Whole-Body Safe Control of Robotic Manipulators with Koopman Neural Dynamics

Anonymous

***Abstract*— Controlling robots with strongly nonlinear, high-dimensional dynamics remains challenging, as direct nonlinear optimization with safety constraints is often intractable in real time. The Koopman operator offers a way to represent nonlinear systems as linear in a lifted space, enabling the use of efficient linear control. We propose a data-driven framework that learns a Koopman embedding and operator from data, and integrates the resulting linear model with the Safe Set Algorithm (SSA). This allows the tracking and safety constraints to be solved in a single quadratic program (QP), ensuring feasibility and optimality without a separate safety filter. We validate the method on a Kinova Gen3 manipulator in simulation, showing accurate tracking and obstacle avoidance, and discuss its potential for sim-to-real transfer with minimal retraining.**

I. INTRODUCTION

Ensuring safe control of robotic manipulators is fundamentally difficult due to the interplay of complex dynamics, high dimensionality, and safety-critical constraints. A first challenge arises from the strong nonlinearity of manipulator dynamics: joint couplings, contact interactions, and nonlinear actuation make direct embedding of such models into optimization-based controllers computationally prohibitive. Even when accurate models are available, nonlinear formulations with safety constraints often lead to nonconvex programs that cannot be solved in real time [1]. A second challenge is on feasibility, which emerges at the boundary of the safe set, where a controller may fail to generate feasible inputs to steer the system back into safety. Finally, when using learned dynamics, approximation errors can propagate into safety constraints, rendering the assumed safe control unsafe. These issues collectively highlight the need for a scalable, data-driven framework that enables safe, efficient whole-body manipulator control.

A range of data-driven methods learn predictive models of nonlinear dynamics, but they rarely translate into real-time safe control. Dynamic Mode Decomposition and its extensions approximate nonlinear systems with linear operators in lifted spaces [2], [3], [4], while sparse-regression methods such as SINDy identify compact governing equations [5]. Deep recurrent and latent-state models extend predictive power [6], and Koopman operator theory provides a principled framework for globally linearizing nonlinear dynamics [7], [8], [9]. Yet these approaches focus on prediction: they do not ensure that modeling errors or online computation limits can meet hard safety constraints during control, a gap that becomes acute for whole-body manipulators.

On the control side, no existing family of methods simultaneously handles high dimensionality, model uncertainty, and strict safety guarantees. Classical nonlinear controllers such as feedback linearization require exact models [10], and trajectory-optimization techniques like Iterative Linear Quadratic Regulator (iLQR) or Nonlinear Model Predictive Control (NMPC) scale poorly as degrees of freedom grow [11]. Reinforcement learning has shown striking successes in manipulation [12], but it demands large data and provides only probabilistic safety at best [13]. Control Barrier Functions (CBFs) offer convex safety filters [14] but can be overly conservative, activating constraints even deep inside the safe set [15]. Safe-Set Algorithms relax this conservatism [16] yet remain difficult to couple with learned or highly nonlinear models in real time. Recent efforts to merge Koopman embeddings with CBFs [17] still separate nominal control from safety filtering, creating feasibility breakdowns near the boundary.

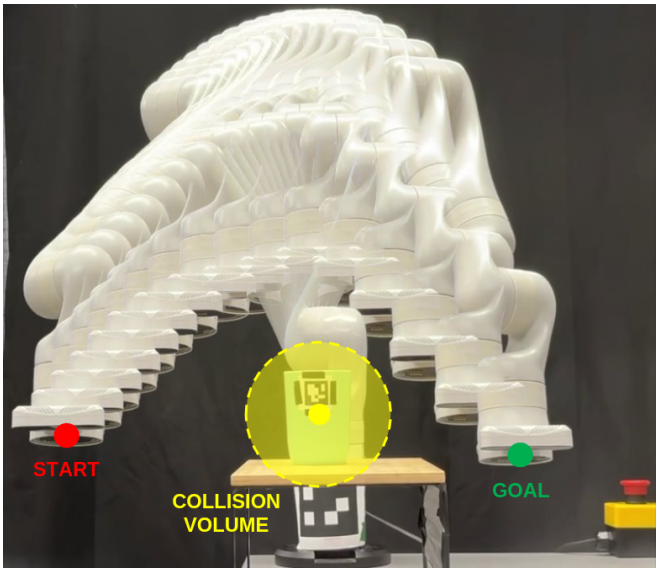

Fig. 1: Kinova Gen3 safely moving from START (red) to GOAL (green) while avoiding the collision volume (yellow).

We propose a data-driven safe control framework that combines Koopman operator theory with whole-body manipulator dynamics. The main contributions are:

Synthesis of Safe Control via Koopman Linearization. We formulate whole-body safe control of robotic manipulators by globally linearizing their nonlinear dynamics through the Koopman operator. Unlike filtering- or projection-based two-stage architectures [18], which apply safety constraints as a post-processing step on nominal controls, our approach synthesizes safety directly into the controller. This unified formulation ensures consistent reasoning about performance

and safety while avoiding the conservatism and inefficiencies of safety filtering.

Safety Index Adaptation for Learned Dynamics. To mitigate infeasible control issues that arise when coupling learned Koopman dynamics with hard safety constraints, we introduce an adversarial fine tuning scheme for the safety index. This procedure adapts the safety specification to the learned dynamics, guaranteeing forward invariance and finite-time convergence to the safe set while significantly reducing the risk of unsolvable QPs.

Adaptation to Real-World Hardware. We demonstrate the effectiveness of the proposed framework through sim-to-real deployment on a Kinova Gen3 manipulator. As illustrated in Fig. 1, the Kinova Gen3 manipulator safely moves from the START position (red) to the GOAL position (green) while avoiding the designated collision volume (yellow). A video demonstration of this experiment is provided in the supplementary material. Our results show that Koopman-based safe control can be transferred with minimal retraining while maintaining both tracking accuracy and safety guarantees in real hardware.

