Selective Prediction in Molecular Structure Retrieval

Workshop on Uncertainty Estimation in Machine Learning

Mira Jürgens

PhD student
Department of Data Science and Mathematical Modeling
Ghent University
Promotor: Willem Waegeman

BioML

GHENT UNIVERSITY

Contributors

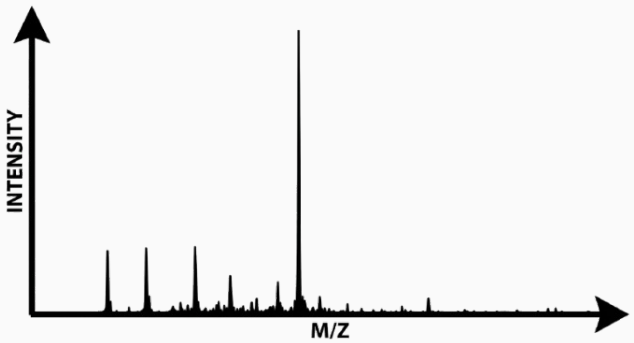
Mira Jürgens
Ghent University

Gaetan De Waele
University of Antwerp

Morteza Rakhshaninejad
Ghent University

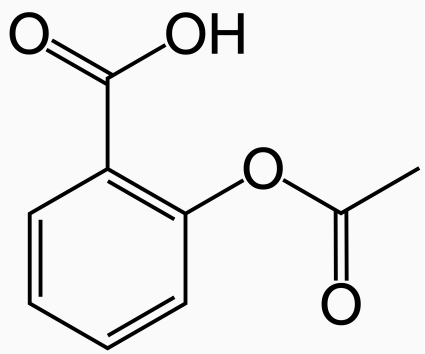
Willem Waegeman
Ghent University

Molecular Identification from Mass Spectrometry



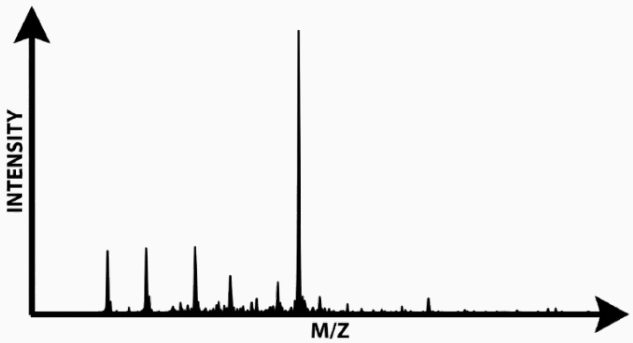
Spectrum ***x***

identify?



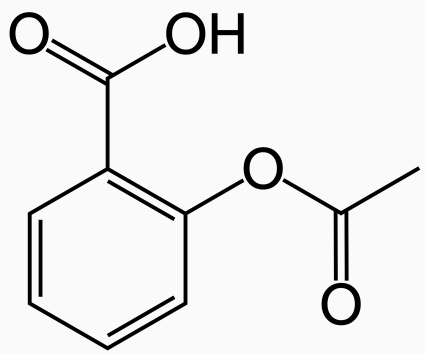
Molecule *y*

Molecular Identification from Mass Spectrometry



Spectrum x

identify?

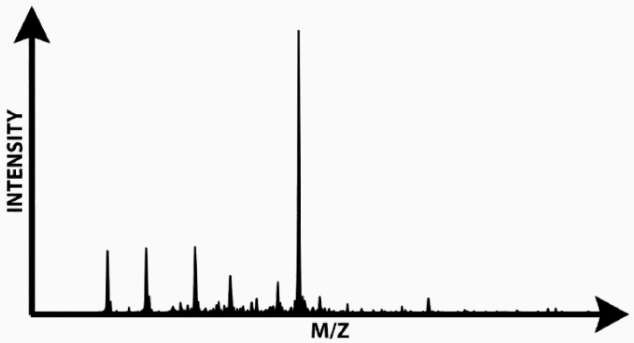


Molecule y

Retrieval task: Given spectrum x, rank a candidate set $\mathcal{C}(x)$ of molecules

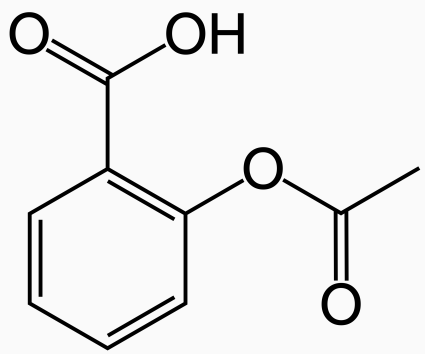
$$\underbrace{> 10^{60}}_{\text{chemical space}} \xrightarrow{\text{filter by mass/formula}} \underbrace{|\mathcal{C}(x)| \leq 256}_{\text{candidates}}$$

Molecular Identification from Mass Spectrometry



Spectrum x

identify?



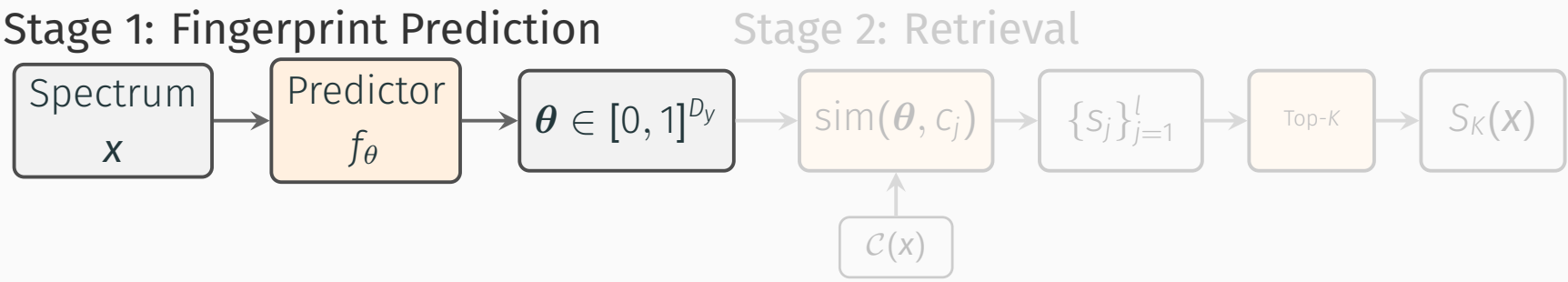
Molecule y

Retrieval task: Given spectrum x, rank a candidate set $\mathcal{C}(x)$ of molecules

$$\underbrace{\geq 10^{60}}_{\text{chemical space}} \xrightarrow{\text{filter by mass/formula}} \underbrace{|\mathcal{C}(x)| \leq 256}_{\text{candidates}}$$

Challenges: Sparse training data ($< 1\%$ coverage), ambiguous mapping (isomers, noise)

