

Generalized Accuracy of Approximated Deep Neural Networks

Jiří Šíma

sima@cs.cas.cz



Institute of Computer Science
Czech Academy of Sciences, Prague, Czechia

joint work with **Jérémie Cabessa**

jeremie.cabessa@uvsq.fr



DAVID Laboratory
Paris-Saclay University, Versailles, France

Accuracy Measure for Approximated DNN Classifiers

DNNs are energy-intensive, limiting their deployment, for example, on battery-powered embedded devices (e.g., smartphones, smartwatches, and augmented reality glasses).

$\longrightarrow$ **reducing the energy cost of DNNs:**

(Sze, Chen, Yang, Emer, Morgan & Claypool Publishers, 2020)

1. Specialized **Hardware Accelerators** for DNN inference, based on GPUs, FPGAs, in-memory computing, etc.

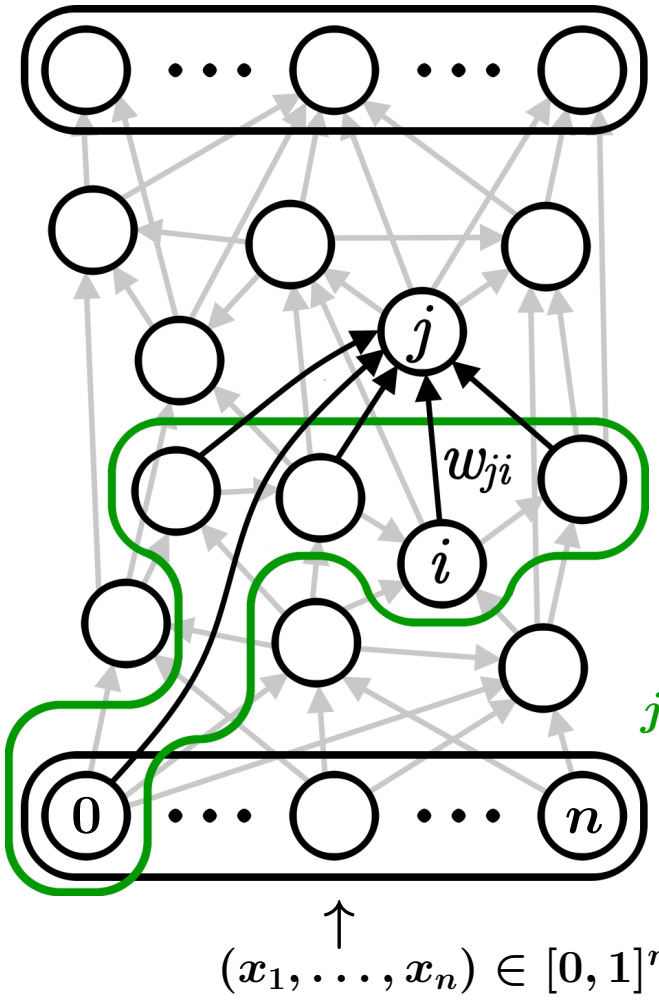
2. **Approximate Computing** in error-tolerant applications (e.g. image classification) save large amounts of energy with minimal accuracy loss by reducing:

- Model Size: pruning, compression, weight sharing, approximate multipliers
- Arithmetic Precision: fixed-point operations, reduced weight bit-width, quantization

The aim of this study: Introduce an **accuracy measure** for approximated DNNs used for classification that would provide stronger reliability guarantees and enable comparison across different approximations.