The proposed unified framework (Fig. 2) tackles nonlinearity, safety-boundary infeasibility, and learned-model approximation errors, enabling scalable, real-time safe control of robotic manipulators.

II. PRELIMINARIES

A. Koopman Operator Theory

Given the nonlinear systems with discrete dynamics:

$$x_{k+1} = f(x_k) \tag{1}$$

Koopman Operator $\mathcal{K}$ is an infinite dimensional linear operator that maps the nonlinear dynamics to linear dynamics [19] and evolves the embedding functions ψ of the state:

$$\mathcal{K}\psi(x) = \psi \circ f(x) \tag{2}$$

Considering a non-autonomous dynamical system with the control input $u_k \in \mathcal{U} \subseteq \mathbb{R}^m$ and system state $x_k \in \mathcal{X} \subset \mathbb{R}^n$, given by $x_{k+1} = f(x_k, u_k)$, the evolution flow follows as:

$$\mathcal{K}\psi(x_k, u_k) = \psi(f(x_k, u_k)) = \psi(x_{k+1}) \tag{3}$$

where $\psi : \mathbb{R}^{n+m} \to \mathbb{R}^{\infty}$. In practice, we constrain the latent space to a finite-dimensional vector space and approximate the Koopman operator, making $\psi : \mathbb{R}^{n+m} \to \mathbb{R}^d$ where $d \in [0, \infty) \subset \mathbb{R}^+$.

The embedding function $\psi(x, u)$ can be separated as $\psi(x, u) = [\psi_x(x, u); \psi_u(x, u)]$, and we focus on the case where $\psi_x(x, u) = \psi_x(x)$ and $\psi_u(x, u) = \hat{u} = u$, where $\psi_x : \mathbb{R}^n \to \mathbb{R}^{n'}$ denotes state embedding and $\hat{u}$ is control in latent space. Under this formulation, the system dynamics can be expressed as:

$$\psi_x(x_{k+1}) = A\psi_x(x_k) + Bu_k \tag{4}$$

where A, B comes from:

$$\mathcal{K} = \begin{bmatrix} A \in \mathbb{R}^{n' \times n'} & B \in \mathbb{R}^{n' \times m} \\ C \in \mathbb{R}^{m \times n'} & D \in \mathbb{R}^{m \times m} \end{bmatrix} \tag{5}$$

We also parameterize state embedding ψ_x as a neural network ψ_ω and denote the lifted state $z_k = \psi_x(x_k) = [x_k; \psi_\omega(x_k)]^\top$. We can preserve the original state data along with neural network embedded state in order to prevent the information loss happening through nonlinear mapping [20]. We can retrieve the original state via:

$$x_k = Pz_k \tag{6}$$

where $P = [I_n; 0] \in \mathbb{R}^{n \times (n+n')}$. Then, the lifted state evolution takes the form:

$$z_{k+1} = Az_k + Bu_k \tag{7}$$

This formulation allows us to design linear controllers for state-constrained control problems (e.g., collision avoidance) by leveraging the linearity of lifted dynamics. In addition, we use end-to-end training framework for the acquisition of embedding function as well as the Koopman operator (ψ_ω, A, B). Readers can refer to [19] for further understanding of Koopman Operator Theory.

B. Safe Control Synthesis through Adversarial Fine Tuning

The goal of safe control is to design a controller that tracks a reference while ensuring forward invariance of the allowable set $\mathcal{A}$ (i.e., trajectories starting in $\mathcal{A}$ remain in $\mathcal{A}$) [21]. For a function $s : \mathbb{R}^n \to \mathbb{R}$ let $\mathcal{S} \coloneqq \{s\}_{\leq 0}$ be its zero-sublevel set, $\partial\mathcal{S} \coloneqq \{s\}_{=0}$ the boundary of this set. Assuming we are given: (1) a control-affine system $\dot{x} = f(x) + g(x)u$ where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are locally Lipschitz continuous on $\mathbb{R}^n$, (2) control set $\mathcal{U}$ is a bounded convex polytope, (3) safety specification $\phi_0 : \mathbb{R}^n \to \mathbb{R}$ implicitly defines the allowable set $\mathcal{A} \coloneqq \{\phi_0\}_{\leq 0}$.

If we directly impose the constraint $\dot{\phi}_0(x) \leq 0$ at the boundary of allowable set $\partial\mathcal{A}$ for a limit–agnostic index ϕ_0 and bounded $\mathcal{U}$, the feasible set can collapse, causing controller saturation or an unsolvable QP. We therefore *reshape* the safety specification by introducing a parameterized index:

$$\phi_\rho = h(\rho, \phi_0) \tag{8}$$

where $h : \mathbb{R} \to \mathbb{R}$ with learnable parameters represented as ρ. At the safe set boundary, constraint is then defined as:

$$\dot{\phi}_\rho(x, u) = \nabla\phi_\rho(x)^\top f(x) + \nabla\phi_\rho(x)^\top g(x)u \leq 0 \quad \forall x \in \partial\mathcal{A} \tag{9}$$

in order to guarantee forward invariance. Readers can refer to [22] for further details.

To ensure feasibility under input limits and model approximation, we adapt ρ via the min–max program:

$$\min_{\rho} \max_{x \in \partial\mathcal{A}} \inf_{u \in \mathcal{U}} \dot{\phi}_\rho(x, u) \tag{10}$$

which can be evaluated over the vertex set of control $v \in \mathcal{V}(\mathcal{U})$ when $\mathcal{U}$ is a polytope [22]. Thus, the objective can be computed as a discrete minimization::

$$\mathcal{L}(\rho, x) \coloneqq \min_{v \in \mathcal{V}(\mathcal{U})} \dot{\phi}_\rho(x, v) \tag{11}$$

This procedure *learns* a safety index that avoids infeasible QPs while preserving forward invariance.

