Using Time-Series Privileged Information for Provably Efficient Learning of Prediction Models

Rickard K.A. Karlsson[*]
Delft University of Technology

Martin Willbo[*]
Research Institute of Sweden

Zeshan Hussain
Massachusetts Institute of Technology

Rahul G. Krishnan
University of Toronto

David Sontag
Massachusetts Institute of Technology

Fredrik D. Johansson
Chalmers University of Technology

[*] equal contribution

Abstract

We study prediction of future outcomes with supervised models that use privileged information during learning. The privileged information comprises samples of time series observed between the baseline time of prediction and the future outcome; this information is only available at training time which differs from the traditional supervised learning. Our question is when using this privileged data leads to more sample-efficient learning of models that use only baseline data for predictions at test time. We give an algorithm for this setting and prove that when the time series are drawn from a non-stationary Gaussian-linear dynamical system of fixed horizon, learning with privileged information is more efficient than learning without it. On synthetic data, we test the limits of our algorithm and theory, both when our assumptions hold and when they are violated. On three diverse real-world datasets, we show that our approach is generally preferable to classical learning, particularly when data is scarce. Finally, we relate our estimator to a distillation approach both theoretically and empirically.

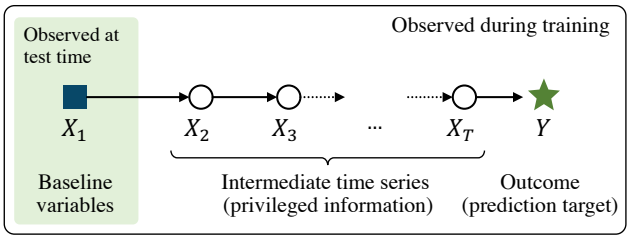


Figure 1: Prediction with intermediate time-series privileged information. The goal at test time is to predict Y based only on X_1 but the learner has access to samples from the full series $X_1, X_2, ..., X_T, Y$ during training.

1 INTRODUCTION

Prediction of future outcomes is a central learning problem in many domains. For example, accurate prediction of the progression of chronic disease allows for identification of patients at higher risk and may be used to trigger interventions. Standard supervised learning algorithms for this task minimize the empirical risk in predicting the outcome using features collected at a baseline time point. When data is scarce, variance can plague this approach and reduce its potential impact. However, in practice, data is often collected not only at the time for prediction and the time of the outcome, but at multiple time points between them; in healthcare, disease markers, lab values and treatments of patients are recorded at regular intervals. These data that could be used for more efficient learning.

Making use of variables for learning that are unavailable at test time has been called *learning using privileged information* (LuPI) (Vapnik and Vashist, 2009) or *learning with side information* (Jonschkowski et al., 2015). A general way to utilize privileged information

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

is via distillation (Lopez-Paz et al., 2016; Hayashi et al., 2019), where a student model is trained to minimize its error in predicting both true labels and soft targets generated by a teacher trained on the privileged information. While these paradigms have shown promise both theoretically (Vapnik and Vashist, 2009) and empirically (Hayashi et al., 2019; Tang et al., 2019), performance guarantees for practical algorithms remain elusive (Serra-Toro et al., 2014).

In particular, it remains unclear *when* learning using privileged information is preferable to learning without it—as discussed by Jonschkowski et al. (2015), incorporating privileged information in learning can harm more than it helps. This work studies a special case in which the privileged information constitutes the intermediate part of a time series starting with baseline features and ending with the target outcome, see Figure 1.

Contributions. We study a strategy that uses privileged information to learn the dynamics of the full time series and makes test-time predictions from baseline by recursively simulating the dynamics and the outcome. Instantiating this idea in Gaussian-linear dynamical systems, we compare it to the best unbiased estimator that uses only baseline data—ordinary least squares regression. In this (well-specified) setting, we prove using a Rao-Blackwell argument (Rao, 1945; Blackwell, 1947) that our recursive strategy is *always* more sample efficient, without bias and with lower variance, compared to learning without privileged information. Additionally, we show that combining this strategy with distillation learning leads to a principled way of trading off bias and variance in the misspecified case.

We study the limits of our theory in synthetic experiments, where our assumptions hold and where they do not. We find that the gap identified by our theory, between our method and the best baseline, grows when more time steps are available and assumptions hold and that it decreases when assumptions are violated (bias is non-zero). Finally, we apply the idea to three diverse real-world problems, where the underlying data-generating process is unknown, showing multiple cases where the approach improves over both non-LuPI and LuPI baselines, and cases where performance is worse.

2 PROBLEM SETTING

We learn models that use baseline variables $X_1 \in \mathbb{R}^d$ to predict outcomes $Y \in \mathbb{R}$, see Figure 1. For a given loss function, $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, our goal is to find a function $h \in \mathcal{H} \subseteq \{h : \mathbb{R}^d \to \mathbb{R}\}$ *of only the baseline variables* X_1, which minimizes the expected risk over the variables with respect to a distribution p,

$$R(h) \coloneqq \mathbb{E}_{X_1,Y}[\mathcal{L}(h(X_1), Y)] \ .$$

This goal is shared with classical supervised learning. However, in our setting, the learner has access to *privileged information* in the form of time series sampled from states $X_2, ..., X_T$, observed chronologically after X_1 and before Y. The information is *privileged* as it is unavailable when the model is used. Time series $x_{i,1}, ..., x_{i,T}, y_i$, indexed by $i = 1, ..., m$, are observed as independent random samples from an unknown distribution $p(X_1, ..., X_T, Y)$. For simplicity,[1] we assume that $X_t \in \mathbb{R}^d$ and define the data matrices $\mathbf{X}_t = [x_{1,t}, ..., x_{m,t}]^\top$, for $t = 1, ..., T$, and $\mathbf{Y} = [y_1, ..., y_m]^\top$, where rows represent different series. We let $D = (\mathbf{X}_1, ..., \mathbf{X}_T, \mathbf{Y})$ denote the full dataset.

Without additional assumptions, learning using privileged information (LuPI) need not lead to smaller risk (Vapnik and Vashist, 2009). Here, we set out to identify conditions on the distribution p under which there is a LuPI algorithm which is provably better than any algorithm learning only from samples of (X_1, Y). Throughout, unless stated otherwise, we assume that the full time series $X_1, X_2, ..., X_T, Y$ is Markov.

Assumption 1 (Markov time series)**.** *For all time points* $t \in \{3, ..., T\}$*,*

$$X_t \perp\!\!\!\perp X_1, ..., X_{t-2} \mid X_{t-1} \quad Y \perp\!\!\!\perp X_1, ..., X_{T-1} \mid X_T \ .$$

Under Assumption 1, X_1 is predictive of Y *only through* the privileged information.

3 LEARNING USING PRIVILEGED TIME SERIES IN LINEAR DYNAMICAL SYSTEMS

A natural strategy for predicting Y in a Markov system is to successively predict X_2 from X_1, then X_3 from the prediction $\hat{X}_2$, and so on. In time-series modeling, this is referred to as *recursive* prediction, in contrast to *direct* prediction (Chevillon, 2007). Unlike time-series modeling, we study prediction of outcomes Y, which are at a fixed time T and distinct in nature from X. A survey of related work is found in Section 5.

We use the recursive strategy in a linear estimator which learns using privileged time series (LuPTS, Algorithm 1). The prediction at each step t is made using a learned linear model $\hat{X}_{t+1} = \hat{A}_t^\top X_t$, with $\hat{A}_t \in \mathbb{R}^{d \times d}$. The final prediction is given by another linear model, $\hat{Y} = \hat{\beta}^\top X_T$, with $\hat{\beta} \in \mathbb{R}^d$, learned from samples of (X_T, Y). At test time, only X_1 is observed, and the models are combined to form $\hat{Y} = (\hat{A}_1 \dots \hat{A}_{T-1}\hat{\beta})^\top X_1$. Algorithm 1 has a flag to indicate whether the transitions are assumed to be stationary. We begin by analyzing the non-stationary case.

[1] That $X_1, ..., X_T$ have the same dimension is not necessary for our main result but simplifies exposition.

Algorithm 1: Learning using privileged time series (LuPTS) in linear dynamical systems

Flag: Stationarity (True / False)
Data: $\{(x_{i,1}, ..., x_{i,T}, y_i)\}_{i=1}^m \sim p^m(X_1, ..., X_T, Y)$

if *Stationarity* **then**
$\tilde{A} = \arg\min_A \sum_{t=1}^{T-1} \sum_{i=1}^{m} \frac{\|A^\top x_{i,t} - x_{i,t+1}\|_2^2}{m(T-1)}$
$\hat{A} = \tilde{A}^{T-1}$
else
for $t = 1, ..., T-1$ **do**
$\hat{A}_t = \arg\min_A \sum_{i=1}^{m} \frac{\|A^\top x_{i,t} - x_{i,t+1}\|_2^2}{m}$
$\hat{A} = \hat{A}_1 \hat{A}_2 \cdots \hat{A}_{T-1}$
$\hat{\beta} = \arg\min_\beta \frac{1}{m} \sum_{i=1}^m (\beta^\top x_{i,T} - y_i)^2$

return $\hat{\theta} = \hat{A}\hat{\beta}$

We study Algorithm 1 in discrete-time Gaussian-linear dynamical systems, where the time series $X_1, ..., X_T$ and the outcome Y evolve according to noisy linear Markov dynamics, as follows.

Assumption 2 (Gaussian-linear system)**.** *The privileged information and outcome evolve as*

$$\begin{aligned} X_t &= A_{t-1}^\top X_{t-1} + \epsilon_t \quad \text{for } t = 2, ..., T \\ Y &= \beta^\top X_T + \epsilon_Y \end{aligned} \tag{1}$$

where A_t are a set of transition matrices that determine the behavior (and stability) of the system. Noise terms are assumed to be zero-mean Normal random variables, $\epsilon_t \sim \mathcal{N}(0, \Sigma)$, for t=2, ..., T, and $\epsilon_Y \sim \mathcal{N}(0, \sigma_Y^2)$. We make no assumption on the distribution of X_1.

Due to the linearity of the transitions and outcome, Y is a linear function of the variable X_t for any t. Most importantly, it is easy to show that Y is also a Gaussian-linear function of X_1,

$$Y = \theta^\top X_1 + \tilde{\epsilon} \quad \text{with} \quad \theta = A_1 \cdots A_{T-1}\beta$$

and

$$\tilde{\epsilon} = \beta^\top \left(\sum_{t=2}^{T-1} \left[\prod_{t'=t}^{T-1} A_{t'} \right]^\top \epsilon_t + \epsilon_T \right) + \epsilon_Y \ .$$

Our goal is now to learn estimates of θ as efficiently as possible, i.e., with the smallest error and/or risk for a given number of samples. It is well-known that the OLS estimator $\hat{\theta}_{\text{OLS}} := (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \mathbf{X}_1^\top \mathbf{Y}$ is the minimum-variance mean-unbiased estimator that is based only on samples of (X_1, Y), see e.g., Johnson et al. (2002, Chapter 7); This makes $\hat{\theta}_{\text{OLS}}$ the strongest possible baseline among estimators that do not make use of privileged information.

3.1 Variance Reduction Through Rao-Blackwellization

Next, we show that, in the Gaussian-linear setting of Assumption 2, with the additional assumption of isotropic noise in transitions, the output of the LuPTS algorithm is a Rao-Blackwell estimator (Rao, 1945; Blackwell, 1947) of θ, which has improved statistical properties over $\hat{\theta}_{\text{OLS}}$. However, to show this, we first prove the following lemma.

Lemma 1. *Let $\hat{K} = (\hat{A}_1, \ldots, \hat{A}_{T-1}, \hat{\beta})$ be the parameters learned by Algorithm 1 without stationarity, and let $(\mathbf{X}_1, \ldots, \mathbf{X}_T, \mathbf{Y})$ be a random dataset from the Gaussian-linear system defined in Assumption 2, with isotropic noise, $\forall t : \epsilon_t \sim \mathcal{N}(0, \sigma_t^2 I)$. Then, for any $t = 2, \ldots, T$ we have that*

$$\mathbb{E}[\mathbf{X}_t | \mathbf{X}_{t-1}, \hat{K}] = \mathbf{X}_{t-1}\hat{A}_{t-1} \quad \mathbb{E}[\mathbf{Y} | \mathbf{X}_T, \hat{K}] = \mathbf{X}_T \hat{\beta} \ .$$

A proof is given in the Appendix. The isotropic noise assumption simplifies the analysis, although it is feasible to prove this lemma in the anisotropic case as well. We give a brief remark in the Appendix highlighting how the analysis differs.

Remark 1. Lemma 1 says that the expected state at t, across datasets of the same size for which Algorithm 1 returns $\hat{A}\hat{\beta}$, is equal to the estimated state at t given the previous state at $t-1$. This is a result of the fact that the same OLS estimator would be obtained if we had samples that were mirrored along the estimated plane of best fit, and that such a dataset is equally likely to occur. Lemma 1 can be used to prove a second lemma, which is found in the Appendix, stating that the output of the algorithm is indeed a Rao-Blackwell estimator, i.e. $\mathbb{E}[\hat{\theta} | \hat{A}, \hat{\beta}] = \hat{A}\hat{\beta}$. With these two lemmas, our main results in Theorem 1 follow using mostly standard arguments (Rao, 1945; Blackwell, 1947).