A Formal Model of Neural Networks (NNs)

The architecture of a network $\mathcal{N}$ is a connected directed acyclic graph (V, E) of ReLU neurons, with each edge $(i, j) \in E \subset V \times V$ weighted by $w_{ji} \in \mathbb{R}$



$Y \subseteq V$ output neurons

$H = V \setminus (X \cup Y)$ hidden neurons

$$y_j = \begin{cases} x_j & \text{if } j \in X \\ R(\xi_j) & \text{if } j \in H \\ \frac{e^{\xi_j}}{\sum_{k \in Y} e^{\xi_k}} & \text{if } j \in Y \text{ sofmax} \end{cases}$$

state (output) of neuron $j \in V$

$\xi_j = \sum_{i \in j_{\leftarrow}} w_{ji} y_i$ excitation of $j \in V \setminus X$

$R(\xi) = \text{ReLU}(\xi) = \max(0, \xi)$ ReLU activation function

$j_{\leftarrow} = \{i \in V \mid (i, j) \in E\}$ inputs to neuron $j \in V \setminus X$

$X = \{0, 1, \ldots, n\} \subseteq V$ input neurons including formal $0 \in X$ $(y_0 \equiv 1)$ for biases w_{j0} for $j \in V \setminus X$

w.l.o.g., excluding (max) pooling layers ($\max(y_1, y_2) = R(y_1 - y_2) + y_2$)

Accuracy of Approximated NNs

$\widetilde{\mathcal{N}}$ is an **approximated** NN of $\mathcal{N}$, sharing the same input neurons ($\widetilde{X} = X$) and the same number of corresponding output neurons ($|\widetilde{Y}| = |Y|$) (tilde notation for $\widetilde{\mathcal{N}}$)

e.g., the weights of $\widetilde{\mathcal{N}}$ are rounded via post-training b-bit quantization

Theorem. *It is NP-hard to maximize the L_1 error $\sum_{j \in Y} |\xi_j - \widetilde{\xi}_j|$ (or the cross-entropy loss) over the entire input space $[0,1]^n$.*

$\rightarrow$ $T \subset [0,1]^n$ is a **dataset** approximating the input probability distribution

each data point $x \in T$ is associated with its corresponding class $c(x) \in Y$

Accuracy of $\mathcal{N}$ on T : $$\alpha_T(\mathcal{N}) = \frac{1}{|T|} \sum_{x \in T} 1\Big(c(x) \in \arg\max_{k \in Y} y_k\Big)$$

where 1 denotes the indicator (equals 1 if its argument is true and 0 otherwise)

We assume that $\mathcal{N}$ solves the classification task perfectly (e.g., $\alpha_T(\mathcal{N}) = 1$).

vs.

Pointwise accuracy $\alpha_T(\widetilde{\mathcal{N}})$ fails to capture local behavior around data points.

Shortcut Weights

The excitation ξ_j of any neuron $j \in V \setminus X$ is a continuous piecewise linear function of the external input (since ReLU is piecewise linear)

$$\rightarrow \quad \xi_j = \sum_{i \in X} W_{ji}\, y_i \quad \text{on a subset } \Xi \subset [0,1]^n \text{ of the input space}$$

W_{ji} is a **shortcut weight** (bias) from input neuron $i \in X$ to neuron $j \in V \setminus X$

For input $x \in [0,1]^n$, let $A(x) = \{j \in H \mid \xi_j > 0\}$ be activated hidden neurons.

$\rightarrow$ The neighborhood Ξ of input x, where ξ_j are linear for all $j \in V \setminus X$,

forms a bounded **convex polytope**—an intersection of half-spaces:

$$\xi_j = \sum_{i \in X} W_{ji}\, y_i \begin{cases} > 0 & \text{if } j \in A(x) \\ \leq 0 & \text{if } j \notin A(x) \end{cases} \quad \text{for } j \in H$$

$$0 \leq y_i \leq 1 \quad \text{for } i \in X$$

Efficient Computation of the shortcut weights via feedforward propagation of the ith one-hot input vector through activated $A(x)$:

$$W_{ji} = w_{ji} + \sum_{k \in j_{\leftarrow} \cap A(x)} w_{jk} W_{ki} \quad \text{for } j \in V \setminus X$$

Correct Polytope and Its Partition

For each data point x in the dataset T, belonging to the class $c = c(x) \in Y$,

the **correct polytope** $\Xi_x \subseteq [0,1]^n$ around x (activating $A(x)$ and $\widetilde{A}(x)$):