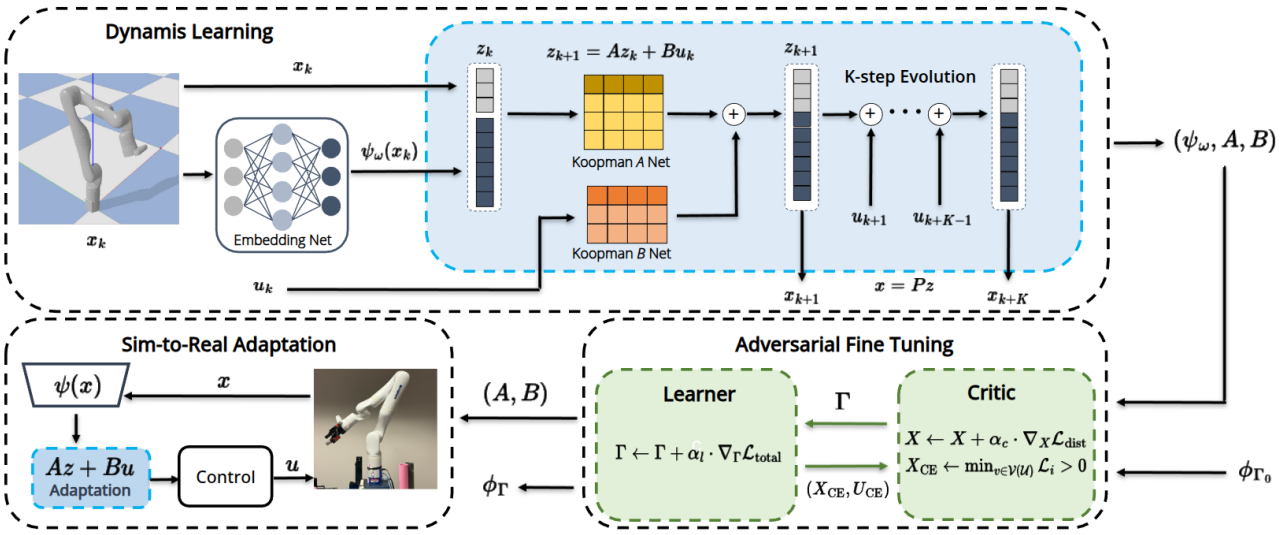

Fig. 2: Overview of training framework for Koopman Safe Control. Neural embedding function and Koopman Operators are first trained, safety index is adapted to the learned dynamics via adversarial fine tuning, and the model is migrated to real environment.

III. PROBLEM FORMULATION

We consider whole-body safe control of an n-DoF manipulator operating amid static/dynamic obstacles. The target system is the manipulator dynamics in generalized coordinates:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \tag{12}$$

where $\theta \in \mathbb{R}^n$ are joint positions, $\dot{\theta}$ joint velocities, $M(\theta)$ the inertia matrix, $C(\theta, \dot{\theta})$ the Coriolis and centrifugal terms, $G(\theta)$ the gravity vector, and τ the applied joint torques. Eq. (12) represents the low-level dynamics that must ultimately be respected, but which are highly nonlinear and high-dimensional to directly embed into real-time safety-critical controllers.

Our control objective is trajectory tracking in the presence of obstacles, under uncertainty in both robot motion and obstacle motion. We consider all obstacles and robot link collision volumes to be spheres. This choice allows (i) distance functions and gradients admit closed-form expressions, enabling efficient evaluation of safety constraints, and (ii) spherical volumes approximate local link geometry while simplifying multi-link safety verification into a minimum-distance calculation between spheres.

We define a signed distance from the ego towards the obstacle as safety specification. For an ego link's center of mass $p_{\text{ego}}(x)$ and obstacle center p_{obs}:

$$\phi_0(x) = d_{\min} - \|p_{\text{ego}}(x) - p_{\text{obs}}\|_2 \tag{13}$$

where $p_{\text{ego}}(x) \in \mathbb{R}^3$ is the Cartesian position of the ego robot as a function of state x, $p_{\text{obs}} \in \mathbb{R}^3$ is the Cartesian position of the obstacle, $d_{\min} \in \mathbb{R}^+$ is the minimum distance to be maintained between ego and obstacle. Safety requires $\phi_0(x) \leq 0$ and resulting safety constraint becomes:

$$\dot{\phi}_0(x_k, u_k) \leq b_\phi(x_k) \tag{14}$$

where the safety bound $b_\phi(x)$ is defined as

$$b_\phi(x) = \begin{cases} 0 & \text{if } x \in \partial\mathcal{A}, \\ -\lambda & \text{if } x \notin \mathcal{A}, \\ \infty & \text{otherwise.} \end{cases} \tag{15}$$

There can be approaches using NMPC with safety constraints to address this problem:

$$\begin{aligned} \min_{u_k} \quad & \sum_{k=1}^{N-1} \|x_k - x_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 + \|x_N - x_N^{\text{des}}\|_{Q_N}^2 \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k), \quad \forall k = 0, \ldots, N-1 \\ & u_k \in [u_{\min}, u_{\max}] \\ & \dot{\phi}_0(x_k, u_k) \leq b_\phi(x_k) \end{aligned} \tag{16}$$

However, solving (16) with full nonlinear dynamics is computationally prohibitive and often infeasible in real time, while local linearizations quickly lose validity away from the expansion point. Safety-filtering methods such as CBFs can be overly conservative or even infeasible at the boundary, and black-box learned models risk false safety guarantees due to approximation error. These drawbacks highlight the need for an approach that embeds safety directly while remaining tractable.

In summary, the problem we address is to synthesize safe, real-time feasible whole-body controllers for manipulators by (i) globally linearizing nonlinear dynamics via Koopman operators, (ii) embedding safety constraints directly within a single MPC optimization, and (iii) adversarially adapting the safety index to ensure non-emptiness of safe control set as well as QP solvability under learned dynamics.

IV. METHODOLOGY

In this section we present a unified, single-MPC formulation for safe whole-body manipulation. The approach (1) learns a globally linear lifted model via Koopman operators, (2) embeds and redesigns the safety constraint so it remains feasible at the boundary, (3) adapts the model for hardware.

