# Pick-to-Learn for Systems and Control: theoretical review with a showcase in reachability analysis

Dario Paccagnan[1], Daniel Marks[1], Marco C. Campi[2], Simone Garatti[3]

*Abstract*— **Data-driven methods have become popular tools for tackling increasingly complex design problems in systems and control. In safety critical settings, deploying these methods requires rigorous safety and performance guarantees. Unfortunately, existing approaches often achieve this requirement at the cost of sacrificing valuable data for testing and calibration, or by restricting the design space, thus leading to suboptimal performances. In this work, we introduce *Pick-to-Learn (P2L) for Systems and Control*, a framework that builds on recent results in sample compression theory to equip *any* data-driven control method with safety and performance guarantees. Crucially, P2L enables the use of *all* available data to *jointly* synthesize and certify the design, eliminating the need to set aside data for calibration or validation purposes. As a result, P2L delivers designs and certificates that improve the current state of the art. We demonstrate this on existing benchmarks in reachability analysis.**

## I. Introduction

The increasing availability of data and rapid advancements in learning-based methods have, over the past years, blurred the boundaries between the fields of *Systems & Control* and *Machine Learning*. Indeed, data-availability and learning-based techniques have not only shown great potential for enhanced control synthesis, but they have also driven fundamental research in new applications arising in fields such as automotive, robotics and medicine.

Due to the critical nature of many control tasks, the deployment of data-driven policies in real-world settings often requires equipping them with safety and performance guarantees. For example, autonomous vehicles, robotics, and industrial automation require controllers that satisfy strict state constraints, often in the face of uncertainty. In learning-based approaches, safety and performance certificates must themselves be based on *data*. This need has motivated much recent work at the interface of statistical learning and control, where guarantees must integrate seamlessly with data-driven design.

Unfortunately, many existing approaches for ensuring safety and performance guarantees either require setting aside valuable data for testing and calibration or constrain the choice of algorithms during design, ultimately leading to suboptimal performance. As a consequence, there remains a strong need to develop broadly applicable, theoretically sound methods that fully exploit the information in the data to *simultaneously* design and provide guarantees.

**Pick-to-Learn for Systems and Control.** This paper presents *Pick-to-Learn (P2L) for Systems and Control*, a novel framework designed to equip any data-driven control method with cutting-edge safety and performance guarantees. The contribution is twofold: (i) we review P2L, introduced earlier by the same authors of this paper in [10], so as to make it available to the community of systems and control; (ii) we showcase its applicability in systems and control problems by deploying P2L for data-driven reachability analysis. Concretely, we focus on the following problem: given a sample of instances of the terminal state of a stochastic system, determine an approximate reachable set and guarantee that, with high probability, future realizations of the terminal state will fall within this set.

P2L builds on a recent breakthrough in sample compression theory [2] and, crucially, enables the use of all available data to jointly synthesize and certify a control policy or a system design. As a result, P2L is well-placed to deliver designs and certificates that surpass the state of the art across a wide range of core problems in systems and control, as demonstrated in this paper through numerical experiments. P2L enjoys the following desirable features:

- **Distribution-free.** P2L comes with formal guarantees of safety/performance without needing access to the data-generating distribution. This is important because, while data may be readily available, the data-generating distribution is often unknown or only partially known.
- **Calibration-free.** P2L does not require to set aside a portion of the dataset for testing or calibration. This is crucial in applications where data are a limited or costly resource because withholding a calibration/test set may worsen the quality of the design.
- **Accurate bounds.** P2L produces accurate probabilistic guarantees, i.e., guarantees that take a small margin from the actual behavior. This is important for properly assessing performance and safety requirements,

This paper was partially supported by the EPSRC grant EP/Y001001/1, funded by the International Science Partnerships Fund (ISPF) and UKRI, by the PRIN 2022 project 2022RRNAEX "The Scenario Approach for Control and Non-Convex Design" (CUP: D53D23001440006), funded by the NextGeneration EU program (Mission 4, Component 2, Investment 1.1), and by the PRIN PNRR project P2022NB77E "A data-driven cooperative framework for the management of distributed energy and water resources" (CUP: D53D23016100001), funded by the NextGeneration EU program (Mission 4, Component 2, Investment 1.1).

[1] D. Paccagnan and D. Marks are with the Department of Computing, Imperial College London, London, UK, {d.paccagnan, dm2020}@imperial.ac.uk.

