

Supplementary Document: Online Model Adaptation with Feedforward Compensation

Anonymous Author(s)

Affiliation

Address

email

1 A Lemma 1: Error Bound of Online Adaptation

2 In this section, we establish the error bound for prediction errors in the general online adaptation.
3 At time step t , we select the critical input-output pairs (X_s, y_{s+1}) from recent L -steps observa-
4 tions. These critical pairs are utilized to update the parameters of the prediction model, resulting in a
5 refined model. Subsequently, predictions are made using the newly optimized parameters.

6 Assuming that the transition function $f : X_t \rightarrow Y_{t+1}$ satisfies the K -Lipschitz continuity condition
7 and the δ time-varying condition.

8 **Bound of ground-truth difference.** Given a transition function $f(t, X)$, if the K -Lipschitz conti-
9 nuity and δ time-varying conditions holds within recent L steps, then the ground-truth value Y_{t+1}
10 and Y_{s+1} has the following property:

$$\|Y_{t+1} - Y_{s+1}\| = \|f(t, X_t) - f(s, X_s)\| \leq K\|X_t - X_s\| + \delta|t - s| \quad (1)$$

11 The proof is shown below:

$$\begin{aligned} \|Y_{t+1} - Y_{s+1}\| &= \|f(t, X_t) - f(s, X_s)\| \\ &= \|f(t, X_t) - f(t, X_s) + f(t, X_s) - f(s, X_s)\| \\ &\leq \|f(t, X_t) - f(t, X_s)\| + \|f(t, X_s) - f(s, X_s)\| \quad (\text{triangle inequality}) \\ &\leq K\|X_t - X_s\| + \|f(t, X_s) - f(s, X_s)\| \quad (K \text{ Lipschitzness}) \\ &\leq K\|X_t - X_s\| + \delta|t - s| \quad (\delta \text{ time varying}) \end{aligned} \quad (2)$$

12 **Error Bound of Online Adaptation.** For time step t , the (prior) prediction error e_{t+1} has the
13 following inequality:

$$\begin{aligned} e_{t+1} &= \|Y_{t+1} - \hat{Y}_{t+1}\| = \|Y_{t+1} - \hat{f}(\theta_t, X_t)\| \\ &= \|Y_{t+1} - Y_{s+1} + Y_{s+1} - \hat{f}(\theta_t, X_s) + \hat{f}(\theta_t, X_s) - \hat{f}(\theta_t, X_t)\| \\ &\leq \|Y_{t+1} - Y_{s+1}\| + \|Y_{s+1} - \hat{f}(\theta_t, X_s)\| + \|\hat{f}(\theta_t, X_s) - \hat{f}(\theta_t, X_t)\| \quad (\text{triangle inequality}) \\ &\leq K\|X_t - X_s\| + \delta|t - s| + \|Y_{s+1} - \hat{f}(\theta_t, X_s)\| + \|\hat{f}(\theta_t, X_s) - \hat{f}(\theta_t, X_t)\| \end{aligned} \quad (3)$$

14 The first two terms come from the difference between ground-truth $Y_{t+1} - Y_{s+1}$, the third term
15 is a (posterior) fitting error for input-output tuple (X_s, Y_{s+1}) , and the last term is the difference
16 between two predictions. Combining the above inequality with the Lipschitz continuity condition
17 for $\hat{f}(\theta_t, X_t)$, we obtain the error bound for general online adaptation is shown below:

$$e_{t+1} \leq (K + \hat{K})\|X_t - X_s\| + \delta|t - s| + \|Y_{s+1} - \hat{f}(\theta_t, X_s)\| \quad (4)$$

18 Then lemma 1 is derived.

19 B Comparison of Error Bound between Feedforward Adaptation and 20 Feedback Adaptation

21 B.1 Lemma 2 (a,b,c): Expected Error Bound

22 Considering the error bound (4), the first two terms are associated with the specific data compensa-
23 tion strategy, while the last term represents the posterior fitting error on the selected samples. In this
24 study, our main focus is on the data compensation strategy, and we do not prioritize the data fitting
25 aspect. Additionally, with a powerful neural network prediction model, achieving a very small fit-
26 ting error (almost zero) is relatively straightforward [1]. Therefore, we can disregard the fitting error
27 when comparing feedforward and feedback adaptation. By neglecting the fitting error, we obtain an
28 approximate upper bound B_e for general online adaptation, as shown below:

$$B_e = (K + \hat{K})\|X_t - X_s\| + \delta|t - s| \quad (5)$$

29 **Error Bound for Feedforward Adaptation.** In feedforward adaptation, the selected input-output
30 pairs are the most similar samples to the current observation $X_s = \arg \min_{X_i} \|X_t - X_i\|$ from
31 L -size buffer, and $s = \arg \min_{i \in [t-L, t-1]} \|X_t - X_i\|$. Then we have an error bound B_e^{ff} for
32 feedforward adaptation:

$$B_e^{ff} = (K + \hat{K})\|X_t - X_s\| + \delta|t - s| \leq (K + \hat{K})\|X_t - X_s\| + \delta L \quad (6)$$

$$X_s = \arg \min_{X_i \in [X_{t-L}, X_{t-1}]} \|X_t - X_i\| \quad (7)$$

33 **Error Bound for Feedback Adaptation.** In feedback adaptation, the selected input-output pairs are
34 the latest observations $X_s = X_{t-1}$ and $s = t - 1$. Then we have an error bound B_e^{fb} for feedback
35 adaptation:

$$B_e^{fb} = (K + \hat{K})\|X_t - X_{t-1}\| + \delta \quad (8)$$

36 **Comparison of the expected error bound between Feedforward and Feedback Adaptation.** Let
37 the expected distance between consecutive samples is D :

$$D := E[\|X_t - X_{t-1}\|]. \quad (9)$$

38 Let the expected minimum sample distance is D^* :

$$D^* := E[\|X_t - X_s\|] = E[\min_{X_i} \|X_t - X_i\|]. \quad (10)$$

39 Then the expected error bound for feedforward adaptation is:

$$E[B_e^{ff}] \leq (K + \hat{K})E[\|X_t - X_s\|] + \delta L = (K + \hat{K})D^* + \delta L. \quad (11)$$

40 The expected error bound for feedback adaptation is:

$$E[B_e^{fb}] = (K + \hat{K})E[\|X_t - X_{t-1}\|] + \delta = (K + \hat{K})D + \delta. \quad (12)$$

41 Consider the conditions that feedforward adaptation has a smaller error bound than feedback adap-
42 tation in expectation. In order to make: $E[B_e^{ff}] < E[B_e^{fb}]$, we have:

$$(K + \hat{K})D^* + \delta L < (K + \hat{K})D + \delta \quad (13)$$

$$\Rightarrow \frac{\delta}{K + \hat{K}} < \frac{D - D^*}{L - 1} \quad (14)$$

43 Equation (14) represents the condition under which feedforward adaptation surpasses feedback
44 adaptation in terms of the expected error bound. Here, the hyperparameter L denotes the prede-
45 fined buffer size. It is important to note that when $L = 1$, feedforward adaptation is equivalent to
46 feedback adaptation. Therefore, our focus is primarily on the case when $L > 1$. From the equation,
47 we observe that if the system exhibits a smaller time-varying property δ compared to the Lipschitz
48 constant K , and a smaller minimum sample distance D^* , feedforward adaptation is more likely
49 to achieve a greater improvement over feedback adaptation. For instance, when $\delta = 0$, we have
50 $E[B_e^{ff}] < E[B_e^{fb}]$ for any K, \hat{K}, D, D^* , and L .

51 By combining (11), (12) and (14), we can conclude Lemma 2 (a,b,c).

52 B.2 Lemma 2 (d): Expected Error Bound on Random-input System

53 Consider a transition function f with randomly sampled input observations. Specifically, input X_t
 54 is a random variable sampled from the uniform distribution: $X_t \sim \mathcal{U}(0, 1)$. In this case, the current
 55 sample X_t and last sample X_{t-1} are independent random variables from $\mathcal{U}(0, 1)$. According to [2],
 56 the expectation of the distance between these two independent and uniform-distributed variables is
 57 $\frac{1}{3}$. Then for feedback adaptation

$$E[\|X_t - X_{t-1}\|] = \frac{1}{3}, \text{ for } X_t, X_{t-1} \sim \mathcal{U}(0, 1) \quad (15)$$