- $\mathcal{N}$ behaves linearly:

$$\xi_j = \sum_{i \in X} W_{ji}\, y_i \begin{cases} \geq 0 & \text{if } j \in A(x) \\ \leq 0 & \text{if } j \notin A(x) \end{cases} \quad j \in H$$

- $\widetilde{\mathcal{N}}$ behaves linearly:

$$\widetilde{\xi}_j = \sum_{i \in X} \widetilde{W}_{ji}\, y_i \begin{cases} \geq 0 & \text{if } j \in \widetilde{A}(x) \\ \leq 0 & \text{if } j \notin \widetilde{A}(x) \end{cases} \quad j \in \widetilde{H}$$

- $\mathcal{N}$ classifies x correctly as $c \in Y$:

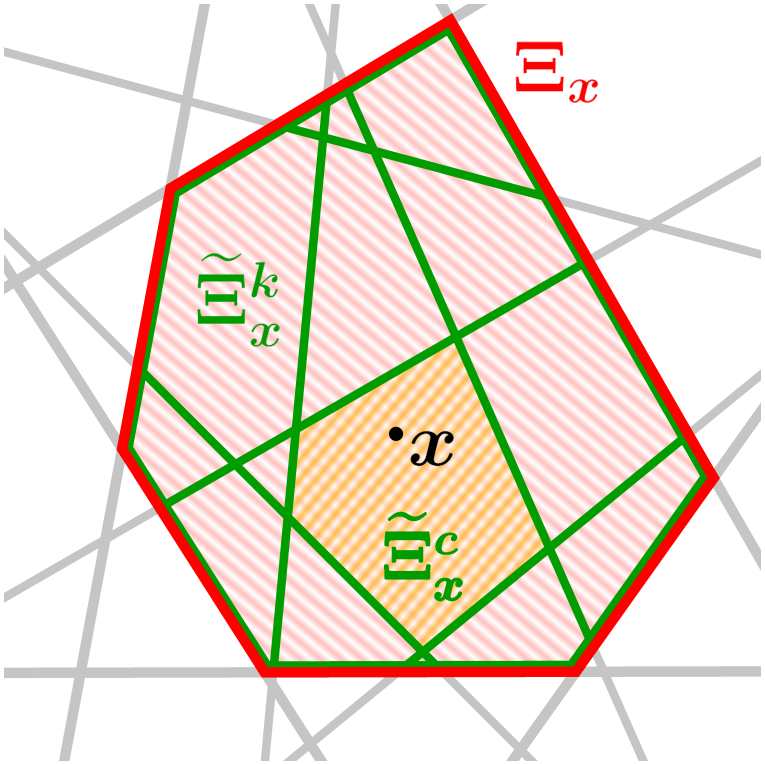
$$\xi_c - \xi_j = \sum_{i \in X} (W_{ci} - W_{ji}) y_i \;\geq\; 0 \quad j \in Y \setminus \{c\}$$

its partition $\Xi_x = \bigcup_{k \in Y} \widetilde{\Xi}^k_x$ into

subpolytopes $\widetilde{\Xi}^k_x$ (incl. correct $\widetilde{\Xi}^c_x$):

- $\widetilde{\mathcal{N}}$ classifies x as $k \in \widetilde{Y}$:

$$\widetilde{\xi}_k - \widetilde{\xi}_j = \sum_{i \in X} (\widetilde{W}_{ki} - \widetilde{W}_{ji}) y_i \;\geq\; 0 \quad j \in \widetilde{Y} \setminus \{k\}$$



Idea of Generalized Accuracy

$$\frac{1}{|T|} \sum_{x \in T} \frac{\text{volume}\left(\widetilde{\Xi}^c_x\right)}{\text{volume}\left(\Xi_x\right)}$$

Estimated Mean Width as a Polytope Volume Surrogate

Computing polytope volumes in high dimensions ($n \sim 1000$) is infeasible (#P-hard; no poly-factor approximation; randomized methods limited to low n).