We evaluate estimates $\hat{\theta}$ using the mean squared error (MSE) w.r.t. θ, and the prediction risk $\overline{R}(\hat{\theta})$, as defined below, where expectations are taken over the randomness in $\hat{\theta}$, determined by the dataset D,

$$\begin{aligned} \text{MSE}(\hat{\theta}) &:= \mathbb{E}_D[\|\hat{\theta} - \theta\|_2^2] \ , \\ \overline{R}(\hat{\theta}) &:= \mathbb{E}_D[\mathbb{E}_{X_1,Y}[(\hat{\theta}^\top X_1 - Y)^2]] \ . \end{aligned}$$

We can now state the following result, relating $\hat{\theta}_{\text{OLS}}$ and $\hat{\theta}_{\text{LuPTS}}$, the output of Algorithm 1.

Theorem 1. *Let $D = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_T, \mathbf{Y})$ be a random dataset with $\hat{\theta}_{\text{OLS}} := (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \mathbf{X}_1^\top \mathbf{Y}$, and let $\hat{\theta}_{\text{LuPTS}} = \hat{A}\hat{\beta}$ be the output of Algorithm 1 without stationarity. Under the Gaussian-linear system of Assumption 2 with isotropic noise as in Lemma 1, $\hat{\theta}_{\text{LuPTS}}$ is unbiased, and*

$$\text{MSE}(\hat{\theta}_{\text{LuPTS}}) = \text{MSE}(\hat{\theta}_{\text{OLS}}) - \mathbb{E}_D[\text{Var}(\hat{\theta}_{\text{OLS}} \mid \hat{\theta}_{\text{LuPTS}})]$$

where Var *is the sum of element-wise conditional variances and the expectation is taken over datasets D, since both estimators are functions of them. Further,*

$$\overline{R}(\hat{\theta}_{\text{LuPTS}}) = \overline{R}(\hat{\theta}_{\text{OLS}}) - \mathop{\mathbb{E}}_{D,X_1}[\mathop{\text{Var}}_{\hat{\theta}_{\text{OLS}}}(\hat{\theta}_{\text{OLS}}X_1 \mid \hat{\theta}_{\text{LuPTS}})] \ .$$

Since the variances are non-negative, $\hat{\theta}_{\text{LuPTS}}$ is at least as good as $\hat{\theta}_{\text{OLS}}$ in both metrics.

Theorem 1 states that, in the Gaussian-linear case, the LuPTS estimator is never worse on average across same-size datasets than the best unbiased estimator learning only from (X_1, Y), irrespective of the distribution of X_1. In other words, *privileged information is provably useful in this case*. If there is significant uncertainty about $\hat{\theta}_{\text{OLS}}$ after $\hat{\theta}_{\text{LuPTS}}$ is determined, $\text{Var}(\hat{\theta}_{\text{OLS}} \mid \hat{\theta}_{\text{LuPTS}})$ is high and LuPTS is favored more strongly.

As a byproduct of the proof of Theorem 1, we further have for the gap in MSE, G,

$$G \coloneqq \mathbb{E}_D[\text{Var}(\hat{\theta}_{\text{OLS}} \mid \hat{\theta}_{\text{LuPTS}})] = \mathbb{E}_D[\|\hat{\theta}_{\text{OLS}} - \hat{\theta}_{\text{LuPTS}}\|^2].$$

We can express this gap more explicitly when $T = 2$.

Corollary 1. *Under Assumption 2, for $T = 2$, with $H_1 = (\mathbf{X}_1^\top \mathbf{X}_1)^{-1}\mathbf{X}_1^\top$, $H_2 = (\mathbf{X}_2^\top \mathbf{X}_2)^{-1}\mathbf{X}_2^\top$, both functions of the dataset D, it holds for the MSE gap G,*

$$G = \mathbb{E}_D[\|(AH_2 - H_1 + H_1\epsilon_2 H_2)\epsilon_Y\|^2] \ .$$

Whenever $\epsilon_Y = 0$, $G = 0$ irrespective of other factors. If $\epsilon_2 = 0$ and A is invertible, $G = 0$ as well ($\epsilon_2 = 0 \implies X_2 = X_1 A$ and $AH_2 = H_1$). In other words, in the case where either the dynamics or the outcome are noiseless, the LuPTS estimator reduces to the OLS estimator. More importantly, if neither noise term is 0, the difference will not be 0 in general. As a consequence, $\text{Var}(\hat{\theta}_{\text{OLS}} \mid \hat{\theta}_{\text{LuPTS}}) > 0$, and LuPTS is strictly preferable over OLS on average. In Section 4, we confirm empirically that LuPTS is more efficient and examine the gap as a function of problem parameters.

Bias & Variance. When the models of both system dynamics and the outcome are well-specified, the gains from the LuPTS estimator come from variance reduction, since both the OLS and LuPTS are unbiased and the irreducible risk due to noise is shared between them. However, in misspecified settings, θ_{LuPTS} may be biased even when θ_{OLS} is not. For example, let $Y = \sqrt{X_2} + \epsilon$ and $X_2 = (X_1)^2$ for X_1 with support on the positive real line. The Markov condition holds, OLS is unbiased, but LuPTS is not. In the misspecified case, the benefits of LuPTS come down to a tradeoff between bias and variance. This is explored in Section 4.3.

Learning From Stationary Systems. When the stationarity flag is false in Algorithm 1, the estimator treats transitions at different time points t, t' as independent mechanisms. Then, while the privileged information provides additional samples with increasing T, the number of functions to estimate increases with T as well. When we apply Algorithm 1 with the stationarity flag set to true, we exploit the assumption that we observe $m \times (T-1)$ (dependent) samples of the same linear transformation. This dependency is the primary reason for why Theorem 1 does not readily extend to the stationary case. However, we observe improvements over baseline and the non-stationary LuPTS model for real-world experiments in Section 4.

3.2 Relation To Distillation Approaches

Generalized distillation (Lopez-Paz et al., 2016) is a technique for learning using privileged information, utilized by Hayashi et al. (2019) in the context of privileged time series. Distillation methods train a student model to minimize its prediction error on both true labels and soft targets provided by a teacher model trained on the privileged data, in the hope to increase sample efficiency by transferring knowledge from teacher to student. However, to the best of our knowledge there are no results proving gains from distillation of privileged information which apply in our setting.

In the linear setting with squared loss, the distillation loss function is defined as

$$\min_{\theta} \ \lambda||\mathbf{Y} - \mathbf{X}_1\theta||_2^2 + (1-\lambda)||\hat{\mathbf{Y}}_{soft} - \mathbf{X}_1\theta||_2^2 \quad (2)$$

where $\lambda \in [0,1]$ and $\hat{\mathbf{Y}}_{soft}$ comprises predictions made by a teacher model. We will now consider the special case of distillation where the LuPTS estimator is used as teacher model, i.e., $\hat{\mathbf{Y}}_{soft} = \mathbf{X}_1\hat{\theta}_{\text{LuPTS}}$. In this case, we can present the following theorem.

Theorem 2. *Let $\hat{\theta}_{\text{LuPTS}}$ be the output of Algorithm 1 and $\hat{\theta}_{\text{OLS}} = (\mathbf{X}_1^\top \mathbf{X}_1)^{-1}\mathbf{X}_1^\top \mathbf{Y}$. For $\hat{\theta}_{Dist}$, the solution to (2) with $\hat{\mathbf{Y}}_{soft} = \mathbf{X}_1\hat{\theta}_{\text{LuPTS}}$ and $\lambda \in [0,1]$, it holds that*

$$\hat{\theta}_{Dist} = \lambda\hat{\theta}_{\text{OLS}} + (1-\lambda)\hat{\theta}_{\text{LuPTS}} \ . \quad (3)$$

Additionally, under Assumption 2, it holds that

$$\text{MSE}(\hat{\theta}_{\text{LuPTS}}) \leq \text{MSE}(\hat{\theta}_{Dist}) \leq \text{MSE}(\hat{\theta}_{\text{OLS}}) \ .$$

A proof can be found in the Appendix. Theorem 2 states that using distillation with a linear student model and LuPTS as teacher leads to an estimate $\hat{\theta}_{Dist}$ that is a convex combination of $\hat{\theta}_{\text{OLS}}$ and $\hat{\theta}_{\text{LuPTS}}$. In the well-specified case (under Assumption 2), since both $\hat{\theta}_{\text{OLS}}$ and $\hat{\theta}_{\text{LuPTS}}$ are unbiased, $\hat{\theta}_{Dist}$ is unbiased as well, and the MSE of $\hat{\theta}_{Dist}$ is bounded between the MSE

of $\hat{\theta}_{\text{OLS}}$ and $\hat{\theta}_{\text{LuPTS}}$. Interestingly, in the misspecified case, eq. (3) shows that using distillation leads to a principled way of trading off bias and variance. For an optimal choice of λ, $\hat{\theta}_{Dist}$ is never worse than using either $\hat{\theta}_{\text{OLS}}$ or $\hat{\theta}_{\text{LuPTS}}$, and may improve on both.

In Section 4.3, we implement two variants of the distillation approach: Distill-Seq, where $\hat{\mathbf{Y}}_{soft}$ comprises predictions made by LuPTS, and Distill-Concat, where $\hat{\mathbf{Y}}_{soft}$ are the predictions made by a traditional linear model trained on a concatenation of the privileged time points, akin to the teacher in Hayashi et al. (2019) (Theorem 2 does not hold for the latter).

4 EXPERIMENTS

We evaluate properties of the LuPTS estimator in a series of experiments[2]. First, in a synthetic setting, we verify our theoretical findings under the assumptions stated in Section 3. An example of what happens when the Markov assumption is violated is also shown. Second, on three real-world datasets, the $\text{PM}_{2.5}$ pollution dataset (Section 4.3) and two clinical datasets (Section 4.4), we study the gain in predictive performance with the LuPTS estimator compared to the baseline OLS estimator. Our results on real-world data demonstrate the bias-variance tradeoff implied by our approach as well as its utility in improving predictive performance for both regression and binary classification tasks. Third, we compare LuPTS to the distillation approaches described in Section 3.2. Finally, we compare the stationary and non-stationary versions of LuPTS, demonstrating that the preferred version depends on the domain and prediction task.

4.1 Experimental Setup

The LuPTS algorithm computes several OLS estimates (see Algorithm 1). All OLS estimates, including the baseline model used for comparison, use the (unregularized) implementation LinearRegression of the Python module scikit-learn (Pedregosa et al., 2011). Although it would be of interest to also study regularized variants of these models, we leave it as future work for a more thorough investigation, both theoretically and experimentally. When extending the algorithm to binary classification tasks, the baseline model and the outcome model in the LuPTS algorithm are implemented using the LogisticRegressionCV class from scikit-learn. We perform 5-fold cross-validation on the training portion to tune the L_2 regularization parameter, which we vary from 1×10^{-4} to 1×10^{4}. Models are evaluated using the coefficient of determination (R^2) for regression tasks and the Area Under the ROC Curve (AUC) for classification tasks. In all plots, Baseline refers to OLS or Logistic Regression (depending on the task), LuPTS refers to the output of Algorithm 1 *without* stationarity, and Stat-LuPTS to the output *with* stationarity.

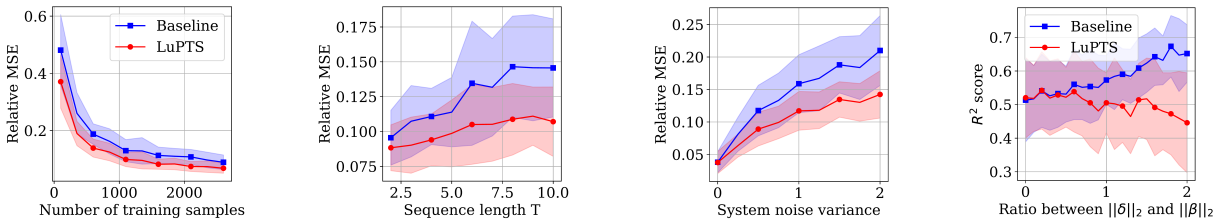
[2]Code available at github.com/RickardKarl/LearningUsingPrivilegedTimeSeries.

4.2 Synthetic Experiments

To verify and further investigate our theoretical results, we sample from a synthetic dynamical system where Markovianity and linearity with additive isotropic Gaussian noise hold. The elements of $A_t \in \mathbb{R}^{d\times d}$ for $t = 1, \ldots, T-1$ and $\beta \in \mathbb{R}^{d\times 1}$ are drawn from a Normal distribution with the exception of the diagonal elements in the transition matrices, which are set to 1. The eigenvalues $(\lambda_1, \ldots, \lambda_d)$ of A_t influence the system's behavior and stability. Unstable linear systems, i.e., those with large eigenvalues, are easier to estimate (Simchowitz et al., 2018), and therefore we enforce the spectral radius $\rho(A_t)$ to equal κ for all t, with $\kappa > 1$. We refer the reader to the Appendix for a more in-depth description of the system generation. For all experiments, we use the following default values unless otherwise stated: $\kappa = 1.5$, $n = 1000$, $T = 10$, $d = 25$, and $\text{Var}(\epsilon_t) = \text{Var}(\epsilon_Y) = 1$ for $t = 1, \ldots, T-1$. Finally, the input distribution is $p(X_1) = \mathcal{N}(\mu = 0, \sigma^2 = 5)$.

