NP-Match: When Neural Processes meet Semi-Supervised Learning

Jianfeng Wang[1], Thomas Lukasiewicz[1], Daniela Massiceti[2], Xiaolin Hu[3], Vladimir Pavlovic[4], Alexandros Neophytou[5]

University of Oxford[1], Microsoft Research[2], Tsinghua University[3], Rutgers University[4], Microsoft[5]

Introduction

Semi-supervised image classification has been widely explored in recent years. Current SOTA methods are deterministic, which have achieved promising results. In contrast, progress on probabilistic approaches in this field lags behind:

- There are only few studies on this task.

- Monte Carlo dropout becomes the only option for implementing the probabilistic model.

Introduction

Our contribution:

- We propose NP-Match, which adjusts Neural processes (NPs) to SSL, and explore its use in semi-supervised large-scale image classification.

- We propose a new uncertainty-guided skew-geometric Jensen-Shannon (JS) divergence for optimizing NP-Match.

- We show that NP-Match achieves competitive results on four public benchmarks. We also show that NP-Match estimates uncertainty faster than the MC-dropout-based probabilistic model, which can improve the training and the test efficiency

Neural Processes

Formally, given a probability space (Ω, Σ, Π) and an index set X , **a stochastic process** can be written as {F (x, ω) : x ∈ X }, where F (· , ω) is a sample function mapping X to another space Y for any point ω ∈ Ω. For each finite sequence $x_{1:n}$, a marginal joint distribution can be defined on the function values F (x_1, ω), F (x_2, ω), . . . , F (x_n, ω).

$$p(y_{1:n} \mid x_{1:n}) = \int \pi(\omega) p(y_{1:n} \mid F(\cdot, \omega), x_{1:n}) d\,\mu(\omega)$$

NPs parameterize the function F (· , ω) with a high-dimensional random vector *z* sampled from a multivariate Gaussian distribution and a neural network g(·).

$$p(y_{1:n} \mid x_{1:n}) = \int \pi(z) p(y_{1:n} \mid g(x_{1:n}, z), x_{1:n}) d\,\mu(z)$$

NP-Match

- NP Model for Semi-Supervised Image Classification
- NP-Match Pipeline
- Uncertainty-Guided Skew-Geometric JS Divergence

NP-Match

- NP Model for Semi-Supervised Image Classification

$$p(y_{1:n} \mid x_{1:n}) = \int \pi(z) p(y_{1:n} \mid g(x_{1:n}, z), x_{1:n}) d\mu(z)$$

categorical distribution

Parameterize the categorical distribution by probability vectors from a classifier that contains a weight matrix (W) and a softmax function (Φ):

