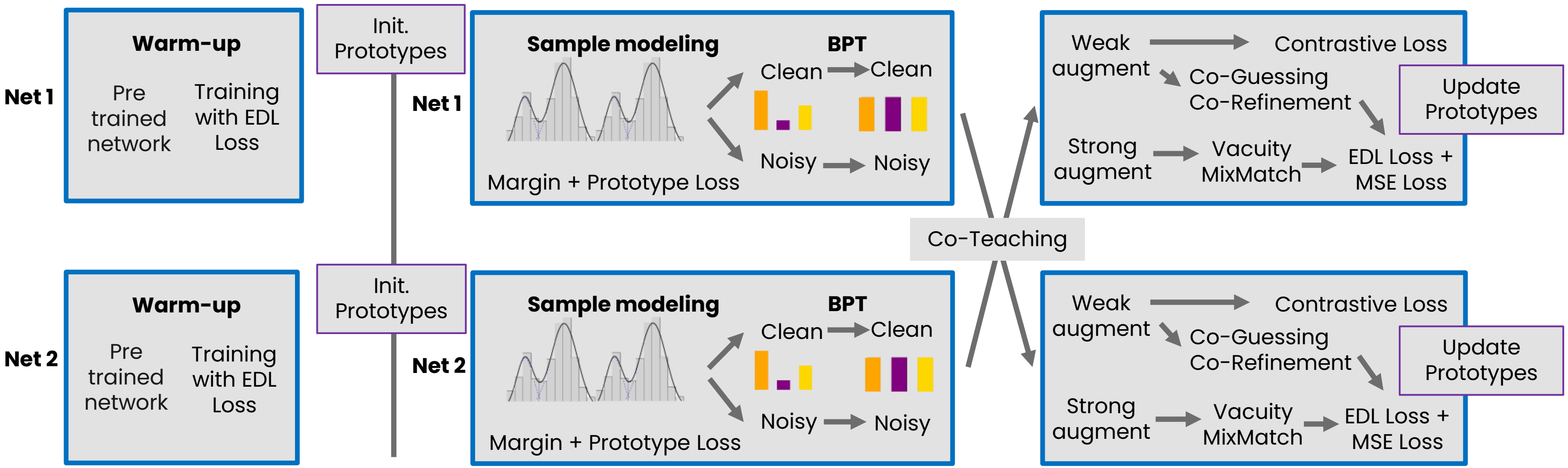
Motivation: Further mitigate noise memorization in the WarmUp stage.



[1]Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A. M. and Litany, O. (2022), Contrast to divide: Self-supervised pre-training for learning with noisy labels, in '2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)', IEEE, p. 387–397.
[2]GitHub - vturrisi/solo-learn: a library of self-supervised methods for visual representation learning powered by Pytorch Lightning

3.7. PRETRAINED WEIGHTS

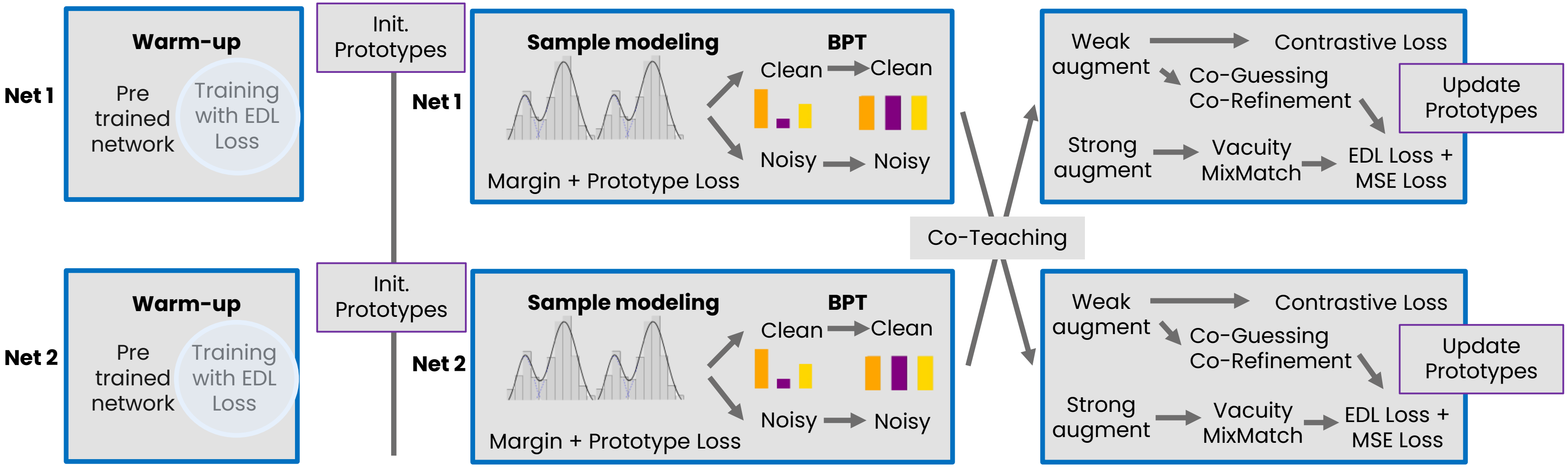
Motivation: Further mitigate noise memorization in the WarmUp stage.