[2] M.C. Campi is with the Dipartimento di Ingegneria dell'Informazione, Università di Brescia, Italy, marco.campi@unibs.it.

[3] S. Garatti is with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy, simone.garatti@polimi.it.

- **Wide applicability.** P2L provides a framework, not a single specific design technique. As a result, it can be applied to a variety of problems, ranging, from reachability analysis to neural optimal control.

At its core, P2L transforms any data-driven algorithm into a *preferent compression scheme*. This enables to leverage recent results in [2] to obtain probabilistic guarantees on any property of interest, e.g., constraint satisfaction, or complex safety guarantees. Specifically, P2L consists of a meta-algorithm constructing a loop around the original data-driven algorithm and feeding it with increasingly larger subsets of the available data, where the data points that least satisfy the property of interest are iteratively added. The degree of compression achieved, i.e., the size of the subset obtained at termination, determines the probabilistic bound on the satisfaction of the property of interest.

The work [2] belongs to the literature of the Scenario Approach, and establishes it as a general methodology to provide tight guarantees for data-driven design schemes that satisfy a suitable compression property. Notably, this is the case for convex and non-convex optimization problems, [1], [5], [6]. However, many modern learning-based algorithms do not meet this compression requirement, so preventing a direct application of these results. By turning any given algorithm into a compressing-inducing scheme, P2L makes it amenable to scenario analysis. In this light, P2L can be seen as an *enabler*, one that greatly broadens the applicability of the Scenario Approach beyond its initial domain of operation. In fact, P2L is particularly well-suited for modern machine learning pipelines with somewhat opaque structures, for which guarantees are difficult to obtain.

### A. Related work

There are various methods that transform available data into decisions, while equipping them with probabilistic guarantees. Below we describe some of the most popular paradigms.

**Conformal Prediction.** Conformal Prediction is a set of tools introduced in [13], [11] and often used in regression and classification tasks to construct guaranteed prediction sets that contain the true outcome with a specified probability. In its calibration-based version, Conformal Prediction has recently gathered momentum within the community of systems and control (see [9] for a recent survey). The main downside in applying this method is that calibration-based Conformal Prediction requires withholding a portion of the dataset in order to generate the desired guarantee.

**Test-Set approach.** Rooted in statistical hypothesis testing, the Test-Set approach is a classical tool to obtain formal probabilistic guarantees on a property of interest, [7]. As the name suggests, this approach requires withholding a portion of the data exclusively for evaluating the quality of the resulting design. While the approach is simple, the reduced size of data used for training may negatively impact the quality of the final design.

## II. Pick-to-Learn for Systems & Control

In this section we revisit in a control-oriented setup the meta-algorithm Pick-to-Learn (P2L) originally introduced in [10]. To this aim, we first introduce the ingredients needed: *a dataset; a synthesis algorithm; a property of interest.*

In more precise terms, we consider a data-driven control problem, where we are given a sample of $N$ data points $\{z_i \in \mathcal{Z}\}_{i=1}^N$ to be used for synthesis purposes. These data are historical observations of uncertain element affecting the problem of interest.[1] For example, $z_1, \dots, z_N$ may represent realizations of the disturbance in a stochastic optimal control problem, which we intend to use for control synthesis. The data are collected in the dataset $D = (z_1, z_2, \dots, z_N)$, which has to be interpreted as a list of elements and is modeled as a realization of $\boldsymbol{D} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N)$. Therein, $\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N$ are independent and identically distributed (i.i.d.) random elements taking value in $\mathcal{Z}$ and defined over a probability space $(\Omega, \mathcal{F}, \mathbb{P})$.[2]

We then assume access to a given data-driven synthesis algorithm $L : D \mapsto h$, which maps any available dataset $D$ of any length into a decision $h$ taken from a feasible set of decisions $\mathcal{H}$, e.g., a control policy from a feasible set of policies. The synthesis algorithm is supposed to be suited for the specific design problem we are facing and its choice may well depend on any information we may have on the problem. For example, $L$ could be gradient descent for policy synthesis in neural-based control design. $L$ is meant to be the algorithm that is used to address the problem at hand and we pose no restrictions on it.[3]

The third ingredient is a property of interest, whose satisfaction depends on the interaction between decision and uncertainty. For its formalization, we introduce a map $\varphi : \mathcal{H} \times \mathcal{Z} \to \{0, 1\}$, where $\varphi(h, z) = 1$ if the decision $h$ satisfies the property of interest for the uncertainty instance $z$ (in this case, we will also use the terminology "property $\varphi$ is satisfied"), and $\varphi(h, z) = 0$ otherwise. For example, when $h$ represents a control policy, property $\varphi$ may be used to encode that the trajectory of the closed loop system associated to a disturbance realization $z$ satisfies a given input-state constraint. The main objective of P2L is to construct decisions through the utilization of $L$ for which formal certifications on the satisfaction of $\varphi$ can be given.