58 The term $E[\min_{X_i} \|X_t - X_i\|]$ represents the expected minimum distance between the current
 59 sample X_t and previous L samples in the buffer, which is $\frac{1}{L+2}$ [2], according to [2]. Then for
 60 feedforward adaptation:

$$E[\|X_t - X_s\|] = E[\min_{X_i \in [X_{t-L}, X_{t-1}]} \|X_t - X_i\|] = \frac{1}{L+2}, \text{ for } X_t, X_i \sim \mathcal{U}(0, 1) \quad (16)$$

61 Let $D = \frac{1}{3}$, $D^* = \frac{1}{L+2}$ on the expected error bound (11) and (12), we obtain the expected error
 62 bound for feedforward and feedback adaptation on the system with random input:

$$E[B_e^{ff}] = (K + \hat{K})D^* + \delta L = \frac{K + \hat{K}}{L+2} + \delta L \quad (17)$$

$$E[B_e^{fb}] = (K + \hat{K})D + \delta = \frac{K + \hat{K}}{3} + \delta \quad (18)$$

63 Consider the conditions that feedforward adaptation has a smaller error bound than feedback adap-
 64 tation in expectation. In order to make: $E[B_e^{ff}] < E[B_e^{fb}]$, we have:

$$(K + \hat{K})D^* + \delta L < (K + \hat{K})D + \delta \quad (19)$$

$$\Rightarrow \frac{K + \hat{K}}{L+2} + \delta L < \frac{K + \hat{K}}{3} + \delta \quad (20)$$

$$\Rightarrow \frac{\delta}{K + \hat{K}} < \frac{1}{3L+6} \quad (21)$$

65 If $L = 1$, the feedforward adaptation is equal to the feedback adaptation. For feedforward adapta-
 66 tion, we have $L > 1$. Then we consider the buffer size $L = 2$ as general settings, then conclude the
 67 conditions for applying feedforward adaptation:

$$\frac{\delta}{K + \hat{K}} < \frac{1}{3L+6} = \frac{1}{12} \approx 0.083 \quad (22)$$

68 In this case, with the optimal buffer size $L = L^* := \sqrt{\frac{K+\hat{K}}{\delta}} - 2$, feedforward adaptation achieves
 69 the smallest expected error bound:

$$E[B_e^{ff}]^* = 2\sqrt{\delta(K + \hat{K})} - 2\delta \quad (23)$$

70 As can be seen, if $\delta \approx 0$, feedforward adaptation could achieve the zero expected error bound with
 71 optimal buffer size L^* , while feedback adaptation cannot converge to zero expected error bound.

72 Thus, given a prediction system f with a random input state, if $\frac{\delta}{K+\hat{K}} < \frac{1}{12}$, with buffer size $L = 2$,
 73 feedforward adaptation achieves the smaller expected error bound than feedback adaptation. In this
 74 case, the optimal buffer size for minimum error bound is $L^* = \sqrt{\frac{K+\hat{K}}{\delta}} - 2$.

75 By combining (17), (18), (22) and (23), one can conclude Lemma 2 (d).

76 B.3 Synthetic Experiments: Linear Time-varying System

77 We design a toy experiment to evaluate Lemma 2. We consider the following linear time-varying
 78 system

$$y_{t+1} = f(x_t) = \sin x_t + \delta t, \quad x_t \sim \mathcal{U}(0, 1)$$

79 Our parameterized prediction model is a one-layer perception with Sigmoid activation function.

$$\hat{y}_t = \hat{f}(V_t, b_t; x_t) = S(V_t x_t) + b_t = \frac{1}{1 + e^{-V_t x_t}} + b_t \quad (24)$$

80 Where $S(\cdot)$ denotes a Sigmoid activation function. We have The Lipschitz constant K and \hat{K} for
81 the ground-truth function f and the one-layer perception \hat{f} :

$$K = \sup \left| \frac{\partial f}{\partial x_t} \right| = \sup |\cos(x_t)| = 1 \quad (25)$$

$$\hat{K} = \sup \left(\left| \frac{\partial \hat{f}}{\partial x_t} \right| \right) = \sup |V_t \cdot S(V_t x_t) \cdot (1 - S(V_t x_t))| = 0.25 \sup |V_t| \quad (26)$$

82 We use SGD as an optimizer in feedback and feedforward adaptation. During training, we keep the
83 $\|V_t\|$ bounded, i.e. $\|V_t\| \leq 1$, then $\hat{K} = 0.25$. We use Lemma 3 (17) and (18) to calculate the error
84 bound for feedback and feedforward adaptation:

$$E[B_e^{fb}] = \frac{5}{12} + \delta \quad (27)$$

$$E[B_e^{ff}] = \frac{5}{4L + 8} + \delta L \quad (28)$$

85 Then we calculate the threshold δ^* (22). If $\delta \leq \delta^*$, feedforward adaptation has a smaller error
86 bound.

