A. Proof of Theorem 1

To prove Thm. 1, we first introduce the theory of scenario optimisation. Let us take a look at the optimization problem below:

$$\begin{aligned} &\min_{\boldsymbol{\gamma}\in\Gamma\subseteq\mathbb{R}^m} \boldsymbol{b}^\top\boldsymbol{\gamma}, \\ &s.t.\ f_{\boldsymbol{\omega}}(\boldsymbol{\gamma}) \le 0,\ \forall\boldsymbol{\omega}\in\Omega, \end{aligned} \tag{4}$$

where $f_{\boldsymbol{\omega}}$ is a convex and continuous function of the m-dimensional optimization variable $\boldsymbol{\gamma}$ for every $\boldsymbol{\omega}\in\Omega$, and both Ω and Γ are convex and closed. It is difficult to solve (4), since there are infinitely many constraints. In [9], Calafiore et al. proposed the following scenario optimisation to solve (4) with a PAC guarantee.

Definition 3 *Let $\mathbb{P}$ be a probability measure on Ω. The scenario approach to handle the optimization problem* (4) *is to solve the following problem. We extract K independent and identically distributed (i.i.d.) samples $(\boldsymbol{\omega}_i)_{i=1}^K$ from Ω according to the probability measure $\mathbb{P}$:*

$$\begin{aligned} &\min_{\boldsymbol{\gamma}\in\Gamma\subseteq\mathbb{R}^m} \boldsymbol{b}^\top\boldsymbol{\gamma}, \\ &\text{s.t.}\ \bigwedge_{i=1}^{K} f_{\boldsymbol{\omega}_i}(\boldsymbol{\gamma}) \le 0. \end{aligned} \tag{5}$$

The scenario optimisation only considers a finite subset of constraints. In [9, 10], a PAC guarantee between the scenario solution in (5) and its original optimization in (4) can be constructed with sufficient samples.

Theorem 2 ([10]) *If (5) is feasible and has an optimal solution $\boldsymbol{\gamma}_K^*$, and*

$$\epsilon \ge \frac{2}{K}(\ln\frac{1}{\eta} + m), \tag{6}$$

where K is the number of samples, and ϵ and η are the pre-defined error rate and the significance level, respectively, then with confidence at least $1-\eta$, the optimal $\boldsymbol{\gamma}_K^*$ satisfies all the constraints in Ω but only at most a fraction of probability measure ϵ, i.e., $\mathbb{P}(f_{\boldsymbol{\omega}}(\boldsymbol{\gamma}_K^*) > 0) \le \epsilon$.

In DEEPPAC [51], scenario optimisation is used for robustness verification of classification DNNs. To adapt Thm. 2 to our settings, where stochastic output is considered, we must describe both the sampling distribution π and the stochasticity in the trajectory prediction model in the probability distribution $\mathbb{P}$.

For label robustness, stochasticity in $\Delta(\mathbf{X})$ only comes from the stochasticity of the output $g(\mathbf{X})$. We regard the model g as a random variable $g(\mathbf{X},\omega) : (\mathit{\Omega},\mathcal{F},\mathrm{Pr}_{\mathbf{X}}) \to (\mathbb{R}^{2\times T_\mathrm{f}},\mathcal{B}(\mathbb{R}^{2\times T_\mathrm{f}}))$, where $(\mathit{\Omega},\mathcal{F},\mathrm{Pr}_{\mathbf{X}})$ is the probability space of the stochasticity in $g(\mathbf{X})$, and $\mathcal{B}(\cdot)$ is the Borel σ-algebra, i.e., the σ-algebra generated by the open sets. Now we consider the product measurable space $(B(\hat{\mathbf{X}},r)\times\mathit{\Omega},\mathcal{B}(B(\hat{\mathbf{X}},r))\times\mathcal{F})$ and we define the probability measure $\mathbb{P}$ on it according to the sampling distribution π and the probability measure $\mathrm{Pr}_{\mathbf{X}}$ in the standard way: For $B\in\mathcal{B}(B(\hat{\mathbf{X}},r))$ and $F\in\mathcal{F}$, the measure of the measurable rectangle $B\times F$ is