To proceed, we assume to have access to a quantifier of the degree of dissatisfaction when $h$ fails to satisfy property $\varphi$. This enables us to identify the data point for which $h$ exhibits the *highest* degree of dissatisfaction, which is instrumental to the use of P2L.[4] For example, given a control policy $h$,

[1]Data points have been also called *scenarios* within the Scenario Approach literature.

[2]Boldface denotes random quantities, which we assume measurable.

[3]Note that $L$ may well be permutation dependent, i.e., feeding $L$ with $(z_1, \dots, z_N)$ or a permutation of it may result in different outputs.

[4]Quantifying the degree of dissatisfaction is not strictly necessary. If only binary information is given and no such quantifier is available, any tie-breaking rule selecting *one* data point among those resulting in the dissatisfaction of $\varphi$ suffices to make P2L working.

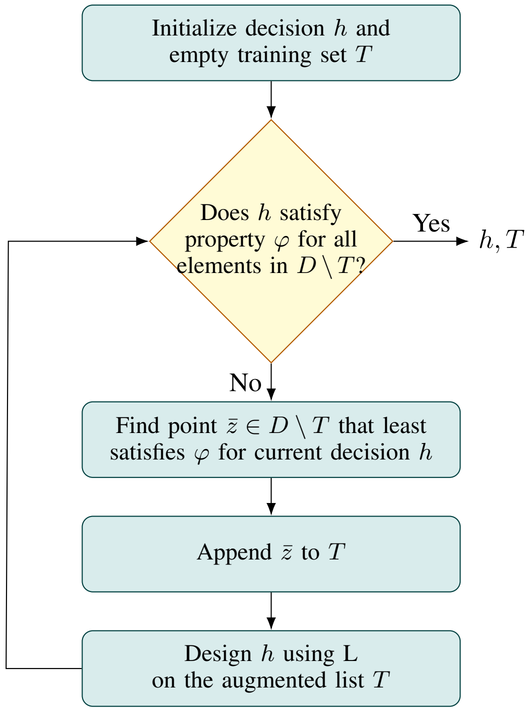



Fig. 1: Schematic presentation of Pick-to-Learn.

the trajectory of the resulting closed loop system associated to the disturbance realization $z_1$ may dissatisfy an input-state constraint "more" than that associated to $z_2$ if violation occurs for a larger time span or if the maximum distance between the trajectory and the constraint is larger. Also in this case, the quantification of the dissatisfaction level is problem dependent and P2L leaves freedom to the user.

We are now ready to introduce the meta-algorithm P2L. Referring to Figure 1, P2L works by iteratively feeding algorithm $L$ with a growing list $T$ of data points taken from $D$, while terminating when the current decision satisfies the property of interest for all the data points that have *not* been yet used for the synthesis step. Until this is not the case, P2L appends to $T$ the data point among those not yet used exhibiting the greatest level of dissatisfaction for the current decision. At termination, P2L returns both the final list $T$ of actively used data points and the corresponding decision $h$.

A more precise formalization of P2L is given in Algorithm 1. Given a synthesis algorithm $L$, a property $\varphi$, and an initial decision $h_0$, the meta-algorithm P2L maps a dataset $D$ to the pair $(h, T) = \text{P2L}(D)$, where $h \in \mathcal{H}$ is the resulting decision and $T \subseteq D$ is a sublist of $D$ called training set. In the initialization (Line 1 in Algorithm 1), the training set $T$ is set to empty and the decision is $h = h_0$. In the main loop (Line 2), if, with the current $h$, property $\varphi$ is satisfied for all elements in $D \setminus T$, the loop terminates. If not, P2L proceeds to compute the element $\bar{z}$ in $D \setminus T$ that exhibits the greatest level of dissatisfaction with the current $h$ (Line 3). Then, $T$ is augmented by appending $\bar{z}$ (Line 4) and the decision is updated by running $L$ with $T$ (Line 5). When