$$\frac{\delta^*}{K + \hat{K}} = \frac{1}{12} \quad (29)$$

$$\Rightarrow \delta^* = \frac{1}{12}(K + \hat{K}) \approx 0.1 \quad (30)$$

87 Thus, in the toy experiment, If $\delta \leq 0.1$, feedforward adaptation has a smaller error bound. The
88 experimental results are shown in Figure 1 of the main paper.

89 C Applications of Feedforward Adaptation

90 When determining whether to apply feedforward adaptation to a system or time-series function,
91 Lemma 2(c) can serve as a criterion. However, estimating the values of δ, K, D, D^* for the system
92 is required. As a straightforward and conservative approach, if $\delta \approx 0$, feedforward adaptation
93 outperforms feedback adaptation for any δ, K, D, D^* . To simplify this decision-making process, we
94 propose a simple criterion based on the widely used stationarity test in time-series analysis.

95 C.1 Stationary time series and ADF test

96 A stationary time series is one that exhibits properties that do not depend on time. Therefore, a sta-
97 tionary time series does not possess trends or seasonality. In the context of a time series (X_t, Y_{t+1}) ,
98 stationarity implies that the transition function $f : X_t \rightarrow Y_{t+1}$ is not explicitly linked to the time
99 step t . In accordance with the δ time-varying condition, which is equivalent to $\delta \approx 0$.

100 The Augmented Dickey-Fuller (ADF) test is a widely used method for detecting the stationarity of a
101 time series [3]. It tests the null hypothesis that a time series is non-stationary or time-dependent (i.e.,
102 it has a unit root), while the alternative hypothesis suggests stationarity, indicating that it cannot be
103 represented by a unit root. The ADF test yields a p-value that is used to assess the test. If the p-value
104 is less than 0.05, we reject the null hypothesis and conclude that the series is stationary. Conversely,
105 if the p-value is greater than or equal to 0.05, we fail to reject the null hypothesis and conclude that
106 the series is non-stationary.

107 C.2 Differencing

108 In many real-world scenarios, time series signals exhibit non-stationarity. Therefore, it is crucial to
109 transform these non-stationary signals into stationary ones in order to apply feedforward adaptation

110 effectively. One approach to achieve this is by computing the differences between consecutive ob-
111 servations, denoted as $d_{t+1} = Y_{t+1} - Y_t$. This process is commonly referred to as differencing [4].
112 Differencing helps stabilize the mean of a time series by eliminating changes in its level and remov-
113 ing trends. By applying differencing, it becomes possible to convert many non-stationary series into
114 stationary ones, thereby facilitating the use of feedforward adaptation

115 C.3 Criterion for applying feedforward adaptation

116 The criterion and procedure for applying feedforward adaptation are presented in Figure 2 of the
117 main paper. In this approach, given a time series (X_t, Y_{t+1}) , such as the training set of the prediction
118 task $f : X_t \rightarrow Y_{t+1}$, we follow a specific process based on the stationarity of the series. If the ADF
119 test indicates that the series is stationary, we directly apply feedforward adaptation to the original
120 series. This involves prediction and adaptation on $Y_{t+1} = f(t, X_t)$. If the series is found to be non-
121 stationary, we employ differencing by calculating the difference between consecutive observations,
122 denoted as $d_{t+1} = Y_{t+1} - Y_t$. We then assess the stationarity of the differenced signal d_{t+1} . If it is
123 determined to be stationary, we proceed with feedforward adaptation on the difference series. This
124 entails prediction and adaptation on d_{t+1} , followed by converting it back to $Y_{t+1} = Y_t + d_{t+1}$ based
125 on the value of Y_t . If the differenced signal remains non-stationary even after differencing, we resort
126 to feedback adaptation for handling the non-stationary signal.

127 D Additional Details of Experiments

128 D.1 Dataset

129 We evaluate the effectiveness of the proposed feedforward adaptation method in three different sce-
130 narios: (1) Human motion prediction in human-robot collaboration, using the THOR dataset and As-
131 sembly dataset; (2) Vehicle trajectory prediction in autonomous driving, using the NGSIM dataset;
132 and (3) Robotic arm trajectory prediction for quality control and monitoring purposes, using the
133 Robot arm trajectory dataset. The specific tasks for each dataset are illustrated in Fig. 1.