A. Safe MPC Formulation via Koopman Dynamics

Although naive models such as single-integrator dynamics [14] have been widely used for safety-critical control, they are ill-suited for manipulators: they ignore actuation limits, joint coupling, and dynamic feasibility, leading to controllers that may certify safety in theory but fail in practice on high-DOF systems. To enable tractable safe control, we will approximate (12) with a globally linear lifted model using Koopman operator theory. Following (2), (3), we define an embedding $\psi : \mathbb{R}^n \to \mathbb{R}^{n+d}$ and lifted state $z_k = [x_k; \psi_\omega(x_k)]^\top$ where $x_k = [p_k; \theta_k]^\top$ where $p_k \in \mathbb{R}^3$ denotes the Cartesian position of the end-effector. The dynamics are approximated as

$$z_{k+1} = Az_k + Bu_k, \qquad x_k = Pz_k \tag{17}$$

where u_k denotes higher-level velocity commands, A, B are Koopman matrices, and P projects the lifted state back to the original coordinates. By wrapping the low-level torque dynamics (12) into the Koopman approximation (17), our controller operates at the velocity-control level. This design choice is crucial: it allows the use of first-order safety indices, simplifies safety constraint construction, and improves computational efficiency.

1) Feedforward Prediction: Given the current state x_t, we predict K steps forward using the learned Koopman operators A and B, which are represented as linear layers. We roll out the dynamics:

$$\hat{z}_{t+k+1} = Az_{t+k} + Bu_{t+k},\ k = [0, ..., K-1] \tag{18}$$
$$z_t = \psi_x(x_t) = [x_t; \psi_\omega(x_t)]^\top \tag{19}$$
$$x_{t+k} = Pz_{t+k} \tag{20}$$
$$u_{t+k} = \hat{u}_{t+k} \tag{21}$$

2) K-steps Prediction Loss: Since we don't have a separate network for control embedding ψ_u, we only train the state embedding network ψ_ω and Koopman matrices A, B end-to-end using a K-step prediction loss. Given dataset $\{x_i, u_i\}$ for $i = 0, \ldots, K$, we compute embedded states $z_i = \psi_x(x_i)$ and predicted lifted states $\hat{z}_i$ via forward rollout. The loss is:

$$\mathcal{L}_{\text{pred}}(\omega) = \sum_{i=1}^{K} \gamma^{i-1} \text{MSE}(z_i, \hat{z}_i) \tag{22}$$

where γ is a decay factor and MSE is the mean squared error[23].

3) Safe Linear MPC Formulation: Given that the target nonlinear control affine dynamics is $\dot{x} = f_q(x) + g_q(x)u$, the time derivative of safety constraint((14), (15)) can be expressed via chain rule:

$$\begin{aligned}\dot{\phi}_0 &= \frac{\partial \phi_0(x)}{\partial x}\, \dot{x}_q \\ &= \frac{\partial \phi_0(x)}{\partial x}\, f_q(x) + \frac{\partial \phi_0(x)}{\partial x}\, g_q(x)\, u\end{aligned} \tag{23}$$

Recall when using Koopman dynamics, the state evolves as (17), so the dynamics of x in terms of z and u can be numerically approximated:

$$\dot{x} \approx \frac{x_{t+1} - x_t}{\Delta t} = \frac{P(Az_k + Bu_k) - Pz_k}{\Delta t} \tag{24}$$

Substituting into the time derivative of the safety index, we obtain:

$$\dot{\phi}_0(x_k, u_k) = \nabla_x \phi_0(x_k)^\top \frac{(PA - P)z_k - PBu_k}{\Delta t} \tag{25}$$

This expression is linear in u and can be directly inserted as a linear constraint in MPC. Considering quadratic cost with desired state x^{des} subject to nonlinear dynamics $x_{k+1} = f(x_k, u_k)$, the system becomes linear in lifted space and we can formulate a QP over horizon N as:

$$\begin{aligned}\min_{u_k} \quad & \sum_{k=1}^{N-1} \|Pz_k - x_k^{\text{des}}\|_Q^2 + \|u_k\|_R^2 + \|Pz_N - x_N^{\text{des}}\|_{Q_N}^2 \\ \text{s.t.} \quad & z_{k+1} = Az_k + Bu_k, \quad \forall k = 0, \ldots, N-1 \\ & \nabla_x \phi_0(x_k)^\top \frac{(PA - P)z_k - PBu_k}{\Delta t} \le b_\phi(x_k) \\ & z_1 = \psi_x(x_1), \quad u_k \in [u_{\min}, u_{\max}]\end{aligned} \tag{26}$$

B. Redesign the Safety Constraint for Persistent Feasibility

Although Koopman dynamics allows us to formulate a single QP for nonlinear systems, extra dimensions introduced by lifted space can cause feasibility issue. Let us take a more rigorous look at $\dot{\phi}_0$. First, matrices A, B can be decomposed as:

$$A = \begin{bmatrix} A_{xx} & A_{x\psi} \\ A_{\psi x} & A_{\psi\psi} \end{bmatrix}, B = \begin{bmatrix} B_x \\ B_\psi \end{bmatrix} \tag{27}$$

Given $z_k = [x_k; \psi_\omega(x_k)]^\top$, the lifted dynamics formulation for original state x becomes:

$$x_{k+1} = A_{xx}x_k + A_{x\psi}\psi_\omega(x_k) + B_x u_k \tag{28}$$

Extra term $A_{x\psi}\psi_\omega(x_k)$ propagates into the time derivative computation of (14) and $\dot{\phi}_0 = \nabla_x \phi_0^\top \dot{x}$ becomes:

$$\frac{(p_{\text{ego}} - p_{\text{obs}})}{\|p_{\text{ego}} - p_{\text{obs}}\|} \frac{\partial p_{\text{ego}}}{\partial x} \frac{(A_{xx}x_k + A_{x\psi}\psi_\omega(x_k) + B_x u_k - x_k)}{\Delta t} \tag{29}$$

Then QP constraints become:

$$\begin{aligned}\frac{(p_{\text{ego}} - p_{\text{obs}})}{\|p_{\text{ego}} - p_{\text{obs}}\|} \frac{\partial p_{\text{ego}}}{\partial x} \frac{(A_{xx}x_k + A_{x\psi}\psi_\omega(x_k) + B_x)}{\Delta t} u_k \le \\ \frac{(p_{\text{ego}} - p_{\text{obs}})}{\|p_{\text{ego}} - p_{\text{obs}}\|} \frac{\partial p_{\text{ego}}}{\partial x} \frac{x_k}{\Delta t}\end{aligned} \tag{30}$$

This becomes problematic at $\partial\mathcal{A}$, since the control set sufficing all constraints can be empty.