SimCLR[1] pretrained weights (self-supervised contrastive learning[2]).

[1]Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A. M. and Litany, O. (2022), Contrast to divide: Self-supervised pre-training for learning with noisy labels, in '2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)', IEEE, p. 387–397.
[2]GitHub - vturrisi/solo-learn: a library of self-supervised methods for visual representation learning powered by Pytorch Lightning

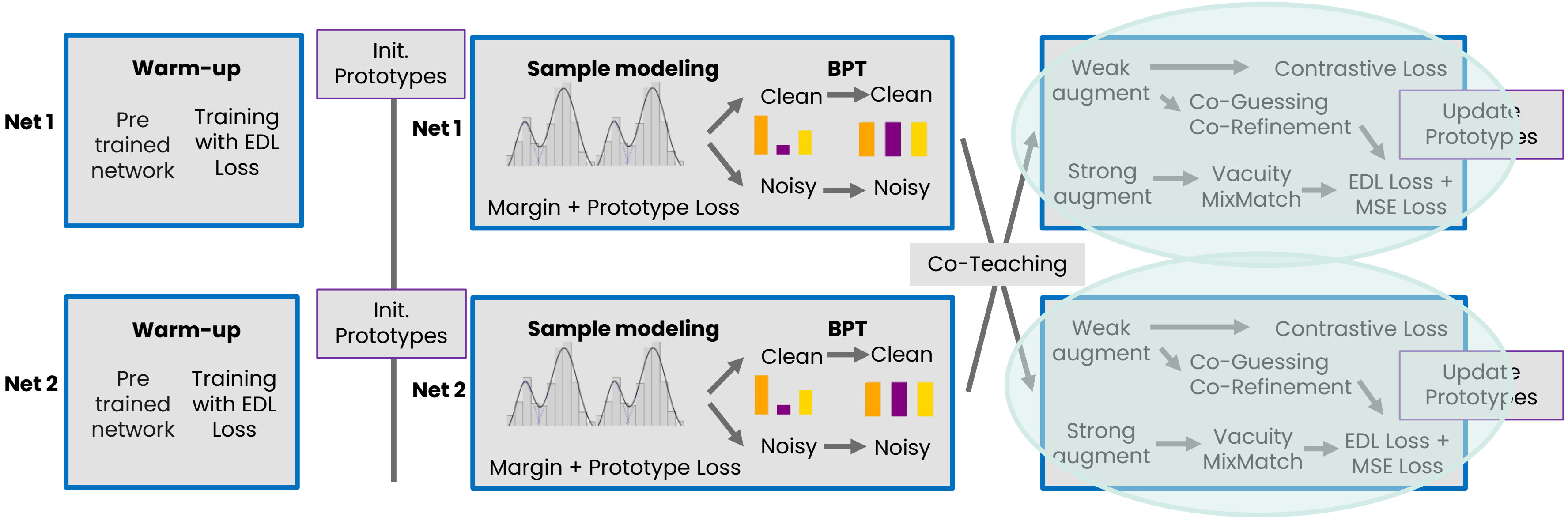
3.8. LOSS COMPONENTS



WarmUp Loss (EDL Loss):