$$\mathbb{P}(B\times F) = \int_B \mathrm{Pr}_{\mathbf{X}}(F)\pi(\mathrm{d}\mathbf{X});$$

it is easy to see that $\mathbb{P}$ is a probability measure on the semi-ring of the measurable rectangles in $\mathcal{B}(B(\hat{\mathbf{X}},r))\times\mathcal{F}$, and thus it can be uniquely extended to a probability measure, still denoted by $\mathbb{P}$, on $(B(\hat{\mathbf{X}},r)\times\mathit{\Omega},\mathcal{B}(B(\hat{\mathbf{X}},r))\times\mathcal{F})$. When sampling in $B(\hat{\mathbf{X}},r)$ according to π, we are actually sampling in the probability space $(B(\hat{\mathbf{X}},r)\times\mathit{\Omega},\mathcal{B}(B(\hat{\mathbf{X}},r))\times\mathcal{F},\mathbb{P})$, so according to Thm. 2, where the dimensionality $m = 2T_\mathrm{p}(N+1)+1$, it suffices to prove Thm. 1.

For pure robustness, stochasticity in $\Delta(\mathbf{X})$ comes from the stochasticity of both $g(\mathbf{X})$ and $g(\hat{\mathbf{X}})$, so we sample in the measurable space $(B(\hat{\mathbf{X}},r)\times\mathit{\Omega}\times\mathit{\Omega},\mathcal{B}(B(\hat{\mathbf{X}},r))\times\mathcal{F}\times\mathcal{F})$, and the probability of a measurable rectangle $B\times F_1\times F_2$ is

$$\mathbb{P}(B\times F_1\times F_2) = \int_B \mathrm{Pr}_{\mathbf{X}}(F_1)\pi(\mathrm{d}\mathbf{X})\cdot\mathrm{Pr}_{\hat{\mathbf{X}}}(F_2).$$

By measure extension, $\mathbb{P}$ is a probability measure on $(B(\hat{\mathbf{X}},r)\times\Omega\times\Omega, \mathcal{B}(B(\hat{\mathbf{X}},r))\times\mathcal{F}\times\mathcal{F})$. With the same dimensionality $m = 2T_{\mathrm{p}}(N+1)+1$, Thm. 1 is proved.

The deduced PAC-model robustness is obviously for label robustness: When $\max_{\mathbf{X}\in B(\hat{\mathbf{X}},r)} \widetilde{\Delta}(\mathbf{X}) + \lambda^* \le s$, with confidence $1-\eta$ we have

$$\mathbb{P}(\Delta(\mathbf{X}) \le s) \ge \mathbb{P}(\Delta(\mathbf{X}) \le \widetilde{\Delta}(\mathbf{X}) + \lambda^*) \ge \mathbb{P}(|\widetilde{\Delta}(\mathbf{X}) - \Delta(\mathbf{X})| \le \lambda^*) \ge 1-\epsilon,$$

which implies that g is PAC-model label-robust in $B(\hat{\mathbf{X}},r)$. As for pure robustness, with the same deduction, when $\max_{\mathbf{X}\in B(\hat{\mathbf{X}},r)} \widetilde{\Delta}(\mathbf{X}) + \lambda^* \le s$, with confidence $1-\eta$ we have $\mathbb{P}(\Delta(\mathbf{X}) \le s) \ge 1-\epsilon$, indicating that with a PAC guarantee, $D(Y,\hat{Y}) \le s$ holds for any $Y \in g(\mathbf{X})$ and any $\hat{Y} \in g(\hat{\mathbf{X}})$, which is a stronger property than the pure robustness defined in Def. 2.

