Distributionally Robust Safety Verification of Neural Networks via Worst-Case CVaR

Masako Kishida, Senior Member, IEEE

Abstract—Ensuring the safety of neural networks under input uncertainty is a fundamental challenge in safety-critical applications. This paper builds on and expands Fazlyab's quadraticconstraint (QC) and semidefinite-programming (SDP) framework for neural network verification to a distributionally robust and tail-risk-aware setting by integrating worst-case Conditional Value-at-Risk (WC-CVaR) over a moment-based ambiguity set with fixed mean and covariance. The resulting conditions remain SDP-checkable and explicitly account for tail risk. This integration broadens input-uncertainty geometry—covering ellipsoids, polytopes, and hyperplanes—and extends applicability to safetycritical domains where tail-event severity matters. Applications to closed-loop reachability of control systems and classification are demonstrated through numerical experiments, illustrating how the risk level ε trades conservatism for tolerance to tail events-while preserving the computational structure of prior QC/SDP methods for neural network verification and robustness analysis.

Index Terms—Distributionally robust optimization, conditional value-at-risk (CVaR), quadratic constraints, semidefinite programming, reachability, neural network verification, tail risk.

I. INTRODUCTION

Risk analysis and management are crucial in the development and operation of safety-critical systems such as automotive systems [2], avionics systems [3], and medical devices [4]. These systems require rigorous safety guarantees despite inherent uncertainties, as their failure can lead to catastrophic consequences, including loss of life, substantial property damage, or environmental harm. Major approaches to handling uncertainties generally fall into two types: deterministic setbased approaches that provide worst-case guarantees [5]–[7] and probabilistic/stochastic approaches that explicitly represent uncertainty using probability distributions and impose performance goals using such as chance constraints [8] or expected values [9]. While effective in many settings, both approaches may fail to capture rare but severe events—tail risks—that are unacceptable in safety-critical contexts.

In recent years, deep neural networks (DNNs) have become ubiquitous in control and decision-making systems. However, DNNs are known to be vulnerable to adversarial perturbations [10]–[12]. This has motivated extensive research on verification and robustness analysis of DNNs [13]–[26]. Existing DNN verification approaches can be broadly categorized into three classes:

This work was supported by JST, PRESTO Grant Number JPMJPR22C3, Japan.

A preliminary version of this paper appears in the proceedings of IEEE Conference on Decision and Control 2025 [1].

The author is with National Institute of Informatics, Tokyo Japan (e-mail: kishida@ieee.org).

ChatGPT was used to improve writing.

- (i) Exact methods, such as SMT/MILP solvers [13], [14] and branch-and-bound algorithms [15]. They provide sound and complete guarantees.
- (ii) Bound-propagation methods, including abstract interpretation and forward-backward propagation [16], as well as recent frameworks that use the backward mode linear relaxation based perturbation analysis [17]. These typically yield conservative bounds.
- (iii) Convex optimization methods, e.g., quadratic-constraint and semidefinite-programming (QC/SDP)-based analysis [18]–[20] and control-theoretic stability analysis of recurrent networks [21]. They offer elegant convex formulations and strong theoretical foundations [22]. Recent work also explores hybrid bound-propagation–SDP approaches that inject SDP-derived inter-neuron coupling into scalable bound propagation [23].

While these approaches have advanced verification significantly, they do not explicitly quantify the severity of lowprobability catastrophic failures.

In this paper, we extend Fazlyab's QC/SDP methods in (iii) by incorporating the Worst-Case Conditional Value-at-Risk (WC-CVaR) [27], thereby enabling tail-risk-aware safety verification under distributional uncertainty. Originally developed in finance [28], [29], CVaR and its worst-case variant have recently been applied in control [30]-[32] and machine learning [33]-[35]. WC-CVaR quantifies the expected loss in the worst ε -tail over all distributions sharing prescribed mean and covariance. This distributional viewpoint is closely related to the literature on distributionally robust optimization (DRO), where ambiguity sets defined by moment information or other statistics are used to model distributional uncertainty [36], [37]. It avoids reliance on a specific model (e.g., Gaussian) and is attractive because (a) moments are often available from data, and (b) the induced verification conditions remain SDPcheckable via QCs. Incorporating WC-CVaR provides a systematic trade-off between conservatism and tail-risk tolerance, while preserving SDP tractability. It also generalizes inputuncertainty descriptions beyond ellipsoids [20] to polytopes and hyperplanes, broadening applicability (Table I).

The contributions of this paper are as follows:

- We extend Fazlyab's QC/SDP methods by incorporating WC-CVaR, yielding a distributionally robust method for neural network safety verification.
- We establish a connection between WC-CVaR and confidence ellipsoid methods [20] (Lemma 4.1), showing that the two safe sets coincide when those sets are restricted to a special ellipsoid.
- We illustrate the proposed methods through numerical

Category	Fazlyab et al., 2022 [18]	Fazlyab et al., 2019 [20]	This paper
Approach	Norm bounded	Confidence ellipsoid propagation	Assessment by Worst-Case Conditional Value-at-Risk (WC-CVaR)
Uncertainty	Bounded sets	Gaussian as well as random vectors with known mean and covariance	Distributional uncertainty: ambiguity set with known mean and covariance, tail risk considered
Input geometry	Those that can be expressed using affine or quadratic functions	Ellipsoids (use Chebyshev's in- equality)	Those that can be expressed using affine or quadratic functions
Output geometry	Those that can be expressed using affine or quadratic functions	Ellipsoids	Those that can be expressed using affine or quadratic functions
Metric	Worst-case safety (all admissible inputs must satisfy specification)	p-level confidence region	WC-CVaR, explicitly accounting for tail risk

TABLE I: Comparison of three QC/SDP-based neural network verification approaches.

experiments on closed-loop reachability and classification problems, highlighting performance under both light- and heavy-tailed input distributions.

The paper is organized as follows: Section II introduces notation and WC-CVaR fundamentals. Section III presents our risk-aware neural network verification framework. Section IV discusses control and classification applications with numerical experiments. Finally, Section V concludes the paper.

