From Noisy Prediction to True Label: Noisy Prediction Calibration via Generative Model

HeeSun Bae*, Seungjae Shin*, Byeonghu Na, JoonHo Jang, Kyungwoo Song, Il-Chul Moon

Correspondence to: Il-Chul Moon <icmoon@kaist.ac.kr>

Learning with Noisy Labels

- Noisy labels are inevitable
  - Large-size dataset is unanimous for the success of DNNs.
  - Yet such large-scale dataset creation is arduous and prone to errors in their label annotations.

Learning with Noisy Labels

- Noisy labels are inevitable
  - Large-size dataset is unanimous for the success of DNNs.
  - Yet such large-scale dataset creation is arduous and prone to errors in their label annotations.

What we want

$$\boxed{R_L(f) \coloneqq E_{(X,Y)\sim P(x,y)}[L(f(x),y)]}$$

Learning with Noisy Labels

- Noisy labels are inevitable
  - Large-size dataset is unanimous for the success of DNNs.
  - Yet such large-scale dataset creation is arduous and prone to errors in their label annotations.

What we get

$$\tilde{R}_L^{emp}(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), \tilde{y}_i)$$

What we want

$$R_L(f) \coloneqq E_{(X,Y)\sim P(x,y)}[L(f(x), y)]$$

Learning with Noisy Labels

- Noisy labels are inevitable
  - Large-size dataset is unanimous for the success of DNNs.
  - Yet such large-scale dataset creation is arduous and prone to errors in their label annotations.

What we get → **What we want**

$$\tilde{R}_L^{emp}(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), \tilde{y}_i) \rightarrow R_L(f) \coloneqq E_{(X,Y)\sim P(x,y)}[L(f(x), y)]$$

Learning with Noisy Labels

- Noisy labels are inevitable
  - Large-size dataset is unanimous for the success of DNNs.
  - Yet such large-scale dataset creation is arduous and prone to errors in their label annotations.

What we get

$$\tilde{R}_L^{emp}(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), \tilde{y}_i)$$

What we want

$$R_L(f) \coloneqq E_{(X,Y)\sim P(x,y)}[L(f(x), y)]$$

- Existing methods are still not robust to label noises: They should solve two problems simultaneously.
  - Train a classifier
  - Manage noisy label problem

Learning with Noisy Labels

- Noisy labels are inevitable
  - Large-size dataset is unanimous for the success of DNNs.
  - Yet such large-scale dataset creation is arduous and prone to errors in their label annotations.

What we get

$$\tilde{R}_L^{emp}(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), \tilde{y}_i)$$