**Algorithm 1** – Meta-algorithm P2L

```
1: Initialize: T = ∅, h = h_0
2: while ∃ z ∈ D \ T : φ(h, z) = 0 do
3:     z̄ ← most unsatisfied point in D \ T
4:     T ← T ∪ {z̄}                          // Augment T
5:     h ← L(T)                            // Synthetize h
6: end while
7: return h, T                 // Decision h and list T
```

P2L terminates, it returns both the decision $h$ and the list $T$. All operations in Algorithm 1 are intended to be operations with lists, so, e.g., $D \setminus T$ is the sublist of $D$ obtained by removing from $D$ the elements appearing in $T$ in the same order and number.

*Remark 1 (Wide-applicability):* Note that the choice of the learning algorithm $L$, property $\varphi$, and initial decision $h_0$ are arbitrary. Perhaps unexpectedly, strong generalization guarantees can be secured at this high level of abstraction (Theorem 1), thus enabling the deployment of P2L across significantly different problems in systems and control. As an example, in this work we apply P2L to reachability analysis.

### A. Probabilistic guarantees

In this section, we equip the decision returned by P2L with formal guarantees on the satisfaction of property $\varphi$ against unseen instances of the uncertainty. Toward this goal, we begin by introducing the notion of risk of a decision.

**Definition 1 (Risk):** The risk of a decision $h \in \mathcal{H}$ is defined as

$$R(h) = \mathbb{P}\{\varphi(\boldsymbol{z}, h) \neq 1\}$$

where $\boldsymbol{z}$ is a random element distributed as, and independent of, any of the $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N$.

The risk of a decision $h$ measures the probability that $h$ does not satisfy property $\varphi$ when facing a new uncertainty instance. For example, if the property encodes the satisfaction of state-input constraint and the decision represents a control policy, the risk measures the probability with which the control policy $h$ does not satisfy such state-input constraint over a new realization of the disturbance $\boldsymbol{z}$.

Because the dataset $D$ is the realization of a random element $\boldsymbol{D} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$, the decision returned by P2L is itself a realization of a random element through $(\boldsymbol{h}, \boldsymbol{T}) = \text{P2L}(\boldsymbol{D})$. Therefore, we aim to make statements of the form

> *"With high confidence $1 - \delta$ with respect to the $N$ i.i.d. draws generating the dataset, the risk of the decision returned by P2L is bounded by $\bar{\varepsilon}(|\boldsymbol{T}|, \delta, N)$"*,

for a suitably defined $\bar{\varepsilon}(\cdot, \cdot, \cdot)$. Note that, the guarantees obtained will depend, crucially, on the length of the returned list $T$, which we denote throughout with $|T|$. Toward providing such guarantees, it is instrumental to introduce the function

$\Psi_{k,\delta}:(0,1)\to\mathbb{R}$ indexed by $k=0,1,\dots,N-1$ and by the confidence parameter $\delta\in(0,1)$:

$$\Psi_{k,\delta}(\varepsilon)=\frac{\delta}{N}\sum_{m=k}^{N-1}\frac{\binom{m}{k}}{\binom{N}{k}}(1-\varepsilon)^{-(N-m)}.$$

We are now ready to state the main result, namely, a bound on the risk of the decision $h$ returned by P2L. See [10] for a proof.

*Theorem 1:* Let $(\boldsymbol{h},\boldsymbol{T})=\text{P2L}(\boldsymbol{D})$ be the output of P2L. Then, for any choice of $\delta\in(0,1)$, it holds that

$$\mathbb{P}\{R(\boldsymbol{h})\le\bar{\varepsilon}(|\boldsymbol{T}|,\delta,N)\}\ge 1-\delta,$$

where, for $k=0,1,\dots,N-1$, $\bar{\varepsilon}(k,\delta,N)$ is defined as the unique solution to the equation $\Psi_{k,\delta}(\varepsilon)=1$ in the interval $[k/N,1]$, while, for $k=N$, it is $\bar{\varepsilon}(N,\delta,N)=1$.