Parameter Recovery Figures 2a, 2b and 2c present the relative MSE, $||\theta - \hat{\theta}||_2^2/||\theta||_2^2$, of the Baseline (OLS) and LuPTS estimates of the synthetic system described above. We investigate the impact of the number of training samples n, sequence length T, and variance of system noise on the MSE by varying one variable and keeping the other two fixed. Compared to the baseline estimates, the LuPTS estimates are closer in general to the true parameter, as predicted by Theorem 1. Both methods improve as we increase the number of training samples, but LuPTS is consistently better or equally good. The difference between them increases for larger T, which can be explained by the fact that there is more uncertainty between X_1 and Y as T gets larger. Notably, when the system noise is removed completely, LuPTS and OLS coincide, as expected.

Breaking The Markov Assumption In Figure 2d, we introduce a coefficient δ, generated in the same way as β, which controls a direct dependence of Y on X_1, i.e., $Y = X_T\beta + X_1\delta$. We scale δ to vary the ratio of the norms $\frac{||\delta||_2}{||\beta||_2}$ (x-axis), and observe that predictions from LuPTS get worse in terms of R^2 score (y-axis) as the ratio increases. In spite of the bias, we see that LuPTS still performs equally well as the baseline for small non-zero ratios. This result can be explained by the fact that the privileged information contains useful knowledge, which offsets the bias when it is small.



(a) Parameter recovery (b) Parameter recovery (c) Parameter recovery (d) Prediction error

Figure 2: Synthetic system showing relative MSE of the parameter estimate or the prediction error and one standard deviation. **Parameter recovery** (Left three; lower is better): Varying either only number of samples n, sequence length T, or noise standard deviation $\text{Var}(\epsilon_t)$. **Breaking the Markov Assumption** (Right-most figure; higher is better): Adding a direct linear relationship between X_1 and Y with coefficient δ. Larger value on x-axis leads to more deviation from Markov assumption.

4.3 Forecasting Air Quality

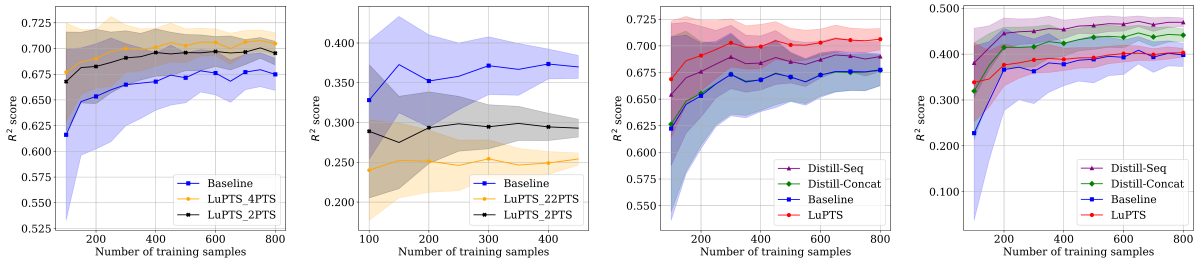
Due to the serious health risks caused by chronic air pollution in China, predicting how air quality changes over time is vital (Kelly et al., 2012). The $\text{PM}_{2.5}$ dataset contains hourly meteorological recordings between the years 2012 and 2015 of the fine particle ($\text{PM}_{2.5}$) concentration in Beijing, Chengdu, Guangzhou, Shanghai and Shenyang (Liang et al., 2016), as well as seven weather features, including temperature, humidity and combined wind direction. We refer the reader to the Appendix for a complete description of the data and pre-processing steps. We also report additional results for all five cities.

We forecast the $\text{PM}_{2.5}$ concentration for several horizons, 6, 12, or 24 hours into the future. Comparing results across different horizons is informative since 1) predictions further into the future are more challenging, and 2) for longer horizons, more time points can be used as privileged information. We further compare LuPTS to Distill-Concat and Distill-Seq, introduced in Section 3.2, as well as non-linear baselines models. For the distillation methods, we tune λ on the validation set, varying it across 0.25, 0.5, and 0.75. At time $t = 1$, we observe the features X_1, which also contains the current $\text{PM}_{2.5}$ concentration. Based on this information, we wish to predict the $\text{PM}_{2.5}$ concentration $T + 1$ hours into the future. The spacing between adjacent intermediate measurements $X_2, \ldots, X_T$—the privileged information—is one hour.

Bias-variance Trade-off When Varying Sequence Length And Privileged Information Results for two forecast horizons with a different number of evenly spaced time points used as privileged information are shown in Figures 3a and 3b. The results depict an interesting example of the bias-variance trade-off of LuPTS. For the 6 hour forecast (Figure 3a), we see improved performance using LuPTS for all sample sizes. In addition, adding more privileged time points is beneficial. For the 24 hour forecast (Figure 3b), LuPTS is consistently worse than baseline. Interestingly, in the case where LuPTS performs worse already, adding more privileged time points is not beneficial. This result may be due to the learned dynamical system being biased, and consequently, the bias compounds when the predicted "roll-out" is longer. This argument also explains why using more privileged time points is subpar if the bias is large already. On the other hand, adding more privileged information reduces the variance, as seen in both Figure 3a and 3b.

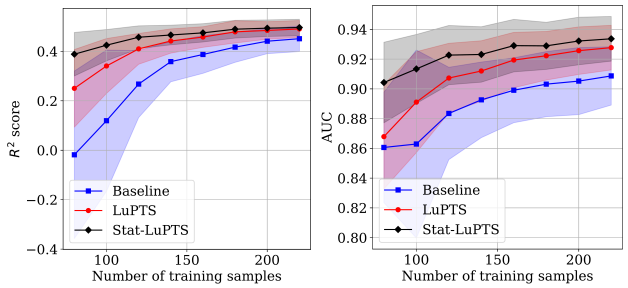
Combining LuPTS And Distillation Can Lead To Even Greater Performance Figure 3c and 3d show the results comparing the distillation-based methods, Distill-Concat and Distill-Seq, which use the same privileged information as LuPTS during training. When forecasting 6 hours ahead (Figure 3c), we see that LuPTS performs better than both distillation-based methods. Distill-Seq, which uses LuPTS as teacher model, also has a higher R^2 score than Distill-Concat, where the latter method lies close to the baseline. As previously posited, the bias is likely small in this case, and the empirical result conforms closely to Theorem 2, which states that the MSE, or equivalently R^2, of Distill-Seq is bounded by the MSEs of LuPTS and OLS in the well-specified case.

For the 12 hour forecast (Figure 3d), the distillation-based methods perform best, with Distill-Seq still outperforming Distill-Concat. As before, LuPTS is preferable to the baseline although the difference is only observed for a lower number of samples. This result is a good example of how, in the misspecified setting, Distill-Seq can do no worse than LuPTS or OLS, given that λ is well chosen. Finally, we find that Distill-Seq does better than Distill-Concat in both cases, which can be attributed to the benefit of using LuPTS as a teacher model to derive better soft targets.



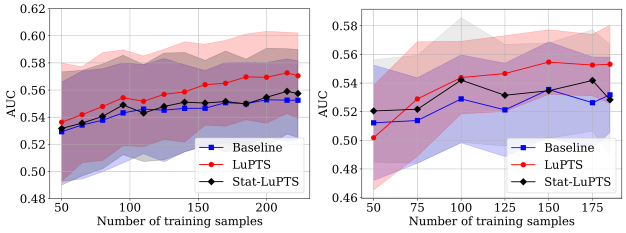
(a) 6 hour forecast Shenyang (b) 24 hour forecast Shenyang (c) 6 hour forecast Shenyang (Distillation) (d) 12 hour forecast Chengdu (Distillation)

Figure 3: 3a, 3b) Changing the amount of privileged information for the LUPTS for different time horizons, where the X in LuPTS_XPTS indicates the number of privileged time points. 3c, 3d) Comparing LuPTS to the distillation-based approaches, which use the same privileged information. Metric used is R^2 (Higher is better); shaded region indicates one standard deviation across 75 iterations.



(a) Predicting MMSE (b) Predicting AD

Figure 4: **Alzheimer's disease progression tasks**. Follow-up at 12, 24 and 36 months after baseline as privileged information. Metrics used are R^2/AUC; shaded region corresponds to one standard deviation across 100 iterations.



(a) Early/Late Progressor (b) Treatment Response

Figure 5: **MM Progression Task** (Left): AUC as a function of the training set size is shown for 4 privileged time points. **Treatment Response Task** (Right) for 2 privileged time points are used. Shaded region for both tasks indicates one standard deviation over all trials.

Comparison to non-linear baselines In addition to the distillation-based baselines, we compare LuPTS to non-linear baselines in the form of random forest (RF) and k-nearest neighbors regression (KNN). Results with a fixed sample size of 200 and a prediction horizon of 6 hours for all cities are shown in Table 1. For all cities, the LuPTS or it's distillation equivalent empirically performs better than these non-linear models without access to privileged information in a setting where linearity, Markovianity or Gaussianity are likely not to hold. One possible reason for this is that non-linear models tend to overfit in low-data settings. See Appendix D for implementation details of the non-linear models and results for a prediction horizon of 12 hours (Table 4).

4.4 Modeling Progression Of Chronic Disease

Alzheimer's Progression Modeling In this experiment, we predict progression of Alzheimer's disease (AD) as measured by two different subject outcomes (i) the diagnostic status (AD or not), and (ii) cognitive function as assessed by the Mini Mental State Examination (MMSE) score (Galea and Woodward, 2005). The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a large multi-site research study tracking the brains of over 2000 participants through genetic, imaging and biospecimen biomarkers.[3] Subjects are followed over several years with measurements taken every three months, with some missingness in the observations.

We observe outcomes (Y) at a fixed follow-up time, 48 months after baseline measurements (X_1) are taken. Privileged information is collected at intermediate time points, here restricted to samples from follow-ups at 12, 24, and 36 months after baseline. See the Appendix for all selected features and details on data pre-processing.

Multiple Myeloma Progression Modeling Given the limited samples available due to the relative rarity of the disease, Multiple Myeloma (MM) provides a suitable setting to demonstrate the utility of using temporal, post-baseline data in improving predictive performance. We use a data registry released by the

[3] ADNI: `http://adni.loni.usc.edu`

Table 1: Comparison of regression methods on the air quality forecasting task with a fixed sample size $n = 200$ and a prediction horizon of 6 hours. Metric used is R^2 (Higher is better); mean value for each method with standard deviation across 200 iterations. The methods with highest R^2 and lowest variance are marked in bold for each city.

Method	Beijing	Shanghai	Shenyang	Chengdu	Guangzhou
Baseline	0.64 ± 0.02	0.58 ± 0.06	0.66 ± 0.04	0.63 ± 0.04	0.45 ± 0.06
LuPTS	0.64 ± 0.03	**0.62 ± 0.03**	**0.70 ± 0.03**	0.65 ± 0.03	0.49 ± 0.04
Stat-LuPTS	0.64 ± 0.02	0.62 ± 0.04	0.69 ± 0.03	0.65 ± 0.03	**0.50 ± 0.04**
Distill-Seq	**0.65 ± 0.02**	**0.62 ± 0.03**	0.68 ± 0.03	**0.67 ± 0.02**	0.49 ± 0.05
Distill-Concat	**0.65 ± 0.02**	0.60 ± 0.04	0.66 ± 0.04	0.66 ± 0.03	0.46 ± 0.07
RF	0.62 ± 0.03	0.58 ± 0.07	0.53 ± 0.06	0.61 ± 0.04	0.48 ± 0.05
KNN	0.57 ± 0.04	0.51 ± 0.23	0.49 ± 0.05	0.51 ± 0.04	0.26 ± 0.09

Multiple Myeloma Research Foundation (MMRF) through the CoMMpass clinical trial (NIH, 2016), which contains de-identified clinical data collected at 2-month intervals for 1143 patients. Preprocessing of the data was done through ML-MMRF, an open-source library provided by Hussain et al. (2021). We refer the reader to the Appendix for a detailed description of the features. We focus on two clinically important predictive tasks for multiple myeloma:

Early/Late Progression Task: The first task is predicting whether or not a patient will progress "early" ($Y = 1$) (before 18 months post-treatment induction) or "late" ($Y = 0$) (after 18 months post-treatment induction). We experiment with using privileged time points within the first "line" (or sequence) of treatment.

Treatment Response Task: The second task is predicting a patient's treatment response (given a fixed treatment policy) as either "Progressive Disease" (PD, $Y = 1$) or "non-Progressive Disease" (non-PD, $Y = 0$). These are used by oncologists to make treatment decisions and assess disease burden. The outcome is recorded after two lines of treatment have been completed. We have privileged information at the end of first line treatment ($t = 2$) and at the end of second line treatment ($t = 3$).

For all tasks outlined in this section, we do repeated (50 repeats) 2-fold cross validation with different training and test splits across multiple training set sizes. Note that when $T = 2$ (one privileged time point), LuPTS and Stat-LuPTS return the same estimator. Hence, only LuPTS is shown in these figures.

LuPTS For Disease Progression Modeling Using LuPTS improves predictive performance for all of the clinical tasks and leads to a reduced variance in estimation, as shown in Figures 4 and 5. This result intuitively implies that it is easier to predict clinical outcomes of chronic diseases from an intermediate set of longitudinal features than from baseline features alone. Indeed, in the context of the MM tasks, the PD category is often determined by temporal changes in a patient's lab values, and recurrence of disease is often measured by temporal changes in a patient's serum immunoglobulins (Kyle and Rajkumar, 2009). We perform additional experiments comparing Stat-LuPTS with LuPTS. For the AD progression tasks, using a stationary transition matrix results in further performance gains (see Figure 4). However, for the MM tasks, Stat-LuPTS does not outperform the baseline model (see Figure 5). This makes sense since the longitudinal dynamics of a myeloma patient may differ across different lines of treatment, justifying a separate transition matrix for each line.