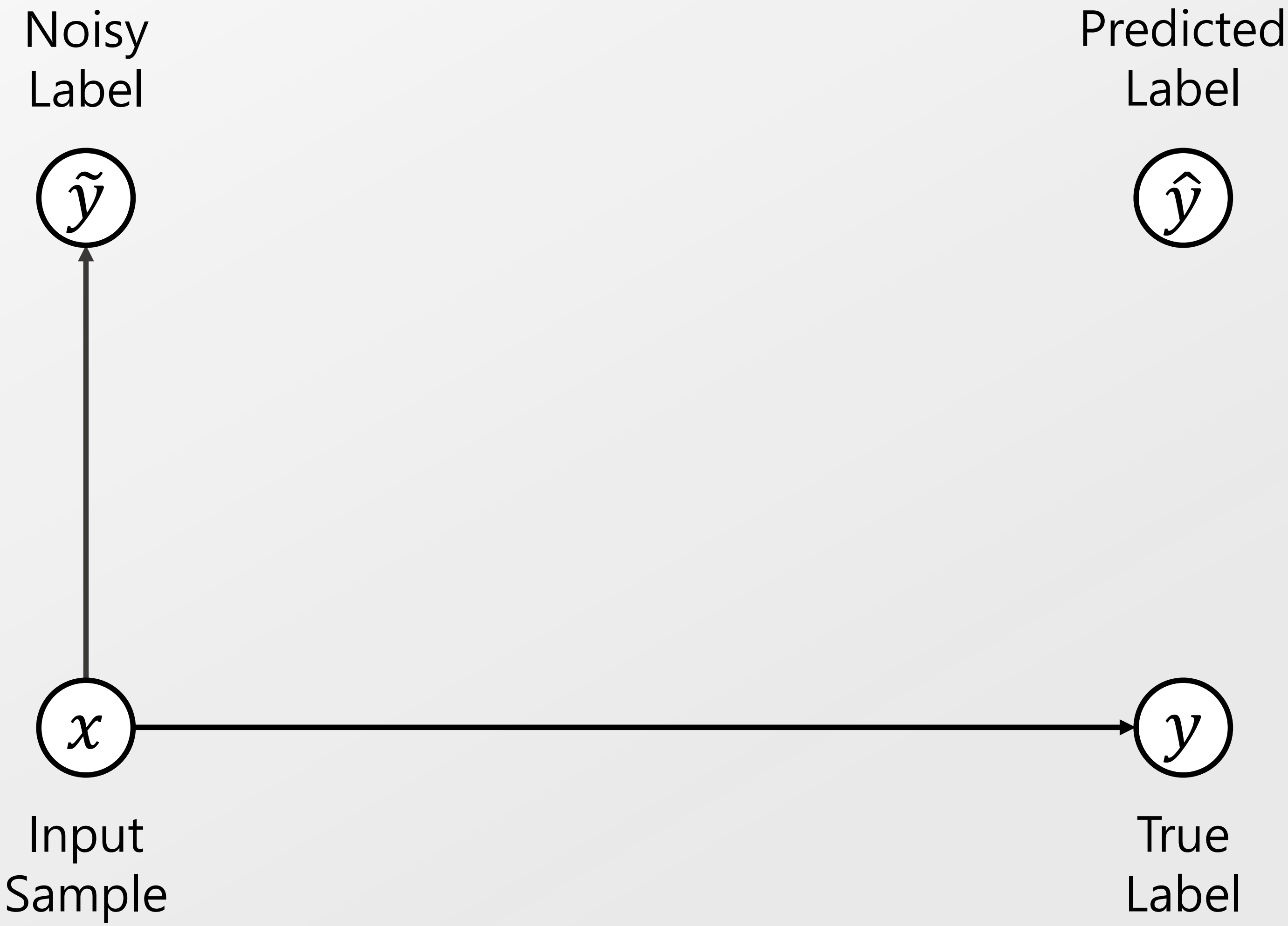
What we want

$$R_L(f) \coloneqq E_{(X,Y)\sim P(x,y)}[L(f(x), y)]$$

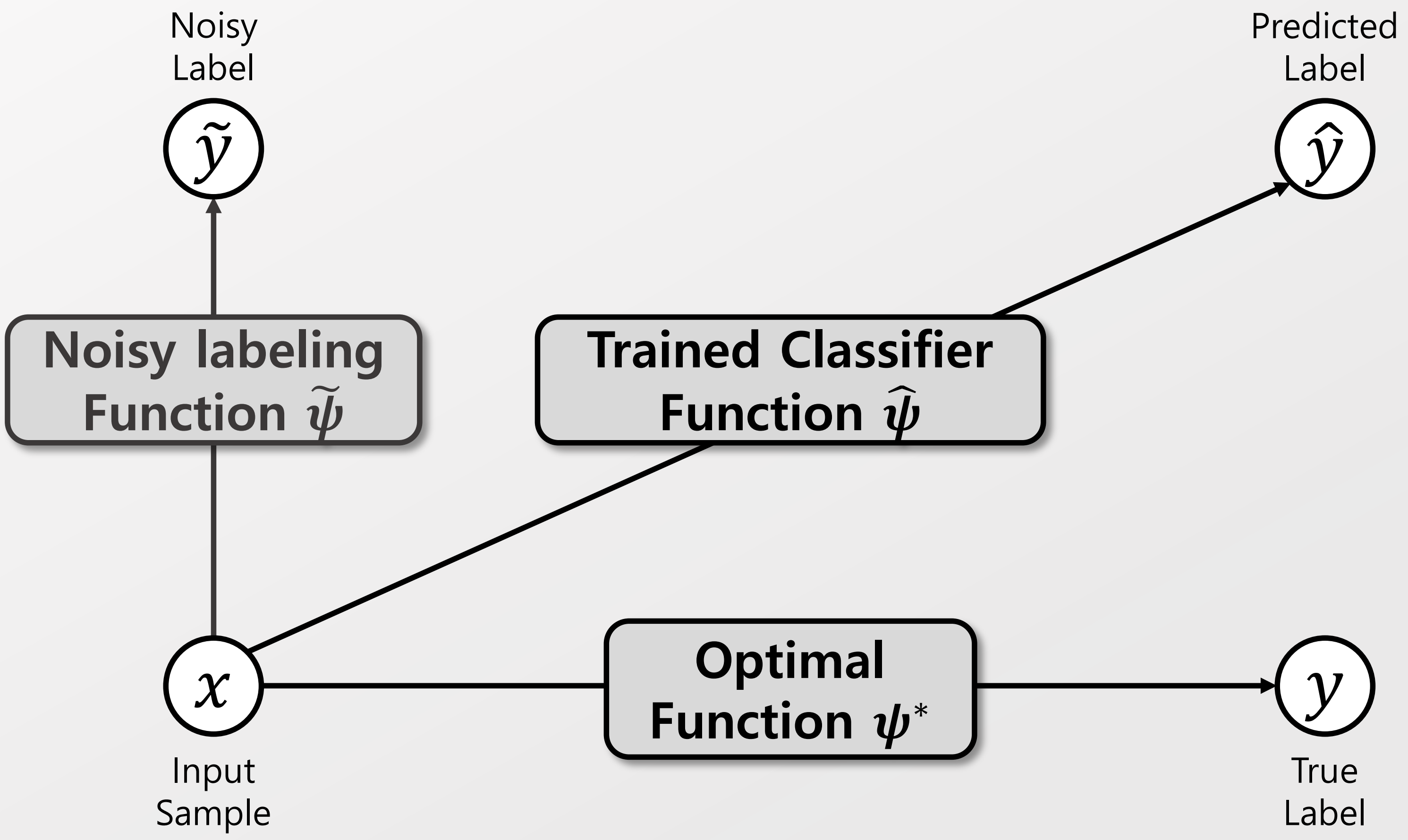
- Existing methods are still not robust to label noises: They should solve two problems simultaneously.
  - Train a classifier
  - Manage noisy label problem

- Modelling of reducing the gap between the prediction of trained classifier and the true latent label is necessary!

Motivation


Noisy Label
$\tilde{y}$
Predicted Label
$\hat{y}$
x
y
Input Sample
True Label

Motivation


Noisy
Label
$\tilde{y}$
Predicted
Label
$\hat{y}$
Noisy labeling
Function $\tilde{\psi}$
Trained Classifier
Function $\hat{\psi}$
x
Optimal
Function ψ^*
y
Input
Sample
True
Label