To adapt our safety specification to learned dynamics, we introduce Learner-Critic architecture for adversarial fine tuning. Let $d = \|p_{\text{ego}}(x) - p_{\text{obs}}\|_2$ and define a nonlinear safety index $\phi(x) : \mathbb{R}^n \to \mathbb{R}$ in Cartesian coordinate as:

$$\phi_{n,\beta}(x) = d_{\min}^n - d^n + \beta d \tag{31}$$

where, $n, \beta \in \mathbb{R}$ are learnable parameters. Let $\Gamma \doteq (n, \beta)$ and learning is formulated as a min-max optimization problem:

$$\min_{\Gamma} \max_{x \in \partial S} \mathcal{L}(\Gamma, x) \tag{32}$$

where the *infeasibility risk* is:

$$\mathcal{L}_{\text{infeas}}(\Gamma, x) \coloneqq \min_{v \in \mathcal{V}(\mathcal{U})} \dot{\phi}_\Gamma(x, v) \tag{33}$$

A central advantage of the selected form (31) lies in the properties of its derivative. The radial derivative $\beta - nd^{n-1}$, visualized in Fig. 3, increases in magnitude between $d_{\min}$ and $d_{\phi=0}$. This ensures that corrective influence becomes stronger as the robot approaches the boundary of the safe set. In practice, this property aligns well with control design requirements: it promotes gentle corrections in the interior while providing decisive repulsive behavior near collision, enabling stable yet responsive safe control.

Unlike the original formulation in [22], where the safety index $\phi(x)$ is scalar and defined over the entire system, our application involves a 7-DOF Kinova arm, where each link must be evaluated for safety independently. We define a linkwise safety index $\phi_i(x)$ for each link $i = 1, \ldots, 7$, and consider all of them as safety constraints. This linkwise definition introduces substantial complexity into both the constraint evaluation and the process of counterexample collection, compared to [22], where a single scalar $\phi(x)$ simplifies the geometry of the safe set boundary $\partial\mathcal{S}$.

To enable adversarial fine tuning under this setup, we modify a Learner–Critic architecture. The Critic attempts to collect boundary states and associated control counterexamples that violate the constraint $\dot{\phi}_i(x, u) \leq 0$ for all active links at the boundary ∂S. However, instead of performing explicit Projected Gradient Descent (PGD) on *infeasibility risk* [24], our implementation uses a two-stage process that achieves an equivalent effect, adapted to the multi-link robotic setting.

First, boundary states are sampled using a differentiable optimization procedure. Specifically, in the presence of an obstacle in 3D space we minimize the distance between any of the robot link's center-of-mass positions and the obstacle center:

$$\mathcal{L}_{\text{dist}}(x) = \sum_{i=1}^{7} \max\{0,\, d_{\min} - \|p_i(x) - p_{\text{obs}}\|_2\} \tag{34}$$

Gradient descent is used to optimize x until this projection loss reaches a small threshold (empirically set to 0.0001), ensuring that the robot is brought close to the safety boundary $\partial\mathcal{S}$. This serves as a practical projection step, and thus effectively making this process a PGD.

Next, for each near-boundary state, the Critic samples control candidates from a set of saturated control vertices $\mathcal{V}(\mathcal{U})$. For each control u, it computes *infeasibility risk*. A counterexample is recorded only if no control in the admissible set satisfies:

$$\mathcal{L}^i_{\text{infeas}}(\Gamma, x) \leq 0 \quad \forall i \in \partial\mathcal{S} \tag{35}$$

for all links i in contact with the boundary.

To prevent the safety index from becoming overly conservative—which could increase infeasibility—we bound the Critic's influence by allowing 10 trials to collect the desired number of counterexamples (e.g., 50). If this quota is not reached, we consider it sufficiently tuned. This guards against overfitting to pathological states, which would otherwise cause the Learner to reshape ϕ in an attempt to satisfy constraints that cannot be satisfied, leading to an overly conservative and distorted safety index.

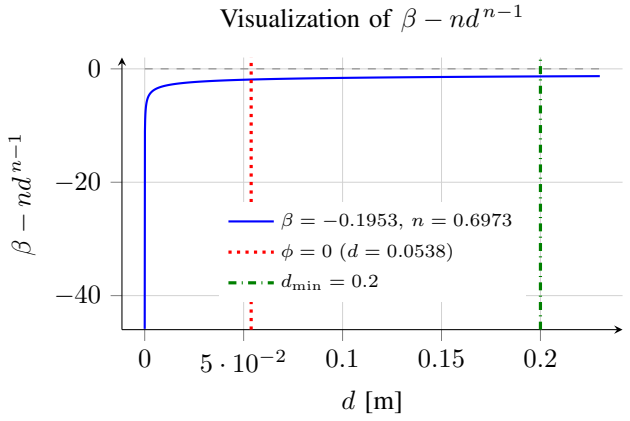

Fig. 3: Visualization of the radial derivative $\beta - nd^{n-1}$ of the learned safety index. Between $d_{\min}$ and $d_{\phi=0}$ the magnitude increases, indicating stronger corrective influence near the safety boundary.