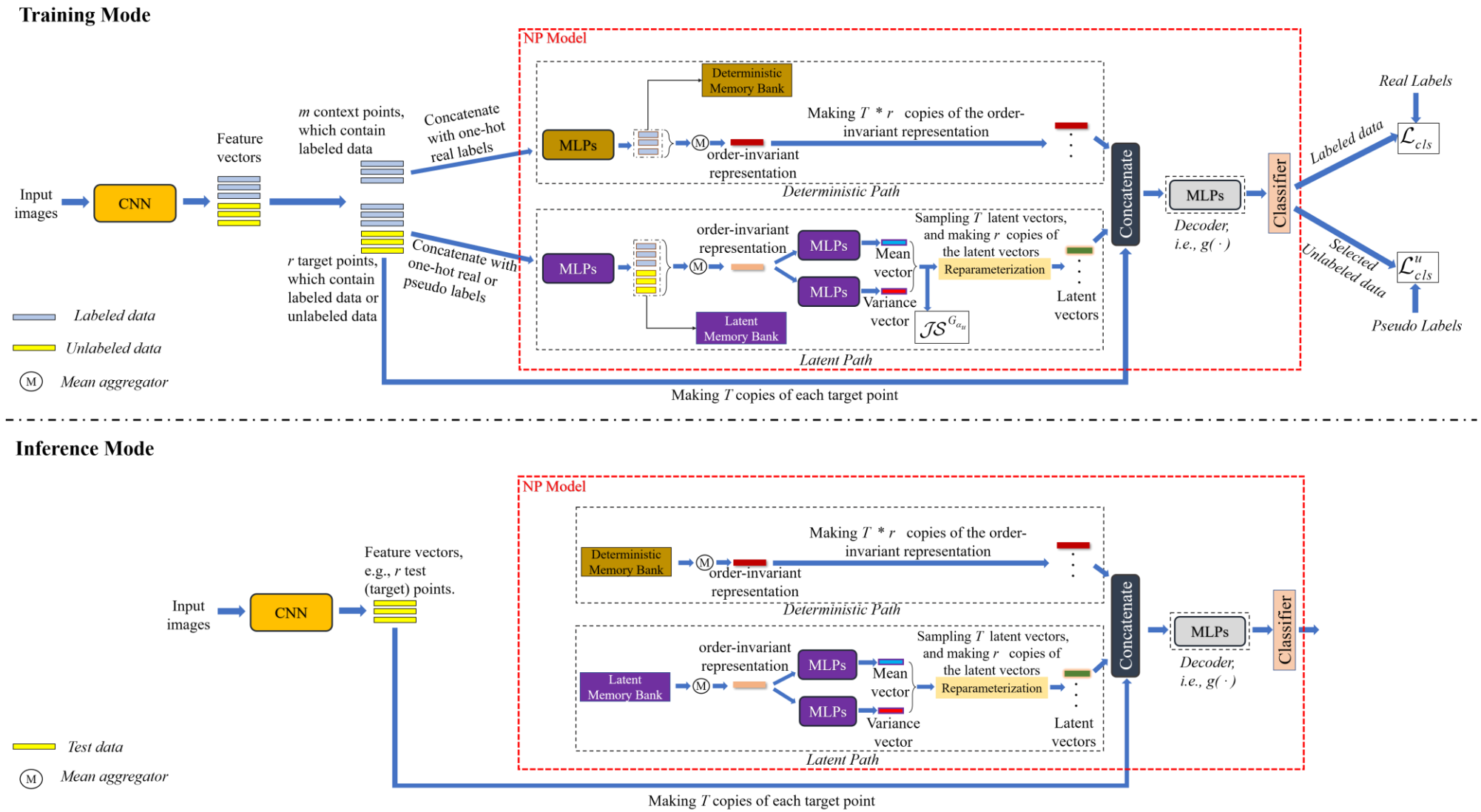
$$p(y_{1:n} \mid g(x_{1:n}, z), x_{1:n}) = Cat(\phi(Wg(x_{1:n}, z)))$$

Evidence Lower Bound (ELBO):

$$\log p(y_{1:n} \mid x_{1:n}) \geq \mathbb{E}_{q(z \mid x_{m+1\,:\,m+r}, y_{m+1\,:\,m+r})}[\sum_{i=m+1}^{m+r} \log p(y_i \mid z, x_i) - \log \frac{q(z \mid x_{m+1\,:\,m+r}, y_{m+1\,:\,m+r})}{q(z \mid x_{1\,:\,m}, y_{1\,:\,m})}] + const$$

NP-Match

- NP-Match Pipeline



NP-Match

- Uncertainty-Guided Skew-Geometric JS Divergence

Definition 1. *Let (Ω, Σ) be a measurable space, where Ω denotes the sample space, and Σ denotes the σ-algebra of measurable events. P and Q are two probability measures defined on the measurable space. Concerning a positive measure*[5]*, which is denoted as μ, the uncertainty-guided skew-geometric JS divergence ($JS^{G_{\alpha_u}}$) can be defined as:*

$$JS^{G_{\alpha_u}}(p,q) = (1-\alpha_u)\int p\, log\frac{p}{G(p,q)_{\alpha_u}}d\mu + \alpha_u \int q\, log\frac{q}{G(p,q)_{\alpha_u}}d\mu,$$

where p and q are the Radon-Nikodym derivatives of P and Q with respect to μ, the scalar $\alpha_u \in [0,1]$ is calculated based on the uncertainty, and $G(p,q)_{\alpha_u} = p^{1-\alpha_u}q^{\alpha_u} \,/\, (\int_\Omega p^{1-\alpha_u}q^{\alpha_u}d\mu)$. The dual form of $JS^{G_{\alpha_u}}$ is given by:

$$JS_*^{G_{\alpha_u}}(p,q) = (1-\alpha_u)\int G(p,q)_{\alpha_u}\, log\frac{G(p,q)_{\alpha_u}}{p}d\mu + \alpha_u \int G(p,q)_{\alpha_u}\, log\frac{G(p,q)_{\alpha_u}}{q}d\mu.$$

