Fast yet Safe: Early-Exiting with Risk Control

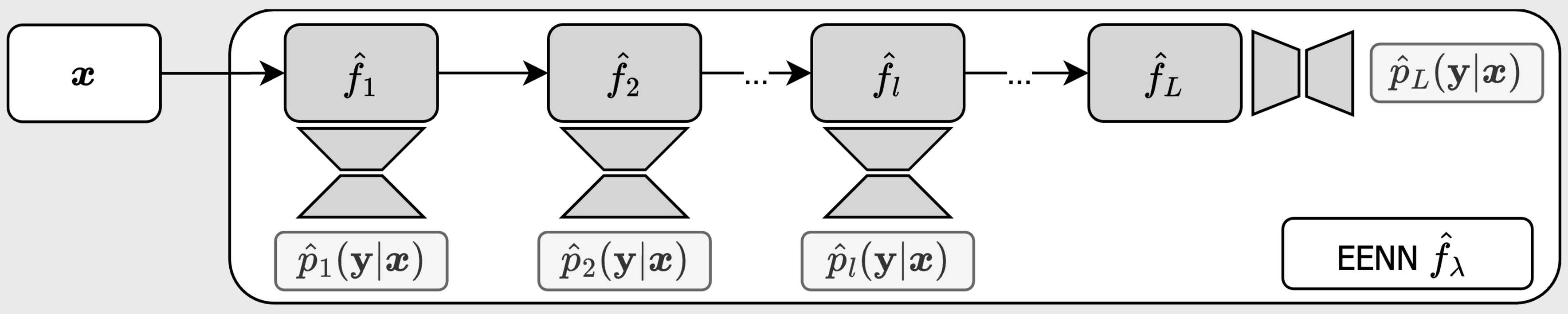
Metod Jazbec[1,*] Alexander Timans[1,*] Tin Hadži Veljković[1]
Kaspar Sakmann[2] Dan Zhang[2] Christian A. Naesseth[1] Eric Nalisnick[1,3]
— [1] University of Amsterdam [2] BCAI [3] Johns Hopkins University —

Motivation

Model inference should be dynamic based on user or data conditions. A simple yet effective solution is to permit intermediate exiting of model layers (EENNs).

- Problem: How to select the EENN's exit condition λ to balance the performance vs. efficiency trade-off.
- Solution (TLDR): Employ post-hoc, distribution-free risk control to resolve the trade-off according to user specifications with statistical guarantees.

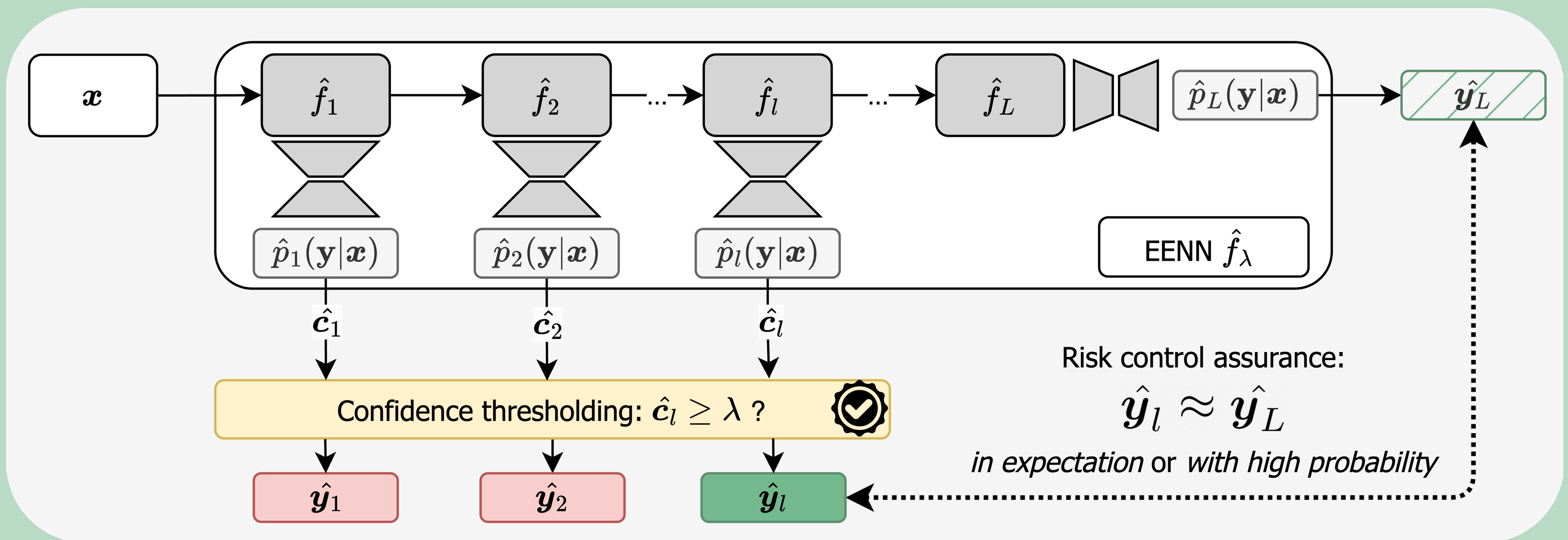
Early-Exit Neural Networks (EENNs)



