Certified Human Trajectory Prediction

Supplementary Material

Here, we provide additional content to complement our main paper. It includes more details of the method regarding the diffusion denoiser in Section 6.1, and the algorithm in Section 6.2. We then provide additional results complementing our experiments. Specifically, we provide results using other metrics in Section 7.1, bring an expanded set of qualitative results in Section 7.2, demonstrate the bound sizes per timestep in Section 7.3 and analyze the bounds with adversarial attacks in Section 7.4.

6. Further details of the method

Here, we provide more details about the methodology of the paper. We first elaborate on the diffusion denoiser, then provide an algorithm for the method.

6.1. Diffusion denoiser

For the denoiser architecture, we used a simple model comprising four residual 1D convolution layers and two linear layers with ReLU activation functions and dropout. This model was then trained using the diffusion approach described below.

At training time, given each input data x^0, our diffusion model selects a step t and then adds Gaussian noise with zero mean and a pre-defined variance to the input to create a noisier version x^1. This process is repeated for t steps resulting in a noisy x^t:

$$q(x^t|x^{t-1}) = x^{t-1} + \mathcal{N}(x^t; \sqrt{1-\beta^t}x^{t-1}, \beta^t\mathbf{I}),$$

where q denotes the forward process, and β^t is the variance of the noise at step t, determined by a scheduler. We utilize a linear noise scheduler in our denoiser. The network learns to reverse the diffusion process and recover the clean signal by predicting the cumulative noise added to x^t.

At inference time, the model starts with a noisy input trajectory x^t. The step t is estimated according to the given σ and the scheduler, and then model iteratively predicts the less noisy signal, reducing the noise step-by-step to get $x^{t-1}, x^{t-2}, \cdots$ until it obtains x^0. This is achieved by subtracting the additive noise learned during training from the output of the previous step, ultimately recovering the original signal.

6.2. Algorithm

Algorithm 1 provides a high-level overview of our method. For an explanation of the notation used, refer to Section 3 of the paper.

Note that rescaling factors are applied at each step to match the expected variance of the diffusion model.

Algorithm 1 Smoothed Trajectory Prediction and its Certified Bounds

Input: Input trajectory X, number of Monte-Carlo samples n, number of predictions k, aggregation operator $\mathcal{A}$, trajectory predictor g, denoiser h, certification radius R, hyperparameter σ, lower bounds $\{l_j\}$, upper bounds $\{u_j\}$
Output: Certified trajectory prediction $\tilde{f}(X)$, the certified bounds
procedure
    Initialize an empty list arr to store predictions
    for $i = 1$ to n **do**
        $\epsilon^i \sim \mathcal{N}(0, \sigma^2 I)$ ▷ Acquire a sample from the Gaussian distribution
        $X^i \leftarrow X + \epsilon^i$ ▷ Generate perturbed inputs
        $f(X^i) = g(h(X^i))$ ▷ Process through denoiser h and predictor g
        if $\mathcal{A}$==Mean **then**
            Clamp the j-th coordinate of $f(.)$ within $[l_j, u_j]$ ▷ Adaptive clamping
        end if
        Append $f^k(X^i)$ to arr ▷ Certify k modes
    end for
    $Y \leftarrow \mathcal{A}(arr)$ ▷ Aggregate the predictions with point-wise mean or median
    if $\mathcal{A}$ == Mean **then** ▷ Bounds for mean
        Compute LB and UB on Y from Equation (1), given $R, \{l_j\}, \{u_j\}$
    else ▷ Bounds for median
        Compute LB and UB on Y from Equation (2), given R
    end if
    return Y, LB, UB ▷ Return prediction and certified bounds
end procedure

7. Additional results

7.1. Results using ADE, ABD and Certified-ADE

In Section 4 of the paper, we mainly reported results in terms of FDE, FBD, and Certified-FDE due to space constraints. Here we provide the results in terms of ADE, ABD, and Certified-ADE in Figure 5 and Figure 6.