It is worth mentioning that, even if we use focused learning detailed in Appendix B, the PAC guarantee given by Thm. 1 will not be violated, since the PAC guarantee is constructed only in the second learning phase, while we only obtain an affine function template with fewer coefficients to be determined in the first learning phase.

B. Focused Learning

We employ a focused learning procedure for PAC model learning first described in [51]. The basic idea involves splitting the model learning stage into two, more manageable, subphases. The first subphase involves extracting $\mathcal{K}$ key features from the model based on the $\mathcal{K}$ largest coefficient magnitudes. In the second subphase we optimize our PAC model with respect to only those previously found key features. The main idea of this procedure is outlined below:

1. *First learning phase*: We learn the scores, i.e., ADEs, $\Delta_{t=1..T_1}$ for T_1 i.i.d. samples from the input region $B(\hat{x},r)$. This LP problem has d variables with T_1 constraints. For large datasets this LP problem is still too large, and so we can instead use linear regression to boost the learning time. After solving the linear problem, we find the $\mathcal{K}$ largest coefficient magnitudes, and denote the set of corresponding features by $Key(\mathcal{K}) \subseteq \{1, x_1, ..., x_d\}$.

2. *Second learning phase*: We learn the scores $\Delta_{t=1..T_2}$ for T_2 i.i.d. samples from $B(\hat{x},r)$. Rather than solving an LP problem for all d variables, we fix the non-key coefficients and generate constraints for only our $\mathcal{K}$ key features. The solution to this LP problem determines the coefficients of these key features $Key(\mathcal{K})$.

With focused learning, rather than optimizing a large LP problem with d variables and T_1+T_2 constraints, we solve only one LP problem with $\mathcal{K} \le d$ variables and T_2 constraints. Moreover, given a predetermined significance η and error rate ϵ, we can determine an appropriate number of key features $\mathcal{K}$ and sample size T_2 satisfying $\mathcal{K} \le \frac{\epsilon T_2}{2} - \ln\frac{1}{\eta} - 1$ [51, Theorem 2.5].

	r	ϵ	η	T_1	T_2
Traj++	0.03	0.01	0.01	30000	12000
MemoNet				20000	12000
AgentFormer				30000	12000
MID				4000	3000

Table 6. Detailed hyperparameter configurations for scenario optimization of each trajectory forecasting model.

C. Robustness Properties for ADE_K

In our experiment, we use a modified version of ADE, the minimum average displacement error of K trajectory samples, which is a standard metric for trajectory prediction [27, 65, 66, 62, 17]. Formally, it is defined as

$$\mathrm{ADE}_K(\mathbb{Y}, Y) = \frac{1}{T}\min_{1\le k\le K}\sum_{t=1}^{T}\|y^{t,(k)} - y^t\|_2,$$

where $\mathbb{Y} = \{(y^{1,(k)}, \ldots, y^{T,(k)}) \mid k = 1, \ldots, K\}$ is a set of K trajectory samples, and $y^{t,(k)}$ is the position at time t in the k-th sample. In the experiment, we choose $K = 20$.

In Sect. 4 of the paper, we focus on the label/pure robustness properties with the metric $D = \mathrm{ADE}$ for simplicity. The robustness properties with ADE_K needs a slight modification. We state it as follows:

Definition 4 (Label Robustness for ADE_K) *Let $\hat{\mathbf{X}} = (\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_N)$ be the past trajectories of the to-be-predicted agent and its N neighbouring agents, and Y_f its ground truth of the future trajectories of the to-be-predicted agent. Given a prediction model g, an evaluation metric D, a safety constant s, then g is label-robust at $\hat{\mathbf{X}}$ w.r.t. the perturbation radius $r > 0$ if for any $X_i \in B(\hat{X}_i, r)$ $(i = 0, 1, \ldots, N)$ and any $\mathbb{Y} \in g(X_0, X_1, \ldots, X_N)$ with $|\mathbb{Y}| = K$, we have $\mathrm{ADE}_K(\mathbb{Y}, Y_\mathrm{f}) \le s$.*