NP-Match

- Uncertainty-Guided Skew-Geometric JS Divergence

Theorem 1. *Given two multivariate Gaussians* $\mathcal{N}_1(\mu_1, \Sigma_1)$ *and* $\mathcal{N}_2(\mu_2, \Sigma_2)$*, the following holds:*

$$JS^{G_{\alpha_u}}(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{2}(tr(\Sigma_{\alpha_u}^{-1}((1-\alpha_u)\Sigma_1 + \alpha_u\Sigma_2)) + (1-\alpha_u)(\mu_{\alpha_u} - \mu_1)^T\Sigma_{\alpha_u}^{-1}(\mu_{\alpha_u} - \mu_1) +$$
$$\alpha_u(\mu_{\alpha_u} - \mu_2)^T\Sigma_{\alpha_u}^{-1}(\mu_{\alpha_u} - \mu_2) + log[\frac{det[\Sigma_{\alpha_u}]}{det[\Sigma_1]^{1-\alpha_u}det[\Sigma_2]^{\alpha_u}}] - D)$$
$$JS_*^{G_{\alpha_u}}(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{2}(log[\frac{det[\Sigma_1]^{1-\alpha_u}det[\Sigma_2]^{\alpha_u}}{det[\Sigma_{\alpha_u}]}] + \alpha_u\mu_2^T\Sigma_2^{-1}\mu_2 - \mu_{\alpha_u}^T\Sigma_{\alpha_u}^{-1}\mu_{\alpha_u} + (1-\alpha_u)\mu_1^T\Sigma_1^{-1}\mu_1),$$

where $\Sigma_{\alpha_u} = ((1-\alpha_u)\Sigma_1^{-1} + \alpha_u\Sigma_2^{-1})^{-1}$ *and* $\mu_{\alpha_u} = \Sigma_{\alpha_u}((1-\alpha_u)\Sigma_1^{-1}\mu_1 + \alpha_u\Sigma_2^{-1}\mu_2)$*, D denotes the number of dimension, and* $det[\cdot]$ *represents the determinant.*

Main Results

Dataset	CIFAR-10			CIFAR-100			STL-10		
Label Amount	40	250	4000	400	2500	10000	40	250	1000
MixMatch (Berthelot et al., 2019)	36.19 (±6.48)	13.63 (±0.59)	6.66 (±0.26)	67.59 (±0.66)	39.76 (±0.48)	27.78 (±0.29)	54.93 (±0.96)	34.52 (±0.32)	21.70 (±0.68)
ReMixMatch (Berthelot et al., 2020)	9.88 (±1.03)	6.30 (±0.05)	4.84 (±0.01)	42.75 (±1.05)	**26.03** (±0.35)	**20.02** (±0.27)	32.12 (±6.24)	12.49 (±1.28)	6.74 (±0.14)
UDA (Xie et al., 2020)	10.62 (±3.75)	5.16 (±0.06)	4.29 (±0.07)	46.39 (±1.59)	27.73 (±0.21)	22.49 (±0.23)	37.42 (±8.44)	9.72 (±1.15)	6.64 (±0.17)
CoMatch (Li et al., 2021)	6.88 (±0.92)	4.90 (±0.35)	4.06 (±0.03)	40.02 (±1.11)	27.01 (±0.21)	21.83 (±0.23)	31.77 (±2.56)	11.56 (±1.27)	8.66 (±0.41)
SemCo (Nassar et al., 2021)	7.87 (±0.22)	5.12 (±0.27)	**3.80** (±0.08)	44.11 (±1.18)	31.93 (±0.33)	24.45 (±0.12)	34.17 (±2.78)	12.23 (±1.40)	7.49 (±0.29)
Meta Pseudo Labels (Pham et al., 2021)	6.93 (±0.17)	4.94 (±0.04)	3.89 (±0.07)	44.23 (±0.99)	27.68 (±0.22)	22.48 (±0.18)	34.29 (±3.29)	9.90 (±0.96)	6.45 (±0.26)
FlexMatch (Zhang et al., 2021)	4.96 (±0.06)	4.98 (±0.09)	4.19 (±0.01)	39.94 (±1.62)	26.49 (±0.20)	21.90 (±0.15)	29.15 (±4.16)	**8.23** (±0.39)	5.77 (±0.18)
UPS (Rizve et al., 2021)	5.26 (±0.29)	5.11 (±0.08)	4.25 (±0.05)	41.07 (±1.66)	27.14 (±0.24)	21.97 (±0.23)	30.82 (±2.16)	9.77 (±0.44)	6.02 (±0.28)
FixMatch (Sohn et al., 2020)	7.47 (±0.28)	**4.86** (±0.05)	4.21 (±0.08)	46.42 (±0.82)	28.03 (±0.16)	22.20 (±0.12)	35.96 (±4.14)	9.81 (±1.04)	6.25 (±0.33)
NP-Match (ours)	**4.91** (±0.04)	4.96 (±0.06)	4.11 (±0.02)	**38.91** (±0.99)	**26.03** (±0.26)	21.22 (±0.13)	**14.20** (±0.67)	9.51 (±0.37)	**5.59** (±0.24)