7.2. More qualitative results

We showed a scenario in the main paper where we showcase the impact of an adversarial attack and imperfect observation on the performance of the predictor. Here, we provide

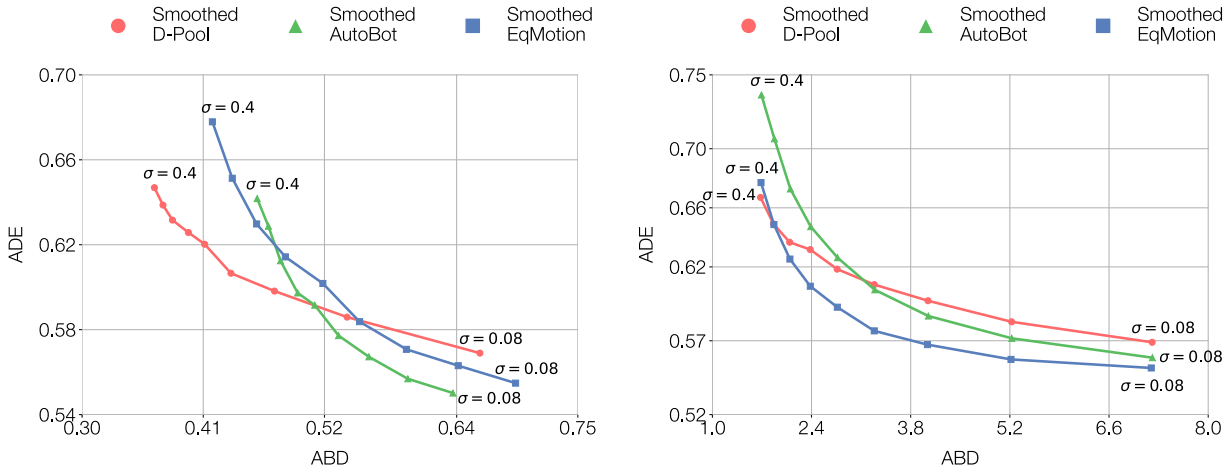

Figure 5. ADE against ABD for median and mean aggregations, respectively. The results are for different smoothed predictors and equally spaced σ within $[0.08, 0.4]$. The bottom left indicates the best performance. The conclusions are similar to the main paper.

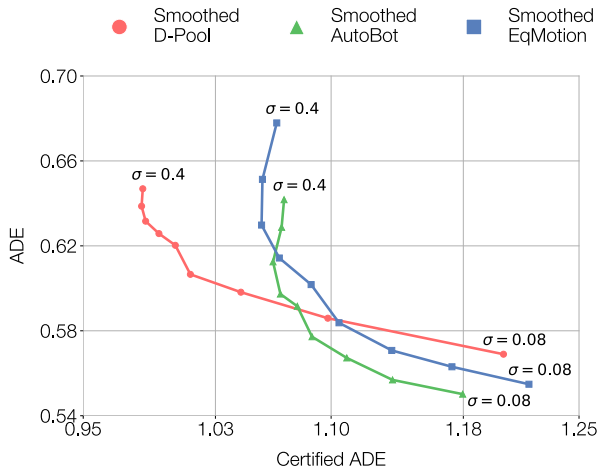

Figure 6. ADE against Certified-ADE. The results are for different smoothed predictors with median aggregation function and equally spaced σ within $[0.08, 0.4]$. The bottom left indicates the best performance. The conclusions are similar to the main paper.

more scenarios in Figure 7. These results demonstrate that the models are vulnerable to different input noises, and certification can provide guaranteed robustness.

In Figure 8, we show qualitative results of EqMotion and smoothed EqMotion. We generate multiple noisy inputs by adding random noise with a magnitude less than 0.1 to an input trajectory and visualize the models' predictions. As evidenced, the original predictor yields highly variable outputs, however, the smoothed predictor predicts within the certified bounds. It is important to note that the certified bounds are functions of the input; consequently, they are larger in some scenarios and smaller in others.