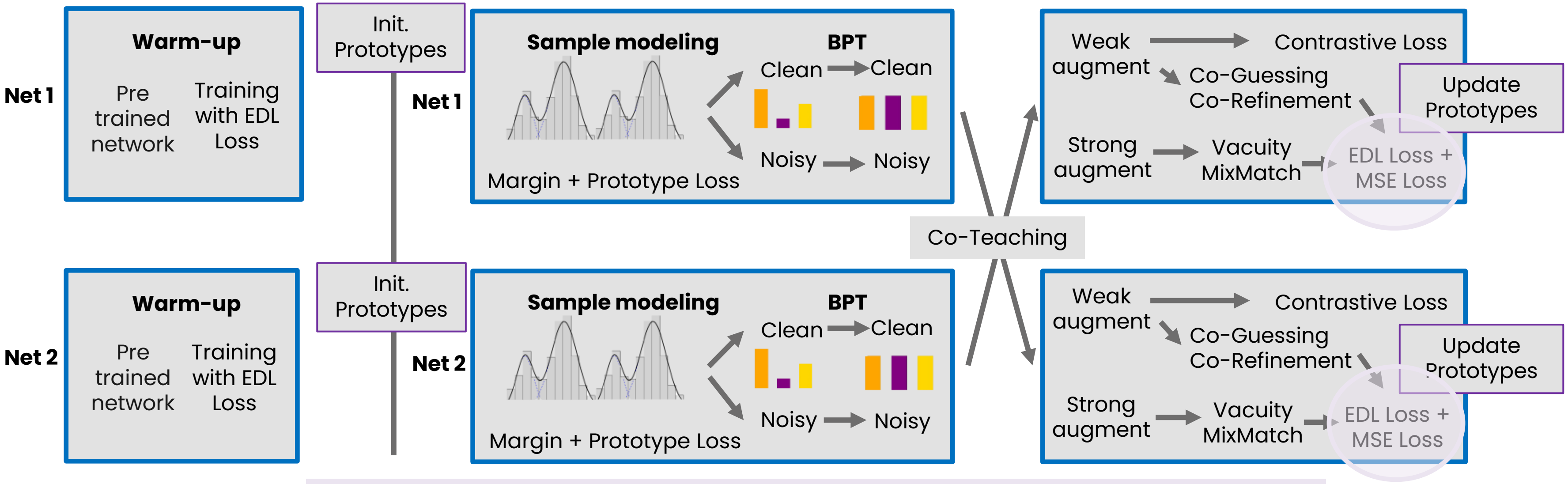
$$L_{warmup} = \frac{1}{N} \sum_{x \in D} \sum_{i} y_i(\log(S) - \log(\alpha_i)) + \eta \, KL \, [Dir(p|\alpha), Dir(p|1)]$$

3.8. LOSS COMPONENTS



$$L = L_{sup} + \lambda_{unsup} L_{unsup} + \lambda_{contr} L_{contr} + \lambda_{prior} L_{prior}$$

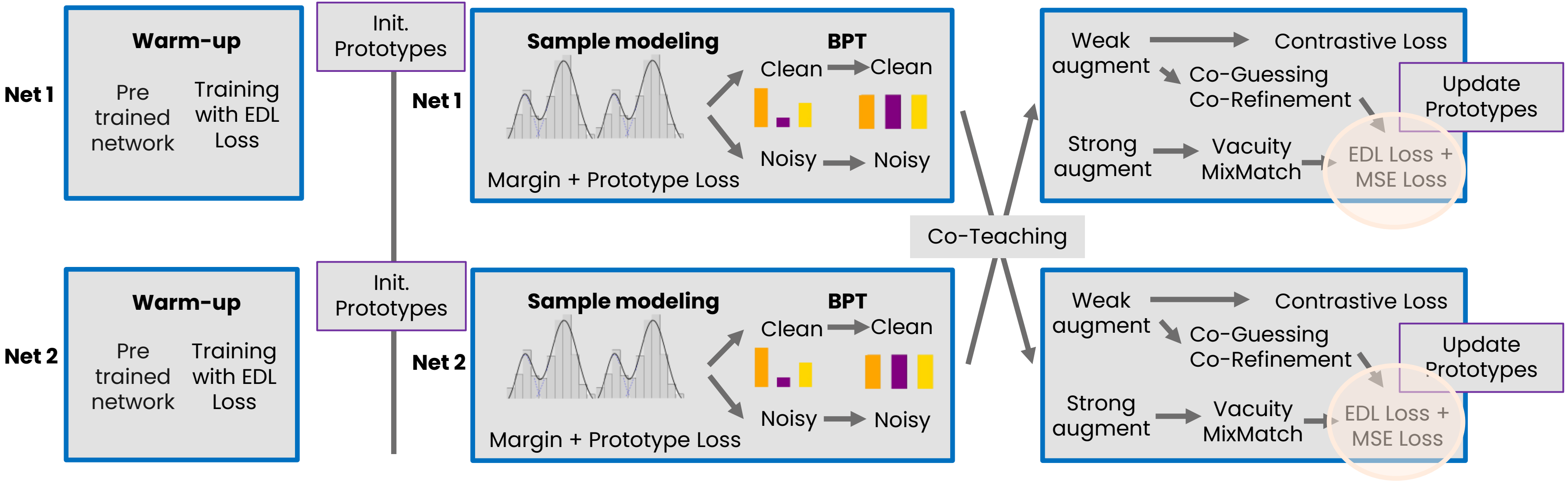
3.8. LOSS COMPONENTS



1. Supervised Loss (EDL Loss):

$$L_{sup} = \frac{1}{|X|} \sum_{x \in X} \sum_{i} y_i (\log(S) - \log(\alpha_i)) + \eta\, KL\, [Dir(p|\alpha), Dir(p|1)]$$