Assessing Feature Importance For MM Early/Late Progression Task In Figure 12 in the Appendix, we show the feature weights of the LuPTS and baseline outcome models in the top and bottom rows of the heatmap, respectively. We find that for the LuPTS estimator, the highest weighted features are the ISS stage and the projected serum M-protein of the patient. This result is consistent with current clinical understanding of myeloma, which measures disease burden based on a patient's M-protein and ISS risk score. On the other hand, we find that the relevant features for the baseline estimator are the myeloma subtypes and not the biomarkers. This is most likely due to the fact that the baseline model takes the biomarkers at the first time step as input, which may be less associated with the overall progression of the patient. This result indicates that using the LuPTS estimator results in a more clinically intuitive explanation for its prediction.

5 RELATED WORK

Making use of information only available at training time was first systematically studied in the context of Learning using Privileged Information (LuPI) (Vapnik

and Vashist, 2009).

We study prediction of future outcomes (Makridakis, 1994; Ing, 2003; Sorjamaa et al., 2007), where the interval between baseline and prediction target is assumed sufficiently long to collect intermediate privileged information. This is related to multi-step prediction in time-series forecasting, with common strategies including direct (Chevillon, 2007) and recursive prediction (Kunitomo and Yamamoto, 1985; Ing, 2003), corresponding to the baseline and LupTS strategies used in this work. However, unlike time-series forecasting, which predicts a future state of a continually evolving variable, our goal is to predict a distinct outcome variable at a fixed finite horizon. Ing (2003) showed for time-series forecasting that in stationary Gaussian-linear systems, recursive prediction is asymptotically preferable to direct prediction. This is consistent with our findings, but these are for the non-stationary case and non-asymptotic.

Our main analysis tool is the Rao-Blackwell theorem (Rao, 1945; Blackwell, 1947), used widely for variance reduction of statistical estimators, such as in the Rao-Blackwellization of MCMC sampling schemes (Casella and Robert, 1996). This use is distinct from ours. It has also been used to improve policy evaluation in RL (Li et al., 2018), general variational inference (Ranganath et al., 2014), and estimation of field goal percentage in basketball (Daly-Grafstein and Bornn, 2019). However, to the best of our knowledge, the result has not previously been used to prove gains from learning using privileged information.

The question of leveraging explicit models of dynamics commonly arises in reinforcement learning (RL) (Sutton and Barto, 2018). In model-based RL, a learned model of system dynamics is used to simulate state transitions in order to predict (long-term) future rewards. This problem maps onto ours when there is a single available action with the reward being given at a fixed future time step. The question of *when* the bias due to the use of a model in model-based RL is preferable to the higher variance model-free RL remains open (Feinberg et al., 2018; Thomas and Brunskill, 2016).

6 DISCUSSION

In this work, we studied prediction of future outcomes in a setting where privileged information is available in the form of a time series observed between prediction and outcome time points. We proved that a recursive estimator that makes use of this privileged information yields improved parameter recovery and improved expected risk compared to the best estimator that does not use privileged information. Through experiments on synthetic and real-world data sets, we showed that our estimator, dubbed LuPTS, often results in better predictive performance and variance reduction in both regression and classification tasks. We also proposed a method for using LuPTS in combination with distillation-based learning to reduce prediction risk in the misspecified case by trading off bias and variance.

Interestingly, compared to prior work on learning using privileged information, our results are qualitatively different. Instead of providing asymptotic bounds on generalization error as done before Vapnik and Vashist (2009); Lopez-Paz et al. (2016) we prove an explicit gap on the improvement in the finite-sample case. A possibly fruitful direction for research could be to further explore the connections between our results and previous results within the topic.

There are some notable limitations to our work. First, the theory is limited to time series from discrete-time linear dynamical systems with isotropic Gaussian noise where the transition matrices for each time step are estimated separately. Furthermore, we assume that the particular structure of the time series is Markov. Extending the theory and algorithm to include non-linear transitions and estimators or exploit stationary series is interesting future work as it will broaden the understanding and applicability of learning using privileged time series. As a start, we provide a general algorithm for arbitrary estimators in the Appendix.

Acknowledgments

The authors thank Alexander D'Amour and Chandler Squires for valuable feedback on initial versions of the manuscript. We also thank the Alzheimer's Neuroimaging Initiative (ADNI) for collecting and providing the data used in this project. In addition, the MMRF data were generated as part of the Multiple Myeloma Research Foundation Personalized Medicine Initiatives (`https://research.themmrf.org` and `www.themmrf.org`). Fredrik Johansson was funded in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Zeshan Hussain was supported by an ASPIRE award from The Mark Foundation for Cancer Research.

References

David Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, pages 105–110, 1947.

George Casella and Christian P Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

Guillaume Chevillon. Direct multi-step estimation and forecasting. *Journal of Economic Surveys*, 21(4): 746–785, 2007.

Daniel Daly-Grafstein and Luke Bornn. Rao-blackwellizing field goal percentage. *Journal of Quantitative Analysis in Sports*, 15(2):85–95, 2019.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I. Jordan, Joseph E. Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

Mary Galea and Michael Woodward. Mini-mental state examination (mmse). *Australian Journal of Physiotherapy*, 51(3):198, 2005.

Shogo Hayashi, Akira Tanimoto, and Hisashi Kashima. Long-term prediction of small time-series data using generalized distillation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

Zeshan Hussain, Rahul G Krishnan, and David Sontag. Neural pharmacodynamic state space modeling. *arXiv preprint arXiv:2102.11218*, 2021.

Ching-Kang Ing. Multistep prediction in autoregressive processes. *Econometric theory*, 19(2):254–279, 2003.

Richard Arnold Johnson, Dean W Wichern, et al. *Applied multivariate statistical analysis*, volume 5–8. Prentice hall Upper Saddle River, NJ, 2002.

Rico Jonschkowski, Sebastian Höfer, and Oliver Brock. Patterns for learning with side information. *arXiv preprint arXiv:1511.06429*, 2015.

Frank J. Kelly, Gary W. Fuller, Heather A. Walton, and Julia C. Fussell. Monitoring air pollution: Use of early warning systems for public health. *Respirology*, 17(1):7–19, 2012.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Naoto Kunitomo and Taku Yamamoto. Properties of predictors in misspecified autoregressive time series models. *Journal of the American Statistical Association*, 80(392):941–950, 1985.

RA Kyle and S Vincent Rajkumar. Criteria for diagnosis, staging, risk stratification and response assessment of multiple myeloma. *Leukemia*, 23(1):3–9, 2009.

Jiajin Li, Baoxiang Wang, and Shengyu Zhang. Policy optimization with second-order advantage information. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5038–5044. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

Xuan Liang, Shuo Li, Shuyi Zhang, Hui Huang, and Song Xi Chen. Pm2.5 data reliability, consistency, and air quality assessment in five chinese cities. *Journal of Geophysical Research: Atmospheres*, 121(17): 10,220–10,236, 2016.

D. Lopez-Paz, B. Schölkopf, L. Bottou, and V. Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, November 2016.

Spyros Makridakis. Time series prediction: Forecasting the future and understanding the past. *International Journal of Forecasting*, 10(3):463–466, 1994.

NIH. Relating clinical outcomes in multiple myeloma to personal assessment of genetic profile (commpass). *Clinical Trials website. https://clinicaltrials.gov/ct2/show/NCT01454297*, 2016.

Antonio Palumbo, Hervé Avet-Loiseau, Stefania Oliva, Henk M Lokhorst, Hartmut Goldschmidt, Laura Rosinol, Paul Richardson, Simona Caltagirone, Juan José Lahuerta, Thierry Facon, et al. Revised international staging system for multiple myeloma: a report from international myeloma working group. *Journal of clinical oncology*, 33(26):2863, 2015.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, November 2011. ISSN 1532-4435.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

Radhakrishna C Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37(3): 81–91, 1945.

John A. Rice. *Mathematical Statistics and Data Analysis*. Belmont, CA: Duxbury Press., 3rd edition, 2006.

Carlos Serra-Toro, V Javier Traver, and Filiberto Pla. Exploring some practical issues of svm+: Is really privileged information that helps? *Pattern Recognition Letters*, 42:40–46, 2014.

Max Simchowitz, Horia Mania, Stephen Tu, Michael I. Jordan, and Benjamin Recht. Learning without mixing: Towards A sharp analysis of linear system identification. *CoRR*, abs/1802.08334, 2018.

Antti Sorjamaa, Jin Hao, Nima Reyhani, Yongnan Ji, and Amaury Lendasse. Methodology for long-term prediction of time series. *Neurocomputing*, 70(16-18): 2861–2869, 2007.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Fengyi Tang, Cao Xiao, Fei Wang, Jiayu Zhou, and Li-wei H Lehman. Retaining privileged information for multi-task learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1369–1377, 2019.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148. PMLR, 2016.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):1–5, 2018.

Appendix

The appendix contains the following sections. We also highlight the key takeaways or descriptions associated with each section.

A. **Proof of Theorem 1**: This section provides the full proof of Theorem 1, including two lemmas that are central to the argument. We also show how one can relax the isotropic noise assumption on the Gaussian-linear system and generalize the first lemma to the anisotropic case.

B. **Proof of Theorem 2**: This section provides a full proof of Theorem 2, showing how using a distillation-approach with LuPTS as a teacher model returns the convex combination of the OLS estimator and output from Algorithm 1. We further show that the MSE of the estimator using LuPTS with distillation is bounded between the MSEs of the OLS and LuPTS estimators.

C. **LuPTS with Non-linear Estimators**: This section includes a bound on the expected risk of the LuPTS estimator in the case where the transition functions and outcome model are non-linear.

D. **Experimental Details**

   D1. **Computational Resources** - We give a brief description of the computational resources that were used to generate the experimental results as well as the running time required to reproduce them.

   D2. **Synthetic Experiments** - We present a more detailed description of how the synthetic data is generated. Additionally, we test empirically in the synthetic setting how the Stat-LuPTS, LuPTS, and baseline OLS estimators compare when stationarity holds and when it does not.

   D3. **Forecasting Air Quality** - We provide a detailed description of the features used for this task as well as the training and evaluation procedure. Finally, we present additional experimental results comparing the LuPTS and baseline OLS estimators for the other Chinese cities.

   D4. **Alzheimer's Progression Modeling** - Along with giving the full list of features and our pre-processing procedures used, we present the results from the main paper in tabular form. This gives a more granular look at which sample sizes LuPTS yields the most gain in AUC and reduction in variance.

   D5. **Multiple Myeloma Progression Modeling** - We give a description of the features used for the Multiple Myeloma experiments as well as the pre-processing procedures used to handle missingness and censorship. We then present a brief description of our evaluation procedure on this dataset. Finally, we end with a qualitative experiment looking at the most highly-weighted features in the LuPTS and baseline outcome models for the early/late progression task.

A PROOF OF THEOREM 1

To prove Theorem 1, we begin by proving the following lemma of OLS estimates.

Lemma 1. Let $K = (\hat{A}_1, \dots, \hat{A}_{T-1}, \hat{\beta})$ be the output of Algorithm 2, and let $(\mathbf{X}_1, \dots, \mathbf{X}_T, \mathbf{Y})$ be a random dataset from the Gaussian-linear Markov dynamical system as defined in Assumption 2, with isotropic noise, $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2 I)$. Then, for any $t = 2, \dots, T$ we have that

$$\mathbb{E}[\mathbf{X}_t \mid \mathbf{X}_{t-1}, K] = \mathbf{X}_{t-1}\hat{A}_{t-1}$$

and

$$\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_T, K] = \mathbf{X}_T\hat{\beta} \ .$$

The main difference between the above equations is the dimensionality of $\mathbf{X}_t$ and $\mathbf{Y}$, respectively, where we have previously stated, without loss of generality, that $\mathbf{X}_t \in \mathbb{R}^{n\times d}$ and $\mathbf{Y} \in \mathbb{R}^{n\times 1}$.

Proof. We will first show $\mathbb{E}[\mathbf{X}_t \mid \mathbf{X}_{t-1}, K] = \mathbf{X}_{t-1}\hat{A}_{t-1}$ and then explain how the same arguments are applied to prove $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_T, K] = \mathbf{X}_T\hat{\beta}$.

Let $\mathbf{R}_t = \mathbf{X}_t - \mathbf{X}_{t-1}\hat{A}_{t-1}$ be the residual of the OLS estimate, $\hat{A}_{t-1}$. We will show that for all $\mathbf{R}_t$, we have that $p(\mathbf{X}_t = \mathbf{X}_{t-1}\hat{A}_{t-1} + \mathbf{R}_t \mid \mathbf{X}_{t-1}, K) = p(\mathbf{X}'_t = \mathbf{X}_{t-1}\hat{A}_{t-1} - \mathbf{R}_t \mid \mathbf{X}_{t-1}, K)$, which implies the statement in the lemma if we assume isotropic Gaussian noise.