## III. Application to Data-driven Reachability

Following the works of [3], [4], [12], in this section we consider a data-driven reachability problem, where draws of the terminal state of a stochastic system are available. The task is to construct an approximate reachable set and to give formal probabilistic guarantees certifying that, with high probability, a newly sampled trajectory will fall inside the constructed set. Naturally, a significant indicator of the quality of such a set is its volume, which should be minimized to reduce conservatism and avoid trivial statements, e.g., identifying the reachable set with the whole state-space.

### A. Problem Formulation

Consider a dynamical system described by the state transition function $\Phi$ mapping the initial state $x_0\in\mathbb{R}^{n_x}$ at time $t_0$ to a final state $z=\Phi(t_1;t_0,x_0,w)$ at time $t_1$, under the effect of the disturbance $w:[t_0,t_1]\to\mathbb{R}^{n_w}$. For instance, when the system is described by an ordinary differential equation $\dot{x}(t)=f(t,x(t),w(t))$, and its initial condition is $x(t_0)=x_0$, then $z$ represents the solution $x(t)$ at time $t_1$. We assume that both the initial state $x_0$ and the disturbance $w$ are uncertain and that they can be modeled as random elements with an unknown probability distribution (possibly introducing correlation among distinct time instants). As a result, the final state becomes a random variable.

In the following, we assume access to a dataset $D=\{z_1,\dots,z_N\}$ containing draws of the terminal state. Formally, $D$ is modeled as a realization of $\boldsymbol{D}=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\dots,\boldsymbol{z}_N\}$, where $\boldsymbol{z}_1,\boldsymbol{z}_2,\dots,\boldsymbol{z}_N$ are independent and identically distributed. This realization can be obtained in various ways, for instance by simulating the system dynamics, or by measuring its terminal state via experiments. Our task is then to utilize this dataset to construct a subset of the state-space, denoted with $S(D)\subseteq\mathbb{R}^n$, such that a desired fraction of the probability mass of $\boldsymbol{z}$ falls in $S(D)$. The problem can be described as follows.

**Data-driven reachability.** Given $N$ i.i.d. draws of the final state collected in the dataset $D$, determine a set $S(D)$ and a bound $\varepsilon(D)$ such that, with confidence $1-\delta$ over the extraction of $\boldsymbol{D}$, it holds that $R(S(\boldsymbol{D}))\le\varepsilon(\boldsymbol{D})$ where $R(S(D))=\mathbb{P}\{\boldsymbol{z}\notin S(D)\}$ and $\boldsymbol{z}$ is a random element distributed as, and independent of, any of the $\boldsymbol{z}_1,\dots,\boldsymbol{z}_N$.

### B. P2L for data-driven reachability

In this section we show how to apply P2L to perform data-driven reachability, and equip the resulting reachable-set with a probabilistic bound in Section III-B. Toward the first objective, recall that the meta-algorithm P2L (Algorithm 1) is fully specified once the following elements are defined: i) a synthesis algorithm $L$, ii) property $\varphi$ and a quantifier for the level of dissatisfaction, iii) an initialization $h_0$. We discuss each of them in the following.

**Synthesis algorithm.** In the context of data-driven reachability, a synthesis algorithm $L$ maps any $T\subseteq D$ to a corresponding set $S\subseteq\mathbb{R}^{n_x}$, with the aim of approximating well the support of the final state distribution. To this purpose, we use the algorithm proposed by [8] and also employed by [4] and [12] in the context of reachability analysis via PAC-Bayes and Conformal Prediction. Such an algorithm utilizes $T$ to construct a so-called *empirical inverse Christoffel function*, from which one determines the set $S$ as a suitable sublevel set.[5] Towards this goal, given an integer $d\in\mathbb{N}$ which plays the role of a hyper-parameter, denote by $v(x)$ the vector of monomials on $\mathbb{R}^{n_x}$ (i.e., the state-space) of degree no higher than $d$. For example, when $n_x=3$ and $d=2$, one has $v(x)=[1,\ x_1,\ x_2,\ x_3,\ x_1x_2,\ x_1x_3,\ x_2x_3]$. For a given $T$, the empirical inverse Christoffel function is then defined as $\kappa^{-1}(x)=v(x)^\top\hat{M}^{-1}v(x)$, where $\hat{M}$ is the matrix of empirical moments, defined by