II. PRELIMINARIES

A. Notation

Let \mathbb{R} , \mathbb{R}^n , $\mathbb{R}^{n \times m}$, and \mathbb{S}^n denote the sets of real numbers, n-dimensional real vectors, $n \times m$ real matrices, and $n \times n$ symmetric matrices, respectively. For $v \in \mathbb{R}^n$, let ||v|| be its Euclidean norm, and for $M \in \mathbb{R}^{n \times m}$, let M^{\top} be its transpose and $\mathrm{Tr}(M)$ its trace. $M \succ 0$ indicates that $M \in \mathbb{S}^n$ is positive definite. The all-ones vector in \mathbb{R}^n and the identity matrix in $\mathbb{R}^{n \times n}$ are denoted by $\mathbf{1}_n$ and I_n the $n \times n$, respectively, and subscripts are dropped when the sizes are clear. For $x \in \mathbb{R}$, define $(x)^+ := \max\{x, 0\}$. $\mathbb{E}_{\mathbb{P}}[x]$ and $\mathrm{Cov}_{\mathbb{P}}[x]$ denote the mean and covariance of a random vector $x \sim \mathbb{P}$, subscripts are dropped when the underlying distribution is clear.

B. Worst-Case Conditional Value-at-Risk

CVaR at risk level ε is the expected value beyond a certain high percentile, measuring the tail risk. Since the exact distribution is often unknown, we consider an ambiguity set of possible distributions \mathcal{P} . WC-CVaR quantifies the worst-case tail risk at level ε over \mathcal{P} . This conservative metric guarantees robustness against any distribution in \mathcal{P} , not just a nominal one.

We first define the ambiguity set and augmented moment matrix. Let $\xi \in \mathbb{R}^n$ be a random vector with distribution \mathbb{P} . The moment-based ambiguity set is defined as

$$\mathcal{P}(\mu, \Sigma) := \{ \mathbb{P} : \mathbb{E}_{\mathbb{P}}[\xi] = \mu, \operatorname{Cov}_{\mathbb{P}}[\xi] = \Sigma \}, \qquad (1)$$

i.e., all distributions that share the same mean μ and covariance Σ , including both light-tailed and heavy-tailed families.

The augmented second-order moment matrix of the lifted variable $\bar{\xi}:=[\xi^\top,\,1]^\top$ is

$$\Omega = \begin{bmatrix} \Sigma + \mu \mu^{\top} & \mu \\ \mu^{\top} & 1 \end{bmatrix}, \tag{2}$$

which compactly represents the second-order information of $\bar{\xi}$. For brevity, we write \mathcal{P} for $\mathcal{P}(\mu, \Sigma)$.

Definition 2.1 (WC-CVaR [27]): For a measurable loss function $L: \mathbb{R}^n \to \mathbb{R}$ and a risk level $\varepsilon \in (0,1)$, the Worst-Case Conditional Value-at-Risk (WC-CVaR) over the ambiguity set \mathcal{P} is defined as:

$$\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\text{-CVaR}_{\varepsilon}[L(\xi)] := \inf_{\beta\in\mathbb{R}} \left\{ \beta + \frac{1}{\varepsilon} \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[(L(\xi) - \beta)^{+}] \right\}.$$
(3)

WC-CVaR quantifies the tail risk under the worst-case probability distribution within \mathcal{P} . In this paper, the risk level ε serves as a tuning parameter, where a smaller ε makes the constraint more conservative, providing a trade-off between risk aversion and performance.

WC-CVaR is one of the coherent risk measure and satisfies the monotonicity property, which we will utilize.

Proposition 2.1 (Monotonicity property [29], [38]): For measurable loss functions $L_1(\xi)$ and $L_2(\xi)$, if $L_1(\xi) \leq L_2(\xi)$ almost surely, then

$$\sup_{\mathbb{P}\in\mathcal{P}}\mathbb{P}\text{-CVaR}_{\varepsilon}[L_1(\xi)]\leq \sup_{\mathbb{P}\in\mathcal{P}}\mathbb{P}\text{-CVaR}_{\varepsilon}[L_2(\xi)].$$

Other coherent risk measures could extend Fazlyab's framework, but WC-CVaR offers superior computational tractability through Proposition 2.2.

Proposition 2.2 (Quadratic Loss Function [27], [39]): For $L(\xi) = \xi^{\top} \Pi \xi + 2\theta^{\top} \xi + \rho$, where $\Pi \in \mathbb{S}^n$, $\theta \in \mathbb{R}^n$, and $\rho \in \mathbb{R}$, the WC-CVaR is given by:

$$\begin{split} \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{P}\text{-CVaR}_{\varepsilon}[L(\xi)] &= \inf_{\beta} \left\{ \beta + \frac{1}{\varepsilon} \mathrm{Tr}(\Omega N) : \\ N &\succcurlyeq 0, \;\; N - \left[\begin{array}{cc} \Pi & \theta \\ \theta^{\top} & \rho - \beta \end{array} \right] \succcurlyeq 0 \right\}. \end{split}$$

III. RISK-AWARE NEURAL NETWORK VERIFICATION

This section proposes risk-aware DNN safety verification method that incorporates WC-CVaR. We also discuss its complexity and extensions.

A. Preparations

In the rest of this paper, let $x \sim \mathbb{P} \in \mathcal{P}(\mu, \Sigma)$ be a random vector of length n. The augmented second-order moment matrix of $[x^{\top}, 1]^{\top}$ is given by (2).

For safety verification, we utilize QCs, which were developed in robust control [40]. In particular, we extend the approaches by Fazlyab et al. [18]–[20] to account for tail risks as follows.