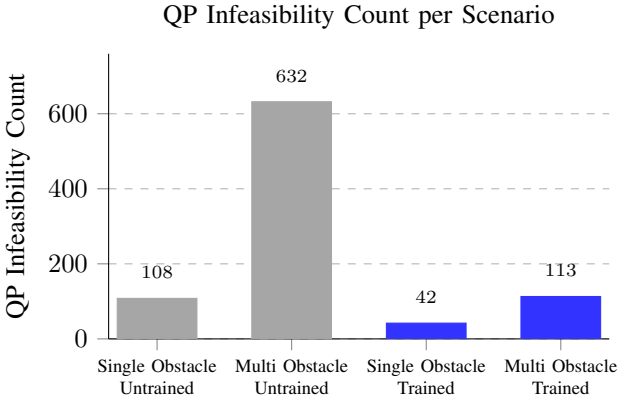

Fig. 4: QP Infeasibility Count Comparison. Single Obstacle includes one dynamic obstacle chasing one of Kinova's links, whereas Multi Obstacle includes seven extra static obstacles placed at random positions.

The Learner then performs a gradient update on the safety index parameters Γ by minimizing a combined loss:

$$\mathcal{L}_{\text{total}} = \frac{1}{N} \sum_{x \in \mathcal{B}} \mathbb{E}_{i \in \partial\mathcal{S}} \left[\mathcal{L}^i_{\text{infeas}}(\Gamma, x) \right] + \mathcal{L}_{\text{reg}} \tag{36}$$

where $\mathcal{B}$ is the batch of counterexamples, $\mathcal{L}_{\text{reg}} = \mu \|\Gamma - \Gamma_0\|_2^2$ is a regularization loss penalizing deviation from heuristically chosen good initial values $\Gamma_0 = (n_0, \beta_0)$. The overall procedure is summarized in Algorithm 1, which outlines the Critic's counterexample collection and the Learner's parameter updates for adversarial fine tuning.

Taken together, our approach preserves the min–max learning structure of adversarial fine tuning and effectively realizes the principles of PGD. The term $\beta - nd^{n-1}$ in Fig. 3 represents the radial derivative of the safety index with respect to the distance towards obstacle d. Between $d_{\min}$ and $d_{\phi=0}$, its magnitude increases, resembling a potential-field–like repulsive trend. This behavior indicates that the

Algorithm 1 Adversarial Fine Tuning

Require: $\Gamma_0 = (n_0, \beta_0)$, p_{obs}, $d_{\min}$ regularization weight μ, batch size N_{batch}, trials for Critic N_{trials}

function COLLECTCE(Γ, p_{obs}, $d_{\min}$, N_{batch}, N_{trials})
  Set learning rate α_c, $N_{\text{collected}} = 0$
  for N_{trials} **do**
    $X \leftarrow$ uniformly sample within operating region
    $X \leftarrow X + \alpha_c \cdot \nabla_X \mathcal{L}_{\text{dist}}$
    $X_{\text{CE}} \leftarrow \min_{v \in \mathcal{V}(\mathcal{U})} \mathcal{L}_i > 0 \forall i = 1, ..., 7$
    if $N_{\text{collected}} \geq N_{\text{batch}}$ **then**
      return $(X_{\text{CE}}, U_{\text{CE}})$
    end if
  end for
  Training complete
end function
function UPDATE(Γ, X_{CE}, U_{CE})
  Set learning rate α_l
  $\Gamma \leftarrow \Gamma + \alpha_l \cdot \nabla_\Gamma \mathcal{L}_{\text{total}}$ (from Eq. (36))
  return Γ
end function
function MAIN
  $\Gamma = (n_0, \beta_0)$
  while not done **do**
    $(X_{\text{CE}}, U_{\text{CE}}) \leftarrow$ CollectCE(n, β)
    $\Gamma \leftarrow$ Learn($\Gamma, X_{\text{CE}}, U_{\text{CE}}$)
  end while
  return Γ
end function

model shapes ϕ to strengthen corrective action near the safety boundary. The effectiveness of adversarial fine tuning in reducing infeasible cases is quantified in Fig. 4, which compares QP infeasibility counts across different obstacle settings before and after training.

C. Sim to Real Adaptation

To bridge the gap between simulation and hardware, we first quantify the sim-to-real mismatch by executing identical reference trajectory tracking experiments in PyBullet simulation and on the real manipulator. The deviation in tracking performance serves as a measure of model discrepancy. Instead of retraining the entire Koopman embedding and operator matrices, we collect hardware data and fine-tune only the A and B matrices of the lifted linear dynamics (17). This lightweight adaptation step efficiently accounts for actuation and unmodeled dynamics differences between simulation and hardware, enabling migration with minimal retraining. As a result, the controller preserves the structure of the learned embedding while maintaining high accuracy in real-world deployment.

D. Guarantees of the Framework

Taken together, our methodology ensures three guarantees: (i) forward invariance of the safe set under adversarially fine-tuned safety indices, (ii) finite-time convergence back into safety from the boundary, and (iii) real-time feasibility of the QP through the unified Koopman-based linear MPC formulation. By avoiding projection-based filtering and mitigating QP infeasibility through optimization-based adaptation, the proposed framework provides both theoretical safety guarantees and practical deployability, validated in both simulation and hardware.

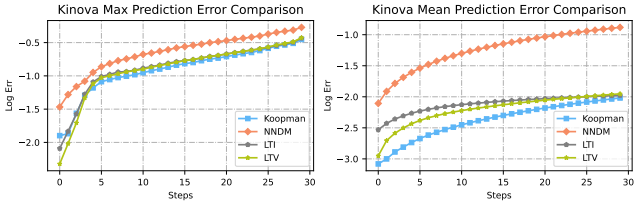

Fig. 5: Prediction Error Comparison. The analytic baselines (LTI and LTV), KDM, and NNDM are compared against PyBullet ground truth. KDM maintains the lowest long-horizon error growth, while the analytic model is competitive in short horizons but diverges.