To show this, we first use Bayes formula:

$$\begin{aligned} p(\mathbf{X}_t \mid \mathbf{X}_{t-1}, K) &= \frac{p(K \mid \mathbf{X}_t, \mathbf{X}_{t-1})p(\mathbf{X}_t \mid \mathbf{X}_{t-1})}{p(K \mid \mathbf{X}_{t-1})} \\ &= \frac{p(\hat{\beta}, \hat{A}_{T-1}, \dots, \hat{A}_t \mid \mathbf{X}_t)p(\hat{A}_{t-1} \mid \mathbf{X}_t, \mathbf{X}_{t-1})p(\hat{A}_1, \dots, \hat{A}_{t-2} \mid \mathbf{X}_{t-1})p(\mathbf{X}_t \mid \mathbf{X}_{t-1})}{p(K \mid \mathbf{X}_{t-1})} \end{aligned}$$

In the second equality, we have used the Markov property, which implies the following statements:

$$\begin{aligned} \hat{A}_1, \dots, \hat{A}_{t-2} &\perp\!\!\!\perp \mathbf{X}_t \mid \mathbf{X}_{t-1} \\ \hat{A}_t, \dots, \hat{A}_{T-1}, \hat{\beta} &\perp\!\!\!\perp \mathbf{X}_{t-1} \mid \mathbf{X}_t \\ \hat{A}_{t-1} \perp\!\!\!\perp \hat{A}_1, \dots, \hat{A}_{t-2}, \hat{A}_t, \dots, \hat{A}_{T-1}\hat{\beta} &\mid \mathbf{X}_t, \mathbf{X}_{t-1}. \end{aligned}$$

For $p(\mathbf{X}_t \mid \mathbf{X}_{t-1}, K) = p(\mathbf{X}'_t \mid \mathbf{X}_{t-1}, K)$ to hold, we look at the factors that depend on $\mathbf{X}_t$. This tells us that we need to prove the following three statements:

(a) $p(\mathbf{X}_t \mid \mathbf{X}_{t-1}) = p(\mathbf{X}'_t \mid \mathbf{X}_{t-1})$

(b) $p(\hat{A}_{t-1} \mid \mathbf{X}_t, \mathbf{X}_{t-1}) = p(\hat{A}_{t-1} \mid \mathbf{X}'_t, \mathbf{X}_{t-1})$

(c) $p(\hat{\beta}, \hat{A}_{T-1}, \dots, \hat{A}_t \mid \mathbf{X}_t) = p(\hat{\beta}, \hat{A}_{T-1}, \dots, \hat{A}_t \mid \mathbf{X}'_t)$

We will now prove each of these statements:

Statement (a): We first define $\boldsymbol{\epsilon}'_t = \boldsymbol{\epsilon} - 2\mathbf{R}_t$ where we have that $\mathbf{X}_t = \mathbf{X}_{t-1}A_{t-1} + \boldsymbol{\epsilon}$. Then, note that

$$\mathbf{X}_{t-1}A_{t-1} + \boldsymbol{\epsilon}'_t = \mathbf{X}_{t-1}A_{t-1} + \boldsymbol{\epsilon} - 2\mathbf{R}_t = \mathbf{X}_t - 2\mathbf{R}_t = \mathbf{X}_{t-1}\hat{A}_{t-1} - \mathbf{R}_t = \mathbf{X}'_t \tag{4}$$

Eq. 4 implies that showing (a) equates to showing that $p(\boldsymbol{\epsilon}) = p(\boldsymbol{\epsilon}'_t)$, since the noise is independent of $\mathbf{X}_{t-1}$. For Gaussian noise, these probabilities are determined by the inner product of the noise, hence it is sufficient to prove

$$\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon} = \boldsymbol{\epsilon}'^\top\boldsymbol{\epsilon}'$$

We have that

$$\boldsymbol{\epsilon}'^\top\boldsymbol{\epsilon}' = \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon} - 4\boldsymbol{\epsilon}^\top\mathbf{R}_t + 4\mathbf{R}_t^\top\mathbf{R}_t$$

and thus, we need to show

$$\mathbf{R}_t^\top(\boldsymbol{\epsilon} - \mathbf{R}_t) = 0 \ .$$

By definition,

$$\mathbf{R}_t = \mathbf{X}_t - \mathbf{X}_{t-1}\hat{A}_{t-1}$$

and so

$$\mathbf{R}_t^\top(\boldsymbol{\epsilon} - \mathbf{R}_t) = \mathbf{R}_t^\top(\mathbf{X}_t - \mathbf{X}_{t-1}A_{t-1} - (\mathbf{X}_t - \mathbf{X}_{t-1}\hat{A}_{t-1})) = -\mathbf{R}_t^\top(\mathbf{X}_{t-1}(A_{t-1} - \hat{A}_{t-1})) = 0$$

since $\mathbf{R}_t^\top\mathbf{X}_{t-1} = 0$ is a property of the OLS estimator. This proves statement (a).

Statement (b): Now, we see that

$$\begin{aligned}\hat{A}'_{t-1} &= (\mathbf{X}_{t-1}^\top\mathbf{X}_{t-1})^{-1}\mathbf{X}_{t-1}^\top\mathbf{X}'_t \\ &= (\mathbf{X}_{t-1}^\top\mathbf{X}_{t-1})^{-1}\mathbf{X}_{t-1}^\top(\mathbf{X}_t - 2\mathbf{R}_t) \\ &= \hat{A}_{t-1} - 2(\mathbf{X}_{t-1}^\top\mathbf{X}_{t-1})^{-1}\mathbf{X}_{t-1}^\top\mathbf{R}_t = \hat{A}_{t-1} \ .\end{aligned}$$

since, again, $\mathbf{X}_{t-1}^\top\mathbf{R}_t = 0$. This implies that the distribution of $\hat{A}_{t-1}$ is the same if conditioned on $\mathbf{X}_t$ or $\mathbf{X}'_t$, which proves statement (b).

Statement (c): We can factorize $p(\hat{\beta}, \hat{A}_{T-1}, \ldots, \hat{A}_t \mid \mathbf{X}_t)$ as

$$p(\hat{\beta} \mid \hat{A}_{T-1}, \ldots, \hat{A}_t, \mathbf{X}_t)p(\hat{A}_{T-1} \mid \hat{A}_{T-2}, \ldots, \hat{A}_t, \mathbf{X}_t)\ldots p(\hat{A}_{t+1} \mid \hat{A}_t, \mathbf{X}_t)p(\hat{A}_t \mid \mathbf{X}_t)$$

Let $C_k = (\hat{A}_k, \hat{A}_{k-1}, \ldots, \hat{A}_t)$ for $k = t, \ldots, T-1$. Then, we can summarize the problem as the following: we need to show that each factor in the above equation is the same for both $\mathbf{X}_t$ and $\mathbf{X}'_t$, i.e.,

$$\begin{aligned}&p(\hat{\beta} \mid C_{T-1}, \mathbf{X}_t) = p(\hat{\beta} \mid C_{T-1}, \mathbf{X}'_t) \\ &p(\hat{A}_k \mid C_{k-1}, \mathbf{X}_t) = p(\hat{A}_k \mid C_{k-1}, \mathbf{X}'_t), \quad k = t+1, \ldots, T-1 \\ &p(\hat{A}_t \mid \mathbf{X}_t) = p(\hat{A}_t \mid \mathbf{X}'_t)\end{aligned}$$

The first two equations could be seen as the distribution of OLS estimators with a (conditional) random design, while the third is the distribution of $\hat{A}_t$ with a fixed design matrix $\mathbf{X}_t$. Assuming mean-zero and uncorrelated Gaussian noise $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2 I)$ and $\epsilon_Y \sim \mathcal{N}(0, \sigma_Y^2 I)$, the distributions of the OLS estimators are known, see Chapter 14 in Rice (2006), and we can show,

$$\begin{aligned}\hat{\beta} \mid C_{T-1}, \mathbf{X}_t \sim &\mathcal{N}\left(\beta, \sigma_Y^2\mathbb{E}\left[\left(\mathbf{X}_T^\top\mathbf{X}_T\right)^{-1} \mid C_{T-1}, \mathbf{X}_t\right]\right) \\ \hat{A}_k^{(\text{row } i)} \mid C_{k-1}, \mathbf{X}_t \sim &\mathcal{N}\left(A_k^{(\text{row } i)}, \sigma_{k+1}^2\mathbb{E}\left[\left(\mathbf{X}_k^\top\mathbf{X}_k\right)^{-1} \mid C_{k-1}, \mathbf{X}_t\right]\right), \quad k = t+1, \ldots, T-1 \\ \hat{A}_t^{(\text{row } i)} \mid \mathbf{X}_t \sim &\mathcal{N}\left(A_t^{(\text{row } i)}, \sigma_{t+1}^2\left(\mathbf{X}_t^\top\mathbf{X}_t\right)^{-1}\right)\end{aligned}$$

where $i = 1, \ldots, d$ corresponds to the OLS estimators which are $d \times d$ matrices. Now, it is sufficient to show that

$$\mathbb{E}\left[\left(\mathbf{X}_k^\top\mathbf{X}_k\right)^{-1} \mid C_{k-1}, \mathbf{X}_t\right] = \mathbb{E}\left[\left(\mathbf{X}_k^\top\mathbf{X}_k\right)^{-1} \mid C_{k-1}, \mathbf{X}'_t\right], \quad k = t+1, \ldots, T$$

and the special case where $\mathbf{X}_t^\top\mathbf{X}_t = \mathbf{X'}_t^\top\mathbf{X}'_t$. For the latter, we have

$$\begin{aligned}(\mathbf{X}_{t-1}\hat{A}_{t-1} \pm \mathbf{R}_t)^\top(\mathbf{X}_{t-1}\hat{A}_{t-1} \pm \mathbf{R}_t) = \\ = (\mathbf{X}_{t-1}\hat{A}_{t-1})^\top(\mathbf{X}_{t-1}\hat{A}_{t-1}) \pm 2(\mathbf{X}_{t-1}\hat{A}_{t-1})^\top\mathbf{R}_t + \mathbf{R}_t^\top\mathbf{R}_t \\ = (\mathbf{X}_{t-1}\hat{A}_{t-1})^\top(\mathbf{X}_{t-1}\hat{A}_{t-1}) + \mathbf{R}_t^\top\mathbf{R}_t\end{aligned}$$

where we have used that the cross term $(\mathbf{X}_{t-1}\hat{A}_{t-1})^\top\mathbf{R}_t = \hat{A}_{t-1}^\top\mathbf{X}_{t-1}^\top\mathbf{R}_t = 0$ because $\mathbf{X}_{t-1}^\top\mathbf{R}_t = 0$. As the cross term is the only thing which differs in $\mathbf{X}_t^\top\mathbf{X}_t$ and $\mathbf{X'}_t^\top\mathbf{X}'_t$, the above derivation implies that they must be equal.

For $\mathbb{E}\left[\left(\mathbf{X}_k^\top \mathbf{X}_k\right)^{-1} \mid C_{k-1}, \mathbf{X}_t\right]$ with $k = t+1, \ldots, T$, we use the same expression as before but observe the following recursive relationship between the inner product of $\mathbf{X}_k$ and $\mathbf{X}_{k-1}$:

$$\mathbf{X}_k^\top \mathbf{X}_k = (\mathbf{X}_{k-1}\hat{A}_{k-1})^\top \mathbf{X}_{k-1}\hat{A}_{k-1} + \mathbf{R}_k^\top \mathbf{R}_k = \hat{A}_{k-1}^\top \underbrace{\mathbf{X}_{k-1}^\top \mathbf{X}_{k-1}}_{\text{Inner product}} \hat{A}_{k-1} + \mathbf{R}_k^\top \mathbf{R}_k$$

Hence, we get that

$$\mathbf{X}_k^\top \mathbf{X}_k = \prod_{i=t}^{k-1} \hat{A}_i^\top (\mathbf{X}_t^\top \mathbf{X}_t) \prod_{i=t}^{k-1} \hat{A}_i + \sum_{j=t+1}^{k-1} \mathbf{R}_j^\top \mathbf{R}_j \prod_{i=j}^{k-1} \hat{A}_i + \mathbf{R}_k^\top \mathbf{R}_k$$

We see that $\mathbf{X}_k^\top \mathbf{X}_k$ is directly dependent on $\mathbf{X}_t^\top \mathbf{X}_t$, noting that the residuals and OLS estimators are fixed given that we condition upon them. Since we already have shown that $\mathbf{X}_t^\top \mathbf{X}_t = \mathbf{X'}_t^\top \mathbf{X}'_t$, this means that $\mathbb{E}\left[\left(\mathbf{X}_k^\top \mathbf{X}_k\right)^{-1} \mid C_{k-1}, \mathbf{X}_t\right] = \mathbb{E}\left[\left(\mathbf{X}_k^\top \mathbf{X}_k\right)^{-1} \mid C_{k-1}, \mathbf{X}'_t\right]$ for $k = t+1, \ldots, T$, which completes the proof for statement (c).

Since we have proven all three statements that were presented in the beginning of this proof, we have shown that $p(\mathbf{X}_t = \mathbf{X}_{t-1}\hat{A}_{t-1} + \mathbf{R}_t \mid \mathbf{X}_{t-1}, K) = p(\mathbf{X}'_t = \mathbf{X}_{t-1}\hat{A}_{t-1} - \mathbf{R}_t \mid \mathbf{X}_{t-1}, K)$.

Finally, to show that $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_T, K] = \mathbf{X}_T\hat{\beta}$, we can use the same arguments as before to show $p(\mathbf{Y} = \mathbf{X}_T\hat{\beta} + \mathbf{R}_Y \mid \mathbf{X}_T, K) = p(\mathbf{Y}' = \mathbf{X}_T\hat{\beta} - \mathbf{R}_Y \mid \mathbf{X}_T, K)$, although only statements (a) and (b) are necessary for this case.