Two-Stage Retrieval Pipeline



$\mathcal{X} \subseteq \mathbb{R}^{D_x}$: MS/MS spectra

$\mathcal{Y} \subseteq \{0,1\}^{D_y}$: fingerprints

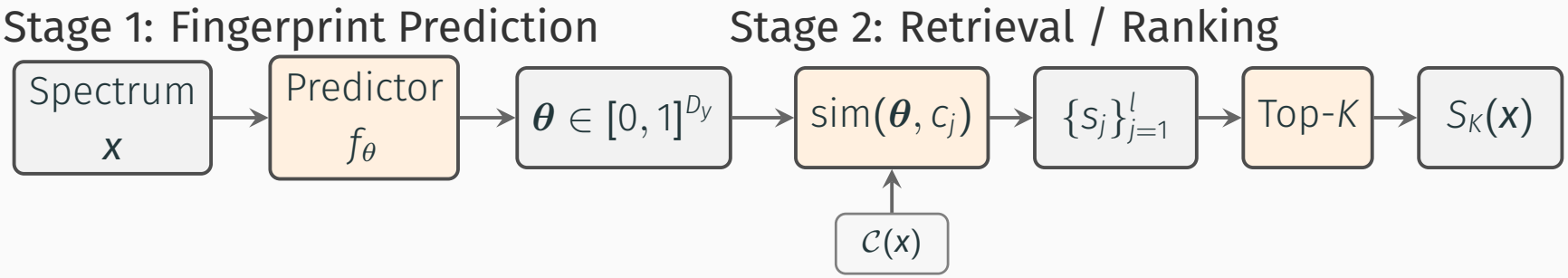
$\boldsymbol{\theta} \in [0,1]^{D_y}$: bit probabilities

$f_\theta : \mathcal{X} \to [0,1]^{D_y}$: fingerprint predictor

Model from De Waele et al.[1]

[1] De Waele et al., "Small Molecule Retrieval from Tandem Mass Spectrometry: What are we optimizing for?"

Two-Stage Retrieval Pipeline



$\mathcal{X} \subseteq \mathbb{R}^{D_x}$: MS/MS spectra

$\mathcal{Y} \subseteq \{0,1\}^{D_y}$: fingerprints

$\theta \in [0,1]^{D_y}$: bit probabilities

$f_\theta : \mathcal{X} \to [0,1]^{D_y}$: fingerprint predictor

$\mathcal{C}(x) \subset \mathcal{Y}$: candidates

$s_j = \mathrm{sim}(\theta, c_j)$: similarity score

$S_K(x) \subset \mathcal{C}(x)$: top-K candidates

Model from De Waele et al.[1]

[1] De Waele et al., "Small Molecule Retrieval from Tandem Mass Spectrometry: What are we optimizing for?"

MassSpecGym Dataset challenge

MassSpecGym: A benchmark for the discovery and identification of molecules

Roman Bushuiev[1,2], **Anton Bushuiev**[2], **Niek F. de Jonge**[3], **Adamo Young**[4], **Fleming Kretschmer**[5], **Raman Samusevich**[1,2], **Janne Heirman**[6], **Fei Wang**[7,8], **Luke Zhang**[9], **Kai Dührkop**[5], **Marcus Ludwig**[10], **Nils A. Haupt**[5], **Apurva Kalia**[11], **Corinna Brungs**[1], **Robin Schmid**[1], **Russell Greiner**[7,8], **Bo Wang**[4], **David S. Wishart**[7,12], **Li-Ping Liu**[11], **Juho Rousu**[13], **Wout Bittremieux**[6], **Hannes Rost**[9], **Tytus D. Mak**[14], **Soha Hassoun**[11,15], **Florian Huber**[16], **Justin J.J. van der Hooft**[3,17], **Michael A. Stravs**[18], **Sebastian Böcker**[5], **Josef Sivic**[2], **Tomáš Pluskal**[1]

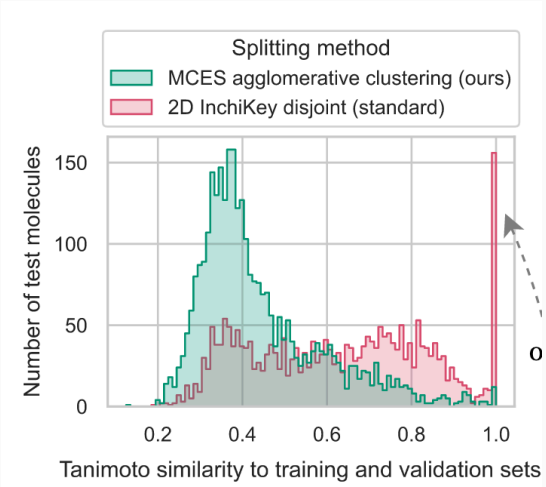
	Hit rate @ 1 ↑	Hit rate @ 5 ↑	Hit rate @ 20 ↑
Random	0.37 (0.24-0.54)	2.01 (1.68-2.39)	8.22 (7.53-8.89)
DeepSets	1.47 (1.18-1.77)	6.21 (5.64-6.82)	19.23 (18.24-20.26)
Fingerprint FFN	2.54 (2.17-2.99)	7.59 (6.96-8.28)	20.00 (19.01-20.98)
DeepSets + Fourier features	5.24 (4.71-5.83)	12.58 (11.80-13.42)	28.21 (27.10-29.36)
MIST	**14.64 (13.82-15.54)**	**34.87 (33.69-36.10)**	**59.15 (57.89-60.39)**

Figure 1: Retrieval results from Massspecgym[2]

[2] R. Bushuiev et al., "MassSpecGym: A benchmark for the discovery and identification of molecules".