Definition 3.1 (Risk-aware QCs): For a symmetric, possibly indefinite matrix $H \in \mathbb{S}^{n+1}$, we define a risk-aware QC (with WC-CVaR risk level ε) by

$$\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\text{-CVaR}_{\varepsilon} \left[\begin{bmatrix} x \\ 1 \end{bmatrix}^{\top} H \begin{bmatrix} x \\ 1 \end{bmatrix} \right] \leq 0. \tag{4}$$

Let $\mathcal{H}_x \subset \mathbb{S}^{n+1}$ be a set of H that satisfies (4). Then, we say the ambiguity set \mathcal{P} satisfies the QC defined by \mathcal{H}_x .

This means that, for most $x \sim \mathbb{P} \in \mathcal{P}$, it holds that

$$\begin{bmatrix} x \\ 1 \end{bmatrix}^{\top} H \begin{bmatrix} x \\ 1 \end{bmatrix} \le 0,$$
 (5)

and even the mean of the worst 100ε % of x satisfies (5).

Here, we note the condition (4) can be checked by solving a SDP using Proposition 2.2.

B. Neural Network Model

We consider an ℓ -hidden-layer feed-forward fully connected neural network $f: \mathbb{R}^n \to \mathbb{R}^m$ defined by

$$x^{0} = x,$$

 $x^{k+1} = \phi(W^{k}x^{k} + b^{k}), \ k = 0, \dots, \ell - 1,$ (6)
 $f(x) = W^{\ell}x^{\ell} + b^{\ell},$

where $x^0=x\in\mathbb{R}^n$ is the input, $x^k\in\mathbb{R}^{n_k}$ is the output of the kth layer, $W^k\in\mathbb{R}^{n_{k+1}\times n_k}$ and $b^k\in\mathbb{R}^{n_{k+1}}$ are the weight matrix and bias vector for the (k+1)th layer, and $W^\ell\in\mathbb{R}^{m\times n_\ell},\ b^\ell\in\mathbb{R}^m$ define the final affine map.

The activation function ϕ is applied element-wise:

$$\phi(x) = \begin{bmatrix} \varphi(x_1) & \dots & \varphi(x_{n_k}) \end{bmatrix}^\top, \ x \in \mathbb{R}^{n_k}. \tag{7}$$

Assuming identical activation functions across layers, the model (6) can be rewritten in a compact form:

$$x = \mathbf{E}^{0}\mathbf{x},$$

$$\mathbf{B}\mathbf{x} = \phi(\mathbf{A}\mathbf{x} + \mathbf{b}),$$

$$f(x) = W^{\ell}\mathbf{E}^{\ell}\mathbf{x} + b^{\ell},$$
(8)

using the entry selector matrices $\mathbf{E}^k \in \mathbb{R}^{n_k \times \sum_{i=0}^{\ell} n_i}$ and

$$\mathbf{x} = [x^{0\top} \cdots x^{\ell\top}]^{\top} \in \mathbb{R}^{\sum_{i=0}^{\ell} n_i},$$

$$x^k = \mathbf{E}^k \mathbf{x}, \ k = 0, \dots, \ell,$$

$$\mathbf{A} = \left[\operatorname{diag}(W^0, W^1, \dots, W^{\ell-1}) \quad 0\right],$$

$$\mathbf{b} = \begin{bmatrix} b^{0^{\top}} & b^{1^{\top}} & \cdots & b^{\ell-1^{\top}} \end{bmatrix}^{\top},$$

$$\mathbf{B} = \begin{bmatrix} 0 & I_{\sum_{i=1}^{\ell} n_i} \end{bmatrix}.$$
(9)

C. Risk-Aware QC for Input

The input $x \sim \mathbb{P} \in \mathcal{P}$ in (8) is characterized using the set \mathcal{H}_x , which contains symmetric indefinite matrices H = -P that satisfy the following risk-aware QC ¹:

$$\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\text{-CVaR}_{\varepsilon} \left[- \begin{bmatrix} x \\ 1 \end{bmatrix}^{\top} P \begin{bmatrix} x \\ 1 \end{bmatrix} \right] \le 0. \tag{10}$$

If (10) holds, then we say \mathcal{P} satisfies the risk-aware QC defined by \mathcal{H}_x .

This QC encodes the risk-aware safety verification's assumption that the inputs lie within a risk-bounded set.

D. QC for Activation Function

Most standard activation functions, including sector-bounded, slope-restricted, and bounded functions, can be described using QCs [18]–[20]. Because the activation function we consider itself is deterministic, we have QCs instead of risk-aware QCs for the activation functions.

Let $\phi: \mathbb{R}^d \to \mathbb{R}^d$, and suppose $\mathcal{Q}_{\phi} \subset \mathcal{S}^{2d+1}$ is the set of symmetric indefinite matrices Q such that the inequality

$$\begin{bmatrix} z \\ \phi(z) \\ 1 \end{bmatrix}^{\top} Q \begin{bmatrix} z \\ \phi(z) \\ 1 \end{bmatrix} \ge 0 \tag{11}$$

holds for all $z \in \mathbb{R}^d$. Then, we say ϕ satisfies QC defined by \mathcal{Q}_{ϕ} .

Intuitively, this QC bounds the possible values of the hidden-layer signals z and their activations $\phi(z)$, ensuring the nonlinearity does not produce extreme outputs outside a known sector.

Constructing a tight \mathcal{Q}_{ϕ} can be challenging. However, ReLU has well-established QCs using slope and repeated nonlinearity constraints, while Sigmoid and Tanh benefit from sector and local QCs for tighter bounds. Preprocessing techniques further improve accuracy by refining local activation bounds [18]. Once \mathcal{Q}_{ϕ} is fixed and ϕ satisfies the QC defined by \mathcal{Q}_{ϕ} , we can utilize (11) to bound the behavior of activation functions. See Appendix V-A for the exact expression of Q in the case of ReLU.

E. Risk-Aware QC for Safety Verification

The risk-aware safe set for the output f(x) of the neural network (8) can be formulated similarly to the input constraint using a symmetric indefinite matrix S. Specifically, we consider the following risk-aware QC:

$$\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\text{-CVaR}_{\epsilon} \begin{bmatrix} x \\ f(x) \\ 1 \end{bmatrix}^{\top} S \begin{bmatrix} x \\ f(x) \\ 1 \end{bmatrix} \le 0.$$
 (12)

If (12) holds, then we say f(x) satisfies the risk-aware QC defined by S.