□

Remark 2. For the anisotropic case, the analysis becomes slightly different. The noise in the data $\mathbf{X}_t$ is $\boldsymbol{\epsilon}_t = [\epsilon_{t,1}, \ldots \epsilon_{t,n}]^\top \in \mathbb{R}^{n \times d}$. The rows corresponds the the noise in a particular sample, while the columns are for the different features. Furthermore, the covariance of the ith feature is $\mathrm{Cov}(\boldsymbol{\epsilon}_t^{(\text{column } i)}) = \sigma_{t,i}^2 I_n$ for $i = 1, \ldots, d$ where I_n is the n-dimensional identity matrix. With anisotropic noise, we have $\sigma_{t,i} \neq \sigma_{t,i'}$ for $i, i' = 1, \ldots, d$. Then, the above lemma will be feasible using a similar analysis as we can show that,

$$\hat{A}_k^{(\text{row } i)} \sim \mathcal{N}\left(A_k^{(\text{row } i)}, \sigma_{k+1,i}^2 \, \mathbb{E}\left[\left(\mathbf{X}_k^\top \mathbf{X}_k\right)^{-1} \mid C_{k-1}, \mathbf{X}_t\right]\right) .$$

Then, the analysis follows as in Lemma 1.

Now, we prove that $\hat{A}\hat{\beta}$ is a sufficient statistic for $\hat{\theta}$.

Lemma 2. *Let $D = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T, \mathbf{Y})$ be a random dataset from a Gaussian-linear Markov dynamical system, as defined in Assumption 2. Then, let $\hat{\theta}_{\mathrm{LuPTS}} = \hat{A}\hat{\beta}$ be the output of Algorithm 1 without stationarity, and $\hat{\theta}_{\mathrm{OLS}} \coloneqq (\mathbf{X}_1^\top \mathbf{X}_1)^{-1}\mathbf{X}_1^\top \mathbf{Y}$. It holds that,*

$$\mathbb{E}_D[\hat{\theta} \mid \hat{A}, \hat{\beta}] = \hat{A}\hat{\beta} .$$

Proof. Let smaller letters $(\mathbf{x}_1, \ldots, \mathbf{x}_T, \mathbf{y})$ indicate a value of the random dataset D and $B = (\mathbf{x}_1^\top \mathbf{x}_1)^{-1}\mathbf{x}_1^\top$. Then,

we have,

$$\mathbb{E}[\hat{\theta} \mid \hat{A}, \hat{\beta}] = \int p(\mathbf{x}_1, \ldots, \mathbf{x}_T, \mathbf{y} \mid \hat{A}, \hat{\beta}) \hat{\theta} d\mathbf{X}_1 \ldots d\mathbf{X}_T d\mathbf{Y}$$

$$= \int p(\mathbf{y} \mid \mathbf{x}_T, \hat{A}, \hat{\beta}) \prod_{t=2}^{T} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \hat{A}, \hat{\beta}) p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) \hat{\theta} d\mathbf{X}_1 \ldots d\mathbf{X}_T d\mathbf{Y} \quad \text{(Markov property)}$$

$$= \int p(\mathbf{y} \mid \mathbf{x}_T, \hat{A}, \hat{\beta}) \prod_{t=2}^{T} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \hat{A}, \hat{\beta}) p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) \underbrace{B\mathbf{y}}_{=\hat{\theta}} d\mathbf{X}_1 \ldots d\mathbf{X}_T d\mathbf{Y} \quad \text{(OLS definition)}$$

$$= \int \prod_{t=2}^{T} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \hat{A}, \hat{\beta}) p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) B \underbrace{\left[\int \mathbf{y} p(\mathbf{y} \mid \mathbf{x}_T, \hat{A}, \hat{\beta}) d\mathbf{Y} \right]}_{=\mathbb{E}[\mathbf{Y} \mid \mathbf{x}_T, \hat{A}, \hat{\beta}] = \mathbf{x}_T \hat{\beta}} d\mathbf{X}_1 \ldots d\mathbf{X}_T \quad \text{(Lemma 1)}$$

$$= \int \prod_{t=2}^{T} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \hat{A}, \hat{\beta}) p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) B \mathbf{x}_T \hat{\beta} d\mathbf{X}_1 \ldots d\mathbf{X}_T$$

$$= \int \prod_{t=2}^{T-1} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \hat{A}, \hat{\beta}) p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) B \underbrace{\left[\int \mathbf{x}_T p(\mathbf{x}_T \mid \mathbf{x}_{T-1}, \hat{A}, \hat{\beta}) d\mathbf{X}_T \right]}_{=\mathbb{E}[\mathbf{X}_T \mid \mathbf{x}_{T-1}, \hat{A}, \hat{\beta}] = \mathbf{x}_{T-1} \hat{A}_{T-1}} \hat{\beta} d\mathbf{X}_1 \ldots d\mathbf{X}_{T-1} \quad \text{(Lemma 1)}$$

$$= \int \prod_{t=2}^{T-1} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \hat{A}, \hat{\beta}) p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) B \mathbf{x}_{T-1} \hat{A}_{T-1} \hat{\beta} d\mathbf{X}_1 \ldots d\mathbf{X}_{T-1}$$

$$= \int \prod_{t=2}^{T-2} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \hat{A}, \hat{\beta}) p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) B \underbrace{\left[\int \mathbf{x}_{T-1} p(\mathbf{x}_{T-1} \mid \mathbf{x}_{T-2}, K) d\mathbf{X}_{T-1} \right]}_{=\mathbb{E}[\mathbf{X}_{T-1} \mid \mathbf{x}_{T-2}, \hat{A}, \hat{\beta}] = \mathbf{x}_{T-2} \hat{A}_{T-2}} \hat{A}_{T-1} \hat{\beta} d\mathbf{X}_1 \ldots d\mathbf{X}_{T-2} \quad \text{(Lemma 1)}$$

$$= \cdots = \quad \text{(recursively)}$$

$$= \int p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) \underbrace{B\mathbf{x}_1}_{=I} \hat{A}_1 \ldots \hat{A}_{T-1} \hat{\beta} d\mathbf{X}_1$$

$$= \hat{A}_1 \ldots \hat{A}_{T-1} \hat{\beta} \int p(\mathbf{x}_1 \mid \hat{A}, \hat{\beta}) d\mathbf{X}_1 = \hat{A}_1 \ldots \hat{A}_{T-1} \hat{\beta}$$

□

For the final result, recall the definition of parameter mean squared error for an estimate $\hat{\theta}$ of θ, where the expectation is taken over the dataset D used to fit $\hat{\theta}$,

$$\mathrm{MSE}(\hat{\theta}) \coloneqq \mathbb{E}_D[\|\hat{\theta} - \theta\|_2^2]$$

and the expected risk for a prediction function h_D of baseline variables X_1 dependent on the random dataset D,

$$\overline{R}(h_D) \coloneqq \mathbb{E}_D[R(h_D)] = \mathbb{E}_D[\mathbb{E}_{X_1,Y}[(f_D(X_1) - Y)^2]] \ .$$

In the linear case, we let $\overline{R}(\hat{\theta})$ denote $\overline{R}(\hat{\theta}^\top(\cdot))$.

Theorem 1. Let $D = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_T, \mathbf{Y})$ be a random dataset with $\hat{\theta}_{\mathrm{OLS}} \coloneqq (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \mathbf{X}_1^\top \mathbf{Y}$, and let $\hat{\theta}_{\mathrm{LuPTS}} = \hat{A}\hat{\beta}$ be the output of Algorithm 1 without stationarity. Under the Gaussian-linear system defined in Assumption 2 with isotropic noise as in Lemma 1, $\hat{\theta}_{\mathrm{LuPTS}}$ is unbiased, and

$$\mathrm{MSE}(\hat{\theta}_{\mathrm{LuPTS}}) = \mathrm{MSE}(\hat{\theta}_{\mathrm{OLS}}) - \mathbb{E}_D[\mathrm{Tr}(\mathrm{Cov}(\hat{\theta}_{\mathrm{OLS}} \mid \hat{\theta}_{\mathrm{LuPTS}}))] \ , \tag{5}$$

where the expectation is taken over random datasets D, since both estimators are functions of them. Further, it holds for the expected risk that over new, unseen samples (X_1, Y),

$$\overline{R}(\hat{\theta}_{\mathrm{LuPTS}}) = \overline{R}(\hat{\theta}_{\mathrm{OLS}}) - \mathbb{E}_{D,X_1}[\mathrm{Var}_{\hat{\theta}_{\mathrm{OLS}}}(\langle \hat{\theta}_{\mathrm{OLS}}, X_1 \rangle \mid \hat{\theta}_{\mathrm{LuPTS}})] \ . \tag{6}$$

Proof. Unbiasedness of $\hat{\theta}_{\text{LuPTS}}$ follows from Lemma 2 or the standard proof for unbiasedness of $\hat{\theta}_{\text{OLS}}$. The remaining result follows from Lemma 2 and standard Rao-Blackwell arguments,

$$\begin{aligned}
\text{MSE}(\hat{A}\hat{\beta}) &= \mathbb{E}_D[\|\hat{A}\hat{\beta} - \theta\|^2] \\
&= \mathbb{E}_D[\|\mathbb{E}[\hat{\theta} \mid \hat{A}\hat{\beta}] - \theta\|^2] \quad \text{(Lemma 2)} \\
&= \mathbb{E}_D[\|\mathbb{E}[\hat{\theta} - \theta \mid \hat{A}\hat{\beta}]\|^2] \\
&= \mathbb{E}_D\left[\sum_{j=1}^{d}(\mathbb{E}[\hat{\theta}_j - \theta_j \mid \hat{A}\hat{\beta}])^2\right] \\
&= \mathbb{E}_D\left[\sum_{j=1}^{d}\left(\mathbb{E}[(\hat{\theta}_j - \theta_j)^2 \mid \hat{A}\hat{\beta}] - \text{Var}\left[\hat{\theta}_j \mid \hat{A}\hat{\beta}\right]\right)\right] \\
&= \mathbb{E}_D[\mathbb{E}[\|\hat{\theta} - \theta\|^2 \mid \hat{A}\hat{\beta}]] - \mathbb{E}_D\left[\sum_{j=1}^{d}\text{Var}\left[\hat{\theta}_j \mid \hat{A}\hat{\beta}\right]\right] \\
&= \text{MSE}(\hat{\theta}) - \mathbb{E}_D\left[\text{Tr}\left(\text{Cov}\left[\hat{\theta} \mid \hat{A}\hat{\beta}\right]\right)\right] \ .
\end{aligned}$$

Recall that X_1 represents a new test point, independent of the dataset D. For the second result, we note that for any estimator $\hat{\theta}$,

$$\mathbb{E}_D[R(\hat{\theta})] = \mathbb{E}_D[\mathbb{E}_{X_1,Y}[(\hat{\theta}^\top X_1 - Y)^2]] = \mathbb{E}_{X_1}[\mathbb{E}_D[\mathbb{E}_{Y|X_1}(\hat{\theta}^\top X_1 - Y)^2 \mid X_1]] \ .$$

Then, if $\hat{\theta}$ is unbiased for the Gaussian linear model, $Y = \theta^\top X_1 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$,

$$\mathbb{E}_D[\mathbb{E}_{Y|X_1}[(\hat{\theta}^\top x_1 - Y)^2 \mid X_1 = x_1]] = \underbrace{\mathbb{E}_D[(\hat{\theta}^\top x_1 - \mathbb{E}_D[\hat{\theta}]^\top x_1)^2]}_{=\text{variance}} + \underbrace{(\mathbb{E}_D[\hat{\theta}]^\top x_1 - \theta^\top x_1)^2}_{=\text{bias}^2=0} + \sigma^2 \ .$$

Since $\mathbb{E}_D[\hat{\theta}] = \theta$, the variance term can then be rewritten as,

$$\mathbb{E}_D[(\hat{\theta}^\top x_1 - \mathbb{E}_D[\hat{\theta}]^\top x_1)^2] = \mathbb{E}_D[\langle \hat{\theta} - \theta, x_1\rangle^2] \ .$$

Then, since $\hat{A}\hat{\beta}$ is an unbiased estimator of θ,

$$\begin{aligned}
\mathbb{E}_D[R(\hat{A}\hat{\beta})] &= \mathbb{E}_{X_1}[\mathbb{E}_D[\langle \hat{A}\hat{\beta} - \theta, X_1\rangle^2]] + \sigma^2 \\
&= \mathbb{E}_{X_1}[\mathbb{E}_D[\langle \mathbb{E}_{\hat{\theta}}[\hat{\theta} \mid \hat{A}\hat{\beta}] - \theta, X_1\rangle^2]] + \sigma^2 \quad \text{(Lemma 2)} \\
&= \mathbb{E}_{X_1}[\mathbb{E}_D[\mathbb{E}_{\hat{\theta}}[\langle \hat{\theta} - \theta, X_1\rangle \mid \hat{A}\hat{\beta}]^2]] + \sigma^2 \\
&= \mathbb{E}_{X_1}[\mathbb{E}_D[\mathbb{E}_{\hat{\theta}}[\langle \hat{\theta} - \theta, X_1\rangle^2 \mid \hat{A}\hat{\beta}] - \text{Var}_{\hat{\theta}}(\langle \hat{\theta} - \theta, X_1\rangle \mid \hat{A}\hat{\beta})]] + \sigma^2 \\
&= \mathbb{E}_D[R(\hat{\theta})] - \mathbb{E}_{D,X_1}[\text{Var}_{\hat{\theta}}(\langle \hat{\theta} - \theta, X_1\rangle \mid \hat{A}\hat{\beta})] \ .
\end{aligned}$$

In the last step, we make use of the fact that $\hat{\theta}$ is unbiased and

$$\mathbb{E}_D[R(\hat{\theta})] = \mathbb{E}_{X_1}[\mathbb{E}_D[\langle \hat{\theta} - \theta, X_1\rangle^2] + \sigma^2 \ .$$

□

B PROOF OF THEOREM 2

We extended the distillation-based method as described by Hayashi et al. (2019) with LuPTS as teacher model, which we called Distill-Seq. In the linear setting with squared loss, the distillation loss function is defined as,

$$\hat{\theta}_{Dist} = \arg\min_{\theta} \lambda||\mathbf{Y} - \mathbf{X}_1\theta||_2^2 + (1-\lambda)||\mathbf{Y}_{soft} - \mathbf{X}_1\theta||_2^2 \tag{7}$$

where $\lambda \in [0, 1]$ and $\mathbf{Y}_{soft}$ is the soft target provided by the teacher. In the case where a student model minimizes the above loss function with LuPTS as teacher model, we can prove the following theorem.