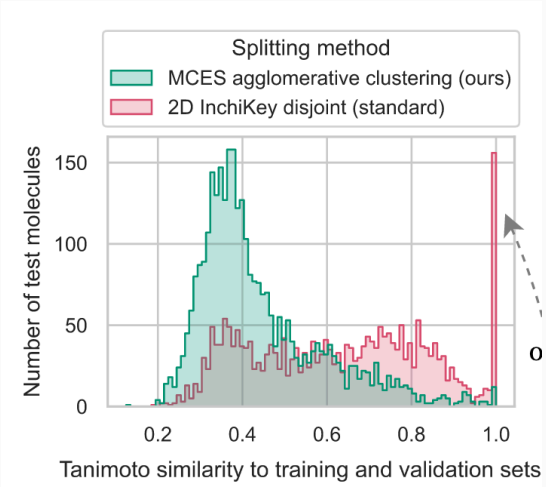
Structural Challenges

MCES-based split

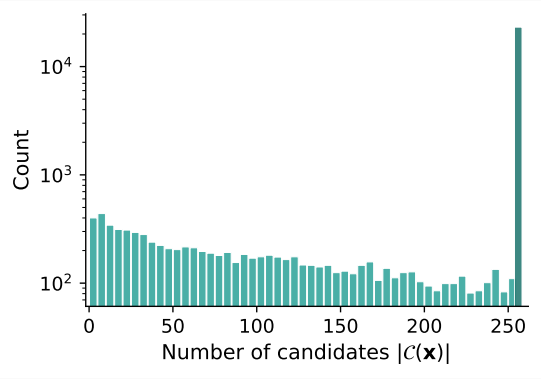


Structural Challenges

MCES-based split

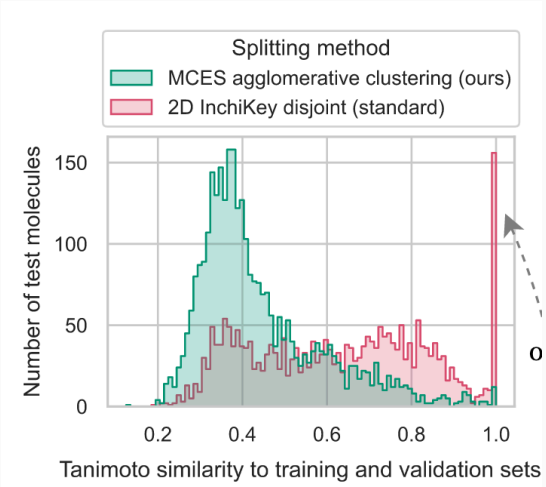


Varying number of candidates

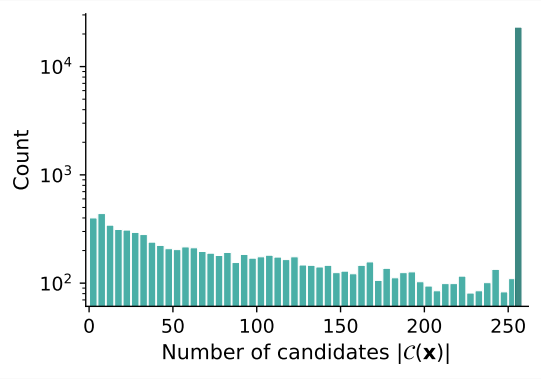


Structural Challenges

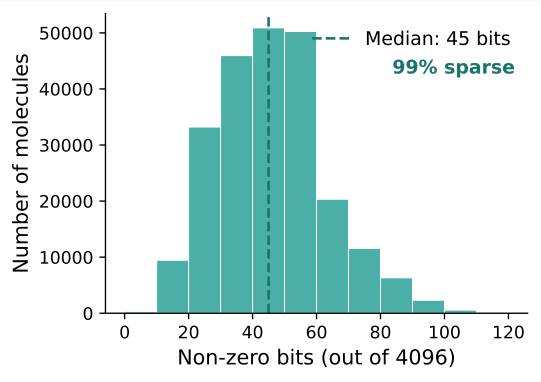
MCES-based split



Varying number of candidates

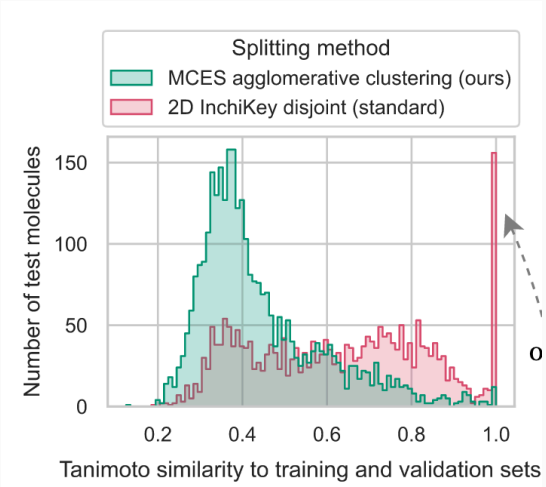


Extreme sparsity

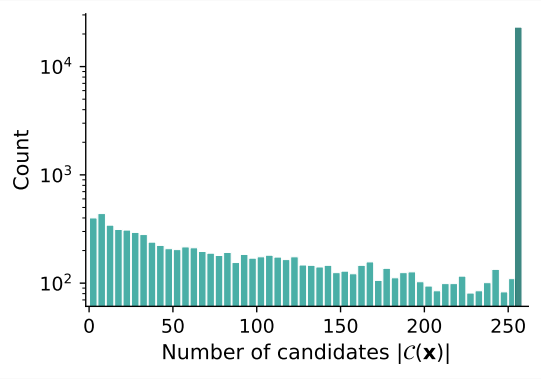


Structural Challenges

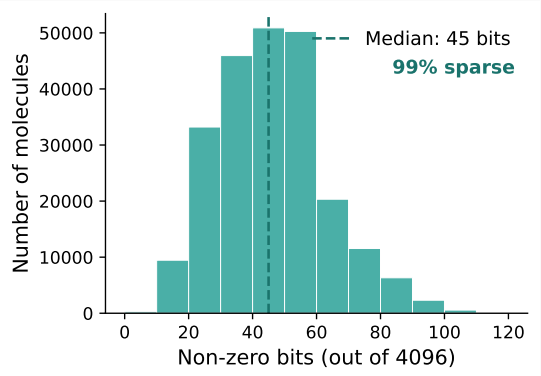
MCES-based split



Varying number of candidates



Extreme sparsity

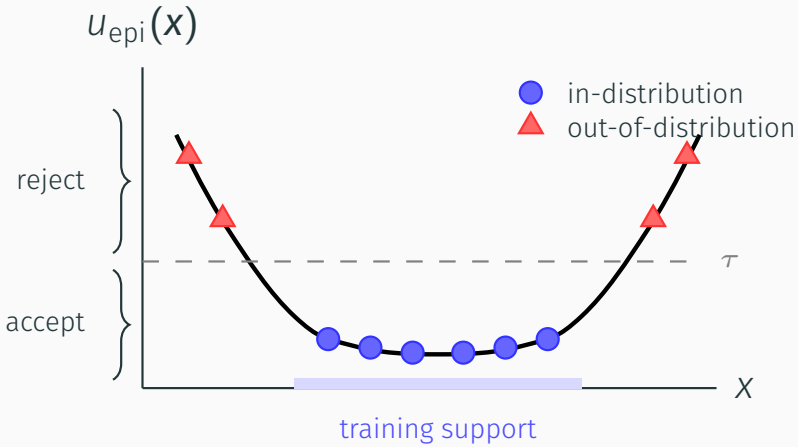


How can we use uncertainty to get guarantees on our predictions?

Selective Prediction in Molecular Retrieval

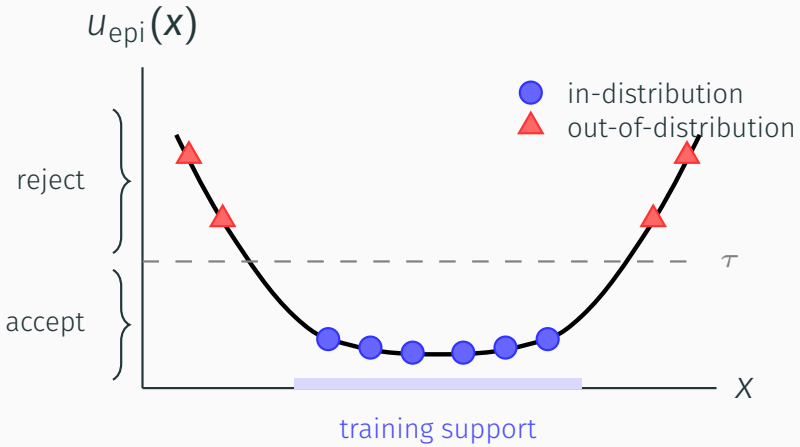
”Downstream Tasks” as evaluation in uncertainty quantification