134 The description of the datasets is shown below.

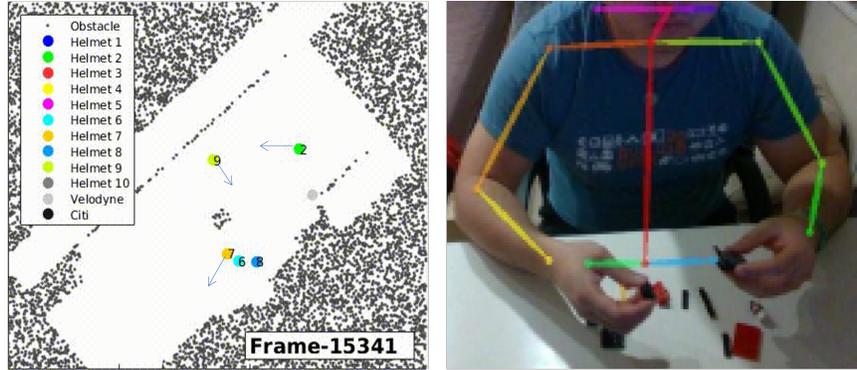
- 135 • **THOR**¹ is a public dataset of human motion trajectories, recorded in a controlled indoor experi-
136 ment [5]. Which includes the motion trajectories with diverse and accurate social human motion
137 data in a shared indoor environment. In our experiments, we use No. 2 ~ 4 agent’s trajectory as
138 a train set and No. 5 ~ 10 agent’s trajectory as a test set.
- 139 • **Assembly** dataset ² records arm motions in assembly tasks. This dataset includes 5 different
140 assembly tasks. Each task requires the human to use LEGO pieces to assemble an object. In our
141 experiments, we use task 1 ~ 2 as a train set and task 3 ~ 5 as a test set.
- 142 • **NGSIM** dataset: US 101 human driving data from Next Generation SIMULATION dataset ³. The
143 dataset contains highway driving trajectories captured by cameras mounted on top of surrounding
144 buildings [6]. In our experiment, we use a subset of the dataset which contains 100 trials of
145 different agents. We use No. 1 ~ 50 trial’s trajectory as a train set and No. 50 ~ 100 trial’s
146 trajectory as a test set.
- 147 • We collect the **Robot arm trajectory** dataset, which records the joint position (De-
148 navit–Hartenberg parameters) of the KINOVA Gen 3 (7 DoF) robotic arm in pick-and-place
149 tasks. This dataset includes 4 pick-and-place tasks for picking objects from different positions
150 on a workbench. In our experiments, we use task 1 ~ 2 as a train set and task 2 ~ 4 as a test set.
151 We will make the dataset publicly available.

152 In our experiments, the prediction model utilizes the most recent 1 second of observations to predict
153 the trajectory for the next 2 seconds. To ensure consistent sampling frequencies, we subsampled the

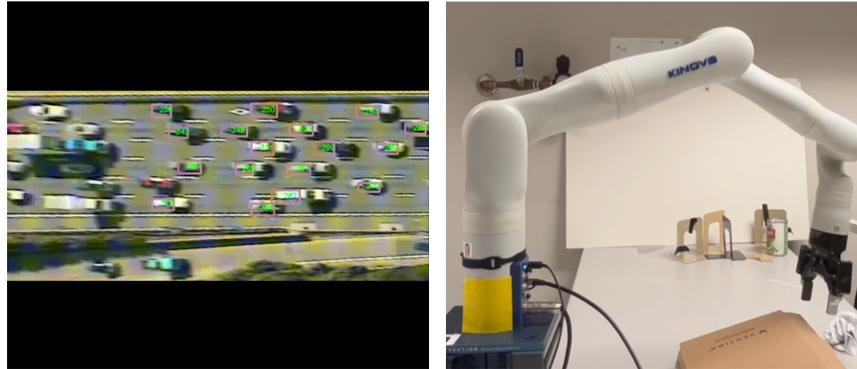
¹<http://thor.oru.se/>

²https://github.com/intelligent-control-lab/Human_Assembly_Data

³<https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm>



(a) THOR human motion prediction dataset (b) Arm motion prediction in assembly tasks



(c) NGSIM vehicle trajectory prediction dataset (d) Robot arm trajectory prediction in pick-and-place tasks

Figure 1: Illustration of tasks in different datasets. Figure (a) is copied from the public website of the THOR dataset <http://thor.oru.se/>; Figure (b) is copied from the website of the Assembly dataset https://github.com/intelligent-control-lab/Human_Assembly_Data; Figure (c) is copied from the public website of the NGSIM dataset <https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Program-I-80-Vide/2577-gpny>; Figure (d) shows a KINOVA robot arm performing pick-and-place tasks in our dataset.