V. EXPERIMENTS

In this section, we conduct experiments using a Kinova Gen3 manipulator to evaluate our approach. For all dynamics models considered—Koopman Dynamics Model (KDM), Neural Network Dynamics Model (NNDM) [16], and the analytic baselines—the state is defined as the concatenation of end-effector position p_k and joint positions q_k:

$$x_k = \begin{bmatrix} p_k \\ q_k \end{bmatrix} \tag{37}$$

a) Analytic Baselines.: We compare two analytic formulations: a time-invariant (LTI) model and a time-varying (LTV) variant. Both share the same state update for the joint angles and end-effector:

$$p_{k+1} = p_k + \Delta t \, J(\bar{q}_k) \, u_k \tag{38}$$
$$q_{k+1} = q_k + \Delta t \, u_k \tag{39}$$

where u_k is the joint velocity command and $J(\bar{q}_k)$ denotes the manipulator Jacobian evaluated at a reference configuration $\bar{q}_k$. For the LTI baseline, $\bar{q}_k = q^*$ is fixed, giving a globally linear but coarse approximation. For the LTV system, $\bar{q}_k = q_k$ is updated at each step, yielding a locally valid linearization that improves tracking while keeping q_k integrated directly. This LTV formulation supports an MPC controller (LTVMPC) that bridges the gap between a static linear model and full nonlinear MPC.

b) Learned Models.: We train NNDM and KDM within PyBullet simulation. Both employ fully connected feedforward networks with hidden layers [256, 256, 256] for either the lifting function ψ_ω (KDM) or direct dynamics mapping (NNDM). Fig. 5 compares the prediction errors across models, showing that the Koopman-based network achieves the lowest prediction error. Once trained, we formulate linear MPC for KDM, LTI, and LTV baselines, and shooting-based NMPC for the other models. For all non-Koopman models, a safety filter is additionally applied to enforce state constraints.

A. Safe Control in 2D Space

This experiment incorporates collision avoidance of the end-effector as a safety constraint. We use the first-order safety index from (13) with $d_{\min} = 0.2$ for NNDM and the

	Single Obstacle	Multi Obstacle (6 total)	Avg. Distance to Target [m]	Avg. Max ϕ Over Links	Avg. Mean ϕ Over Links	Avg. Min Dist. to Obstacle [m]	Cumulative Cost
	Avg. Comp. Time [s]	Avg. Comp. Time [s]					
KMPC (Ours)	9.66e-3 ± 0.15e-3	1.496e-2 ± 0.23e-3	0.071860	-0.03828	-0.21492	0.21913	13154
LTIMPC	5.19e-3 ± 0.43e-3	0.629e-2 ± 0.79e-3	0.572857	-0.15237	-0.33137	0.32237	307378578
LTVMPC	14.98e-3 ± 0.12e-3	0.629e-2 ± 0.79e-3	0.144116	-0.09723	-0.18294	0.26745	93292
NMPC-10	40.49e-3 ± 0.96e-3	5.299e-2 ± 3.06e-3	0.116499	-0.16023	-0.25239	-	81261
NMPC-100	412.59e-3 ± 227.05e-3	1.04704 ± 0.03220	0.009787	-0.12234	-0.25224	-	160

TABLE I: Computation Time and Performance Comparison. The two left columns report average computation time for safe control, while the four right columns summarize results from the 3D safe control experiment. Lower ϕ values and larger minimum distances indicate safer behavior. The symbol "–" indicates that the robot collided with the obstacle during the experiment. KMPC achieves better cost performance comparing with LTIMPC and LTVMPC by 23,367.69 and 7.09 times, while it also achieves faster speed comparing with NMPC-10 and NMPC-100 by 4.19 and 42.71 times at single obstacle setting, Overall, KMPC achieves the best trade-off between performance and safety, outperforming alternative NMPC and analytic baselines.

analytic baseline, and the adapted safety index from (31) for KDM. The KDM control law solves the QP in (26), whereas MPC for the other models solve equivalent tracking problem in original state space with the safety filter formulated as:

$$\begin{aligned} \min_{u_k} \quad & \|u_k - u_k^{\text{ref}}\|^2 \\ \text{s.t.} \quad & \nabla_x \phi_0(x_k)^\top \frac{x_{k+1} - x_k}{\Delta t} \leq b_\phi(x). \end{aligned} \tag{40}$$

We employ a 9-step horizon for all models to ensure consistent comparison. Results depicted in Fig. 6 demonstrate effective obstacle avoidance and accurate tracking of KDM-MPC.

B. Safe Control in 3D Space

In this scenario, the robot is tasked with tracking a reference end-effector trajectory in full 3D space while avoiding static obstacles placed along the path. This setting requires simultaneous trajectory tracking and obstacle avoidance across all seven links. The MPC optimization problem is identical except for the linkwise-extended safety constraint:

$$\dot{\phi}_i(x, u) \leq b_\phi(x), \quad \forall i = 1, ..., 7. \tag{41}$$

The performance results in Table I illustrate robust trajectory tracking with obstacle evasion, validating the practicality of KMPC frameworks for safe robot control. The proposed KMPC achieves the best overall trade-off, yielding significantly lower cost (23,368× and 7.1× better than LTIMPC and LTVMPC) and faster computation (4.2× and 42.7× faster than NMPC-10 and NMPC-100) under the single-obstacle setting.

C. Sim-to-Real Adaptation

To evaluate the robustness of our approach beyond simulation, we deployed the fine-tuned Koopman-based safe control pipeline directly on the Kinova Gen3 manipulator. Figure 7 compares prediction errors before and after fine-tuning, showing substantial reduction in both joint angle and end-effector position errors. These results confirm that the proposed methodology not only improves model accuracy in simulation but also transfers effectively to real hardware without retraining the embedding. Our experiments demonstrate that the unified pipeline, combining Koopman lifting and adversarial safety index tuning, achieves reliable performance on the physical system. A supplementary demo video is provided, illustrating safe control execution on hardware and validating the practical deployability of the proposed framework.