Out-of-Distribution Detection

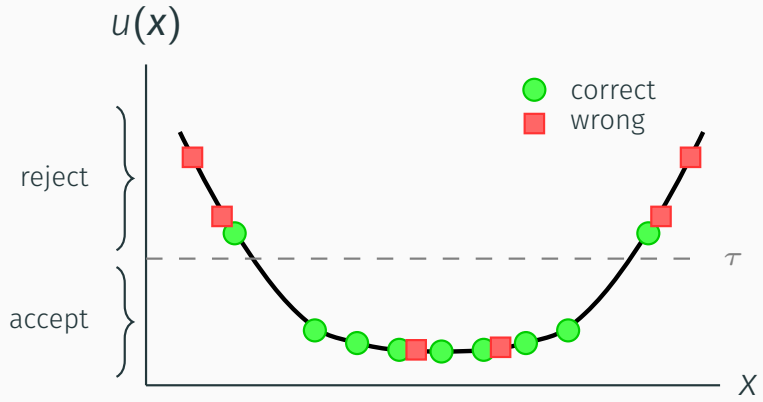


”Downstream Tasks” as evaluation in uncertainty quantification

Out-of-Distribution Detection

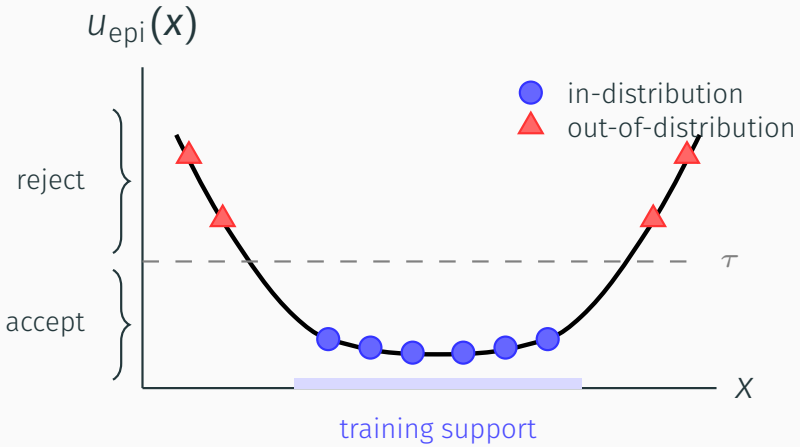


Selective Prediction

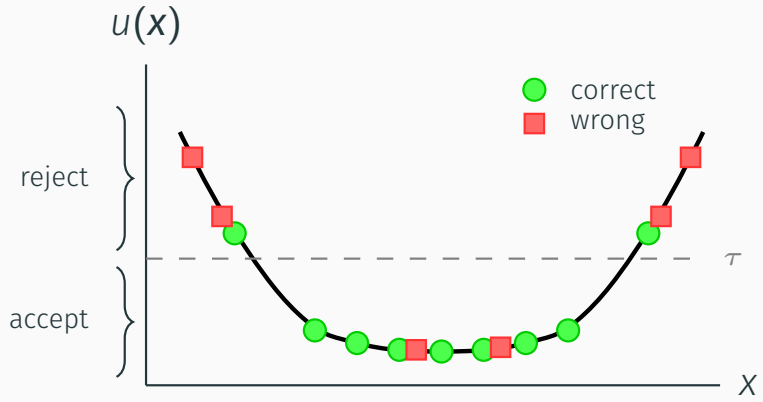


”Downstream Tasks” as evaluation in uncertainty quantification

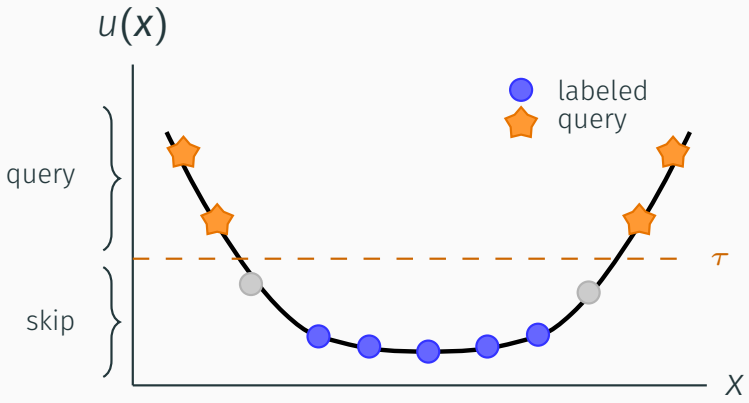
Out-of-Distribution Detection



Selective Prediction



Active Learning



Selective prediction setting

Following Geifman et al.[3], we define

- $f : \mathcal{X} \to \mathcal{Y}$ as our **predictor**,
- $g_\tau : \mathcal{X} \to \{0, 1\}$ as our **selector**, and

$$(f, g_\tau)(x) = \begin{cases} f(x), & g_\tau(x) = 1 \\ \text{reject}, & g(x) = 0 \end{cases}$$

as the **selective predictor**.

[3] Geifman and El-Yaniv, "Selective Classification for Deep Neural Networks".

Selective prediction setting

Following Geifman et al.[3], we define

- $f : \mathcal{X} \to \mathcal{Y}$ as our **predictor**,
- $g_\tau : \mathcal{X} \to \{0, 1\}$ as our **selector**, and

$$(f, g_\tau)(x) = \begin{cases} f(x), & g_\tau(x) = 1 \\ \text{reject}, & g(x) = 0 \end{cases}$$

as the **selective predictor**. Having an *uncertainty score* $u_f : \mathcal{X} \to \mathbb{R}$, g_τ can be defined as

$$g_\tau(x) := g_\tau(x, u_f) = \begin{cases} 1, & u_f(x) < \tau \\ 0, & u_f(x) \geq \tau. \end{cases}$$

[3] Geifman and El-Yaniv, "Selective Classification for Deep Neural Networks".

Selective prediction - Evaluation

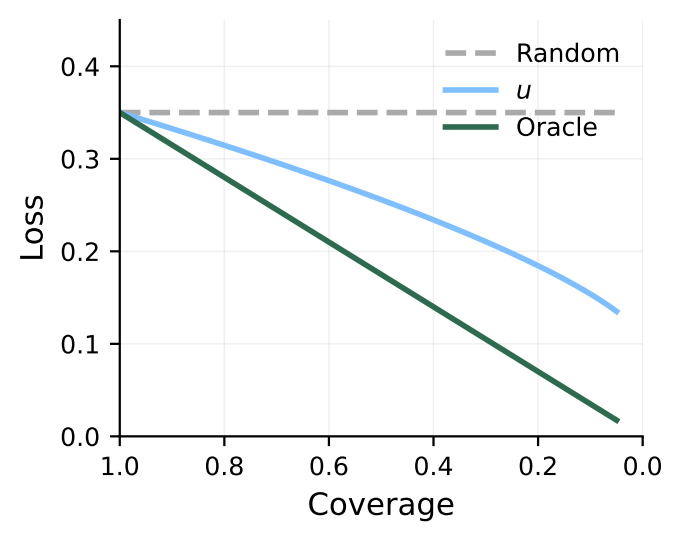
Risk-coverage-curve

Define the *coverage* of the selector g_τ as

$$\phi(g_\tau) = \mathbb{E}[g_\tau(X)]$$

and the *selective risk* as

$$R(f, g) = \frac{\mathbb{E}\left[\ell(f(x), y)g(x)\right]}{\phi(g)}.$$



Selective prediction - Evaluation

Risk-coverage-curve

Define the *coverage* of the selector g_τ as

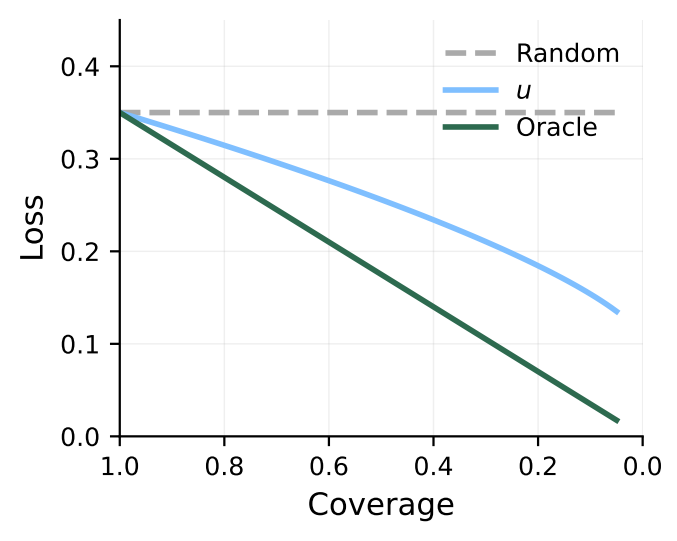
$$\phi(g_\tau) = \mathbb{E}[g_\tau(X)]$$