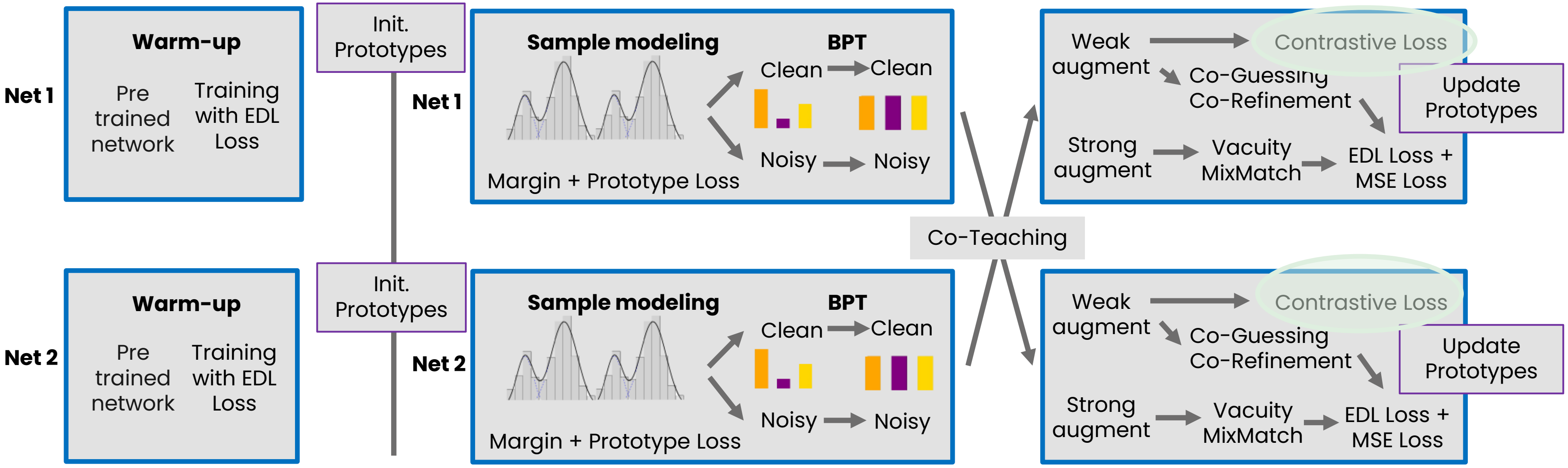
3.8. LOSS COMPONENTS



2. Unsupervised Loss (MSE):

$$L_{unsup} = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2$$

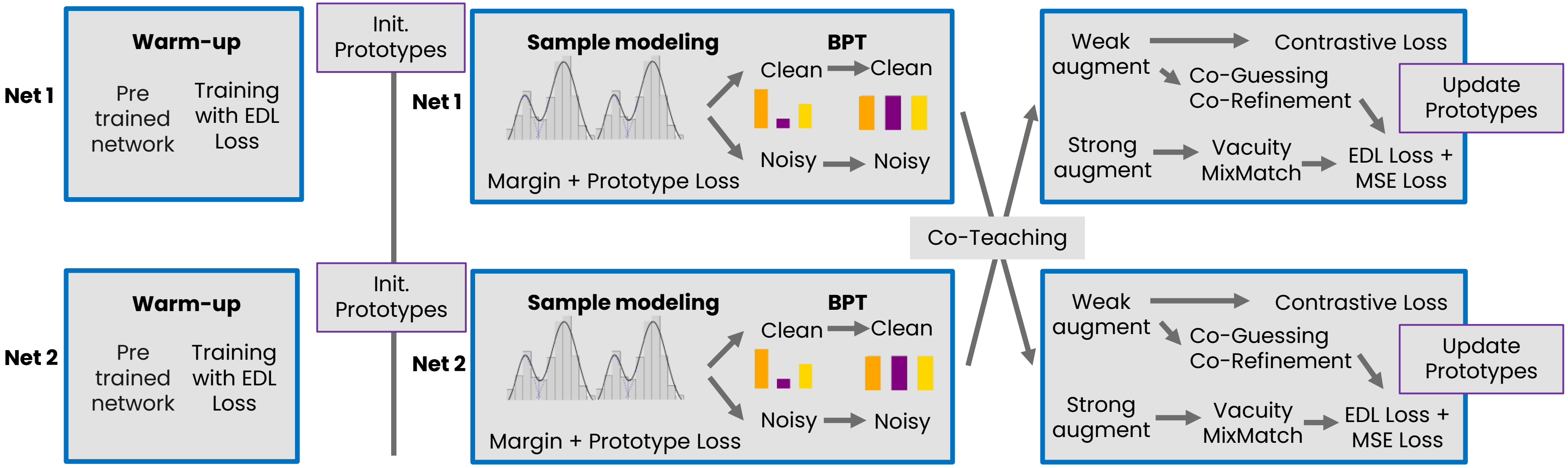
3.8. LOSS COMPONENTS



3. Contrastive Loss (PLR):

$$L_{contr} = -\frac{1}{2N}\sum_{i=1}^{N}\log\left(\frac{\exp(z_i^t\ z_{i+N})}{\sum_{j\neq i}\exp(z_i^t\ z_j)}\right) - \frac{1}{2N}\sum_{i=N+1}^{2N}\log\left(\frac{\exp(z_i^t\ z_{i-N})}{\sum_{j\neq i}\exp(z_i^t\ z_j)}\right)$$

3.8. LOSS COMPONENTS



4. Prior regularization (KL divergence):

$$L_{prior} = \sum_{i=1}^{C} \pi_c \, \log\left(\frac{\pi_c}{p_c}\right), \; \pi_c = \frac{1}{C}$$

1. INTRODUCTION
2. THEORETICAL FRAMEWORK
3. PROPOSED MODEL
4. **RESULTS AND ANALYSIS**
5. CONCLUSIONS

4.1. RESULTS

- Benchmark datasets: **CIFAR-100, CIFAR-100N**.
- **Symmetric**, **IDN** and **real noise**.
- Architecture backbone: **Resnet18**.
- WarmUp: **5** epochs.
- Total training: **200** epochs.

Method	**20% Sym.**	**50% Sym.**	**80% Sym.**
DivideMix (Li et al., 2020)	77.30	74.60	60.20
LongReMix (Cordeiro et al., 2023)	78.61	75.87	62.24
BPT-PLR (Zhang et al., 2024)	78.85	78.02	69.31
DPC (Zong et al., 2024)	**81.0**	78.5	66.4
CCLM (Tatjer et al., 2024)	80.2	76.6	67.1
ULC (Huang et al., 2022)	77.3	74.9	61.2
Bayesian DivideMix++ (Nagarajan et al., 2024)	80.02	78.31	70.01
Ours	79.03	**78.72**	**72.92**
Difference			