Table 1. Comparison with SOTA results on CIFAR-10, CIFAR-100, and STL-10. The error rates are reported with standard deviation.

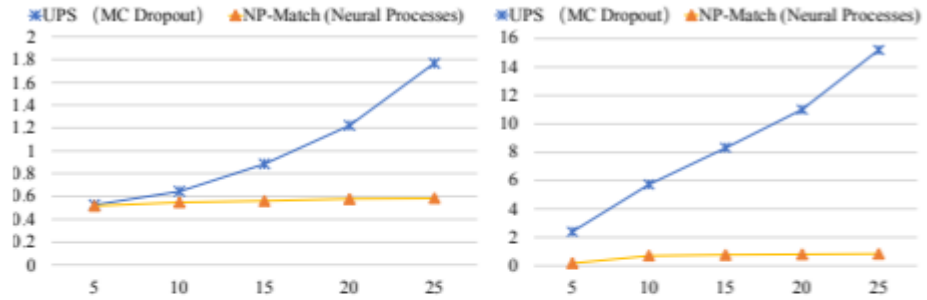
	Method	Top-1	Top-5
Deterministic Methods	FixMatch (Sohn et al., 2020)	43.66	21.80
	FlexMatch (Zhang et al., 2021)	41.85	19.48
	CoMatch (Li et al., 2021)	42.17	19.64
Probabilistic Methods	UPS (Rizve et al., 2021)	42.69	20.23
	NP-Match	**41.78**	**19.33**

Table 3. Error rates of SOTA methods on ImageNet.

Main Results

Dataset	CIFAR-10			STL-10		
Label Amount	40	250	4000	40	250	1000
UPS (MC Dropout)	7.96	7.02	**5.82**	17.23	9.65	5.69
NP-Match	**7.23**	**6.85**	5.89	**12.45**	**8.72**	**5.28**

Table 2. Expected UCEs (%) of the MC-dropout-based model (i.e., UPS (Rizve et al., 2021)) and of NP-Match on the test sets of CIFAR-10 and STL-10.



(a) WRN-28-2 on CIFAR-10 (b) WRN-28-8 on CIFAR-100

Figure 3. Time consumption of estimating uncertainty for the MC-dropout-based model (i.e., UPS (Rizve et al., 2021)) and NP-Match. The horizontal axis refers to the number of predictions used for the uncertainty quantification, and the vertical axis indicates the time consumption (sec).

Ablation Studies

Dataset	CIFAR-10			CIFAR-100			STL-10		
Label Amount	40	250	4000	400	2500	10000	40	250	1000
NP-Match with KL	5.32 (±0.06)	5.20 (±0.02)	4.36 (±0.03)	39.15 (±1.53)	26.48 (±0.23)	21.51 (±0.17)	14.67 (±0.38)	9.92 (±0.24)	6.21 (±0.23)
NP-Match with $JS_*^{G_{\alpha_u}}$	4.93 (±0.02)	**4.87** (±0.03)	4.19 (±0.04)	**38.67** (±1.29)	26.24 (±0.17)	21.33 (±0.10)	14.45 (±0.55)	**9.48** (±0.28)	**5.47** (±0.19)
NP-Match with $JS^{G_{\alpha_u}}$	**4.91** (±0.04)	4.96 (±0.06)	**4.11** (±0.02)	38.91 (±0.99)	**26.03** (±0.26)	**21.22** (±0.13)	**14.20** (±0.67)	9.51 (±0.37)	5.59 (±0.24)

Table 4. Ablation studies of the proposed uncertainty-guided skew-geometric JS divergence and its dual form.