Theorem 2. Let $\hat{\theta}_{\text{LuPTS}}$ be the output of Algorithm 1 and $\hat{\theta}_{\text{OLS}} = (\mathbf{X}_1^\top \mathbf{X}_1)^{-1}\mathbf{X}_1^\top \mathbf{Y}$. Let $\hat{\theta}_{Dist}$ be the solution to (7) with $\hat{\mathbf{Y}}_{soft} = \mathbf{X}_1\hat{\theta}_{\text{LuPTS}}$ and $\lambda \in [0,1]$. Then, it holds that

$$\hat{\theta}_{Dist} = \lambda\hat{\theta}_{\text{OLS}} + (1-\lambda)\hat{\theta}_{\text{LuPTS}} \ . \tag{8}$$

Additionally, under Assumption 2, it holds that

$$\text{MSE}(\hat{\theta}_{\text{LuPTS}}) \leq \text{MSE}(\hat{\theta}_{Dist}) \leq \text{MSE}(\hat{\theta}_{\text{OLS}}) \ . \tag{9}$$

Proof. We will first show the first part of the theorem, namely that equation (8) holds. Then, as a consequence, we will proceed with proving that equation (9) holds.

Since the optimization problem in (7) is convex, we can compute the derivative with respect to θ and find the value for which the derivative is zero,

$$\begin{aligned}
\frac{d}{d\theta} &= \left(\lambda||\mathbf{Y} - \mathbf{X}_1\theta||_2^2 + (1-\lambda)||\mathbf{X}_1\hat{\theta}_{\text{LuPTS}} - \mathbf{X}_1\theta||_2^2\right) \\
&= \left(2\lambda\mathbf{X}_1^\top(\mathbf{Y} - \mathbf{X}_1\theta) + 2(1-\lambda)\mathbf{X}_1^\top(\mathbf{X}_1\hat{\theta}_{\text{LuPTS}} - \mathbf{X}_1\theta)\right) \\
&= 2\mathbf{X}_1^\top\left(\underbrace{\lambda\mathbf{Y} + (1-\lambda)\mathbf{X}_1\hat{\theta}_{\text{LuPTS}}}_{\tilde{\mathbf{Y}}} - \mathbf{X}_1\theta\right) = 0 \ .
\end{aligned}$$

The solution is given by the OLS estimate

$$\hat{\theta}_{Dist} = (\mathbf{X}_1^\top\mathbf{X}_1)^{-1}\mathbf{X}_1^\top\tilde{\mathbf{Y}}$$

where we can expand $\tilde{\mathbf{Y}}$ to get the following,

$$\begin{aligned}
\hat{\theta}_{Dist} &= (\mathbf{X}_1^\top\mathbf{X}_1)^{-1}\mathbf{X}_1^\top\left(\lambda\mathbf{Y} + (1-\lambda)\mathbf{X}_1\hat{\theta}_{\text{LuPTS}}\right) \\
&= \lambda(\mathbf{X}_1^\top\mathbf{X}_1)^{-1}\mathbf{X}_1^\top\mathbf{Y} + (1-\lambda)(\mathbf{X}_1^\top\mathbf{X}_1)^{-1}\mathbf{X}_1^\top\mathbf{X}_1\hat{\theta}_{\text{LuPTS}} \\
&= \lambda\hat{\theta}_{\text{OLS}} + (1-\lambda)\hat{\theta}_{\text{LuPTS}} \ .
\end{aligned}$$

This proves the first part of the theorem. Next, we prove equation (9), which will be done element-wise, i.e. we prove the statement for $\hat{\theta}_{Dist}^{(j)}$ for some $j = 1, \ldots, d$.

First, due to equation (7), we note that since $\hat{\theta}_{\text{OLS}}$ and $\hat{\theta}_{\text{LuPTS}}$ are unbiased estimators, $\hat{\theta}_{Dist}$ is also unbiased. Hence, we can write

$$\text{MSE}(\hat{\theta}_{Dist}^{(j)}) = \text{Var}(\hat{\theta}_{Dist}^{(j)}) + \underbrace{\text{Bias}(\hat{\theta}_{Dist}^{(j)})^2}_{=0} = \text{Var}(\hat{\theta}_{Dist}^{(j)})$$

Then, we use equation (8) again to rewrite the variance of $\hat{\theta}_{Dist}^{(j)}$,

$$\begin{aligned}
\text{Var}(\hat{\theta}_{Dist}^{(j)}) &= \text{Var}(\lambda\hat{\theta}_{\text{OLS}}^{(j)} + (1-\lambda)\hat{\theta}_{\text{LuPTS}}^{(j)}) \\
&= \lambda^2\text{Var}(\hat{\theta}_{\text{OLS}}^{(j)}) + (1-\lambda)^2\text{Var}(\hat{\theta}_{\text{LuPTS}}^{(j)}) + 2\lambda(1-\lambda)\text{Cov}(\hat{\theta}_{\text{OLS}}^{(j)}, \hat{\theta}_{\text{LuPTS}}^{(j)}) \ .
\end{aligned}$$

We shall focus on the covariance term, and using the law of total covariance we can show the following,

$$\begin{aligned}
\text{Cov}(\hat{\theta}_{\text{OLS}}^{(j)}, \hat{\theta}_{\text{LuPTS}}^{(j)}) =& \mathbb{E}_{\hat{\theta}_{\text{LuPTS}}^{(j)}}\left[\text{Cov}(\hat{\theta}_{\text{OLS}}^{(j)}, \hat{\theta}_{\text{LuPTS}}^{(j)} \mid \hat{\theta}_{\text{LuPTS}}^{(j)})\right] \\
&+ \text{Cov}_{\hat{\theta}_{\text{LuPTS}}^{(j)}}\left(\mathbb{E}[\hat{\theta}_{\text{OLS}}^{(j)} \mid \hat{\theta}_{\text{LuPTS}}^{(j)}], \mathbb{E}[\hat{\theta}_{\text{LuPTS}}^{(j)} \mid \hat{\theta}_{\text{LuPTS}}^{(j)}]\right) \\
=& \mathbb{E}_{\hat{\theta}_{\text{LuPTS}}^{(j)}}\left[\mathbb{E}\left[(\hat{\theta}_{\text{OLS}}^{(j)} - \mathbb{E}[\hat{\theta}_{\text{OLS}}^{(j)} \mid \hat{\theta}_{\text{LuPTS}}^{(j)}])\underbrace{(\hat{\theta}_{\text{LuPTS}}^{(j)} - \mathbb{E}[\hat{\theta}_{\text{LuPTS}}^{(j)} \mid \hat{\theta}_{\text{LuPTS}}^{(j)}])}_{=0}\right] \mid \hat{\theta}_{\text{LuPTS}}^{(j)})\right] \\
&+ \text{Cov}_{\hat{\theta}_{\text{LuPTS}}^{(j)}}\left(\hat{\theta}_{\text{LuPTS}}^{(j)}, \hat{\theta}_{\text{LuPTS}}^{(j)}\right) \qquad \text{by Lemma 2 and definition of covariance} \\
=& \text{Var}(\hat{\theta}_{\text{LuPTS}}^{(j)}) \ .
\end{aligned}$$

Algorithm 2: Learning using privileged time series (LuPTS)

Parameters: Function classes $\mathcal{F}$, $\mathcal{G}$, loss function L
Data: $D = \{(x_{1,1}, ..., x_{1,T}, y_1), ..., (x_{m,1}, ..., x_{m,T}, y_m)\} \sim p^m(X_1, ..., X_T, Y)$

for $t = 1, ..., T-1$ **do**
  $\hat{f}_t = \arg\min_{f_t \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} L(f_t(x_{i,t}), x_{i,t+1})$
$\hat{g} = \arg\min_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} L(g(x_{i,T}), y_i)$

return $h = \hat{g} \circ \hat{f}_{T-1} \circ ... \circ \hat{f}_1$

Hence, we end up with

$$\begin{aligned}
\mathrm{Var}(\hat{\theta}^{(j)}_{Dist}) &= \lambda^2 \mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{OLS}}) + (1-\lambda)^2 \mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{LuPTS}}) + 2\lambda(1-\lambda)\mathrm{Cov}(\hat{\theta}^{(j)}_{\mathrm{OLS}}, \hat{\theta}^{(j)}_{\mathrm{LuPTS}}) \\
&= \lambda^2 \mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{OLS}}) + (1-\lambda)^2 \mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{LuPTS}}) + 2\lambda(1-\lambda)\mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{LuPTS}}) \\
&= \lambda^2 \mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{OLS}}) + (1-\lambda^2)\mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{LuPTS}}) \ .
\end{aligned}$$

Looking at the last line of the previous equation, we know from Theorem 1 that $\mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{LuPTS}}) \leq \mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{OLS}})$. Hence, the lower bound of $\mathrm{Var}(\hat{\theta}^{(j)}_{Dist})$ is obtained by setting $\lambda = 0$ and, similarly, the upper bound is obtained when $\lambda = 1$. This is possible since we can choose $\lambda \in [0, 1]$ freely. This leads to the following results,

$$\mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{LuPTS}}) \leq \mathrm{Var}(\hat{\theta}^{(j)}_{Dist}) \leq \mathrm{Var}(\hat{\theta}^{(j)}_{\mathrm{OLS}})$$

which, due to the unbiasedness of the estimators, can be written as

$$\mathrm{MSE}(\hat{\theta}^{(j)}_{\mathrm{LuPTS}}) \leq \mathrm{MSE}(\hat{\theta}^{(j)}_{Dist}) \leq \mathrm{MSE}(\hat{\theta}^{(j)}_{\mathrm{OLS}}) \ .$$

Lastly, since $\mathrm{MSE}(\hat{\theta}_{\mathrm{LuPTS}}) = \sum_{j=1}^{d} \mathrm{MSE}(\hat{\theta}^{(j)}_{\mathrm{LuPTS}})$ and that the inequality holds element-wise, we have that,

$$\mathrm{MSE}(\hat{\theta}_{\mathrm{LuPTS}}) \leq \mathrm{MSE}(\hat{\theta}_{Dist}) \leq \mathrm{MSE}(\hat{\theta}_{\mathrm{OLS}}) \ .$$

□

C LuPTS WITH NON-LINEAR ESTIMATORS

Under Assumption 1 (Markovianity), it is natural to consider the following generalized (non-linear) procedure: a) For each time-step t, fit a transition function f_t predicting X_{t+1} from X_t, b) Fit g to predict Y from X_T, c) Return $h = g \circ f_{T-1} \circ \cdots \circ f_1$. This approach outlined in Algorithm 2. The idea may be compared to model-based value estimates in reinforcement learning (Sutton and Barto, 2018), in which predictions of future rewards are based on simulating roll-outs under a learned policy and model of state dynamics.

In the general case, without assumptions on the data-generating process or the hypothesis classes $\mathcal{F}$ and $\mathcal{G}$, we may bound the expected risk of the LuPTS estimator in terms of the risk accumulated in simulating the system dynamics through f, and that of the outcome model g.

Theorem 3 (Risk expansion)**.** *Let $\hat{h} = \hat{g} \circ \hat{f}$ be the output of Algorithm 2 with $\hat{f} = \hat{f}_{T-1} \circ \hat{f}_{T-1} \circ \cdots \circ \hat{f}_1$ the estimated system dynamics and $\hat{g}$ the prediction model of Y from X_T. Then,*

$$R(\hat{f} \circ \hat{g}) \leq R_{X_T}(\hat{f}) + R_Y(\hat{g}) + 2\sqrt{R_{X_T}(\hat{f}) R_Y(\hat{g})} \tag{10}$$

where $R(\hat{f} \circ \hat{g}) = \mathbb{E}[(Y - \hat{g}(\hat{f}(X_1)))^2]$ is the expected risk of predicting Y from X_1,

$$R_{X_T}(f) = \mathbb{E}\left[\left(\hat{g}(\hat{f}(X_1)) - \hat{g}(X_T)\right)^2\right] \quad \textit{and} \quad R_Y(\hat{g}) = \mathbb{E}\left[(Y - \hat{g}(X_T))^2\right] \ .$$

Here, $R_{X_T}(f)$ is the mean squared error in predictions of Y that stems from errors in the learned dynamical system while $R_Y(\hat{g})$ is due to the error in the outcome model $\hat{g}$.