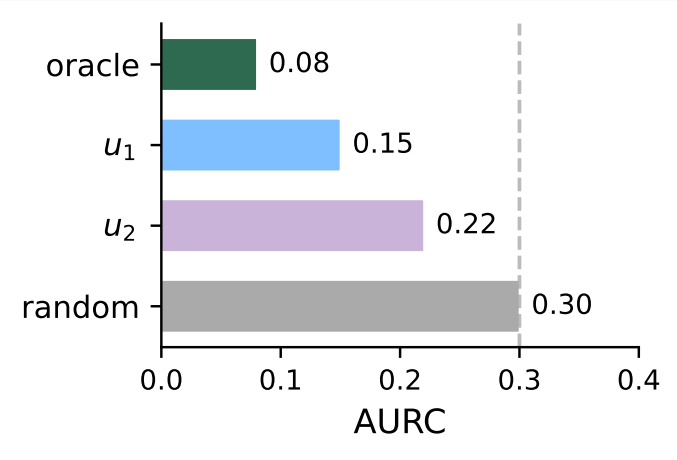
and the *selective risk* as

$$R(f, g) = \frac{\mathbb{E}\left[\ell(f(x), y)g(x)\right]}{\phi(g)}.$$

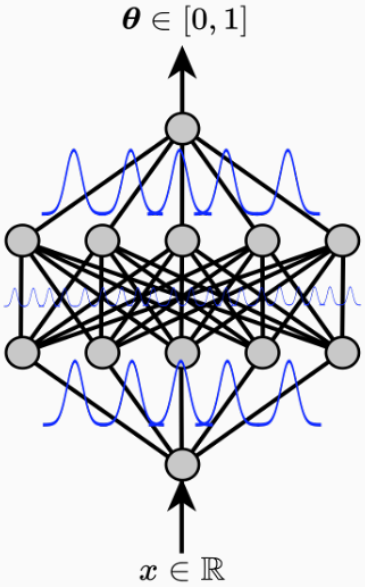
Area-under-risk-coverage-curve

$$\text{AURC} = \int_0^1 R(f, g_\tau)d\tau$$



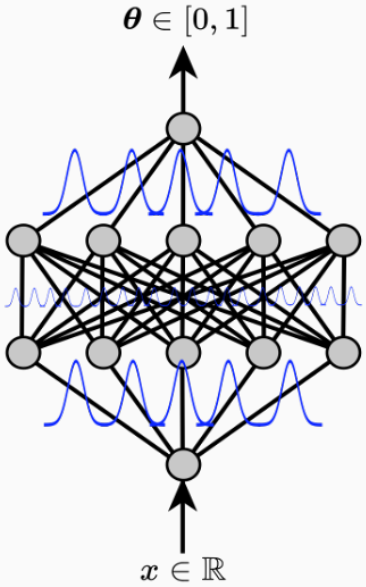


Levels of uncertainty



Use a *second-order distribution* to represent uncertainty:

$$q(\boldsymbol{\theta}|\boldsymbol{x}) : \mathcal{X} \to \mathbb{P}(\Theta).$$



Use a *second-order distribution* to represent uncertainty:

$$q(\boldsymbol{\theta}|\boldsymbol{x}) : \mathcal{X} \to \mathbb{P}(\Theta).$$

Standard: $x \to f_\theta \to \boldsymbol{\theta}$ single prediction

Second-order model: $x \to q(\boldsymbol{\theta}|x) \to \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(M)}$ *M* samples

Here: use Deep Ensembles, MC Dropout, Laplace approximation to sample from $q(\boldsymbol{\theta}|\boldsymbol{x})$.

On which *levels* can we model uncertainty?

On the fingerprint level

Assuming **independence between bits**, we model $\theta_d \sim q_d(\theta_d|\boldsymbol{x})$, $d = 1, \ldots D_y$.
Calculate *uncertainty per bit*.

[4] Depeweg et al., "Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning".

On the fingerprint level

Assuming **independence between bits**, we model $\theta_d \sim q_d(\theta_d|\boldsymbol{x})$, $d = 1, \ldots D_y$. Calculate *uncertainty per bit*.

Information theoretic decomposition[4] (bit-level)

Assume $q(\boldsymbol{\theta}|\boldsymbol{x}) = \prod_{d=1}^{D} q_d(\theta_d|\boldsymbol{x})$ and $y_d|\theta_d \sim \text{Bernoulli}(\theta_d)$.

$$\mathbb{H}\Big[\mathbb{E}_{\theta_i \sim q_i}\big[y_d|\theta_d\big]\Big] = \underbrace{\mathbb{E}_{\theta_d \sim q_d}\Big[\mathbb{H}[y_d|\theta_d]\Big]}_{\text{aleatoric}} + \underbrace{\mathbb{E}_{\theta_d \sim q_d}\Big[D_{KL}\big(\text{Bern}(\theta_d) \,\|\, \text{Bern}(\bar{\theta}_d)\big)\Big]}_{\text{epistemic}},$$

where $\bar{\theta}_d = \mathbb{E}_{\theta_d}[\theta_d]$ is the **aggregated bit probability**.

[4] Depeweg et al., "Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning".

On the fingerprint level

Assuming **independence between bits**, we model $\theta_d \sim q_d(\theta_d|\boldsymbol{x})$, $d = 1, \ldots D_y$. Calculate *uncertainty per bit*.

Information theoretic decomposition[4] (bit-level)

Assume $q(\boldsymbol{\theta}|\boldsymbol{x}) = \prod_{d=1}^{D} q_d(\theta_d|\boldsymbol{x})$ and $y_d|\theta_d \sim \text{Bernoulli}(\theta_d)$.

$$\mathbb{H}\Big[\mathbb{E}_{\theta_i \sim q_i}\big[y_d|\theta_d\big]\Big] = \underbrace{\mathbb{E}_{\theta_d \sim q_d}\Big[\mathbb{H}[y_d|\theta_d]\Big]}_{\text{aleatoric}} + \underbrace{\mathbb{E}_{\theta_d \sim q_d}\Big[D_{KL}\big(\text{Bern}(\theta_d) \,\|\, \text{Bern}(\bar{\theta}_d)\big)\Big]}_{\text{epistemic}},$$

where $\bar{\theta}_d = \mathbb{E}_{\theta_d}[\theta_d]$ is the **aggregated bit probability**.

Fingerprint level aggregation: Sum over bit, i.e.

$$u_{tot}(\boldsymbol{x}) = \sum_{d=1}^{D} u_{tot,d}(\boldsymbol{x}), \quad u_{al}(\boldsymbol{x}) = \sum_{d=1}^{D} u_{al,d}(\boldsymbol{x}),$$

$$u_{ep}(\boldsymbol{x}) = \sum_{d=1}^{D} u_{ep,d}(\boldsymbol{x}).$$

[4] Depeweg et al., "Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning".