$$\hat{M}=\frac{1}{|T|}\sum_{z_i\in T}v(z_i)v(z_i)^\top.$$

The set $S$ is then obtained by computing the $\alpha$-sublevel set of the empirical inverse Christoffel function, where we take $\alpha=\max_{z_i\in T}\kappa^{-1}(z_i)$. That is, we "slice" the empirical inverse Christoffel function at the largest value it takes over points in $T$. Correspondingly, we define the set

$$S=\{x\in\mathbb{R}^{n_x}\ \text{s.t.}\ v(x)^\top\hat{M}^{-1}v(x)\le\alpha\}.$$

Here, it will be convenient to let the algorithm $L$ return the decision $h=(S,\kappa^{-1})$, i.e., both the set $S$ and the empirical inverse Christoffel function as the latter will be useful when defining the quantifier of the level of dissatisfaction.

**Property $\varphi$ and quantifier of the level of dissatisfaction.** Recall that our goal is to bound the probability with which a newly extracted draw falls outside the set constructed by P2L. This corresponds to bounding the risk defined in (1), where the property $\varphi$ of interest is as follows: given a decision $h=(S,\kappa^{-1})$ and draw $z$, $\varphi(h,z)=1$ if $z\in S$, and zero else. In light of that, we wish P2L to terminate when

[5] We refer the reader to [8] for the theoretical underpinnings of this method, e.g., see [8, Thm 3.13] for the convergence of $S$ to the true support of the distribution when the sample size grows. Here, we provide a minimal, but self sufficient exposition.

all points in $D \setminus T$ fall inside the set $S$. When this is not the case, we utilize the value of $\kappa^{-1}(z_i)$ to quantify the level of dissatisfaction and P2L selects the point in $D \setminus T$ which attains the highest value of $\kappa^{-1}$.

**Initialization.** We utilize a small fraction of $D$ to initialize P2L. Specifically, we use $N_i << N$ datapoints to compute an initial set and empirical inverse Christoffel function via the above-described synthesis algorithm.

**Probabilistic bound.** Bounding the probability with which a newly extracted terminal state falls outside the set $S$ returned by P2L is obtained as a corollary of Theorem 1 for the choice of property $\varphi$ previously introduced.

*Corollary 1:* Let $S(D)$ be the reachable set produced by P2L on dataset $D$, with learning algorithm, quantifier of dissatisfaction, and initialization as in the above. Let $T$ denote the compressed set returned by P2L. Then, with confidence $1-\delta$ over the realization of $\boldsymbol{D}$ it holds that

$$R(S(\boldsymbol{D})) \leq \bar{\varepsilon}(|\boldsymbol{T}|, \delta, N - N_i).$$

### C. Numerical example

We consider the benchmark employed in [4], [12] and compare vis-a-vis the performance of P2L against state-of-the-art algorithms based on Conformal Prediction and the Test-Set approach. Specifically, the works [4], [12] consider the Duffing oscillator

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1(1-x_1^2) - \alpha x_2 + \gamma \cos(\omega t), \end{cases}$$

with $\alpha = 0.05$, $\gamma = 0.4$, $\omega = 1.3$, where the initial state $x_0$ is uncertain and we are given access to $N$ i.i.d. realization of it. Exploiting knowledge of the above dynamics, it is then immediate to construct the set of i.i.d. draws $D = \{z_1, \ldots, z_N\}$, where each $z_i$ consists of a terminal state obtained at time $t_1 = 100$, from the corresponding initial condition $x_0$ at $t_0 = 0$. The goal is to leverage this dataset and construct a set $S(D)$ over $\mathbb{R}^2$ containing as much probabilistic mass of $\boldsymbol{z}$ as possible. We do so using three different methods, all of which are compared using the same confidence level $\delta = 0.01$, the same number of datapoints $N = 2000$ and the same choice of $d = 10$ for the inverse Christoffel function. For each method, we compute the volume of the resulting set $S(D)$, and the actual risk, by generating 500000 terminal states and computing the fraction of them falling outside $S(D)$. We repeat this whole experiment for 2000 times, in order to give statistical significance to our findings. Before presenting the results, we provide details on how each individual method is implemented.

*Pick-to-Learn.* We run the meta-algorithm P2L presented in Algorithm 1 with the choice of learning algorithm, quantifier of dissatisfaction, and initialization previously described. Here we take $N_i = 200$, i.e., 10% of the available dataset. With the choice of $\delta = 0.01$ and $N = 2000$ as per above, the average of the upper bound to the probability of observing a new terminal state lying outside $S(D)$ returned by P2L is $0.034$, with a standard deviation of $0.0029$.