This QC ensures that the network's final-layer output remains within a set constrained by a risk-aware bound.

¹The negative sign follows the convention in [18]–[20].

F. Sufficient Risk-aware Safety Condition

With the neural network model (8) and QCs (10)-(12) defined, we now present the main result. The following theorem states sufficient conditions for verifying that a neural network meets the risk-aware safety specification (12) under input uncertainties (10).

Theorem 3.1: Consider the neural network (8). Suppose that \mathcal{H}_x and \mathcal{Q}_ϕ are given. Assume that \mathcal{P} satisfies the risk-aware QC defined by \mathcal{H}_x and $\phi(z)$ satisfies the QC defined by \mathcal{Q}_ϕ . If a symmetric matrix S satisfies the linear matrix inequality (LMI)

$$M_{\rm in}(P) + M_{\rm mid}(Q) + M_{\rm out}(S) \le 0, \tag{13}$$

where

$$M_{\text{in}}(P) = \begin{bmatrix} \mathbf{E}^{0} & 0 \\ 0 & 1 \end{bmatrix}^{\mathsf{T}} P \begin{bmatrix} \mathbf{E}^{0} & 0 \\ 0 & 1 \end{bmatrix},$$

$$M_{\text{mid}}(Q) = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{B} & 0 \\ 0 & 1 \end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{B} & 0 \\ 0 & 1 \end{bmatrix},$$

$$M_{\text{out}}(S) = \begin{bmatrix} \mathbf{E}^{0} & 0 \\ W^{\ell} \mathbf{E}^{\ell} & \mathbf{b}^{\ell} \\ 0 & 1 \end{bmatrix}^{\mathsf{T}} S \begin{bmatrix} \mathbf{E}^{0} & 0 \\ W^{\ell} \mathbf{E}^{\ell} & \mathbf{b}^{\ell} \\ 0 & 1 \end{bmatrix}$$

$$(14)$$

for some $-P \in \mathcal{H}_x$, $Q \in Q_{\phi}$, then the output satisfies the risk-aware safety constraint (12).

Here, S encodes a fixed safety specification to be checked, and the optimization variables are P and Q. In practice, once the input set is specified, the matrix P is typically fixed, and one may optimize over Q to reduce conservatism in the output bound.

G. Computing a Minimum-volume Ellipsoidal Safe Set

For a fixed neural network and input constraint with P, when the output variable for safety verification is defined as a linear map of the input and the output of the neural network,

$$y = C \begin{bmatrix} x \\ f(x) \end{bmatrix} \in \mathbb{R}^{m_y}, \tag{15}$$

a minimum volume ellipsoidal safe set for y, s.t.

$$\sup_{\mathbb{P} \in \mathcal{P}} \mathbb{P}\text{-CVaR}_{\varepsilon} \left[y^{\top} E^{-1} y - 1 \right] \le 0. \tag{16}$$

with $E \succ 0$, can be obtained via convex optimization

$$\min_{E \succ 0} -\log \det(E^{-1})$$
s.t. $M_{\rm in}(P) + M_{\rm mid}(Q) + M_{\rm out}(S(E)) \preceq 0$, (17)

where

$$S(E) = \begin{bmatrix} C^{\top} E^{-1} C & 0\\ 0 & -1 \end{bmatrix}. \tag{18}$$

H. Choosing ε

In safety critical applications, ε should reflect the tolerated tail after accounting for dataset size and uncertainty in μ and Σ . A smaller ε tightens certification but may increase conservatism. We recommend validating with empirical CVaR on held-out data at the certified ε .

I. Complexity and scalability

The neural network (8) has $\ell+1$ layers with widths $\{n_k\}_{k=0}^\ell$ $(n=n_0 \text{ input},\ m=n_{\ell+1} \text{ output}).$

The LMI condition (13) in Theorem 3.1 checks a fixed safety specification S, which in practice is solved as an SDP feasibility problem. The optimization variables are the multipliers $-P \in \mathcal{H}_x$ and $Q \in \mathcal{Q}_{\phi}$, while S is given. The aggregated variable (in Appendix V-B) is

$$\bar{x} := [x_0^\top, x_1^\top, \dots, x_\ell^\top, 1]^\top \in \mathbb{R}^{\bar{n}}$$
 (19)

and the lifted dimension is

$$\bar{n} = \sum_{k=0}^{\ell} n_k + 1. \tag{20}$$

The number of variables in Q grows as $O(\sum_{k=1}^{\ell-1} n_k)$ for diagonal multipliers and $O(\sum_{k=1}^{\ell-1} n_k^2)$ for repeated-nonlinearity multipliers.

The SDP in (17) searches for an ellipsoidal safe set. Here S is parameterized by a matrix $E \succ 0$ of size $m_y \times m_y$, and the optimization variables are Q and E (with P fixed in practice). The matrix E adds $m_y(m_y+1)/2$ scalar variables to the problem, on top of those from the activation multipliers in Q.

Both SDP use LMIs of size \bar{n} , so the practical bottleneck is the scaling of the multipliers and the output dimension.

J. Extensions beyond standard feed-forward MLPs

Although our discussions have focused on feed-forward multilayer perceptrons (MLPs), the proposed WC-CVaR + QC/SDP framework extends naturally to more general neural network architectures.

- Residual neural network (ResNet): A residual block augments a linear transformation with an identity skip connection, e.g., $x^{k+1} = \phi(W^k x^k + b^k) + x^k$. Thus, the verification conditions retain the same SDP form with B replaced by a block-difference operator; only $M_{\rm mid}(Q)$ is modified.
- Recurrent Neural Networks (RNNs): An RNN takes the form $h^k = \phi(Wx^k + Uh^{k-1} + b)$ with hidden state h^k . Over a finite horizon, time-unfolding preserves the SDP structure with a block-banded lifting. Modify $M_{\rm mid}(Q)$ by substituting $(\mathbf{A}, \mathbf{b}, \mathbf{B})$ with their recurrent counterparts, include h^0 in the input QC.