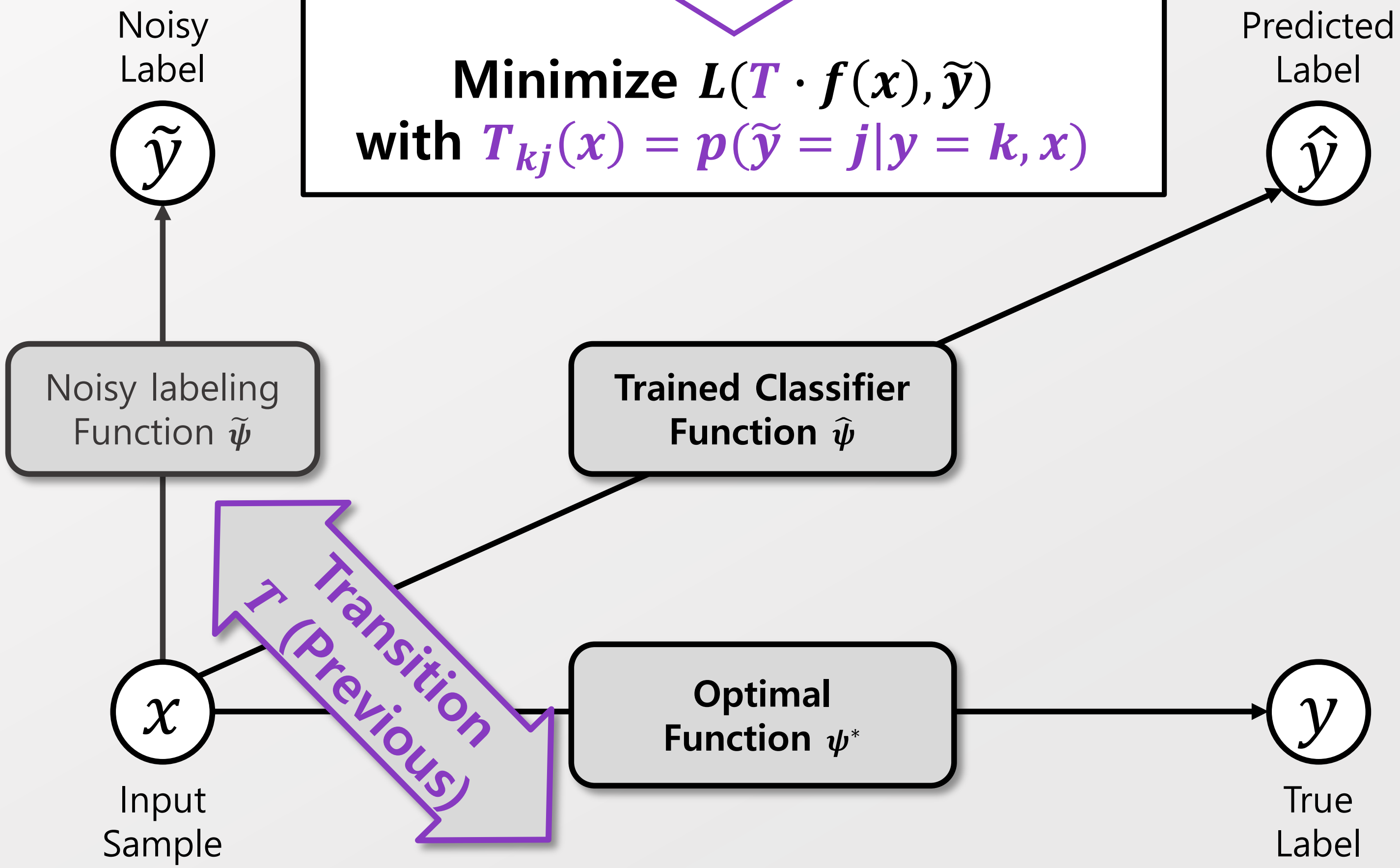
Motivation

Minimize $L(f(x), \tilde{y})$

⇩

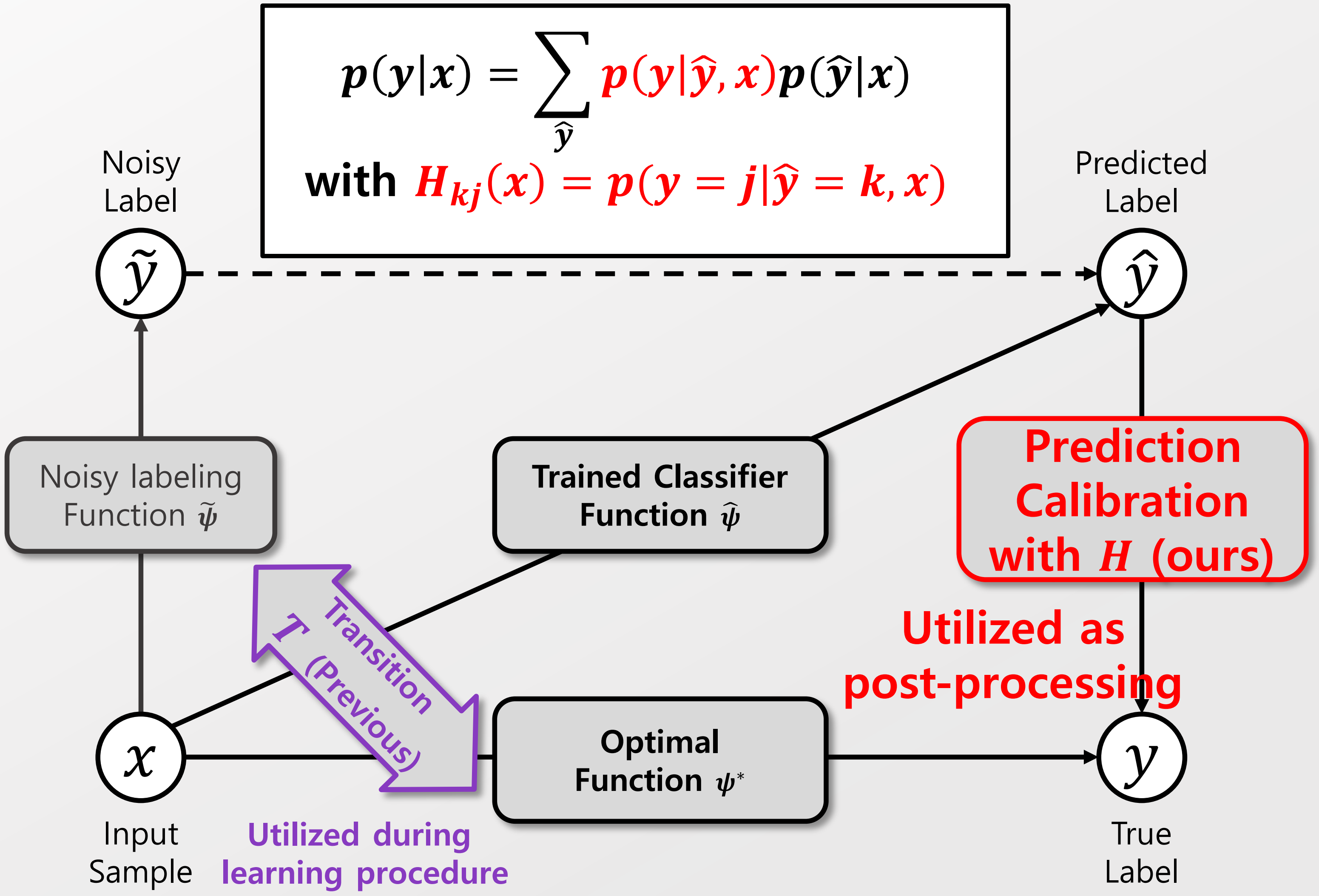
Minimize $L(T \cdot f(x), \tilde{y})$

with $T_{kj}(x) = p(\tilde{y} = j | y = k, x)$



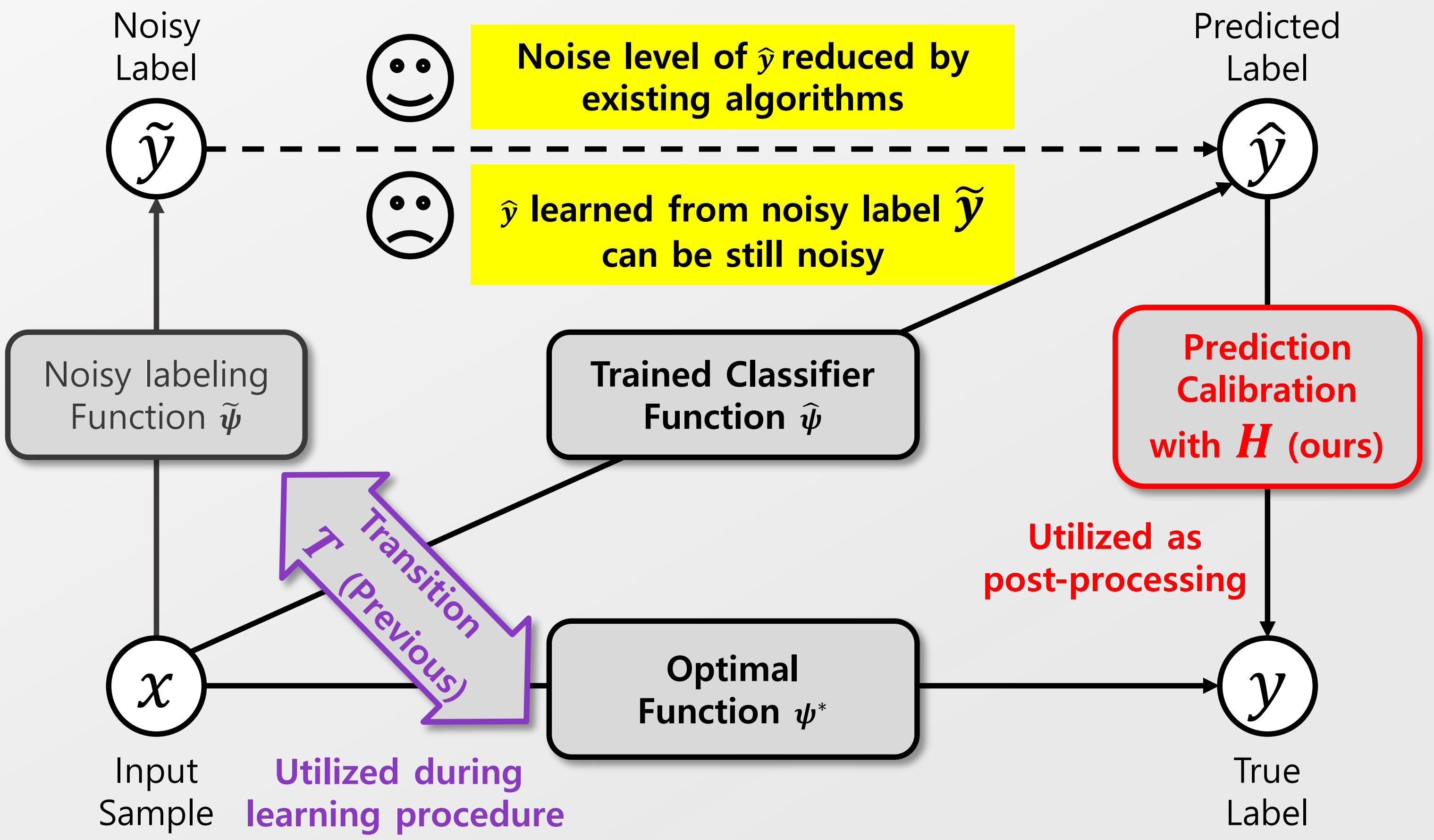
Utilized during learning procedure

Motivation



$$p(y|x) = \sum_{\hat{y}} p(y|\hat{y}, x)p(\hat{y}|x)$$
with $H_{kj}(x) = p(y = j|\hat{y} = k, x)$
Noisy Label
$\tilde{y}$
Predicted Label
$\hat{y}$
Noisy labeling Function $\tilde{\psi}$
Trained Classifier Function $\hat{\psi}$
Prediction Calibration with H (ours)
Transition T (Previous)
Utilized as post-processing
x
Optimal Function ψ^*
y
Input Sample
Utilized during learning procedure
True Label

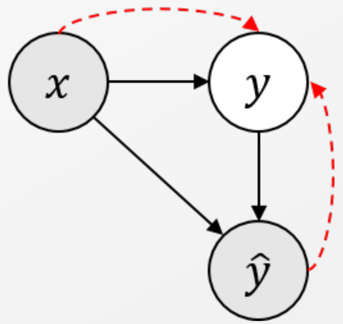
Motivation



Noisy Label
$\tilde{y}$
Noise level of $\hat{y}$ reduced by existing algorithms