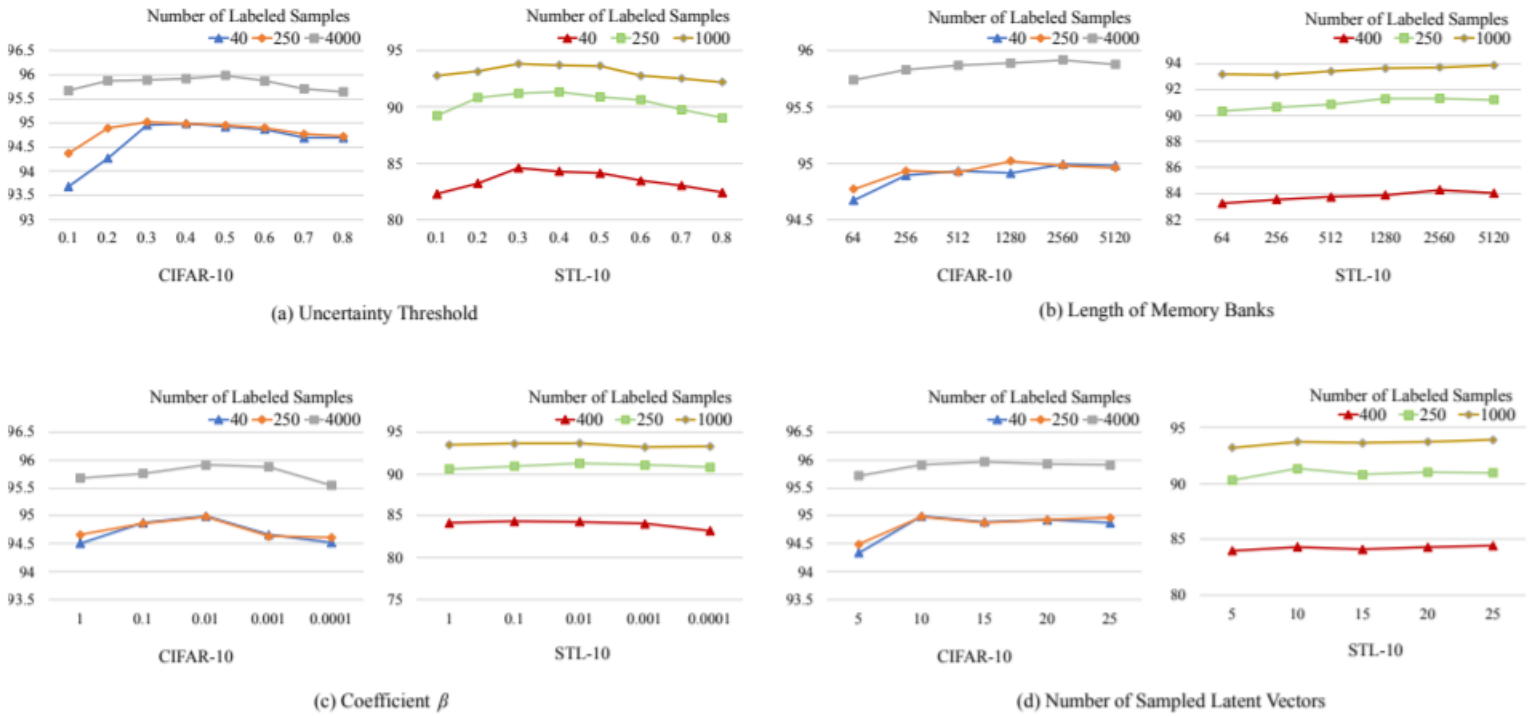


Figure 4. Performance for different hyperparameters.

Conclusion and Future Works

- We proposed the application of neural processes (NPs) to semi-supervised learning (SSL), designing a new framework called NP-Match, and explored its use in semi-supervised large-scale image classification.

- To better adapt NP-Match to the SSL task, we proposed a new divergence term, called uncertainty-guided skew-geometric JS divergence, which further improves the performance of NP-Match.

- We demonstrated the effectiveness of NP-Match and the proposed divergence term for SSL in extensive experiments.

Conclusion and Future Works

- Due to the successful application of NPs to semi-supervised image classification, it is valuable to explore NPs in other SSL tasks, such as object detection and segmentation.

- Many successful NPs variants have been proposed since the original NPs. It is valuable to explore these in SSL for image classification.

Source codes are available at: https://github.com/Jianf-Wang/NP-Match

Thanks for watching!!!