More generally, any architecture that can be expressed as linear operators composed with slope-/sector-bounded nonlinearities—possibly with skip connections, convolutions, graph propagations, or time unrolling—fits the same QC/SDP template by replacing $(\mathbf{A}, \mathbf{b}, \mathbf{B})$ accordingly.

These extensions demonstrate that the proposed risk-aware verification framework is not restricted to simple feed-forward MLPs, while preserving the tractable SDP structure of Theorem 3.1.

IV. APPLICATIONS

Here we show how the obtained results can be used for closed-loop reachability and classification problems. Throughout this section, we restrict activation functions to ReLU.

A. Closed-loop Reachability

Consider a discrete-time linear time-invariant system with a neural network controller, as in [18]:

$$x^{+} = Ax + Bf(x), \tag{21}$$

where $x \in \mathbb{R}^n$ is the current state, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are the system matrices, and $f: \mathbb{R}^n \to \mathbb{R}^m$ is a neural network controller with x as its input.

Here, we consider obtaining a minimum volume ellipsoidal safe set for the output $y = x^+$, with input set being the set of possible current states x. In this case, C in (15) is

$$C = \begin{bmatrix} A & B \end{bmatrix}. \tag{22}$$

Numerical experiments were performed with the system parameters

$$A = \begin{bmatrix} 0.2 & 0 \\ 0.1 & 0.3 \end{bmatrix}, B = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \tag{23}$$

and a neural network controller that approximates a stabilizing controller $f(x) \approx Kx$ with $K = [-1 \ 2]$, using 2 neurons in the input layer, 3 neurons in the hidden layer, and 1 neuron in the output layer.

We compared three approaches:

- 1) Norm bounded [18],
- 2) Confidence level [20],
- 3) Risk-aware (proposed)

To illustrate sampled inputs and outputs, we used the followings:

- For 1), the inputs are random vectors x drawn uniformly from the unit disk, with mean $\mu = 0$ and covariance $\Sigma = 1/4I$. Those represents samples in the deterministic input region.
- \bullet For 2) and 3), the inputs are random vectors x drawn from uniform, normal and t-distributions, all with mean $\mu = 0$ and covariance $\Sigma = 1/4I$.

In each case, the outputs are computed by (21).

We set the input QC with

$$P = \begin{bmatrix} -I & 0 \\ 0 & r \end{bmatrix},\tag{24}$$

where

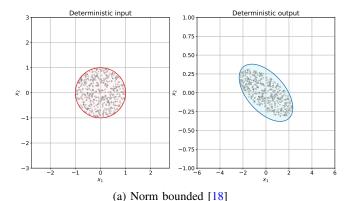
- Case 1): r=1• Case 2): $r=\frac{1}{2(1-p)}$ Case 3): $r=\sup_{\mathbb{P}\in\mathcal{P}}\mathbb{P}\text{-CVaR}_{\epsilon}\left[x^2\right]=\frac{1}{2\varepsilon},$

Although the chance-constrained [20] and WC-CVaR formulation are derived from different risk measures, our case provides identical bounds.

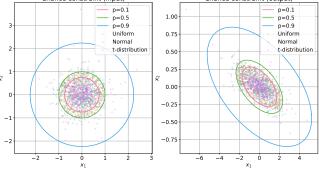
Lemma 4.1: Suppose the input random vector is drawn from a distribution with mean $\mu = 0$ and covariance Σ . If we fix the shape of input bound to be a scaled ellipsoid $\{x: x^T \Sigma^{-1} x \leq x\}$ k} with some k > 0, then the risk-aware input set with ε is identical to the ellipsoidal chance-constraint input set with

Proof: See appendix V-C.

Furthermore, both of the corresponding minimum-volume ellipsoidal safe sets can be computed using (17), thus coincide.







(b) Confidence ellipsoid [20]

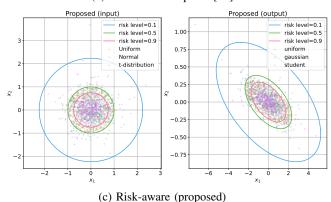


Fig. 1: Closed-loop reachability (left column: inputs, right column: outputs). Bounds are shown with random samples x in the input space and $x^+ = Ax + Bf(x)$ in the output space

For the confidence levels in 2), we plot p = 0.1, 0.5 and 0.9 and for the risk levels in 3), we plot $\varepsilon = 0.1, 0.5$ and 0.9.

Figures 1a)-c) illustrate the ellipsoidal safe sets obtained by the three approaches. The norm bounded approach ensures that all samples remain within the bound, as expected in Fig. 1(a). As expected, the confidence ellipsoid method (Fig. 1b) and the proposed risk-aware method (Fig. 1c) produce identical ellipsoids. We also observe that a t-distribution, which has a long tail, has samples out of those safe sets.

While this may seem to undermine its advantage, unlike [20], which is limited to ellipsoidal inputs, the proposed approach also applies to polytope and zonotope input, offering greater flexibility, such as classification problem we see the next.

B. Classifications

We consider a neural network classifier $f:\mathbb{R}^n\to\mathbb{R}^m$ that assigns input x to the class with the highest score: $C(x)= \underset{0\leq i\leq m-1}{\operatorname{argmax}}_{0\leq i\leq m-1}f_i(x)$ [18], [41]. A classifier is risk-aware if the ambiguity set $\mathcal P$ does not alter the classification decision with respect to its mean μ , i.e., for input $x\sim\mathbb P\in\mathcal P$, risk-aware classifier satisfies:

$$\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\text{-CVaR}_{\varepsilon} \left[f_c(x) - f_i(x) \right] \le 0, \ \forall i \ne c.$$
 (25)

Without loss of generality, we set $c = C(\mu) = 0$. Then, because these constraints are in the form of polytope, using the results of [18], it can be written in the form of (12) using

$$S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & S_{\text{sub}}^{\top} \Gamma S_{\text{sub}} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ S_{\text{sub}} = \begin{bmatrix} 0 & 0 \\ \mathbf{1}_{m-1} & -I_{m-1} \end{bmatrix}, \ (26)$$

where $\Gamma \in \mathbb{S}^m$, $\Gamma \geq 0$, $\Gamma_{ii} = 0$ with appropriate S_{sub} . Thus, the use of Theorem 3.1 verifies the satisfaction of risk-aware classification.