154 THOR and Assembly datasets to 20Hz. For these datasets, we set the input horizon to 20 and the
 155 prediction horizon to 40. The NGSIM dataset has a sampling frequency of 15Hz, so we adjusted the
 156 input horizon to 15 and the prediction horizon to 30 accordingly. As for the Robot arm trajectory
 157 dataset, we subsampled it to a sampling frequency of 25Hz and set the input horizon to 25 and the
 158 prediction horizon to 50.

159 D.2 Stationarity test of datasets

160 As discussed in Appendix C, we use the ADF method to test the stationarity of the time-series data
 161 and check the slow-varying property of its transition function. If the p-value of the ADF test is less
 162 than 0.05, we can reject the null hypothesis and conclude that the time series is stationary. If the
 163 p-value of the ADF test is greater than 0.05, we cannot reject the null hypothesis and conclude that
 164 the time series is non-stationary.

165 The results of the ADF test are shown Table 1. It can be observed that the original raw series
 166 of the THOR and Assembly datasets exhibit stationarity, indicating that the transition function of
 167 these datasets is slow time-varying. Therefore, feedforward adaptation can be directly applied to the
 168 THOR and Assembly datasets. On the other hand, the original raw series of the NGSIM and Robot
 169 arm datasets are non-stationary, but the difference series demonstrates stationarity. This implies that

Table 1: ADF test results for raw time-series and the difference signal on Thor, Assembly, NGSIM, Robot arm datasets.

Dataset	THOR	Assembly	NGSIM	Robot arm
P value on Raw Series	5e-3 (stationary)	4e-3 (stationary)	0.34 (nonstationary)	0.09 (nonstationary)
P value on Difference	1e-20	7e-21	0 (stationary)	0.008 (stationary)

170 the transition function of the difference signal for the NGSIM and Robot arm datasets is slow time-
 171 varying. Consequently, feedforward adaptation can be applied to the difference series in NGSIM
 172 and Robot arm datasets, which is equivalent to predicting velocity instead of the raw trajectory.

173 D.3 Experimental design

174 **Parameterized Prediction models.** We utilize a Multi-layer Perceptron (MLP) with a direct mul-
 175 tistep (DMS) prediction strategy [7]. The choice of MLP with DMS is motivated by the superior
 176 performance of a simple MLP over many larger Transformer-based models, as reported in [7]. Our
 177 MLP architecture consists of two layers. The first layer can be considered as an Encoder, denoted
 178 as $X_t = W \cdot X_t$. Following the encoder, the MLP incorporates layer normalization, an activation
 179 function, and a final linear projection represented as $Y_{t+1} = V \cdot \text{Relu}(\text{LayerNorm}(X_t))$. The layer
 180 normalization and the final projection can be viewed as a decoder. It is worth noting that we do not
 181 flatten the input for the MLP. The expression $X_t = W \cdot X_t$ represents a linear layer applied along
 182 the temporal axis.

183 **Baselines.** We compare the proposed method with four baselines.

- 184 • **w/o adapt** directly conduct prediction without adaptation. Which is a lower bound for adaptation
 185 methods.
- 186 • **Feedback adaptation** selected the latest sample to optimize the model [8]. Which is the most
 187 important baseline for us.
- 188 • **Random adaptation** is the same as the Experience Replay with Reservoir Sampling [9]. Which
 189 is a method that selects the critical pair from the L -size buffer with random sampling.
- 190 • **Full adaptation** is a method that uses all samples from the buffer to adapt the model, which is
 191 similar to offline training.

192 **Hyperparameters.** For offline training, we follow the strategy in [7]. In adaptation, we set the
 193 learning rate of SGD as $\eta = 0.001$. Buffer size for feedforward adaptation is $L = 1000$. For
 194 uncertainty estimation, we set $\tilde{\delta} = 0, \tilde{K} = 1$.