*Conformal Prediction.* We run Conformal Prediction exactly as presented in [12], and give it the benefit of experimenting with different fractions of the dataset (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%) in order to construct the calibration set.[6] In order to compare with P2L on equal footing, for each such fraction, we employ the conformal bound using the empirical quantile that returns the risk bound closest to that returned by P2L (selecting a higher quantile would achieve better bound but larger volume, while lower quantiles would achieve a worse bound but smaller volume; none of which is comparable). Doing so, both methods return an almost identical certificate, and therefore can be fairly compared by analyzing the volume of the resulting sets.

*Test-Set.* We run the Test-Set method [7], also giving it the benefit of experimenting with different fractions of dataset for training vs testing (again, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%). We then select the fraction that, among all those achieving a risk bound no larger than that of P2L, returns the set $S$ with smallest volume.

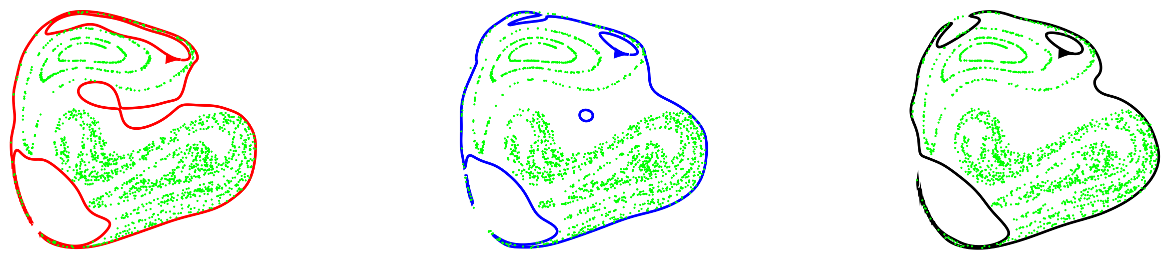
**Results.** Figure 3 depicts the empirical distribution of the volume of the sets $S$ obtained by the three approaches (top) and the empirical distributions of their actual risk (bottom). Averages and standard deviations of these values are reported in Table I. In Figure 2 we present the sets with median volume obtained by the three approaches so as to visualize the impact of the volume of $S$.

| | **P2L** | Conf | T-S |
|---|---|---|---|
| Volume | **0.362 ± 0.009** | 0.410 ± 0.012 | 0.405 ± 0.009 |
| Actual risk | **0.007 ± 0.002** | 0.023 ± 0.005 | 0.027 ± 0.008 |

TABLE I: Mean ± standard deviations of the reachable sets' volumes and actual risks. P2L outperforms the other methods across all metrics (**bold**).

**Comparison.** As we have fixed the value of $\delta$, $N$ and $\varepsilon$ to be the same for all algorithms, the methods are now directly comparable. Taking into account the top and bottom panels in Figure 3, it emerges clearly that, for a given identical probabilistic certificate, P2L outperforms all other approaches both in terms of i) constructing a set $S$ with smaller volume (top panel), corresponding to a less conservative estimate, and of simultaneously ii) trapping a larger amount of the probabilistic mass of $\boldsymbol{z}$ into it (bottom panel), corresponding to better post-training safety. Note that the difference in the volume of the sets returned by the three algorithms is substantial, as can be visually appreciated from Figure 2.

[6]Note that, testing different configurations and selecting the best one as done here, formally requires summing the individual confidence levels used in each test, resulting in an overall confidence for Conformal Prediction that is lower than that obtained by P2L (specifically, $1 - 9 \cdot \delta = 0.91$ vs. 0.99). As we will see, in spite of giving Conformal Prediction, and later Test-Set, this advantage, P2L will outperform them.



(a) P2L, volume = 0.3616 (b) Conformal, volume = 0.4101 (c) Test-Set, volume = 0.4063

Fig. 2: Reachable set with median volume obtained by different methods. The set generated by P2L is visibly less conservative, while resulting in the same certificate.