We compare the performance of the proposed approach on different distributions, uniform, normal, Weibull, power law, lognormal, and student's t-distribution. Experiments were performed using the scikit-learn Digits dataset (8×8 grayscale images of handwritten digits, 1797 samples) [42], [43]. A neural network classifier with 64 input neurons (corresponding to the 8×8 input images), one hidden layer of 32 neurons, and 10 output neurons (one for each digit class 0-9, i.e., $f_i(x)$ is for digit class i for $i=0,\ldots,9$), which achieves 98% test accuracy was used. We set the risk-level $\varepsilon=0.2$.

For robustness analysis, we first obtained P for the risk-aware input QC such that P along with S as in (26) and Q in (11) of the neural network classifier satisfy the condition (13). Then, we checked if this P satisfies the risk-aware QC (10) with the mean and covariance of the images labeled as digit "6". From Theorem 3.1, we expect the output satisfies (12) if the test samples were generated from various distributions while preserving the mean and covariance of the images labeled as digit "6".

Figure 2(a) shows the distribution of the difference values $f_6 - \max_{i \neq 6} f_i$ across various input distributions, where f_6 denotes the output for class "6" and $\max_{i \neq 6} f_i$ represents the maximum output among the other classes. A positive difference indicates correct classification, while a negative difference corresponds to a misclassification. The results highlight different levels of robustness: for instance, the uniform and normal distributions achieve perfect classification, whereas heavy-tailed distributions such as the t-distribution lead to more frequent misclassifications.

The statistical values of Fig. 2(a) are summarized in Table II. The mean/median columns show that uniform and power law have the largest typical margins, while heavy-tailed families exhibit visible skew (median > mean), indicating occasional low or negative margins that pull the mean down; this skew is strongest for t-distribution. The standard deviation column separates stability: uniform distribution is the most concentrated (smallest spread), normal distribution is also tight, whereas t-distribution is the most variable. The Positive

Ratio (PR), defined as $PR = \frac{1}{N} \sum_{k=1}^{N} \mathbf{1} \{ P_{\text{diff}}(x^{(k)}) > 0 \}$ with $P_{\text{diff}} := f_6 - \max_{i \neq 6} f_i$ and $x^{(k)}$ the k-th sample, then flags outright errors: all but student's t-distribution are near or at 1, i.e., misclassifications are rare for those inputs. Because PR saturates for several distributions, the decisive column is CVaR(0.20): uniform distribution sustains the largest positive tail margins (safest worst 20%), normal distribution remains safely positive, the other light-to-moderate tails (power law, lognormal, Weibull) are positive but weaker, and t-distribution alone turns negative—revealing tail failures despite acceptable central statistics.

Figure 2(b) depicts the average feature pattern of samples correctly classified as "6," capturing the typical shape of the digit. In contrast, Figure 2(c) presents a misclassified "6" sample, where atypical handwriting leads the classifier to fail. Figure 2(d) plots the output values f_i for each class on this misclassified sample. It can be observed that both classes 6 and 8 yield positive activation values, 3.52 and 5.22, respectively, whereas the remaining classes have negative outputs. This indicates that the network confuses the digit "6" with "8," providing insight into the source of the misclassification, which aligns with human trends.

V. CONCLUSIONS AND FUTURE WORKS

This paper introduced a risk-aware safety verification framework for neural networks under input uncertainties using WC-CVaR. Our approach provides risk-aware safety guarantees by controlling the expected severity of worst-case outcomes beyond a specified risk level. It allows us to balance conservatism with tail risk tolerance and to handle diverse input geometries beyond ellipsoids, while maintaining computational tractability at the same level as existing approaches. Experiments across reachability and classification tasks confirmed that our approach effectively handles heavy-tailed distributions.

One possible future work is to broaden ambiguity sets beyond moments, and another would be to validate on safetycritical closed-loop systems while exploring controller synthesis with WC-CVaR.

APPENDIX

A. Expression of Q for ReLU Function

QC for ReLU is given in [18] as follows: The function $\phi(z) = \max(0, z)$ satisfies the QC

$$\begin{bmatrix} z \\ \phi(z) \\ 1 \end{bmatrix}^{\top} \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12}^{\top} & Q_{22} & Q_{23} \\ Q_{13}^{\top} & Q_{23}^{\top} & Q_{33} \end{bmatrix} \begin{bmatrix} z \\ \phi(z) \\ 1 \end{bmatrix} \ge 0$$

for all $z \in \mathbb{R}^n$, where

$$Q_{11} = 0, \quad Q_{12} = T, \quad Q_{13} = -\nu,$$

$$Q_{22} = -2T, \quad Q_{23} = \nu + \eta, \quad Q_{33} = 0,$$

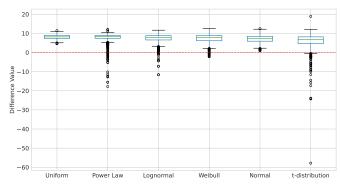
$$T = \sum_{i=1}^{n} \lambda_i e_i e_i^\top + \sum_{1 \le i < j \le n} \lambda_{ij} (e_i - e_j) (e_i - e_j)^\top,$$

$$\lambda_{ij} \ge 0, \quad \nu, \eta \ge 0.$$

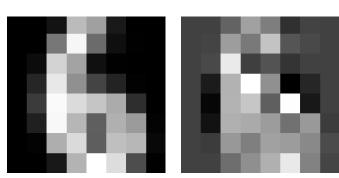
Here, $e_i \in \mathbb{R}^n$ is the *i*th unit vector, $\lambda_i \in \mathbb{R}$, and $\nu, \eta \in \mathbb{R}^n$. \mathcal{Q}_{ϕ} is the set of matrices Q in this form.