Figure 9 presents qualitative results across varying σ in the smoothed function. It demonstrates the trade-off between accuracy and the bound size. As the sigma value increases, the perturbation overwhelms the original input, resulting in a signal whose median aligns closely with the noise median, which is zero. Therefore, the bounds become tighter, but the accuracy drops. Among the various smoothing functions depicted, the one with $\sigma = 0.16$ appears to maintain a better balance, offering sufficiently tight bounds without significantly compromising accuracy, while the function with $\sigma = 0.32$ demonstrates relatively lower accuracy.

7.3. Certified bound per timestep

Until now, we have reported the final and average certified bounds. However, each prediction timestep has a different bound. Figure 10 shows the bounds for each timestep. As expected, later timesteps have larger bounds, correlating with their potentially greater variations.

7.4. Analyzing the bounds with adversarial attacks

Additionally, we conduct two experiments on a subset of Trajnet++ dataset to further investigate the certified bounds presented in the paper.

We first compare the Certified-FDE of Smoothed EqMotion to those of the original EqMotion model. Remind that Certified-FDE is the guaranteed worst-case FDE happening given input deviations (we use worst-case FDE and Certified-FDE interchangeably in this subsection). However, there is no guarantee for the worst-case FDE of the original model. In order to determine a lower-bound for the worst-case FDE of the original model, we employ an adver-