**P2L better exploits the data.** Since the algorithm utilized to design the reachable set is identical across the three methods tested (they all use the same inverse Christoffel function approach), the fact that P2L outperforms the others can not be ascribed to this. This advantage comes from a better exploitation of the data as P2L uses all data to *jointly* learn the set $S$ and provide a risk bound. This is not the case for Test-Set, for which the data used to train cannot be used to test. This is also not the case for Conformal Prediction, where the data used to construct the Christoffel function cannot be used to obtain a probabilistic bound.

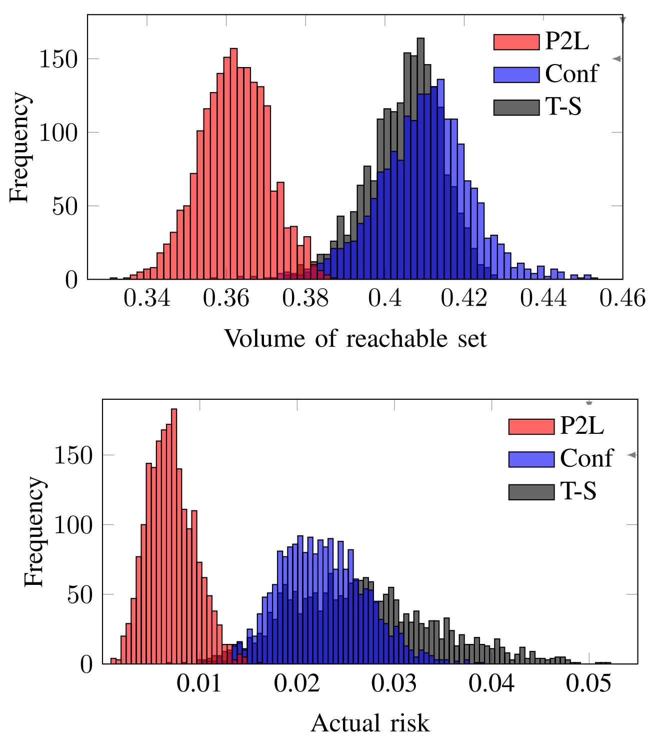



Fig. 3: Empirical distributions of volumes (*top panel*) and of actual risks (*bottom panel*) achieved by the considered approaches. P2L returns sets with lower volume and lower risk as compared to both Conformal Prediction and Test-Set.

## References


[1] M.C. Campi and S. Garatti. Wait-and-judge scenario optimization. *Mathematical Programming*, 167(1):155–189, 2018.

[2] M.C. Campi and S. Garatti. Compression, generalization and learning. *Journal of Machine Learning Research*, 24(339):1–74, 2023.

[3] A. Devonport, F. Yang, L. El Ghaoui, and M. Arcak. Data-driven reachability analysis with christoffel functions. In *Proceedings of the 60th IEEE Conference on Decision and Control*, pages 5067–5072, 2021.

[4] A. Devonport, F. Yang, L. El Ghaoui, and M. Arcak. Data-driven reachability and support estimation with christoffel functions. *IEEE Transactions on Automatic Control*, 68(9):5216–5229, 2023.

[5] S. Garatti and M.C. Campi. Risk and complexity in scenario optimization. *Mathematical Programming*, 191:243–279, 2022.

[6] S. Garatti and M.C. Campi. Non-convex scenario optimization. *Mathematical Programming*, 209:557–608, 2025.

[7] J. Langford. Tutorial on practical prediction theory for classification. *Journal of machine learning research*, 6(3), 2005.

[8] J.B. Lasserre and E. Pauwels. The empirical christoffel function with applications in data analysis. *Advances in Computational Mathematics*, 45:1439–1468, 2019.

[9] L. Lindemann, Y. Zhao, X. Yu, G.J. Pappas, and J.V. Deshmukh. Formal verification and control with conformal prediction. *Arxiv*, pages 1–50, 2024.

[10] D. Paccagnan, M.C. Campi, and S. Garatti. The pick-to-learn algorithm: Empowering compression for tight generalization bounds and improved post-training performance. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 18165–18185. Curran Associates, Inc., 2023.

[11] G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3):371–421, 2008.

[12] A. Tebjou, G. Frehse, and F. Chamroukhi. Data-driven reachability using christoffel functions and conformal prediction. In *Proceedings of the 12th Symposium on Conformal and Probabilistic Prediction with Applications*, volume 204 of *Proceedings of Machine Learning Research*, pages 194–213, 2023.

[13] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.