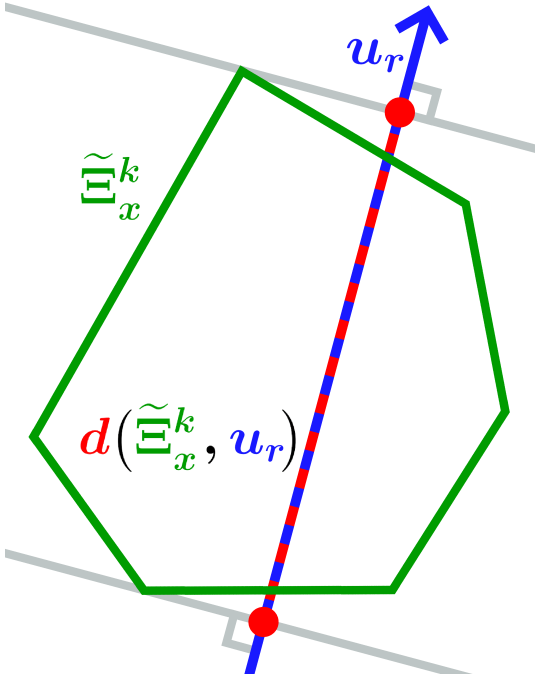
$\rightarrow$ surrogate measure: the **mean width** $\overline{d}(\widetilde{\Xi}_x^k)$ of polytope $\widetilde{\Xi}_x^k$

$$\overline{d}(\widetilde{\Xi}_x^k) = \frac{1}{\omega_{n-1}} \int_{\mathcal{S}_{n-1}} d(\widetilde{\Xi}_x^k, u)\, du \quad \sim \quad \widetilde{d}(\widetilde{\Xi}_x^k) = \frac{1}{N} \sum_{r=1}^{N} d(\widetilde{\Xi}_x^k, u_r)$$

(ω_{n-1} denotes the surface area of the unit sphere $\mathcal{S}_{n-1} \subset \mathbb{R}^n$)

$$d\left(\widetilde{\Xi}_x^k, u_r\right) = \max_{y \in \widetilde{\Xi}_x^k} \sum_{i=1}^{n} u_{ri} y_i - \min_{y \in \widetilde{\Xi}_x^k} \sum_{i=1}^{n} u_{ri} y_i$$

- u_r is an independent uniform sample from $\mathcal{S}_{n-1}$
- evaluated by two linear programs over $\widetilde{\Xi}_x^k$



$$\frac{\text{volume}\left(\widetilde{\Xi}_x^c\right)}{\text{volume}\left(\Xi_x\right)} \quad \rightsquigarrow \quad \frac{\widetilde{d}_0\left(\widetilde{\Xi}_x^c\right)}{\sum_{k \in Y} \widetilde{d}_0\left(\widetilde{\Xi}_x^k\right)}$$

where $\widetilde{d}_0\left(\widetilde{\Xi}_x^k\right)$ equals $\widetilde{d}\left(\widetilde{\Xi}_x^k\right)$ if $\text{volume}\left(\widetilde{\Xi}_x^k\right) > 0$ and 0 otherwise (degenerate polytopes)

Generalized Accuracy

$$\gamma_T\left(\widetilde{\mathcal{N}}\right) = \frac{\sum_{x \in T} \widetilde{d}_0\left(\widetilde{\Xi}_x^{c(x)}\right)}{\sum_{x \in T,\, k \in Y} \widetilde{d}_0\left(\widetilde{\Xi}_x^{k}\right)} \qquad \text{averaged over the dataset } T$$

$0 \leq \gamma_T(\widetilde{\mathcal{N}}) \leq 1$, where $\gamma_T(\widetilde{\mathcal{N}}) = 1$ corresponds to a perfect approximation

It depends on the correct polytopes $\Xi_x = \bigcup_{k \in Y} \widetilde{\Xi}_x^k$ whose sizes vary with $\widetilde{\mathcal{N}}$,
i.e., $\gamma_T(\widetilde{\mathcal{N}})$ not directly comparable across different approximated networks $\widetilde{\mathcal{N}}$.