On the retrieval level

For spectrum $\boldsymbol{x}$ with candidate set $\mathcal{C}(\boldsymbol{x}) = \{c_1, \ldots, c_M\}$, each sample $\boldsymbol{\theta}^{(s)} \sim q(\boldsymbol{\theta}|\boldsymbol{x})$ induces a *distribution over candidates*:

$$p_c(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\exp(\mathrm{sim}(\boldsymbol{\theta}, c)/T)}{\sum_{c' \in \mathcal{C}(\boldsymbol{x})} \exp(\mathrm{sim}(\boldsymbol{\theta}, c')/T)}, \quad c \in \mathcal{C}(\boldsymbol{x})$$

On the retrieval level

For spectrum $\boldsymbol{x}$ with candidate set $\mathcal{C}(\boldsymbol{x}) = \{c_1, \ldots, c_M\}$, each sample $\boldsymbol{\theta}^{(s)} \sim q(\boldsymbol{\theta}|\boldsymbol{x})$ induces a *distribution over candidates*:

$$p_c(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\exp(\mathrm{sim}(\boldsymbol{\theta}, c)/T)}{\sum_{c' \in \mathcal{C}(\boldsymbol{x})} \exp(\mathrm{sim}(\boldsymbol{\theta}, c')/T)}, \quad c \in \mathcal{C}(\boldsymbol{x})$$

Information-theoretic decomposition (retrieval)

Let $\hat{y} \in \mathcal{C}(\boldsymbol{x})$ be the predicted candidate. Then:

$$\underbrace{\mathbb{H}\big[\bar{p}(\boldsymbol{x})\big]}_{\text{total}} = \underbrace{\mathbb{E}_{\boldsymbol{\theta}\sim q}\Big[\mathbb{H}\big[p(\boldsymbol{x}, \boldsymbol{\theta})\big]\Big]}_{\text{aleatoric}} + \underbrace{\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[D_{KL}(p(\boldsymbol{x}, \boldsymbol{\theta}) \,\|\, \bar{p}(\boldsymbol{x})\right]}_{\text{epistemic}}$$

where $\bar{p}_c(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\theta}}[p_c(\boldsymbol{x}, \boldsymbol{\theta})]$ is the **aggregated candidate probability**.

Alternative: Rank-based uncertainty

Rank variance (retrieval level)

For each sample $\boldsymbol{\theta}^{(s)}$, let $r^{(s)}(c)$ denote the rank of candidate c. Define:

$$u^{(k)}_{\text{rank-var}}(\boldsymbol{x}) = \frac{1}{k} \sum_{c \in \text{Top-}k} \text{Var}_s\left[r^{(s)}(c)\right]$$

Alternative: Rank-based uncertainty

Rank variance (retrieval level)

For each sample $\boldsymbol{\theta}^{(s)}$, let $r^{(s)}(c)$ denote the rank of candidate c. Define:

$$u^{(k)}_{\text{rank-var}}(\boldsymbol{x}) = \frac{1}{k} \sum_{c \in \text{Top-}k} \text{Var}_s\left[r^{(s)}(c)\right]$$

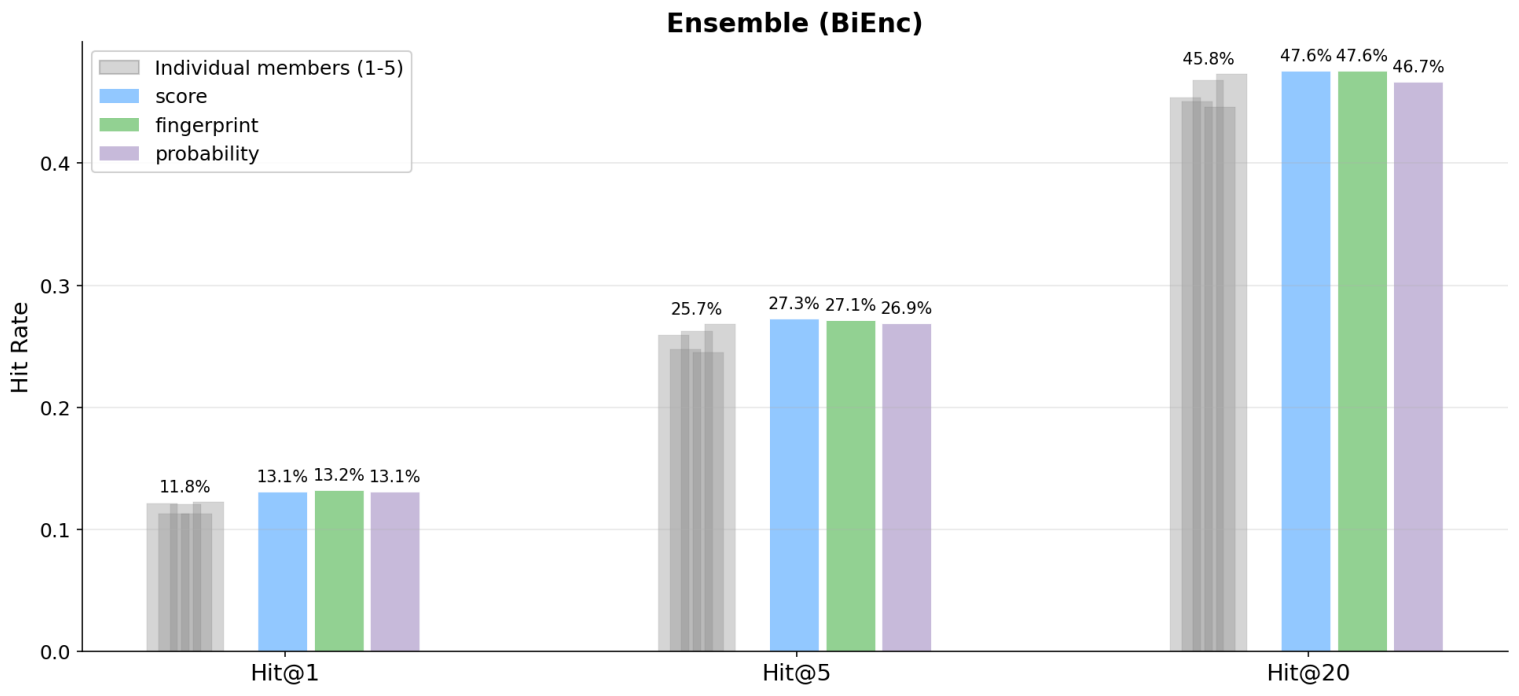
Why rank variance?

- Directly measures *ranking stability*
- Bypasses softmax compression

Experimental results

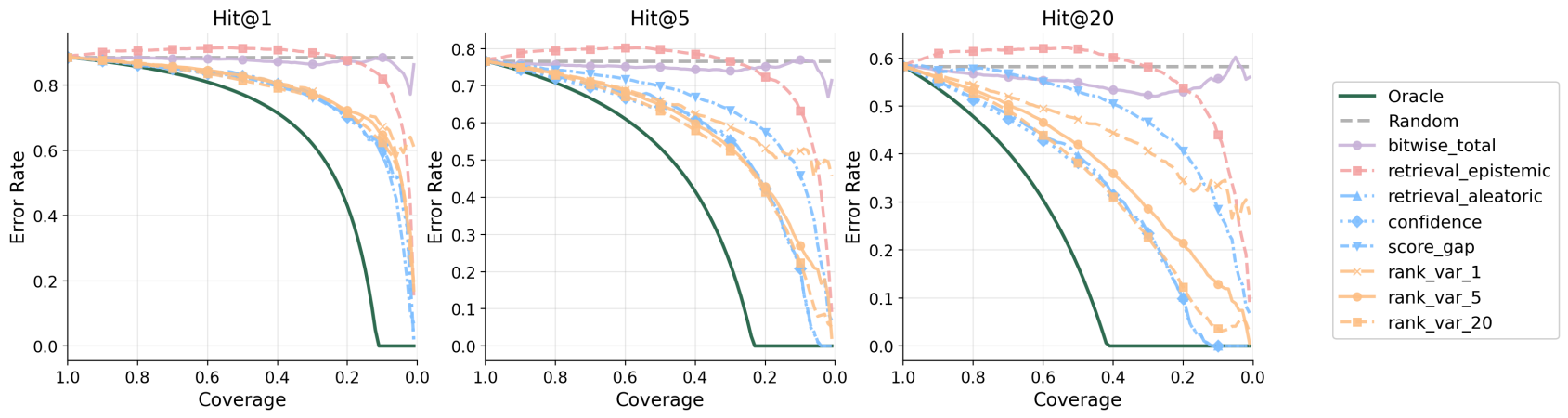
Aggregation of samples leads to improved hit rate