Test accuracy (%) on Symmetric noise (CIFAR-100).

4.1. RESULTS

- Benchmark datasets: **CIFAR-100, CIFAR-100N**.
- **Symmetric**, **IDN** and **real noise**.
- Architecture backbone: **Resnet18**.
- WarmUp: **5** epochs.
- Total training: **200** epochs.

Method	20% Sym.	50% Sym.	80% Sym.
DivideMix (Li et al., 2020)	77.30	74.60	60.20
LongReMix (Cordeiro et al., 2023)	78.61	75.87	62.24
BPT-PLR (Zhang et al., 2024)	78.85	78.02	69.31
DPC (Zong et al., 2024)	**81.0**	78.5	66.4
CCLM (Tatjer et al., 2024)	80.2	76.6	67.1
ULC (Huang et al., 2022)	77.3	74.9	61.2
Bayesian DivideMix++ (Nagarajan et al., 2024)	80.02	78.31	70.01
Ours	79.03	**78.72**	**72.92**
Difference	+1.73	+4.12	+12.72

Test accuracy (%) on Symmetric noise (CIFAR-100).

4.1. RESULTS

- Benchmark datasets: **CIFAR-100, CIFAR-100N**.
- **Symmetric**, **IDN** and **real noise**.
- Architecture backbone: **Resnet18**.
- WarmUp: **5** epochs.
- Total training: **200** epochs.

Method	**20% Sym.**	**50% Sym.**	**80% Sym.**
DivideMix (Li et al., 2020)	77.30	74.60	60.20
LongReMix (Cordeiro et al., 2023)	78.61	75.87	62.24
BPT-PLR (Zhang et al., 2024)	78.85	78.02	69.31
DPC (Zong et al., 2024)	**81.0**	78.5	66.4
CCLM (Tatjer et al., 2024)	80.2	76.6	67.1
ULC (Huang et al., 2022)	77.3	74.9	61.2
Bayesian DivideMix++ (Nagarajan et al., 2024)	80.02	78.31	70.01
Ours	79.03	**78.72**	**72.92**
Difference	-2.0	+0.2	+2.9

Test accuracy (%) on Symmetric noise (CIFAR-100).