$\rightarrow$ **Extended** correct polytope $\overline{\Xi}_x \supseteq \Xi_x$ ($\overline{\Xi}_x \subseteq [0,1]^n$) around x unique for all $\widetilde{\mathcal{N}}$
where $\mathcal{N}$ behaves linearly (not necessarily $\widetilde{\mathcal{N}}$)
and classifies x as $c = c(x) \in Y$

To evaluate $\gamma_T(\widetilde{\mathcal{N}})$, we partition the extended $\overline{\Xi}_x$ into the correct polytopes Ξ_y
(where $\widetilde{\mathcal{N}}$ behaves linearly) by augmenting the dataset T, using

the **Hit-and-Run Markov Chain Monte Carlo** method

Hit-and-Run Markov Chain Monte Carlo (MCMC) Method

For each data point $x_0 = x \in T$, we generate $T'_x = \{x_1, \ldots, x_P\}$ as

$$x_r = x_{r-1} + \lambda_r u_r \qquad \text{for } r \in \{1, \ldots, P\}$$

where u_r from $\mathcal{S}_{n-1}$ and λ_r such that $x_r \in \overline{\Xi}_x$, both sampled uniformly at random

equivalence relation $\sim$ on T'_x: $x_r \sim x_s$ iff $\Xi_{x_r} = \Xi_{x_s}$ (i.e., $\widetilde{A}(x_r) = \widetilde{A}(x_s)$)

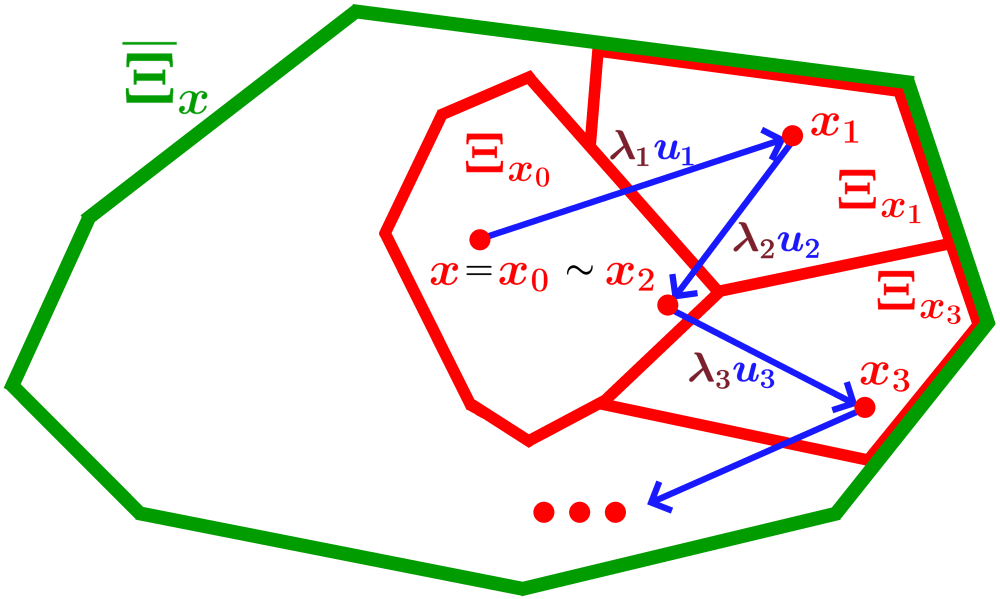
Let $T_x \subseteq T'_x$ be representatives of the equivalence classes of $T'_x/\!\sim$, with $x_0 \in T_x$.

$\rightarrow$ We augment the dataset as $T_P = \bigcup_{x \in T} T_x \supseteq T = T_0$ for sufficiently large P.