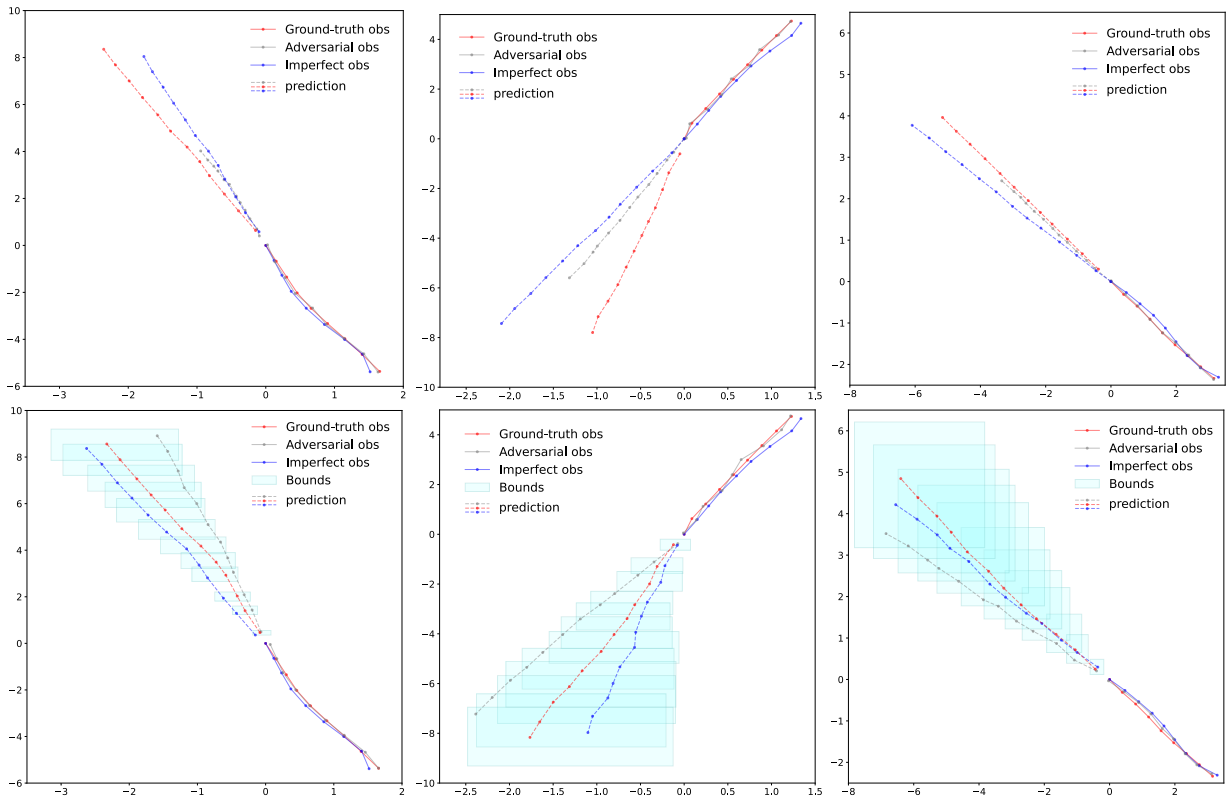

Figure 7. Comparing the performance of the original predictor (on the top) and the smoothed predictor (on the bottom). The red trajectories depict original observations, the blue trajectories represent predictions with imperfect observations coming from detection and tracking algorithms on real-world data, and the gray ones show the predictions given adversaries.

sarial attack. Note that it is a lower-bound since our attack is one potential attack and not necessarily the strongest possible attack, and the worst-case FDE could potentially be higher with other adversarial attack approaches. We employ the PGD attack [41], constrain the L_2 norm of perturbations to 0.1, similar to the value of R in our main experiments, and use a subset of trajnet++ dataset. The objective is to find perturbations that would increase the FDE for the EqMotion model. This attack demonstrated that applying adversarial perturbations could raise the FDE of EqMotion from 1.12 to 1.73. On the other hand, the Certified-FDE for the Smoothed EqMotion is 1.87. This shows that while a lower-bound for the worst-case FDE of the original model is 1.73, the guaranteed worst-case FDE for the Smoothed predictor is 1.87 which is within the same range but guaranteed. This means that any attack to the smoothed predictor, as long as it adheres to the L_2 norm constraint, will result in outputs that fall within the guaranteed worst-case FDE.