$\hat{y}$ learned from noisy label $\tilde{y}$ can be still noisy
Predicted Label
$\hat{y}$
Noisy labeling Function $\tilde{\psi}$
Trained Classifier Function $\hat{\psi}$
Prediction Calibration with H (ours)
Transition T (Previous)
Utilized as post-processing
x
Optimal Function ψ^*
y
Input Sample
Utilized during learning procedure
True Label

NPC: Noisy Prediction Calibration

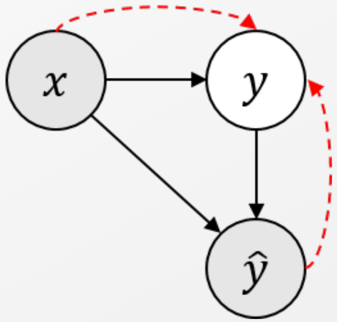
- Bayesian Network



- Generative Process
  1. $y \sim Dir(\alpha_x)$
  2. $\hat{y} \sim Multi(\pi_{x,y})$

NPC: Noisy Prediction Calibration

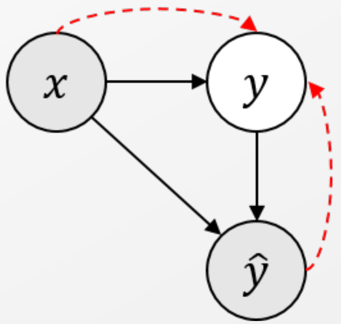
- Bayesian Network



- Generative Process
  1. $y \sim Dir(\alpha_x)$
  2. $\tilde{y} \sim Multi(\pi_{x,y})$
- **$p(y|\hat{y}, x)$ is intractable!**
  - → Minimize KL$\left(q(y|\hat{y}, x) \middle| p(y|\hat{y}, x)\right)$