Partition of the extended polytope

$$\overline{\Xi}_x = \bigcup_{y \in T^*_x} \Xi_y$$

where $T^*_x = \overset{\text{a.s.}}{\lim}_{P \to \infty} T_x < \infty$ $\quad (|T'_x/\!\sim| < 2^{|\widetilde{H}|})$

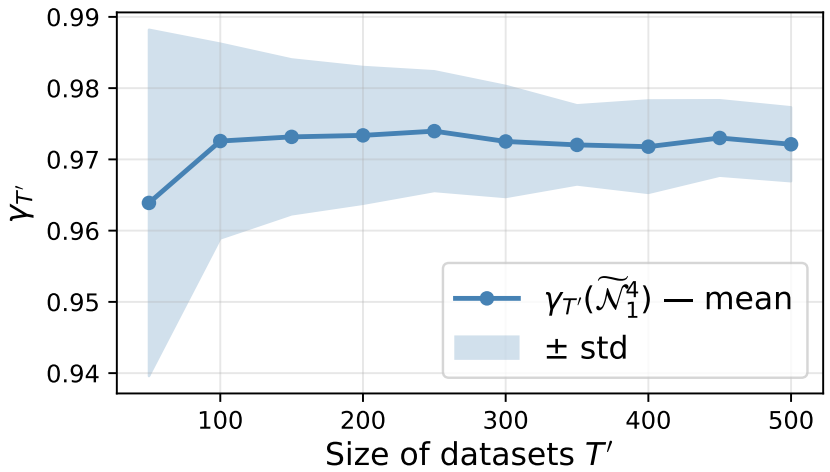


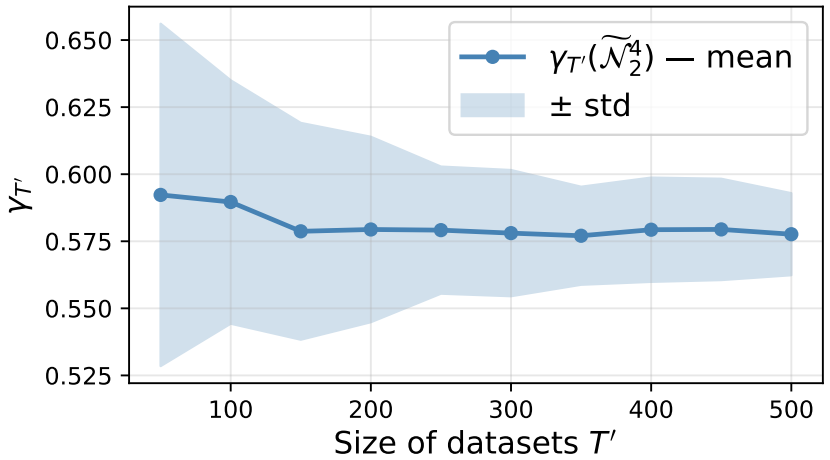
Generalized accuracy $\gamma_{T^*}(\widetilde{\mathcal{N}})$ is evaluated over $T^* = \bigcup_{x \in T} T^*_x \sim T_P$.

Experiments: Correctness of the $\gamma_T(\widetilde{\mathcal{N}})$ Definition

- Fashion-MNIST Classification Task: 28×28 grayscale images, 10 clothing classes
- Five-layer $\mathcal{N}_1$ (784–1024–512–256–128–10) & Convolutional $\mathcal{N}_2$ ($\mapsto$ ten-layer NN 784–7840–7840–3920–1960–3920–3920–1960–980–96–10), both trained with 32-bit weights using 60,000 training and 10,000 validation images
- Dataset T: 1,000 (out of training) data points with $\alpha_T(\mathcal{N}_1) = \alpha_T(\mathcal{N}_2) = 1$
- Approximated Networks: $\widetilde{\mathcal{N}}_1^b$ and $\widetilde{\mathcal{N}}_2^b$ derived from $\mathcal{N}_1$ and $\mathcal{N}_2$ via uniform symmetric post-training b-bit quantization of weights, for $b \in \{4, 6, 8, 10, 12, 16\}$