4.1. RESULTS

- Benchmark datasets: **CIFAR-100, CIFAR-100N**.
- **Symmetric**, **IDN** and **real noise**.
- Architecture backbone: **Resnet18**.
- WarmUp: **5** epochs.
- Total training: **200** epochs.

Method	**20% IDN**	**40% IDN**	**60% IDN**
DISC (Li et al., 2023)	80.12	78.44	69.57
SL (Kim et al., 2024)	**80.94**	78.60	-
SplitNet (Kim et al., 2025)	80.45	76.97	70.20
Ours	78.85	**78.84**	**77.44**
Difference			

Test accuracy (%) on instance-dependent noise (CIFAR-100).

4.1. RESULTS

- Benchmark datasets: **CIFAR-100, CIFAR-100N**.
- **Symmetric**, **IDN** and **real noise**.
- Architecture backbone: **Resnet18**.
- WarmUp: **5** epochs.
- Total training: **200** epochs.

Method	**20% IDN**	**40% IDN**	**60% IDN**
DISC (Li et al., 2023)	80.12	78.44	69.57
SL (Kim et al., 2024)	**80.94**	78.60	-
SplitNet (Kim et al., 2025)	80.45	76.97	70.20
Ours	78.85	**78.84**	**77.44**
Difference	-2.09	+0.24	+7.24

Test accuracy (%) on instance-dependent noise (CIFAR-100).

4.1. RESULTS

- Benchmark datasets: **CIFAR-100, CIFAR-100N**.
- **Symmetric**, **IDN** and **real noise**.
- Architecture backbone: **Resnet18**.
- WarmUp: **5** epochs.
- Total training: **200** epochs.

Method	CIFAR-100N
CCLM (Tatjer et al., 2024)	70.6
Ours	**70.78**

Test accuracy (%) on real noise (CIFAR-100N).

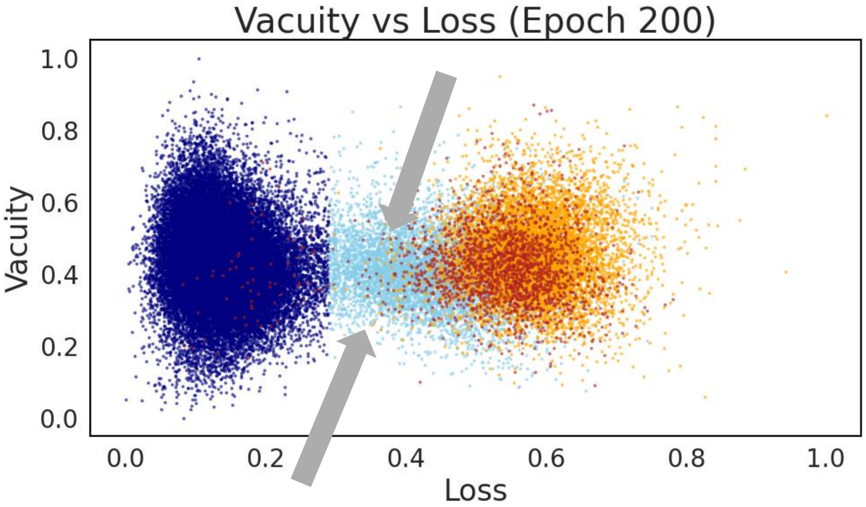
4.2. ABLATION STUDIES

Method	80% Sym.
w/o Pretrained	66.30
w/o Contrastive Loss	72.57
w/o Vacuity MixMatch	72.79
Ours (Full Method)	**72.92**

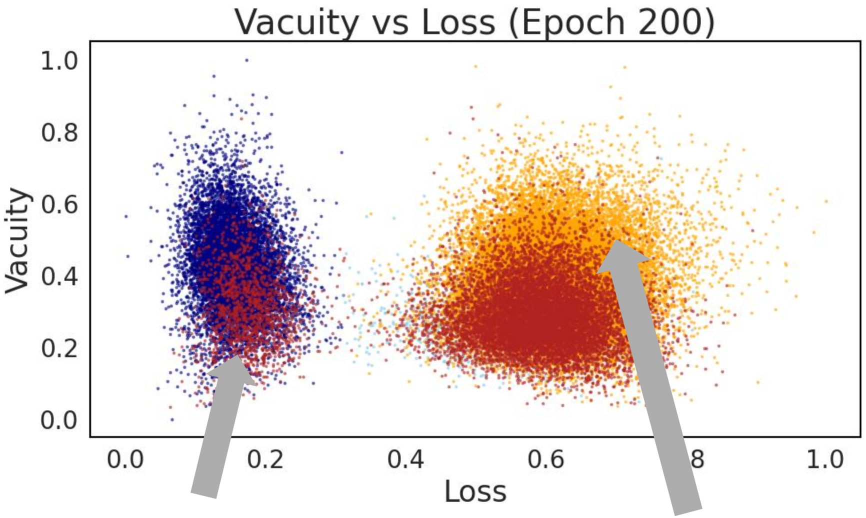
Ablation studies. Test accuracy (%) on 80% symmetric noise (CIFAR-100).