Marginal monotonicity assumption:

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{P}}[\ell(\hat{p}_l(\mathbf{y}|\boldsymbol{x}),\boldsymbol{y})] \geq \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{P}}[(\ell(\hat{p}_{l+1}(\mathbf{y}|\boldsymbol{x}),\boldsymbol{y})] \quad \forall l = 1,\dots,L-1$$

Early-Exiting with Risk Control



Empirical threshold: $\hat{\lambda}_{\text{emp}} := \min\{\lambda \in \Lambda : \hat{\mathcal{R}}(\lambda;\mathcal{D}_{cal}) \leq \epsilon\}$

- No guarantees !

Conformal Risk Control (CRC): $\hat{\lambda}_{\text{CRC}} := \min\left\{\lambda \in \Lambda : \frac{n}{n+1}\hat{\mathcal{R}}(\lambda;\mathcal{D}_{cal}) + \frac{B}{n+1} \leq \epsilon\right\}$

- Risk control in expectation: $\mathbb{E}_{\mathcal{D}_{cal}\sim\mathcal{P}^n}[\mathcal{R}(\hat{\lambda}_{\text{CRC}})] \leq \epsilon$

Upper Confidence Bound (UCB): $\hat{\lambda}_{\text{UCB}} := \min\{\lambda \in \Lambda : \hat{\mathcal{R}}^+(\lambda';\mathcal{D}_{cal}) < \epsilon, \forall\lambda' \geq \lambda\}$

- Risk control w. high probability: $\mathbb{P}_{\mathcal{D}_{cal}\sim\mathcal{P}^n}(\mathcal{R}(\hat{\lambda}_{\text{UCB}}) \leq \epsilon) \geq 1-\delta$

Framework

INPUT

- Exit threshold candidates $\lambda \in [0,1]$
- Early-exit risk of the form

$$\mathcal{R}(\lambda) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{P}}\left[\ell(\hat{\boldsymbol{o}}_\lambda(\boldsymbol{x}),\boldsymbol{y}) - \ell(\hat{\boldsymbol{o}}_L(\boldsymbol{x}),\boldsymbol{y})\right],$$
$$\hat{\boldsymbol{o}}_l(\boldsymbol{x}) = \hat{\boldsymbol{y}}_l \quad \text{or} \quad \hat{\boldsymbol{o}}_l(\boldsymbol{x}) = \hat{p}_l(\mathbf{y}|\boldsymbol{x})$$