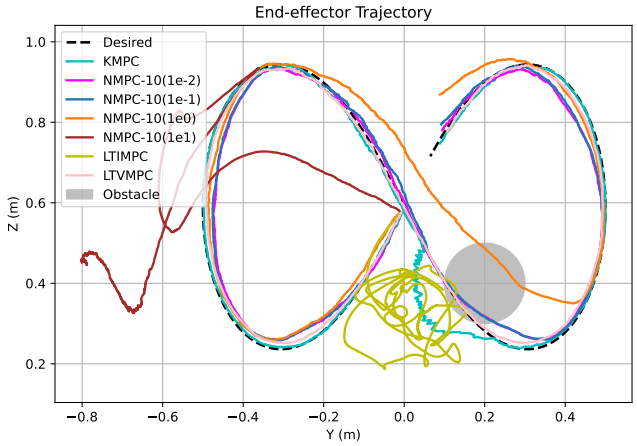

Fig. 6: Safe Tracking Comparison. NMPC-10 is nonlinear MPC with 10 times shooting, values in parentheses represent slack weight for safety filter relaxation since hard constraints cause infeasibility for projection. KMPC outperforms NMPC-10 both in tracking and safety constraint satisfaction.

Collectively, these experiments demonstrate the versatility and effectiveness of our proposed Koopman-based control methodology, from simple trajectory tracking to dynamic safety-critical scenarios. It is worth noting that since safety specification is derived from Koopman space, successful evasion also depicts the accuracy in model approximation.

VI. FUTURE WORK

While the proposed Koopman-based control framework shows promising results, several limitations remain and motivate future directions.

A. Higher-Order Safety Constraints

The current formulation relies on a first-order safety index based on positional information. This may be insufficient for dynamic or fast-evolving environments. Future work will explore higher-order safety indices incorporating velocity and curvature, which require torque or acceleration control rather than velocity control.

B. Scaling to High-Dimensional Systems

Our experiments focused on control of the Kinova Gen3. We aim to extend this framework to higher dimensional

Fig. 7: Prediction error comparison for Kinova-Gen3 before and after fine-tuning. Top: (Left) Scatter plot of original versus fine-tuned model errors, where nearly all points lie below the $y = x$ line (red), indicating consistent improvement; (Right) Average prediction error across horizons ($\log_{10}$ scale), showing that fine-tuned Koopman dynamics maintain significantly lower error over multiple steps. Bottom: Error distribution over all samples, comparing before and after fine-tuning for (Left) joint angles and (Right) end-effector positions, both of which show markedly reduced errors after fine-tuning.

systems—such as humanoid, quadruped—targeting whole-body safe control.

These extensions will further improve the scalability, safety, and real-world viability of Koopman-based safe control.

VII. CONCLUSION

This project presents a data-driven safe control framework that combines Koopman operator theory with the Safe Set Algorithm to enable efficient, verifiable control of nonlinear robotic systems. By lifting nonlinear dynamics into a latent linear space via a neural Koopman embedding, we exploit classical linear control techniques while enforcing safety constraints through a single quadratic program.

We validate the approach on the Kinova Gen3 manipulator in tasks ranging from trajectory tracking to dynamic obstacle avoidance with multi-step MPC, demonstrating reliable tracking and real-time safety. The framework's linear structure also facilitates adaptive extensions and hardware deployment. Looking forward, we aim to scale to higher-dimensional systems and incorporate higher-order safety indices.

Overall, Koopman-based safe control offers a scalable and interpretable alternative to conventional model-free methods, particularly in safety-critical robotics.

REFERENCES

[1] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear mpc: bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020.

[2] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[3] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.

[4] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[5] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[6] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," *Advances in neural information processing systems*, vol. 28, 2015.

[7] F. Li, A. Abuduweili, Y. Sun, R. Chen, W. Zhao, and C. Liu, "Continual learning and lifting of koopman dynamics for linear control of legged robots," in *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, ser. Proceedings of Machine Learning Research, N. Ozay, L. Balzano, D. Panagou, and A. Abate, Eds., vol. 283. PMLR, 04–06 Jun 2025, pp. 136–148.

[8] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

[9] H. Chen, A. ABUDUWEILI, A. Agrawal, Y. Han, H. Ravichandar, C. Liu, and J. Ichnowski, "Korol: Learning visualizable object feature with koopman operator rollout for manipulation," in *Conference on Robot Learning*. PMLR, 2025, pp. 4509–4524.

[10] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[11] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4906–4913.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[13] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," 2019. [Online]. Available: https://arxiv.org/abs/1903.08792

[14] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[15] J. Xie, L. Hu, J. Qin, J. Yang, and H. Gao, "Safety-critical control with control barrier functions: A hierarchical optimization framework," *arXiv preprint arXiv:2410.15877*, 2024.

[16] T. Wei and C. Liu, "Safe control with neural network dynamic models," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 739–750.

[17] V. Zinage and E. Bakolas, "Neural koopman control barrier functions for safety-critical control of unknown nonlinear systems," 2022. [Online]. Available: https://arxiv.org/abs/2209.07685

[18] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[19] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern koopman theory for dynamical systems," *arXiv preprint arXiv:2102.12086*, 2021.

[20] H. Shi and M. Q. H. Meng, "Deep koopman operator with control for nonlinear systems," 2022. [Online]. Available: https://arxiv.org/abs/2202.08004

[21] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human-robot interactions," in *Dynamic Systems and Control Conference*, vol. 46209. American Society of Mechanical Engineers, 2014, p. V003T42A003.

[22] S. Liu, C. Liu, and J. Dolan, "Safe control under input limits with neural control barrier functions," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 1970–1980. [Online]. Available: https://proceedings.mlr.press/v205/liu23e.html

[23] H. Shi and M. Q.-H. Meng, "Deep koopman operator with control for nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7700–7707, 2022.

[24] 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.