4.3. ANALYSIS

Vacuity (20% Sym Noise)



Vacuity (80% Sym Noise)



Incorrect clean samples:
middle loss values.

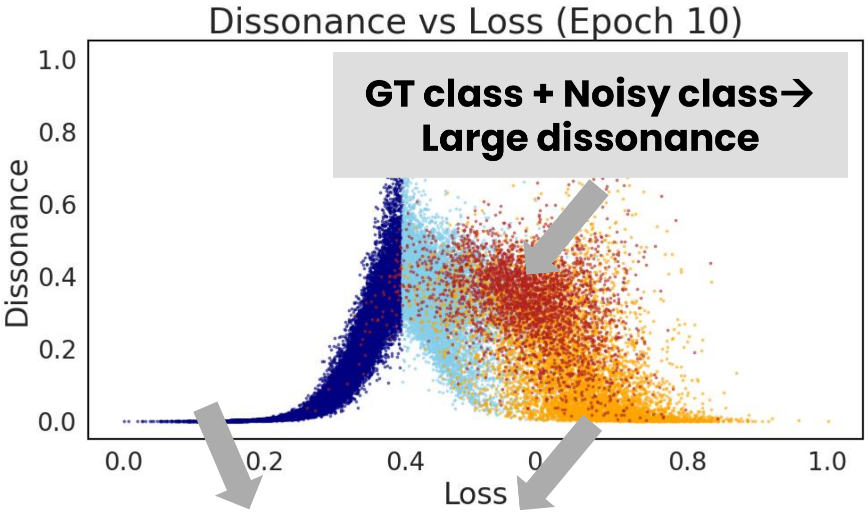
Clean & Correct
Clean & Incorrect

Noisy samples
detected as clean.

High-vacuity →
correct prediction.

4.3. ANALYSIS

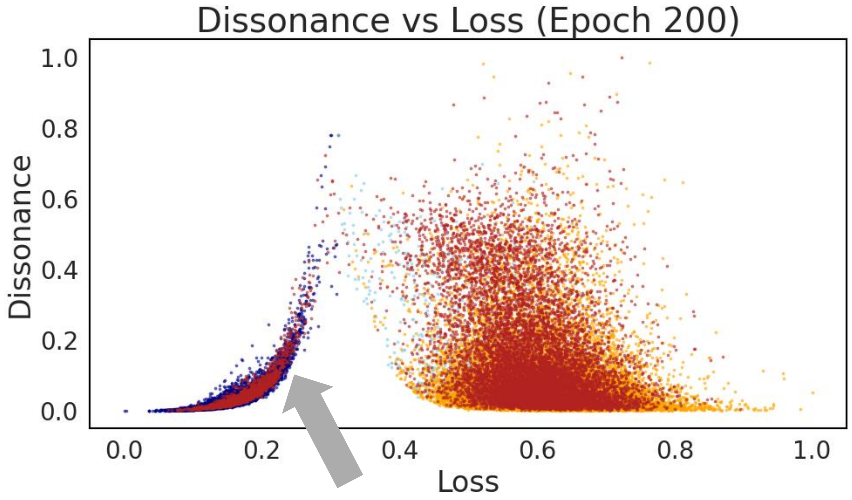
Dissonance (20% Sym Noise)



GT class + Noisy class→ Large dissonance

Low dissonance → correct prediction.

Dissonance (80% Sym Noise)



No difference of clean vs noisy samples.

1. INTRODUCTION
2. THEORETICAL FRAMEWORK
3. PROPOSED MODEL
4. RESULTS AND ANALYSIS
5. CONCLUSIONS

5.1. CONCLUSIONS

- Proposed a **Dirichlet-based approach** to Learning with Noisy Labels, building on DivideMix.
- Enhanced with self-supervised **pre training**, **contrastive learning**, and improved clean **sample selection**.

- Achieved **SOTA performance** and **outperformed** all competitors in **high-noise** settings.
- Provided **analysis** on the **evolution of uncertainty**.

- Our model highlights the **benefits** of incorporating **EDL** and **contrastive learning** in the LNL field.

Future Work: Understanding the relationship between EDL uncertainty measures and the informativeness or relevance of training samples.

Thank you!