- User-defined risk settings

$$\ell : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}, \ \epsilon \in (0,1), \ \delta \in (0,1)$$

OUTPUT

- Risk-controlling exit threshold $\hat{\lambda} \in [0,1]$

Options

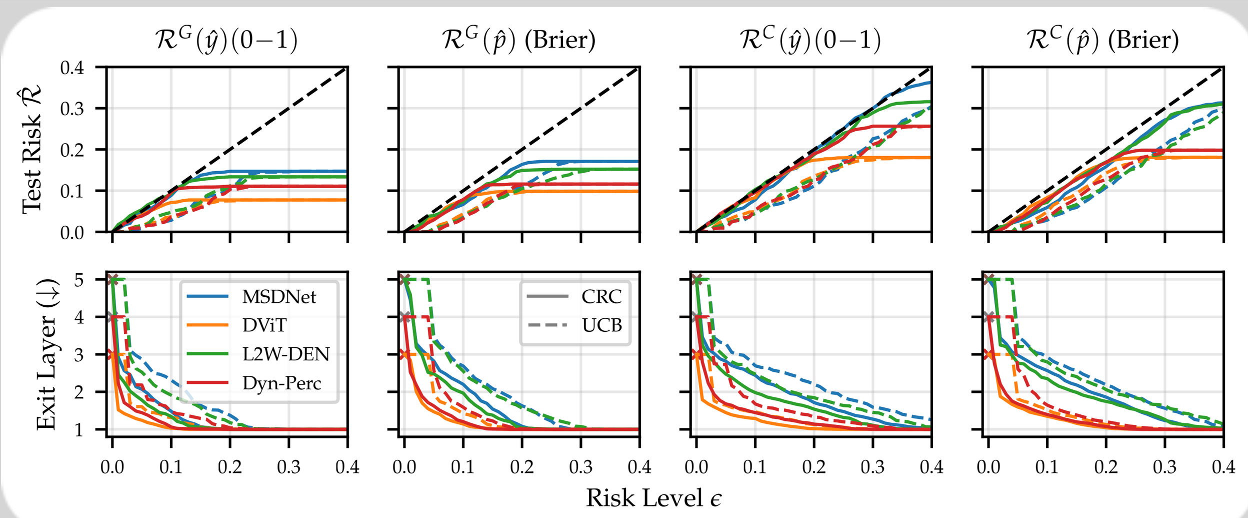
- Prediction control with task-specific losses
- Predictive distribution control with 'Brier score' loss
- Labelled and unlabelled data

Experiments

- Verify that risk is controlled on test data, i.e. $\hat{\mathcal{R}}(\hat{\lambda};\mathcal{D}_{test}) \leq \epsilon$ (across multiple trials)
- Assess obtained efficiency gains in terms of average exit layer (across samples & multiple trials)