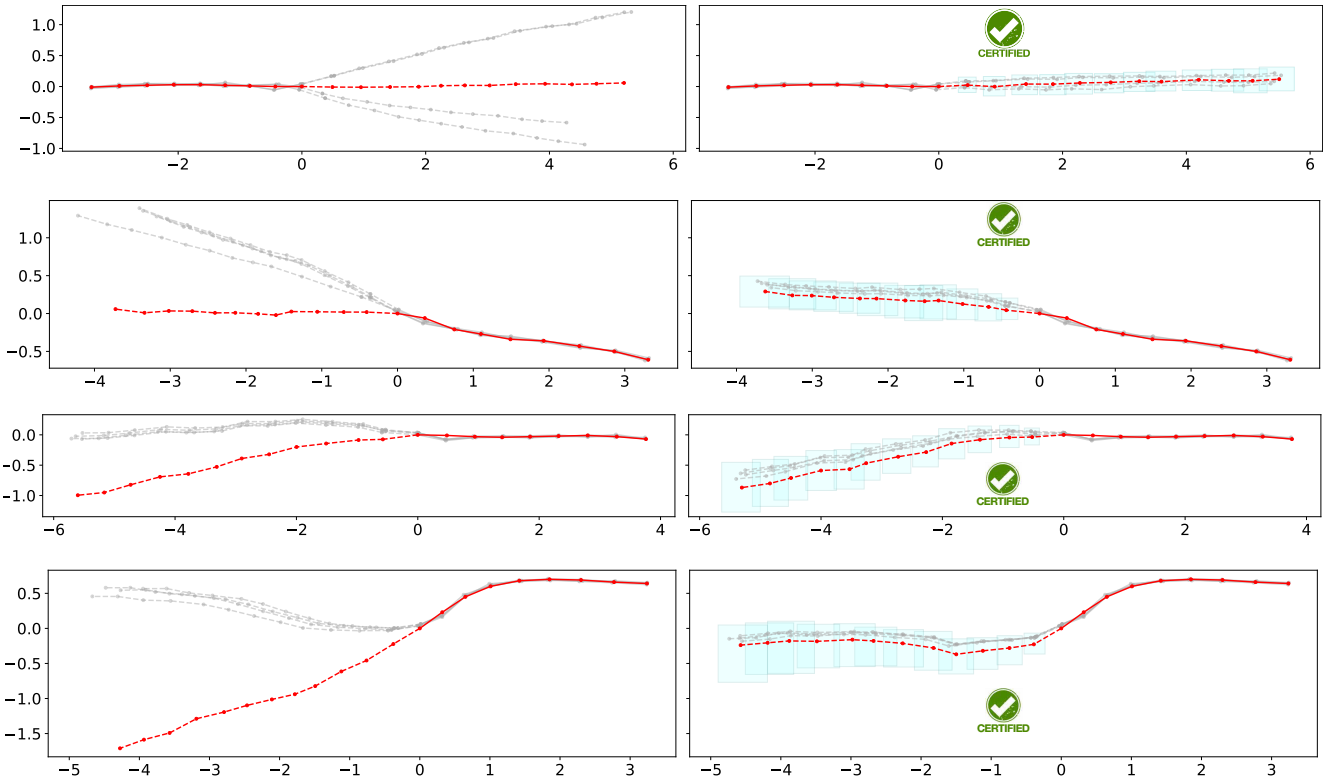

Figure 8. Qualitative results of the original predictor compared with the smoothed predictor. The red trajectories depict clean inputs and the corresponding predictions, and the gray trajectories represent noisy inputs and predictions. The left part showcases the outputs of the original predictor, revealing unbounded predictions. In contrast, the right part demonstrates the outputs of the smoothed predictor, underscoring our ability to certify bounds on predicted outputs.

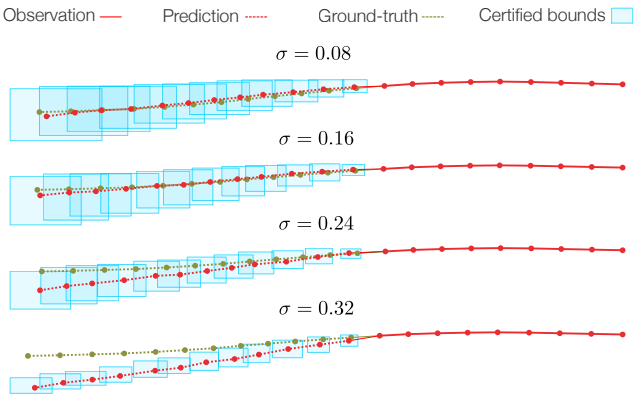

Figure 9. Qualitative results of our model for different values of σ. It shows the outputs of the smoothed EqMotion for one randomly selected data sample in the dataset.The ground-truth predictions are depicted in green, while the observation and the model's predictions are in red. The figure shows that increasing σ tightens the bound at the cost of dropping the accuracy.

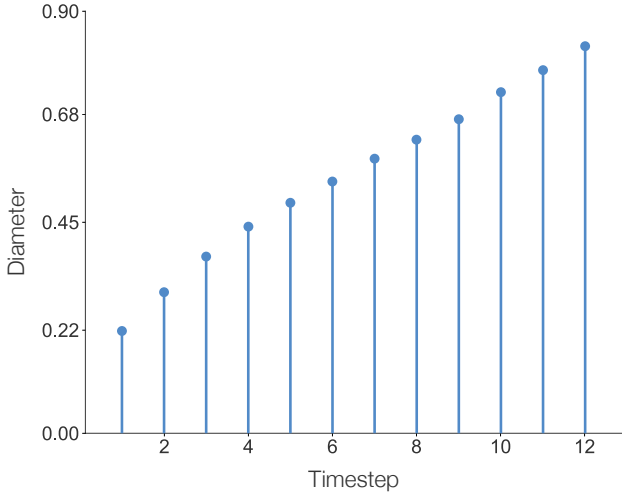

Figure 10. Certified bound per timestep. We report the distance of the farthest point in the certified bound to the predicted trajectory for different timesteps as the bound diameter. Smoothed EqMotion with $\sigma = 0.2$ is employed for this experiment. It shows that later timesteps have higher bounds due to their larger output variation.