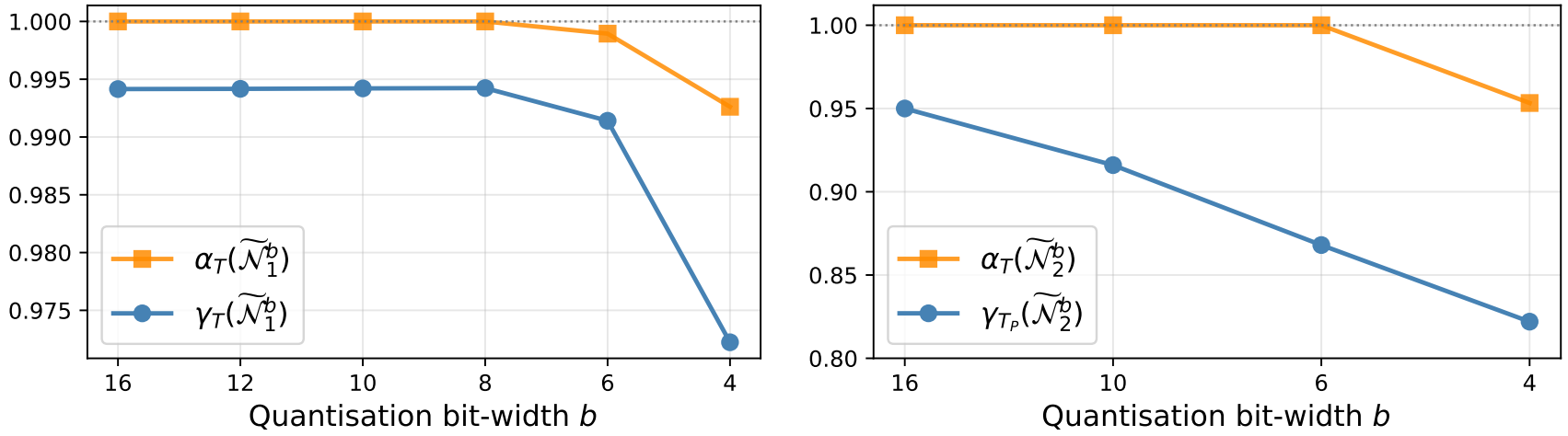
Correctness of the Generalized Accuracy: $\gamma_{T'}(\widetilde{\mathcal{N}}_1^4)$ and $\gamma_{T'}(\widetilde{\mathcal{N}}_2^4)$ converge for samples $T' \subset T$ of sizes $|T'| = 50, 100, \ldots, 500$, averaged over 100 samplings, $N = 200$ random directions for mean width $\longrightarrow$ $|T'| = 150$, $N = 100$ suffice





Experiments: Pointwise vs. Generalized Accuracy

the effect of reducing the weight bitwidth b on the generalized accuracy γ_T compared to the standard pointwise accuracy α_T for the approximated $\widetilde{\mathcal{N}}_1^b$ and $\widetilde{\mathcal{N}}_2^b$



$\gamma_T(\widetilde{\mathcal{N}}_1^b)$ computed for the dataset T (since $\widetilde{d}(\Xi_x)/\widetilde{d}(\overline{\Xi}_x) \approx 1$ for all $x \in T$)

$\gamma_{T_P}(\widetilde{\mathcal{N}}_2^b)$ evaluated on a sampled T ($|T| = 150$, $N = 100$) augmented to T_P via MCMC with large enough P so that each extended polytope $\overline{\Xi}_x$ is covered by at most 5 distinct correct polytopes Ξ_y (e.g., $\widetilde{d}(\Xi_x)/\widetilde{d}(\overline{\Xi}_x) \approx 0.5$ for $b = 4$)

$\rightarrow \quad |T_P| = 159, 158, 206, 900$ for $b = 16, 10, 6, 4$

Conclusion: The generalized accuracy $\gamma_T(\widetilde{\mathcal{N}}^b)$ reveals degraded generalization near data points in T with decreasing weight bitwidth b, even when standard pointwise accuracy $\alpha_T(\widetilde{\mathcal{N}}^b)$ remains largely preserved.