TABLE II: Summary statistics of $P_{\text{diff}} = f_6 - \max_{i \neq 6} f_i$ by input distribution.

Distribution	Mean	Median	Std. Dev.	Positive Ratio	CVaR (0.20)
Uniform	8.105	8.178	1.135	1.000	6.450
Power law	7.637	8.364	2.797	0.973	3.941
Lognormal	7.424	7.923	2.379	0.986	3.869
Weibull	7.324	7.774	2.266	0.985	3.834
Normal	7.116	7.210	1.927	1.000	4.280
Student's t	5.841	6.803	4.344	0.943	-0.092

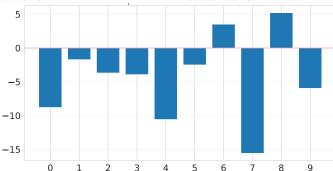


(a) Difference values, $f_6 - \max_{i \neq 6} f_i$: A positive difference indicates correct classification



(b) Mean of figures that are labeled '6'

(c) A figure labeled '6' that was misclassified '8'.



(d) Output values of each class f_i for misclassified sample in (c)

Fig. 2: Classifications

B. Proof of Theorem 3.1

We note that the theorem itself is essentially identical to Fazlyab's and the computational cost to check the sufficient condition remains the same as in [18]. However, a new proof is required for our risk-aware case to incorporate WC-CVaR QCs.

Proof: Let define a vector
$$\bar{x} = [x^{0^\top} \ x^{1^\top} \ \cdots \ x^{\ell^\top} \ 1]^\top =$$

 $[\mathbf{x}^{\top} \ 1]^{\top}$. From (13), it follows that

$$-\bar{x}^{\top} M_{\text{in}}(P) \bar{x} - \underbrace{\bar{x}^{\top} M_{\text{mid}}(Q) \bar{x}}_{>0, ::(11)} - \bar{x}^{\top} M_{\text{out}}(S) \bar{x} \ge 0,$$

which implies

$$-\bar{x}^{\top} M_{\text{in}}(P) \bar{x} \ge \bar{x}^{\top} M_{\text{out}}(S) \bar{x}.$$

Proposition 2.1 implies that

$$\underbrace{\sup_{\mathbb{P} \in \mathcal{P}} \mathbb{P}\text{-CVaR}_{\epsilon} \left[-\bar{x}^{\top} M_{\text{in}}(P) \bar{x} \right]}_{\leq 0, \ \because (10)} \geq \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{P}\text{-CVaR}_{\epsilon} \left[\bar{x}^{\top} M_{\text{out}}(S) \bar{x} \right].$$

Using
$$\bar{x}^{\top}M_{\mathrm{out}}(S)\bar{x} = \begin{bmatrix} x \\ f(x) \\ 1 \end{bmatrix}^{\top} S \begin{bmatrix} x \\ f(x) \\ 1 \end{bmatrix}$$
, we obtain the satisfaction of (12).

C. Proof of Lemma 4.1

Let's see the intuition first. Let $z = \Sigma^{-1/2}x$ and $Y = ||z||^2$. Then $\mathbb{E}[z] = 0$, $\mathrm{Cov}(z) = I$, and $\mathbb{E}[Y] = n$.

Intuitively, among all distributions with $\mathbb{E}[z] = 0$ and $\operatorname{Cov}(z) = I$, both tail measures $\mathbb{E}[\mathbf{1}\{Y > t\}]$ (where **1** is an indicator) and $\mathbb{E}[(Y - \beta)^+]$ achieve their worst case when all probability mass is concentrated at just two radii: some mass at the origin and the rest at one fixed radius r, i.e.,

$$\mathbb{P}(Y=r^2)=q, \qquad \mathbb{P}(Y=0)=1-q.$$

Geometrically, a fraction q of the mass lies on the sphere of radius r, and the remainder is at the origin.

Because both problems reduce to the same one–dimensional radius tradeoff, the safe sets coincide once the ellipsoid's shape is fixed and only its scale k.

Proof: From Lemma 2.4 in [32], we have

$$\sup_{\mathbb{P} \in \mathcal{P}} \mathbb{P}\text{-CVaR}_{\varepsilon}[\boldsymbol{x}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}] = \frac{1}{\varepsilon}\text{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}) = \frac{n}{\varepsilon},$$

recalling n is the dimension of x.

On the other hand, Lemma 2 in [20], a p-level confidence region of x is

$$\left\{ x : x^{\top} \Sigma^{-1} x \le \frac{n}{1-p} \right\}.$$

Thus, the boundary of the risk-aware safe set with ε is coincides with that of the confidence level set with 1-p.

This equivalence extends whenever one can define a variable z with $\mathbb{E}[z]=0$ and $\mathrm{Cov}(z)=I$, while fixing the ellipsoidal shape and tuning only its scale.