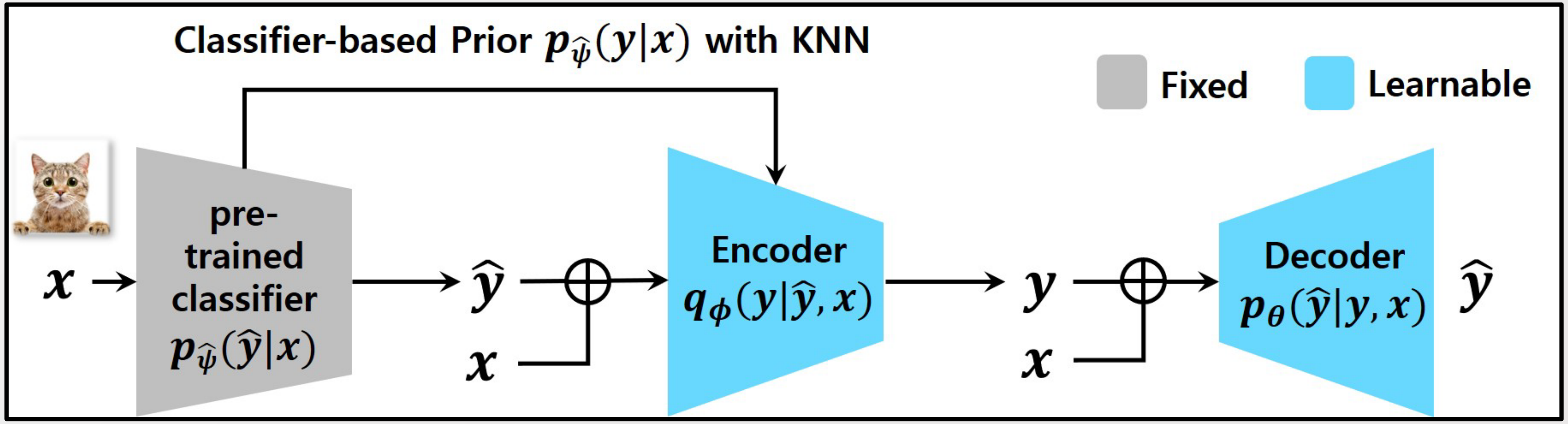
NPC: Noisy Prediction Calibration

- Bayesian Network



- Generative Process
  1. $y \sim Dir(\alpha_x)$
  2. $\tilde{y} \sim Multi(\pi_{x,y})$
- **$\mathbf{p}(\boldsymbol{y}|\hat{\boldsymbol{y}}, \boldsymbol{x})$ is intractable!**
  - → Minimize KL$\left(q(y|\hat{y}, x) \middle| p(y|\hat{y}, x)\right)$

- Neural Network structure of NPC $\quad \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\hat{\boldsymbol{y}}} \boldsymbol{q_\phi}(\boldsymbol{y}|\hat{\boldsymbol{y}}, \boldsymbol{x})\, \boldsymbol{p_{\hat{\psi}}}(\hat{\boldsymbol{y}}|\boldsymbol{x})$



NPC: Noisy Prediction Calibration

- Although NPC works as a post-processing algorithm, H provides a same pathway to correct the noisy classifier as T.

$$\boldsymbol{p(y|x) = \sum_{\hat{y}} q_{\phi}(y|\hat{y},x)\, p_{\hat{\psi}}(\hat{y}|x)}$$

$$H_{kj}(x) = \frac{p(y=j|x)}{p(\hat{y}=k|x)} \sum_i p(\hat{y}=k|\tilde{y}=i,x) T_{ij}(x)$$

Noisy Classifier Output

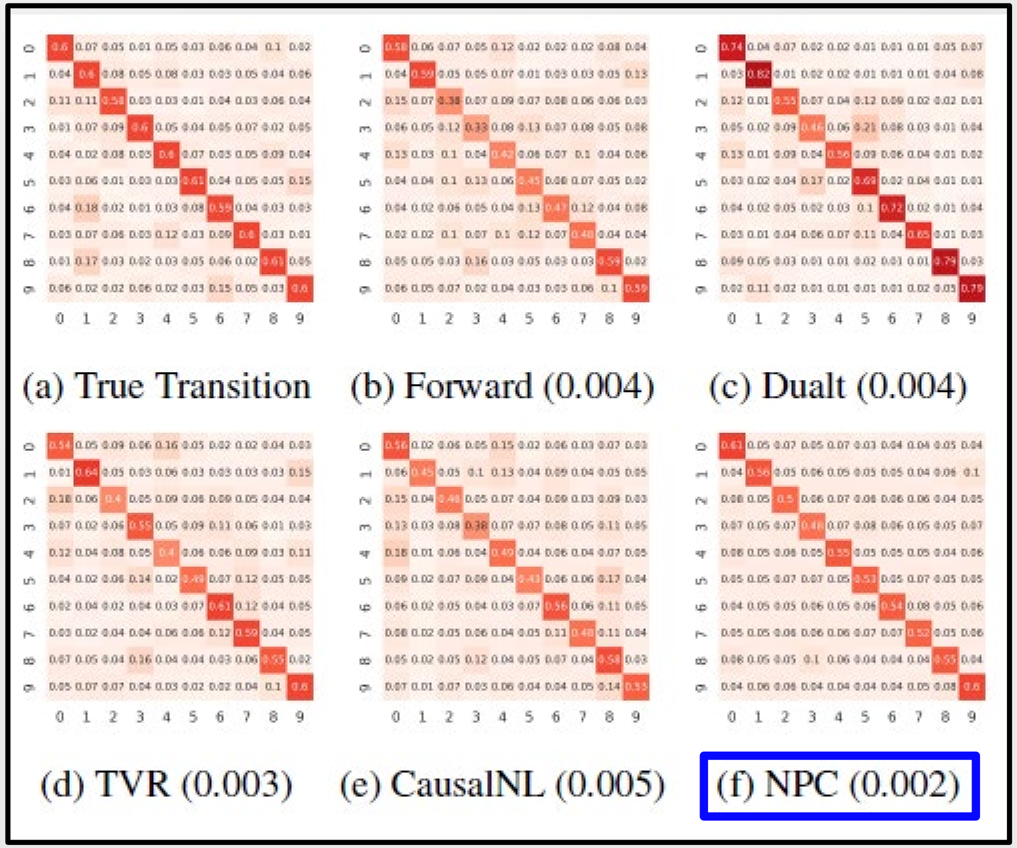
Trainable Function

NPC: Noisy Prediction Calibration

- Although NPC works as a post-processing algorithm, H provides a same pathway to correct the noisy classifier as T.

$$H_{kj}(x) = \frac{p(y=j|x)}{p(\hat{y}=k|x)} \sum_i p(\hat{y}=k|\tilde{y}=i,x) T_{ij}(x)$$

- NPC can approximate T good enough.
  - Values in parentheses are the MSE between the estimation and the truth.
  - NPC can also generate the transition matrix with comparable quality.