Definition 5 (Pure Robustness for ADE_K) *Let $\hat{\mathbf{X}} = (\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_N)$ be the past trajectories of the to-be-predicted agent and its N neighbouring agents. Given a prediction model g, an evaluation metric D, a safety constant s, then g is purely robust at $\hat{\mathbf{X}}$ w.r.t. the perturbation radius $r > 0$ if for any $X_i \in B(\hat{X}_i, r)$ $(i = 0, 1, \ldots, N)$ and any any $\mathbb{Y} \in g(X_0, X_1, \ldots, X_N)$ with $|\mathbb{Y}| = K$, there exists $\hat{Y} \in g(\hat{\mathbf{X}})$, s.t. $\mathrm{ADE}_K(\mathbb{Y}, \hat{Y}) \le s$.*

The PAC guarantee constructed in Thm. 1 will not be violated, where we only need to modify the measurable space as $(B(\hat{\mathbf{X}}, r) \times \Omega^K, \mathcal{B}(B(\hat{\mathbf{X}}, r)) \times \mathcal{F}^K)$ for label robustness, or $(B(\hat{\mathbf{X}}, r) \times \Omega^K \times \Omega, \mathcal{B}(B(\hat{\mathbf{X}}, r)) \times \mathcal{F}^K \times \mathcal{F})$ for pure robustness. The probability $\mathbb{P}$, as the independent coupling, can be constructed in a quite similar way as that in Appendix B, first defined on the semi-ring of the measurable rectangles, and then uniquely extended to the σ-algebra generated by it.

D. Experiments on SDD

We conduct experiments on samples from the Stanford Drone Dataset (SDD) with $r = 2$ pixels, $\eta = 0.01$ and $\epsilon = 0.01$. As shown in Tab. 7, TRAJPAC shows the similar good performance as in ETH/UCY. The learning time of our method in SDD is also as little as in ETH/UCY.

Scene	ID	Label Robustness			Pure Robustness		
		Traj++	Memo	MID	Traj++	Memo	MID
quad_0	(84, 5)	✗†	✓	✓	✗†	✓	✓
quad_3	(84, 9)	✗†	○	✗	✗†	○	○
nexus_5	(588, 10)	✗†	✓	○	✗†	○	✓

Table 7. Label/pure robustness verification on SDD with the ADE_{20} metric, where the safety constant is 50 pixels. Marks are the same as Tab. 1.

E. Experiments with varying values of r

$r = 0.03$m is an empirical value. We chose this value as perturbation radius because it is small enough, yet it already has a significant impact on the accuracy of predictions. We also conducted experiments with $r = 0.05$ and 0.1. Please refer to Tab. 8. As the perturbation radius grows larger, the ADE PAC bounds for different models generally expand yet they are still tight comparing to maximum sampled ADE. This demonstrates that the ADE PAC bound can remain unaffected with the perturbation radius increasing.

methods	r	label	bound	max	adver	pure	bound	max	adver
Memo	0.03	✓	0.98	0.81	0.73	✓	0.35	0.23	0.13
	0.05	○	1.12	0.87	0.82	✓	0.47	0.32	0.25
	0.1	✗	1.49	1.14	1.19	○	0.72	0.49	0.41
Traj++	0.03	○	1.01	0.86	0.65	✓	0.45	0.37	0.18
	0.05	○	1.07	0.86	0.86	✓	0.48	0.38	0.20
	0.1	○	1.12	0.91	0.84	○	0.57	0.43	0.14

Table 8. Label/pure robustness verification of Memonet/Trajectron++ on Zara1 (4430,69) (from UCY) with the corresponding ADE values (bound/max/adver indicate PAC bound/max sampled/adversarial) in three different perturbation radii.