REFERENCES

- [1] M. Kishida, "Risk-aware safety verification and robustness analysis of neural network," in *IEEE Conference on Decision and Control*, 2025.
- [2] E. De Gelder, H. Elrofai, A. K. Saberi, J.-P. Paardekooper, O. O. Den Camp, and B. De Schutter, "Risk quantification for automated driving systems in real-world driving scenarios," *Ieee Access*, vol. 9, pp. 168 953–168 970, 2021.
- [3] M. Brunner, D. N. Darwesh, and B. Annighoefer, "A safety process for self-adaptive safety-critical plug&fly avionics," in *IEEE/AIAA 40th Digital Avionics Systems Conference*. IEEE, 2021, pp. 1–10.
- [4] H. Alemzadeh, R. K. Iyer, Z. Kalbarczyk, and J. Raman, "Analysis of safety-critical computer failures in medical devices," *IEEE Security & Privacy*, vol. 11, no. 4, pp. 14–26, 2013.
- [5] K. Zhou, J. Doyle, and K. Glover, Robust and Optimal Control, ser. Feher/Prentice Hall Digital and. Prentice Hall, 1996. [Online]. Available: https://books.google.co.jp/books?id=RPSOQgAACAAJ
- [6] G. E. Dullerud and F. Paganini, A course in robust control theory: a convex approach. Springer Science & Business Media, 2013, vol. 36.
- [7] F. Blanchini and S. Miani, Set-theoretic methods in control. Springer, 2015.
- [8] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," SIAM Journal on Optimization, vol. 17, no. 4, pp. 969–996, 2007.
- [9] G. Welch, G. Bishop et al., "An introduction to the kalman filter," 1995.
- [10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.
- [11] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," in *Proceedings of the ACM SIGSAC* Conference on Computer and Communications Security, 2018, pp. 2154–2156.
- [12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083, 2017.
- [13] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *International conference on computer aided verification*. Springer, 2017, pp. 97–117.
- [14] R. Ehlers, "Formal verification of piece-wise linear feed-forward neural networks," in *International symposium on automated technology for* verification and analysis. Springer, 2017, pp. 269–286.
- [15] R. Bunel, J. Lu, I. Turkaslan, P. H. Torr, P. Kohli, and M. P. Kumar, "Branch and bound for piecewise linear neural network verification," *Journal of Machine Learning Research*, vol. 21, no. 42, pp. 1–39, 2020.
- [16] Z. Shi, H. Zhang, K.-W. Chang, M. Huang, and C.-J. Hsieh, "Robustness verification for transformers," in *International Conference on Learning Representations*, 2020.
- [17] K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin, and C.-J. Hsieh, "Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=nVZtXBI6LNn
- [18] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 1–15, 2022.
- [19] ——, "An introduction to neural network analysis via semidefinite programming," in *IEEE Conference on Decision and Control*, 2021, pp. 6341–6350.
- [20] ——, "Probabilistic verification and reachability analysis of neural networks via semidefinite programming," in *IEEE Conference on Decision* and Control, 2019, pp. 2726–2731.
- [21] S. V. Noori, B. Hu, G. Dullerud, and P. Seiler, "Stability and performance analysis of discrete-time relu recurrent neural networks," in *IEEE Conference on Decision and Control*, 2024, pp. 8626–8632.
- [22] H. Salman, G. Yang, H. Zhang, C.-J. Hsieh, and P. Zhang, "A convex relaxation barrier to tight robustness verification of neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [23] H.-M. Chiu, H. Chen, H. Zhang, and R. Y. Zhang, "SDP-CROWN: Efficient bound propagation for neural network verification with tightness of semidefinite programming," in Forty-second International Conference on Machine Learning, 2025. [Online]. Available: https://openreview.net/forum?id=5liHhkgvAn
- [24] L. Weng, P.-Y. Chen, L. Nguyen, M. Squillante, A. Boopathy, I. Oseledets, and L. Daniel, "Proven: Verifying robustness of neural networks with a probabilistic approach," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6727–6736.

- [25] J. Pilipovsky, V. Sivaramakrishnan, M. Oishi, and P. Tsiotras, "Probabilistic verification of relu neural networks via characteristic functions," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 966–979.
- [26] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *International conference on computer aided* verification. Springer, 2017, pp. 3–29.
- [27] S. Zymler, D. Kuhn, and B. Rustem, "Distributionally robust joint chance constraints with second-order moment information," *Mathematical Programming*, vol. 137, pp. 167–198, 2013.
- [28] R. T. Rockafellar and S. Uryasev, "Optimization of conditional valueat-risk," *Journal of Risk*, vol. 2, pp. 21–41, 2000.
- [29] S. Zhu and M. Fukushima, "Worst-case conditional value-at-risk with application to robust portfolio management," *Operations research*, vol. 57, no. 5, pp. 1155–1168, 2009.
- [30] M. Cleaveland, L. Lindemann, R. Ivanov, and G. J. Pappas, "Risk verification of stochastic systems with neural network controllers," *Artificial Intelligence*, vol. 313, p. 103782, 2022.
- [31] M. Kishida, "Risk-aware stability, ultimate boundedness, and positive invariance," *IEEE Trans. on Automatic Control*, vol. 69, no. 1, pp. 681– 688, 2024.
- [32] M. Kishida and A. Cetinkaya, "Risk-aware linear quadratic control using conditional value-at-risk," *IEEE Transactions on Automatic Control*, vol. 68, no. 1, pp. 416–423, 2023.
- [33] A. Takeda and M. Sugiyama, "ν-support vector machine as conditional value-at-risk minimization," in *Proceedings of the international confer*ence on Machine learning, 2008, pp. 1056–1063.
- [34] T. Hiraoka, T. Imagawa, T. Mori, T. Onishi, and Y. Tsuruoka, "Learning robust options by conditional value at risk optimization," Advances in Neural Information Processing Systems, vol. 32, 2019.
- [35] T. Soma and Y. Yoshida, "Statistical learning with conditional value at risk," arXiv preprint arXiv:2002.05826, 2020.
- [36] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations research*, vol. 58, no. 3, pp. 595–612, 2010.
- [37] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally robust convex optimization," *Operations research*, vol. 62, no. 6, pp. 1358–1376, 2014.
- [38] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, pp. 203–228., 1999.
- [39] S. Zymler, D. Kuhn, and B. Rustem, "Worst-case value at risk of nonlinear portfolios," *Management Science*, vol. 59, no. 1, pp. 172–188, 2013.
- [40] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE transactions on automatic control*, vol. 42, no. 6, pp. 819–830, 2002.
- [41] M. Hein and M. Andriushchenko, "Formal guarantees on the robustness of a classifier against adversarial manipulation," Advances in neural information processing systems, vol. 30, 2017.
- [42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [43] E. Alpaydin and C. Kaynak, "Optical recognition of handwritten digits," UCI Machine Learning Repository, 1998, dOI: 10.24432/C50P49.