(a) True Transition (b) Forward (0.004) (c) Dualt (0.004)

(d) TVR (0.003) (e) CausalNL (0.005) (f) NPC (0.002)

Experiment Result

- Test accuracy : Synthetic Datasets

Model	MNIST		Fashion-MNIST									CIFAR-10								
	Clean	IDN	Clean	SN		ASN		IDN		SRIDN		Clean	SN		ASN		IDN		SRIDN	
	-	40%	-	20%	80%	20%	40%	20%	40%	20%	40%	-	20%	80%	20%	40%	20%	40%	20%	40%
CE	97.8	66.3	87.1	74.0	27.0	81.0	77.3	68.4	52.1	81.0	67.3	86.9	73.1	15.1	80.2	71.4	72.9	53.9	72.6	61.8
w/ NPC	98.2	89.0	88.4	84.0	35.8	85.9	86.2	82.5	74.5	81.8	69.4	89.0	80.8	17.0	84.7	78.8	80.9	59.9	74.3	64.3
Joint	93.0	93.6	82.8	82.0	6.0	82.1	82.3	82.7	82.4	80.6	74.6	83.0	78.9	8.3	81.5	76.8	80.4	64.5	70.6	62.2
w/ NPC	94.0	94.6	83.6	82.7	6.0	82.9	82.9	83.4	83.0	81.1	75.5	84.4	80.2	8.3	83.0	77.7	80.7	69.1	72.0	63.6
Coteaching	98.0	87.5	87.0	82.5	64.2	88.2	73.6	81.8	75.4	84.0	75.0	88.5	82.5	29.7	86.5	76.6	81.5	75.2	75.3	66.6
w/ NPC	98.3	90.6	88.3	85.8	66.0	88.5	73.6	85.1	78.7	84.2	75.3	89.2	85.3	32.1	87.1	76.8	84.8	78.5	76.1	67.2
JoCoR	97.8	93.3	88.7	86.0	27.6	88.9	79.4	86.3	83.2	81.9	71.3	89.1	83.6	24.8	82.6	73.3	82.8	75.3	75.2	66.1
w/ NPC	98.3	96.1	89.8	88.0	31.5	89.2	82.7	88.0	85.7	82.2	72.3	89.3	86.0	27.0	85.1	79.0	85.8	80.1	75.9	66.7
CORES2	97.0	48.8	87.2	74.6	8.9	77.6	74.3	80.0	58.1	81.3	71.2	87.1	70.1	31.2	79.0	71.2	70.3	50.9	72.8	62.0
w/ NPC	98.0	67.2	88.5	84.3	10.2	82.5	81.0	84.0	69.6	82.2	74.9	88.2	80.4	30.7	84.2	80.4	80.4	65.6	74.2	64.1
SCE	97.7	66.6	87.0	74.0	27.0	82.0	77.4	68.3	52.0	81.1	67.5	86.9	73.1	15.1	80.2	71.4	72.9	53.9	72.6	61.8
w/ NPC	98.2	88.7	88.3	83.7	35.5	86.4	86.7	82.0	75.2	81.8	69.7	87.4	75.0	15.2	81.5	75.2	75.4	55.6	72.9	62.5
Early Stop	96.5	73.3	87.5	83.6	49.5	84.1	76.6	79.5	55.4	83.3	72.6	83.0	79.1	18.0	80.9	70.6	77.1	62.5	71.4	60.6
w/ NPC	97.9	90.8	88.7	85.9	62.9	87.6	87.1	84.3	75.3	84.0	76.0	84.0	82.5	18.2	81.2	72.0	79.4	65.1	72.1	63.0
LS	97.8	66.2	87.5	73.9	27.8	81.5	77.0	69.0	52.5	81.1	67.5	86.9	73.1	15.1	80.2	71.4	72.9	53.9	72.6	61.8
w/ NPC	98.2	88.6	88.6	83.7	35.2	86.0	86.4	82.2	74.7	81.6	69.5	89.0	80.8	15.5	84.7	78.8	80.9	59.9	74.3	64.3
REL	98.0	90.7	88.1	84.6	70.1	82.8	76.2	84.6	75.5	83.7	78.1	80.7	74.9	21.2	72.8	69.9	75.5	51.8	69.3	63.8
w/ NPC	97.9	95.5	86.9	85.0	70.3	85.3	83.0	83.8	80.1	82.9	78.3	83.4	78.6	26.0	75.9	76.1	78.5	51.2	70.7	64.2
Forward	98.0	67.9	88.5	77.4	24.3	83.3	79.2	75.2	56.9	82.4	69.5	85.3	71.8	16.9	78.2	70.1	70.2	54.5	73.2	63.5
w/ NPC	98.4	91.1	89.6	85.3	33.0	87.2	86.8	86.8	80.5	83.3	73.7	88.7	81.5	17.2	83.8	74.5	80.3	63.3	74.8	65.0
DualT	96.7	94.3	86.3	84.5	10.0	86.9	83.1	85.1	68.5	82.7	73.2	84.3	79.3	7.6	80.6	77.1	78.6	71.2	68.7	63.1
w/ NPC	97.8	96.6	88.2	85.9	10.0	87.6	84.3	86.3	72.3	83.4	74.9	86.0	83.0	8.4	83.0	77.5	81.0	77.3	70.1	64.0
TVR	97.7	64.4	87.0	72.6	24.9	80.6	76.4	66.3	51.7	81.4	67.7	86.7	71.9	15.2	78.5	71.2	72.3	53.6	72.2	62.2
w/ NPC	98.1	84.5	88.3	82.3	31.9	84.9	85.3	79.8	73.6	82.1	70.3	88.3	80.8	15.7	84.1	76.5	80.8	60.7	74.5	64.5
CausalNL	98.1	85.2	88.1	84.0	51.5	88.8	87.4	83.4	75.2	82.0	71.2	89.6	79.9	17.0	84.6	74.8	79.9	60.4	74.6	63.5
w/ NPC	98.6	94.5	89.4	87.0	58.9	89.3	88.7	87.6	83.3	83.3	74.1	89.7	81.2	18.8	85.0	74.8	81.2	71.9	75.3	63.9