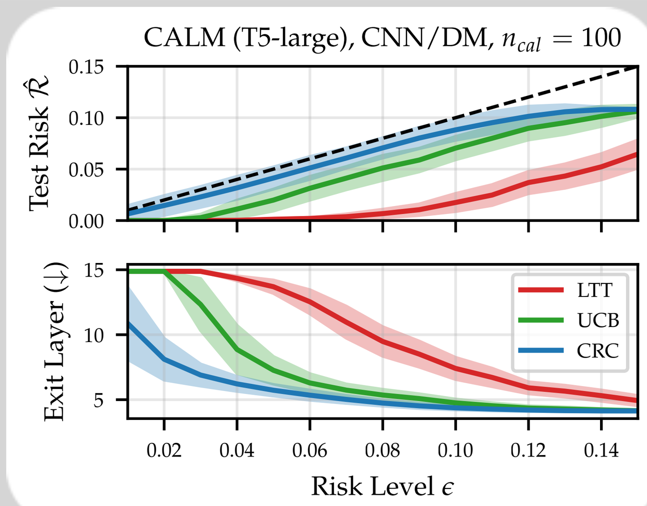
Image Classification

- Generalizes across varying black-box early-exit architectures



Language Modeling

- Outperforms existing method Learn-then-Test (LTT) used by CALM



Semantic Segmentation

- Generalizes across varying confidence measures

	Risk Level ϵ	$\mathcal{R}^G(\hat{\boldsymbol{y}})$ (mIoU) 0.01	0.05	0.1	$\mathcal{R}^G(\hat{p})$ (Brier) 0.01	0.05	0.1
Mean	**Top-1**	6.3	33.7	53.5	0.0	13.6	43.4
	Top-Diff	9.3	35.5	54.4	0.0	17.5	44.3
	Entropy	5.2	36.0	54.3	0.0	17.9	41.0
Patch	**Top-1**	10.0	35.7	53.3	0.0	18.4	45.3
	Top-Diff	10.0	35.2	53.4	0.0	19.4	45.9
	Entropy	9.1	34.8	53.5	0.0	18.0	45.8

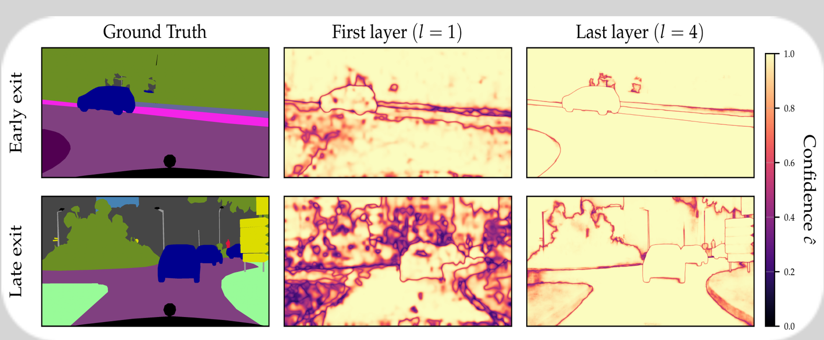
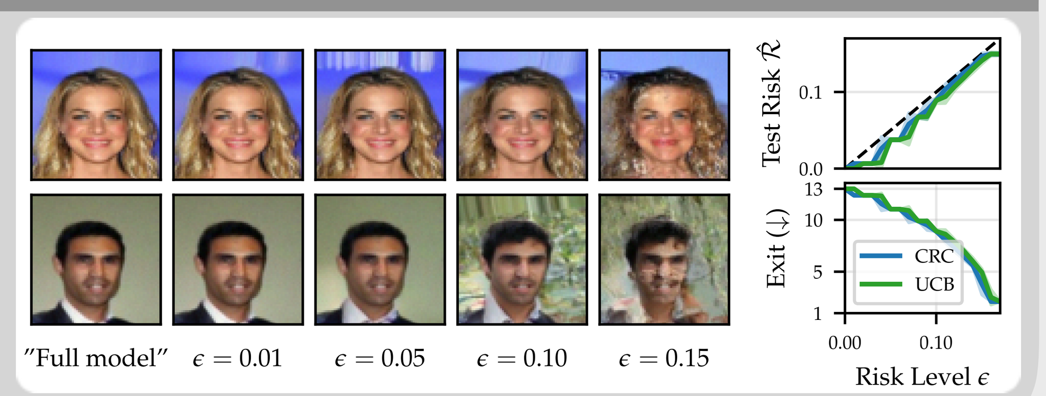


Image Generation

- Applicable to novel tasks (Diffusion)



References

- Bates et al. (2021). Distribution-free, risk-controlling prediction sets (JACM)
- Angelopoulos et al. (2024). Conformal Risk Control (ICLR)
- Angelopoulos et al. (2021). Learn then Test: Calibrating Predictive Algorithms to Achieve Risk Control (Preprint)
- Schuster et al. (2022). Confident Adaptive Language Modeling (NeurIPS)