195 D.4 Prediction output and Prediction Error

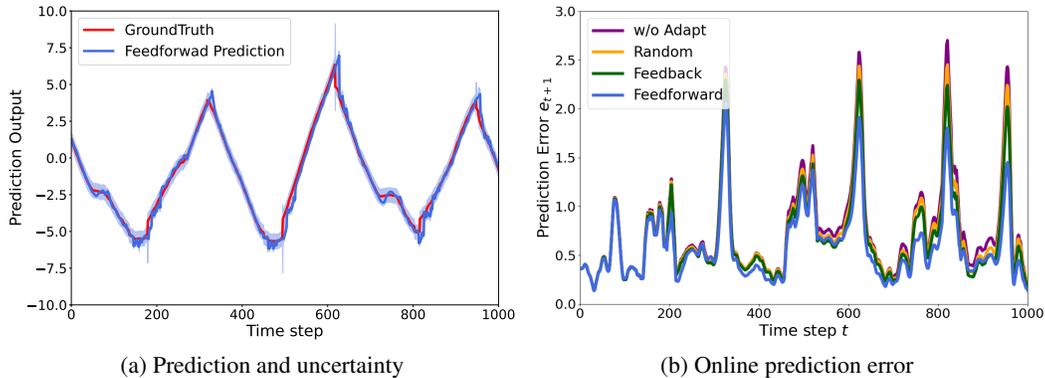


Figure 2: Experimental results on THOR dataset.

196 Figure 2a shows the prediction output (blue curve), ground truth label (red curve), and uncertainty
 197 estimation (blue dashed region) on the THOR dataset. Figure 2b shows the real prediction error

198 for different adaptation methods over time. Notably, feedforward adaptation exhibits the lowest
 199 prediction error among them.

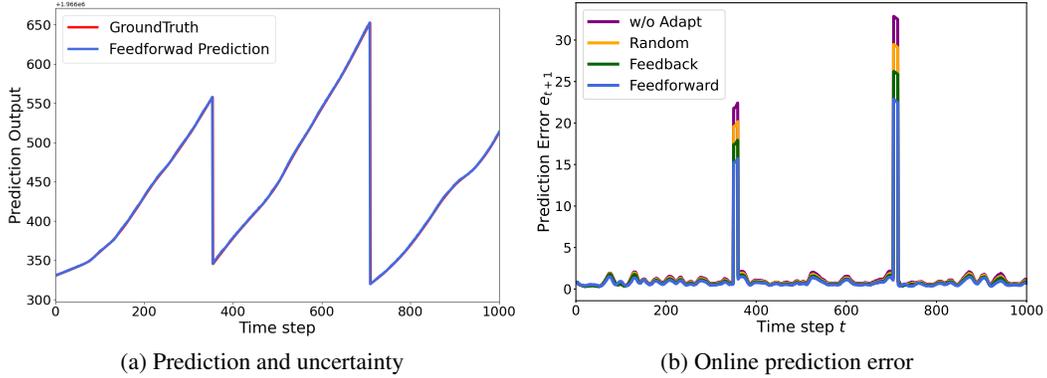


Figure 3: Experimental results on NGSIM dataset.

200 Figure 3a shows the prediction output (blue curve), ground truth label (red curve), and uncertainty
 201 estimation (blue dashed region) on the NGSIM dataset. Figure 3b shows the real prediction error
 202 for different adaptation methods over time. Notably, feedforward adaptation exhibits the lowest
 203 prediction error among them.

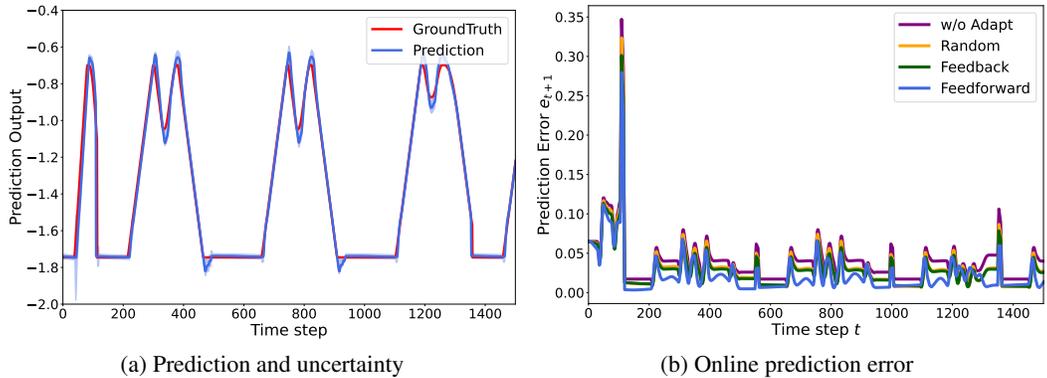


Figure 4: Experimental results on Robot arm dataset.