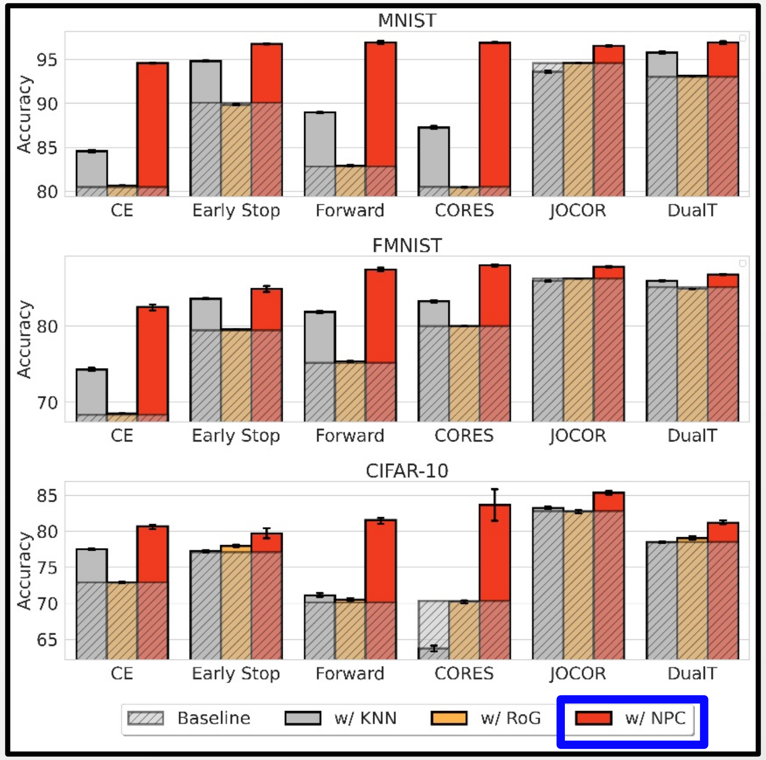
Experiment Result

- Test accuracy : Real Datasets

	Food-101		Clothing1M	
Method	w.o/ NPC	w/ NPC	w.o/ NPC	w/ NPC
CE	78.37	**80.21**±0.2	68.14	**70.83**±0.1
Early Stop	73.22	**76.80**±0.3	67.07	**70.21**±0.1
SCE	75.23	**78.26**±0.3	67.77	**70.36**±0.1
REL	78.96	78.95±0.4	62.53	**64.83**±0.1
Forward	83.76	83.77±0.3	66.86	**70.02**±0.1
DualT	57.46	**61.82**±0.7	70.18	69.99±0.4
TVR	77.34	**79.37**±0.1	67.18	**69.44**±0.1
CausalNL	86.08	**86.29**±0.0	68.31	**69.90**±0.2

Experiment Result

- NPC as a post-processor



- NPC shows the best performances **among post-processors**

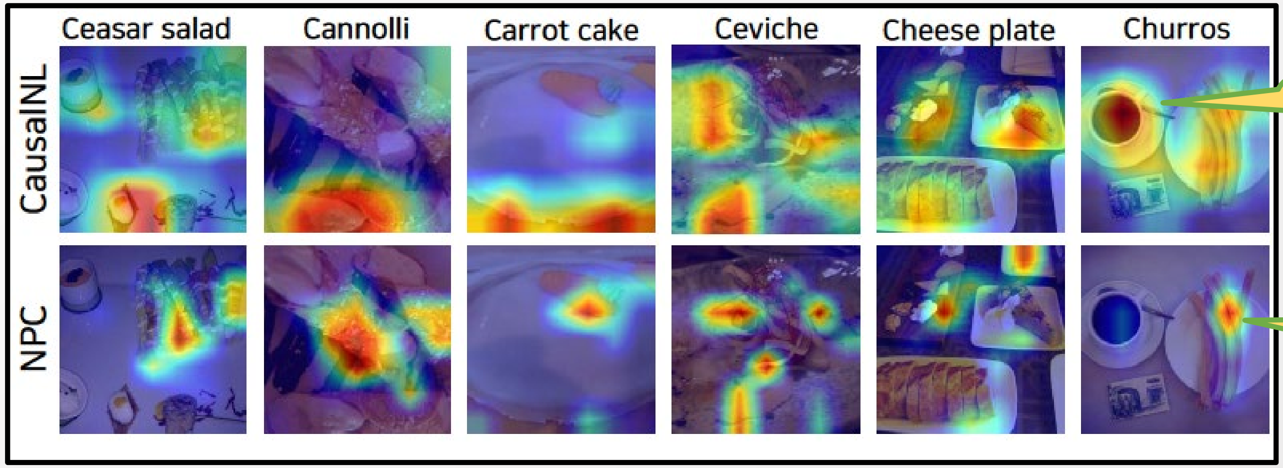
Method	Label Correction			
Noise	Joint	LRT	MLC	CauseNL
SN	80.0±0.6	82.9±0.2	71.1±1.9	77.2±1.5
IDN	78.6±1.3	82.5±0.2	72.2±2.6	78.4±1.7

Method	Post-processing			
Noise	LRT*	MLC*	CauseNL*	NPC
SN	82.7±0.1	82.2±1.9	83.5±0.5	**85.3**±0.3
IDN	82.9±0.2	82.1±0.4	83.3±0.5	**84.8**±0.1

- NPC achieves better accuracy than Label Correction methods
- Asterisks represent label correction to model prediction (application as post-processor)