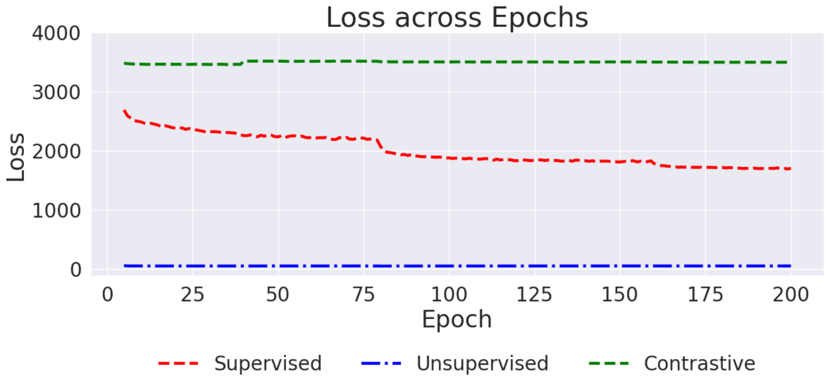
EVIDENTIAL DEEP LEARNING FOR HIGH-CONFIDENCE SAMPLE SELECTION IN NOISY LABEL LEARNING

Marc Pascual Roig

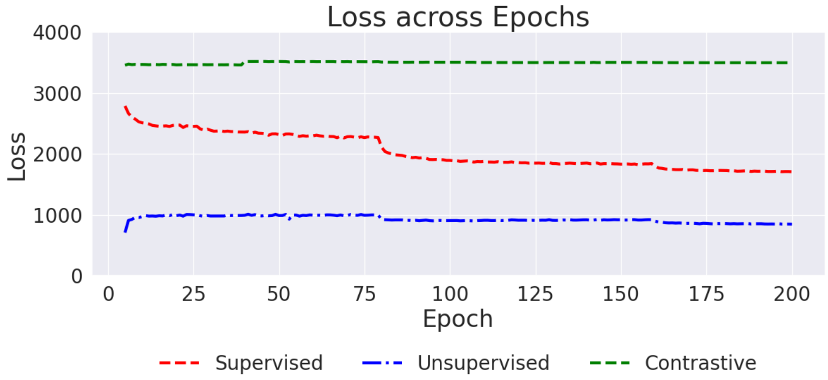
Thesis supervisors:
Prof. Petia Radeva
Dr. Bhalaji Nagarajan

Master's Thesis (Master's Degree in Artificial Intelligence)
UB - UPC - URV

LOSS CURVES



Loss across Epochs
4000
3000
2000
1000
0
Loss
0
25
50
75
100
125
150
175
200
Epoch
Supervised
Unsupervised
Contrastive



Loss across Epochs
4000
3000
2000
1000
0
Loss
0
25
50
75
100
125
150
175
200
Epoch
Supervised
Unsupervised
Contrastive

HYPERPARAMETERS

Hyperparameter	Value
λ_{contr}	1
λ_{prior}	1
α (Beta distribution)	4
Temperature	0.5
GMM threshold (τ)	0.5
Number of epochs	200
Batch size	64
Learning rate (epoch 0–79)	0.02
Learning rate (epoch 80–159)	0.002
Learning rate (epoch 160–200)	0.0002
Optimizer	SGD
Momentum	0.9
Weight decay	5×10^{-4}

HYPERPARAMETERS

Noise Level	$\lambda_{unsupervised}$
20%	25
40%	150
40.2% (CIFAR-100N)	150
50%	150
60%	150
80%	500

Contrastive Loss hyperparameter	**Value**
Feature space	128
k	1,2,3

TUNING OF EDL LOSS

Method	Activation	Weight	20%	50%	80%
CE	Softmax	–	75.40	74.63	**63.76**
EDL	Softplus	0.001	–	–	–
EDL	Softplus	0.01	**78.27**	74.58	54.85
EDL	Softplus	0.1	77.11	75.77	61.63
EDL	Exp	0.001	76.38	75.36	61.71
EDL	Exp	0.01	76.21	**75.80**	63.40
EDL	Exp	0.1	–	–	–

Method	Activation	Weight	20%	50%	80%
CE	Softmax	–	78.36	76.93	**66.02**
EDL	Exp	0.01	**79.09**	**77.76**	64.64

SUSTAINABILITY AND ETHICS

- **Environmental impact:** Estimated **81.6 kg CO_2** emitted over ~2400 GPU hours using an RTX 2080 Ti during 5 months of experimentation.

- **Economic Impact**: The project had no direct financial costs. Our method reduces future annotation costs via robust uncertainty-aware learning.

- **Social impact**: reducing manual data cleaning and promoting more reliable AI systems.

The proposed method uses **self-supervised learning** and **uncertainty estimation (EDL)** to mitigate human biases and improve fairness.