204 Figure 4a shows the prediction output (blue curve), ground truth label (red curve), and uncertainty
 205 estimation (blue dashed region) on the Robot arm dataset. Figure 4b shows the real prediction error
 206 for different adaptation methods over time. Notably, feedforward adaptation exhibits the lowest
 207 prediction error among them.

208 **D.5 Study of the sample selection strategy of different adaptation methods**

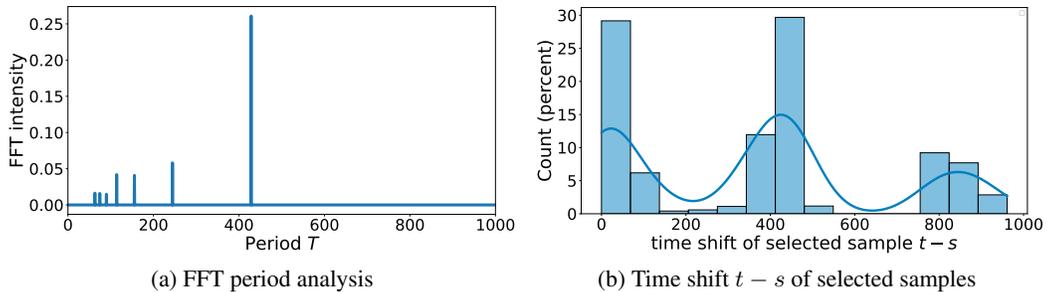


Figure 5: Experimental results on Robot arm dataset. (a) FFT period analysis. (b) Timeshift $t - s$ between current sample X_t and selected sample X_s in feedforward adaptation.

209 Feedforward adaptation selects samples with the smallest sample difference $\min_{X_i} |X_t - X_i|$. This
210 selection strategy allows feedforward adaptation to inherently capture the periodicity in time-series
211 data when faced with periodic patterns. In the case of the robot arm dataset, as depicted in Figure
212 4a, we observe an approximate periodicity of $T \approx 420$. This is evident from the FFT (Fast Fourier
213 Transform) period analysis depicted in Figure 5a. In Figure 5b, we demonstrate how many samples
214 were chosen from $(t - s) \approx 420$ steps earlier during the feedforward compensation process, aligning
215 with the repetition period of $T \approx 420$. Feedforward adaptation’s selection of the most similar
216 samples to the current sample facilitates the extraction of hidden periodic patterns within the input
217 signal over time. Consequently, the distribution of $t - s$ exhibits similarity to the FFT period analysis.

218 References

- 219 [1] Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-
220 parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR,
221 2019.
- 222 [2] R. Pyke. Spacings. *Journal of the Royal Statistical Society: Series B (Methodological)*, 27(3):
223 395–436, 1965.
- 224 [3] D. A. Dickey and W. A. Fuller. Distribution of the estimators for autoregressive time series with
225 a unit root. *Journal of the American statistical association*, 74(366a):427–431, 1979.
- 226 [4] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.
- 227 [5] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilien-
228 thal. Thör: Human-robot navigation data collection and accurate motion trajectories dataset.
229 *IEEE Robotics and Automation Letters*, 5(2):676–682, 2020.
- 230 [6] J. Colyar and J. Halkias. Us highway 101 dataset. *Federal Highway Administration (FHWA)*,
231 *Tech. Rep. FHWA-HRT-07-030*, 2007.
- 232 [7] A. Zeng, M. Chen, L. Zhang, and Q. Xu. Are transformers effective for time series forecasting?
233 *arXiv preprint arXiv:2205.13504*, 2022.
- 234 [8] A. Abuduweili and C. Liu. Robust nonlinear adaptation algorithms for multitask prediction
235 networks. *International Journal of Adaptive Control and Signal Processing*, 35(3):314–341,
236 2021.
- 237 [9] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ran-
238 zato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.