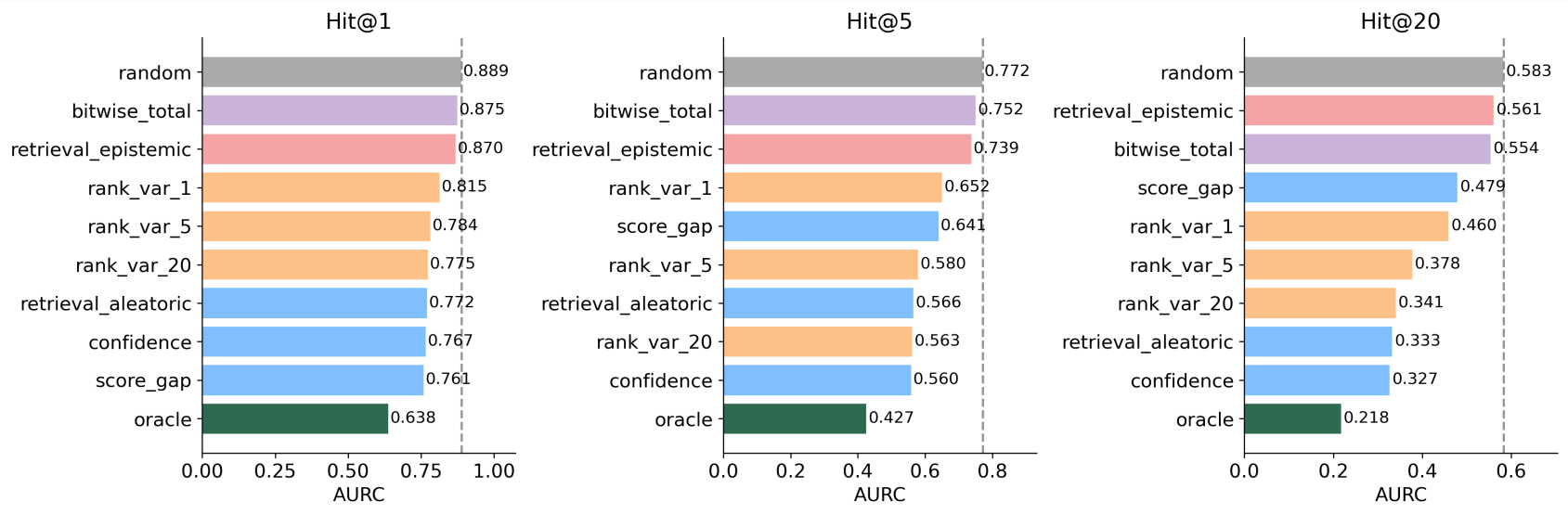
Example: Deep Ensemble



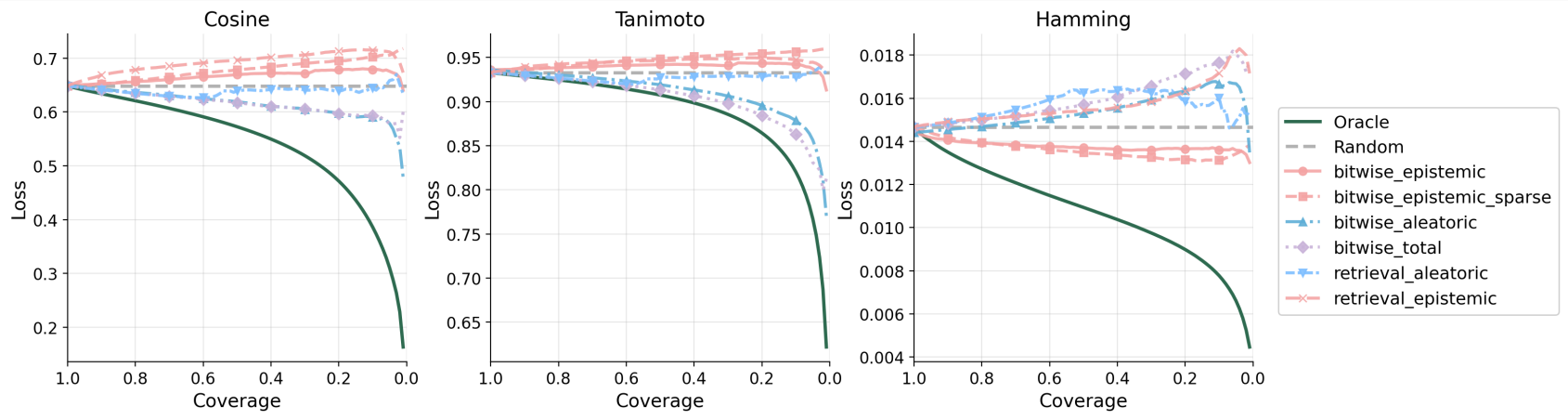
Different aggregation techniques do not seem to greatly influence performance

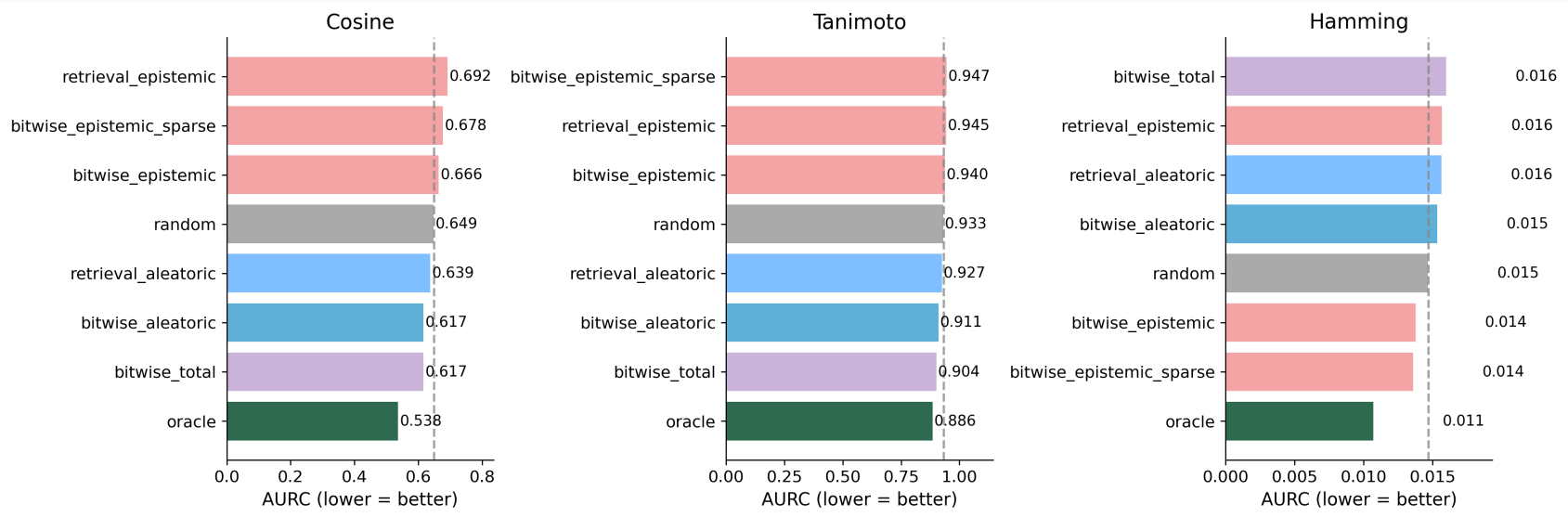
Retrieval: ”First order” retrieval uncertainties beat ”second-order” ones





Fingerprint similarity: Bitwise uncertainties most meaningful





From evaluation to guarantees

Can we guarantee a maximum error rate on the selected samples?