Experiment Result

- NPC as a Generative Model

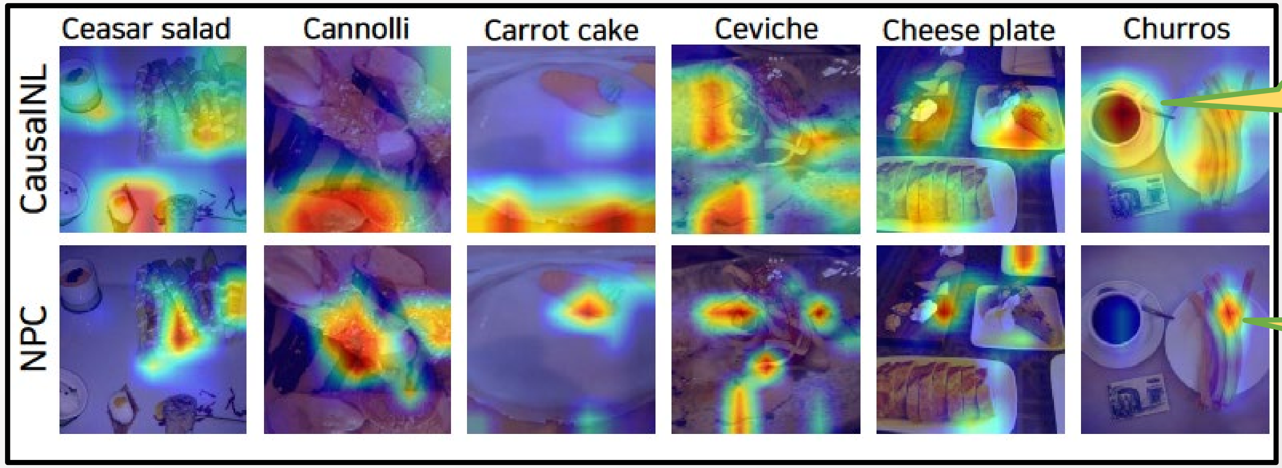


CausalNL: Total Image

NPC: Class-related

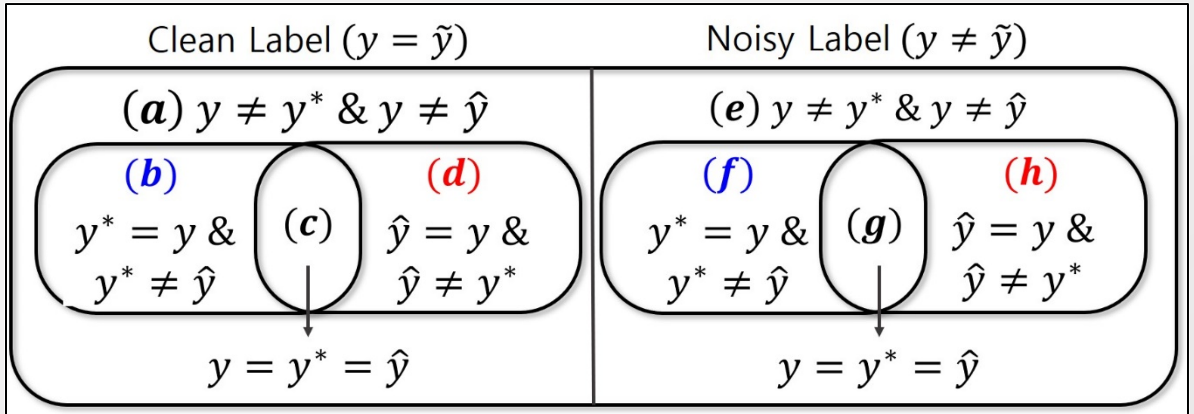
Experiment Result

- NPC as a Generative Model



CausalNL: Total Image

NPC: Class-related



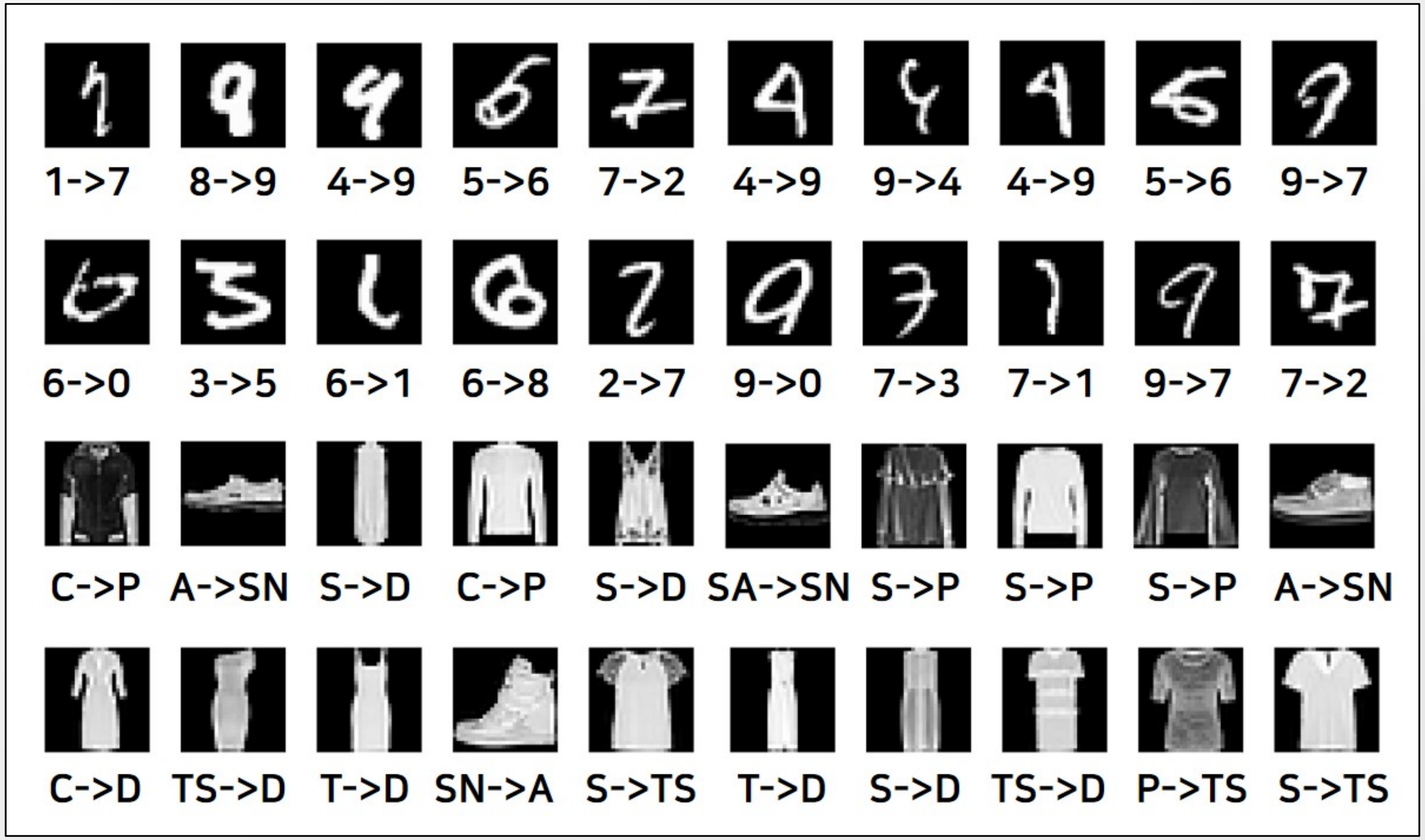
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
NPC	8	89	39799	86	9035	949	15	19
CausalNL*	39	58	32459	7426	2446	7538	31	3

- A good post processor should increase ▢ and decrease ▢

- NPC a cautious corrector
- CausalNL more risk-taker

Experiment Result

- NPC identifies potential noises in benchmarks



Conclusion

- We introduce novel post-processing method 'NPC' (Noisy Prediction Calibration)
  - NPC models the relation between output of a classifier and the true label via generative model.
  - NPC consistently boosts the classification performances of pre-trained models from diverse algorithms.
  - The prediction calibration scheme of NPC can be applied on various fields of machine learning.

Classifier Training (In-Processing)	→	Prediction Calibration (Post-Processing)
• Computationally inefficient for models with too many parameters. (e.g. CLIP, GPT-3) • It often hinge upon heuristics or assumptions (e.g. simple pattern at the early learning)	→	• Model-agnostic algorithm which only requires the model prediction. • Modeling objective is defined based on true latent label (Y)

Thank you