Selective risk control

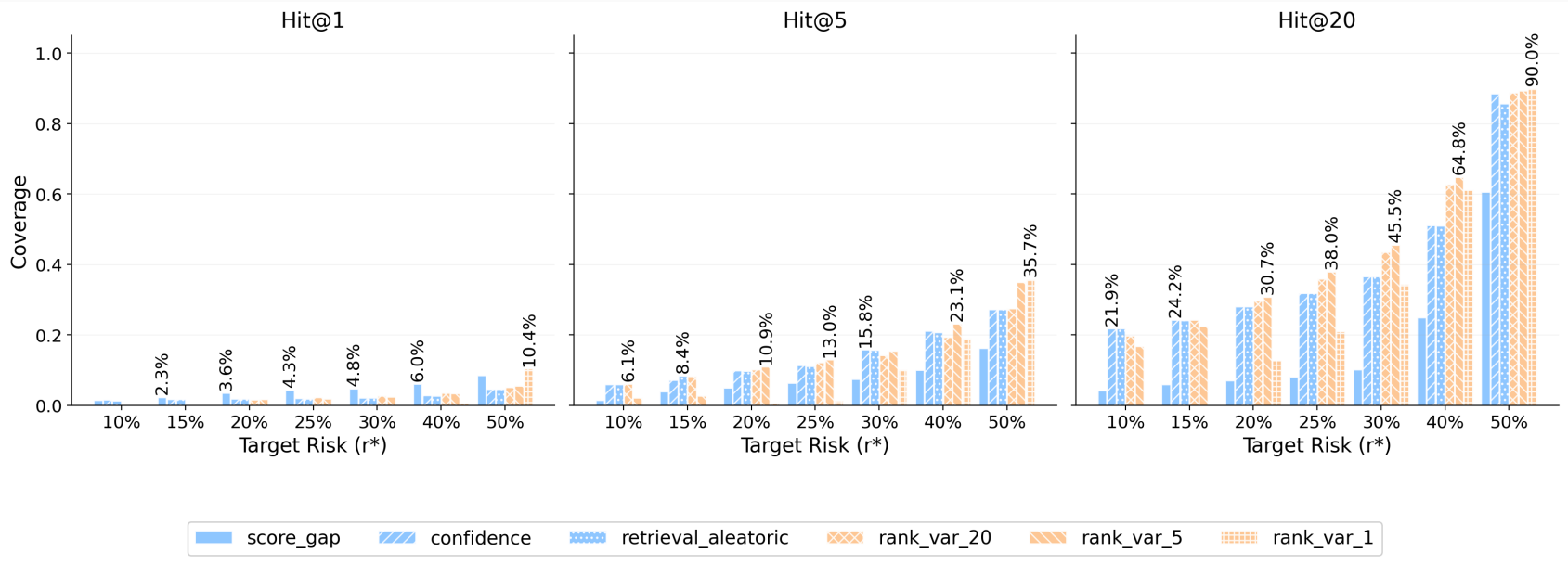
For a given target risk rate r^* and $\delta > 0$, find threshold τ^* such that

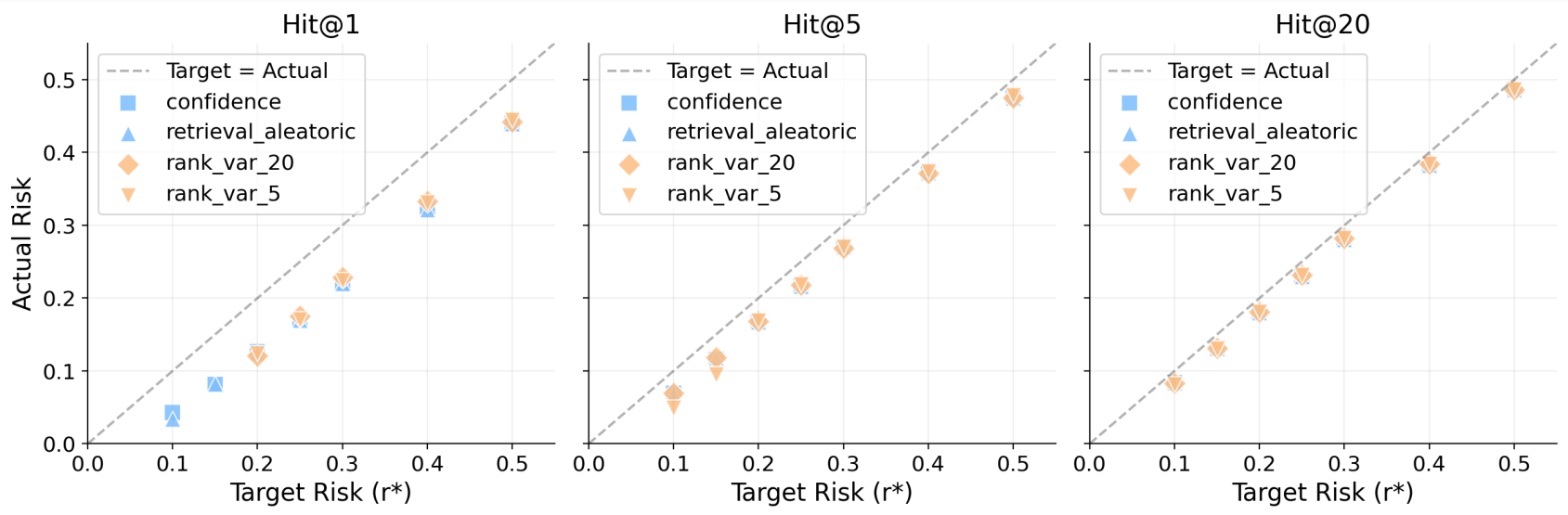
$$\mathbb{P}(R(f, g_{\tau^*}) \geq r^*) < \delta.$$

Method: Use calibration set to find τ^* via binary search and generalisation bounds[5].

[5] Geifman and El-Yaniv, "Selective Classification for Deep Neural Networks".

Selective risk control guarantees for hit rate





Open questions and discussion

Open questions

- What are targeted uncertainty measures for the two-step retrieval task?
- How to properly integrate bitwise and retrieval uncertainty?

Conclusion

- Complex two-stage problem requires a careful modeling of uncertainty
- Similar to Hofman et al.[6]: *"One Size does not fit all"*.

[6] Hofman et al., "Uncertainty Quantification for Machine Learning: One Size Does Not Fit All".