Proof. As seen in (Feinberg et al., 2018).

$$\begin{aligned}
\mathbb{E}\left[\left(\hat{g}(\hat{f}(x_1)) - y\right)^2\right] =& \mathbb{E}\left[\left(\hat{g}(\hat{f}(x_1)) - y + \hat{g}(x_T) - \hat{g}(x_T)\right)^2\right] \\
=& \mathbb{E}\left[\left((\hat{g}(\hat{f}(x_1)) - \hat{g}(x_T)) - (y - \hat{g}(x_T))\right)^2\right] \\
=& \mathbb{E}\left[\left(\hat{g}(\hat{f}(x_1)) - \hat{g}(x_T)\right)^2\right] + \mathbb{E}\left[(y - \hat{g}(x_T))^2\right] \\
& - 2\mathbb{E}\left[\left(\hat{g}(\hat{f}(x_1)) - \hat{g}(x_T)\right)(y - \hat{g}(x_T))\right] \\
& \leq \mathbb{E}\left[\left(\hat{g}(\hat{f}(x_1)) - \hat{g}(x_T)\right)^2\right] + \mathbb{E}\left[(y - \hat{g}(x_T))^2\right] \\
& + 2\sqrt{\mathbb{E}\left[\left(\hat{g}(\hat{f}(x_1)) - \hat{g}(x_T)\right)^2\right]\mathbb{E}\left[(y - \hat{g}(x_T))^2\right]}
\end{aligned} \tag{11}$$

The first equalities are algebra, and the inequality step comes from the Cauchy-Schwarz inequality for random variables. □

Remark 3. A similar result appears in Feinberg et al. (2018) for the case of model-based value expansion for model-free reinforcement learning. Although the bound cannot be compared directly to the risk of the baseline method, it gives an indication about how the LuPTS algorithm behaves. $R_{X_T}(f)$ can be expected to increase as T becomes larger since X_T gets "further away" from X_1, making X_T more difficult to predict. Meanwhile, $R_Y(\hat{g})$ is unaffected by this.

D EXPERIMENT DETAILS

D.1 Computational Resources

All procedures for pre-processing real-world datasets, generating synthetic datasets, training and evaluating models are implemented in Python with the help of standard scientific modules such as NumPy and scikit-learn. The experiments are run on mid-tier laptops generally utilizing one CPU core. Running times for each individual experiment under this setup rarely exceed a couple of minutes. The full set of experiments can be reproduced in less than 48 hours.

D.2 Synthetic Experiments

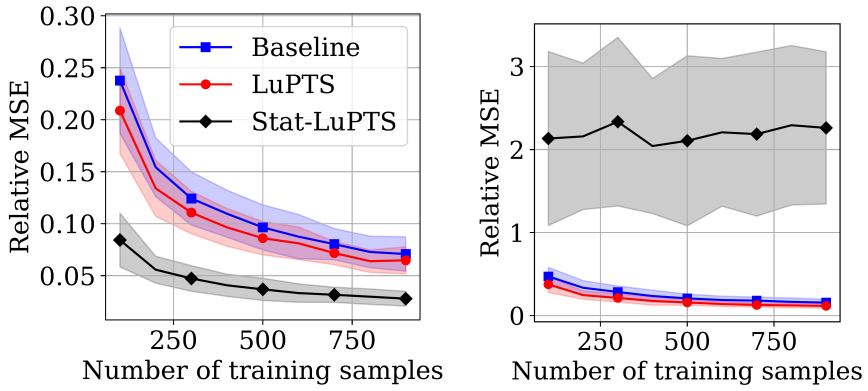
We give a detailed description of how the synthetic data we use in the experiments is generated. As a reminder, the Gaussian-linear dynamical system of interest is

$$\begin{aligned}
X_t &= A_{t-1}^\top X_{t-1} + \epsilon_t, \quad \text{for } t = 2, \ldots, T \\
Y &= \beta^\top X_T + \epsilon_Y \ .
\end{aligned}$$

To verify and further investigate our theoretical results, we sample from a synthetic dynamical system where Markovianity and linearity with additive isotropic Gaussian noise hold. The parameters $A_t \in \mathbb{R}^{d\times d}$ and $\beta \in \mathbb{R}^{d\times 1}$ were generated in the following way: For each $t = 1, \ldots, T-1$, all elements in A_t are sampled independently from a Normal distribution $\mathcal{N}(\mu = 0, \sigma = 0.2)$, except for the diagonal elements of A_t which were set to 1. The linear parameter for the outcome model β was sampled from the same distribution, i.e. $\beta_j \sim \mathcal{N}(\mu = 0, \sigma = 0.2)$ for $j = 1, \ldots, d$.

As mentioned, the eigenvalues of A_t influence the system's behavior and stability, hence we enforce the spectral radius $\forall t : \rho(A_t) = \kappa = 1.5$ for all t for the experiments in Section 4.2. This is by factorizing A_t into its spectral decomposition $U_t \Lambda_t U_t^{-1}$ and computing $\Lambda_t^{(new)} = \frac{\kappa}{\rho(A_t)}\Lambda_t$. Then, an update $A_t = U_t \Lambda_t^{(new)} U_t^{-1}$ is performed where κ becomes the new spectral radius of A_t.

For all experiments, we use the following default values unless otherwise stated: $\kappa = 1.5$, $n = 1000$, $T = 10$, $d = 25$, and $\mathrm{Var}(\epsilon_t) = \mathrm{Var}(\epsilon_Y) = 1$ for $t = 1, \ldots, T-1$. Finally, the input distribution is $p(X_1) = \mathcal{N}(\mu = 0, \sigma^2 = 5)$.



(a) Stationarity is true. (b) Stationarity is violated.

Figure 6: Parameter recovery with or without stationarity when varying the number of training samples n. R^2 used as metric; shaded region corresponds to one standard deviation over 200 iterations.

Additional Experiments: Testing Stationary Systems

For the experiments in Section 4.2, we solely consider synthetic systems with time-dependent transition A_t. We also include an experiment for systems where the transitions are stationary, that is $A_t = A_{t'}, t, t' = 1, \ldots, T-1$. The generation process is identical with the exception that only one transition matrix is sampled, A. In the case with more than one privileged time point, this enables us to evaluate the potential benefits that using Stat-LuPTS (instead of the non-stationary variant) have in a setting where the assumptions holds true.

When the stationary assumptions is true (Figure 6a), LuPTS does better than baseline, as before. More importantly, Stat-LuPTS is closer to the true parameter estimate than both of them. Meanwhile, as expected, when breaking the stationary assumption (Figure 6b), Stat-LuPTS performs significantly worse while LuPTS and baseline remain about the same. These experiments indicate that the stationary variant of LuPTS is preferable when the stationarity assumption is true.

D.3 Forecasting Air Quality

Implementation Of Distillation Methods We implement the distillation models and loss function, as described in Section 3.2, in PyTorch v1.7. The loss function is optimized using Adam (Kingma and Ba, 2014), and the models are trained for 200 epochs. Error bars on all our plots are generated by training and evaluating on different train/test splits over 100 iterations.

Pre-processing Due to the prevalence of missing values for the $PM_{2.5}$ concentration levels in the dataset, the first pre-processing step is to extract all non-overlapping sequences of length T that have no missing values for the $PM_{2.5}$ concentration. In addition, we enforce a rule that there must be at least a gap of six hours between adjacent sequences to decrease correlations between them. Finally, dummy encoding was used for the categorical features in the dataset.

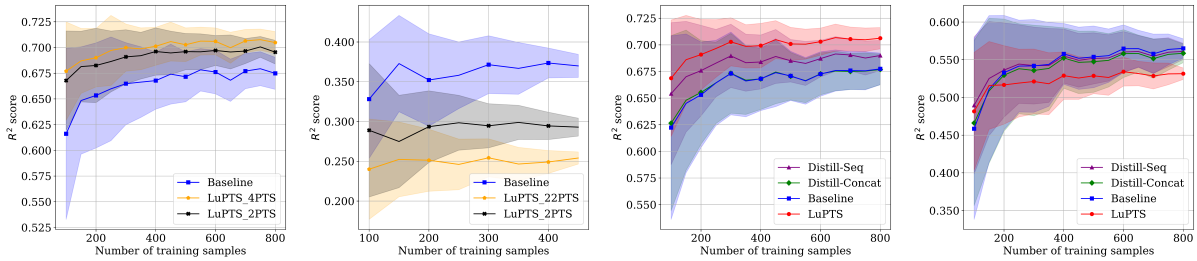
Evaluation During training, the dataset was split into a training and test set portion consisting of 80% and 20% of the data respectively. The training procedure on the dataset for the forecasting task is the following: We vary the number of training samples, and for each sample size, data points are randomly sampled from the training set without replacement. Then, before training the algorithms on this set, we apply zero-mean unit-variance standardization and mean imputation where applicable. Each algorithm is then evaluated after training on a held-out test set, which is the same for every run, and the corresponding R^2 score is noted. This process is iterated 200 times per sample size.

Additional Experiments

In this section, we show additional experiments performed on the air quality forecasting task.

Table 2: Features in the $PM_{2.5}$ dataset.

Feature	Type
Temperature	Numerical
Humidity	Numerial
Dew Point	Numerical
Pressure	Numerical
Cumulated wind speed	Numerical
PM2.5 concentration	Numerical
Season	Categorical
Combined wind direction	Categorical



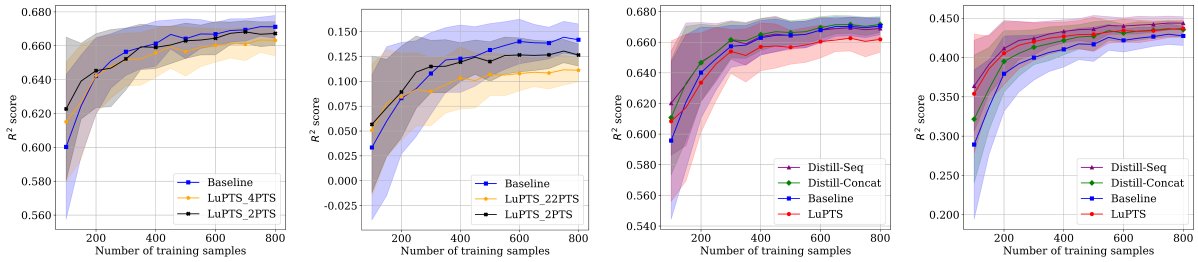
(a) 6 hour forecast (b) 24 hour forecast (c) 6 hour forecast (d) 12 hour forecast

Figure 7: **Shenyang:** 7a, 7b) Changing the amount of privileged information for the LUPTS for different time horizons, where the X in LuPTS_XPTS indicates the number of privileged time points. 7c, 7d) Comparing LuPTS to the distillation-based approaches, which use the same privileged information. Metric used is R^2 (Higher is better); shaded region indicates one standard deviation across 75 iterations.

More Cities We present the same results as shown in Section 4.3 for all of the Chinese cities in the dataset; Shenyang, Beijing, Chengdu, Shanghai and Guangzhou. These are shown in Figure 7, 8, 9, 10 and 11, respectively.

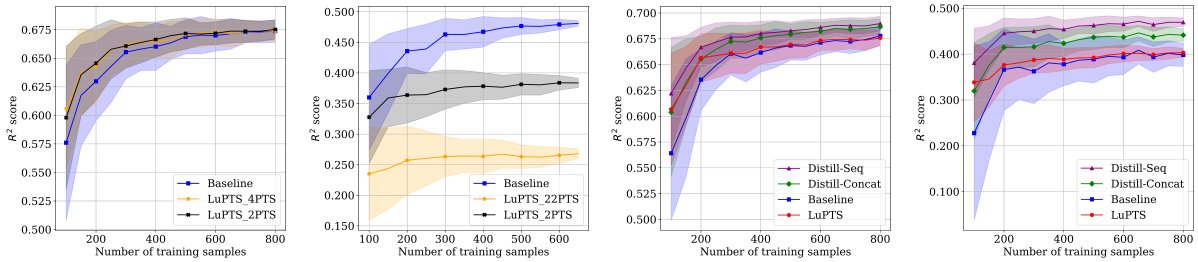
Comparison To Non-linear Baselines In addition to the distillation-based baselines in the main paper, we compare LuPTS to non-linear baselines in the form of random forest (RF) and k-nearest neighbors regression (KNN). These can be found in Table 3 and Table 4 where we have a fixed sample size $n = 200$ and a prediction horizon of either 6 or 12 hours. First, we describe the implementation details for RF and KNN regression. We use the RandomForestRegressor and KNeighborsRegressor implementations of the Python module scikit-learn Pedregosa et al. (2011). The model parameters are tuned using randomized search with 2-fold cross validation. For the RandomForestRegressor we tune the number of trees, max depth of the trees, whether bootstrap sampling is used, the minimum number of samples required to split a node and the minimum number of samples required to be at a leaf node. For the KNeighborsRegressor we tune the number of neighbors used, weight function used in prediction, the size of the leaves and the power parameter for the Minkowski metric ($p = 1$ or $p = 2$). The specific ranges for each parameter can be found in the attached code to this paper.

For the 6 hour predictions (see Table 3), we see that all linear methods (Baseline, LuPTS variants and distillation-based variants) perform better than both RF and KNN for all cities, although the gap between LuPTS and RF is relatively small for Beijing and Guangzhou in particular. For the 12 hour predictions (see Table 4), the results look similar except for Guangzhou where RF performs the best among all the methods. A possible explanation for why the non-linear methods in almost all cases perform worse could be due to the low-sample regime which is not as suitable for their larger flexibility. In particular, a benefit of the linear methods is that they either do not need model parameter tuning at all or to a smaller degree in comparison to the non-linear methods.



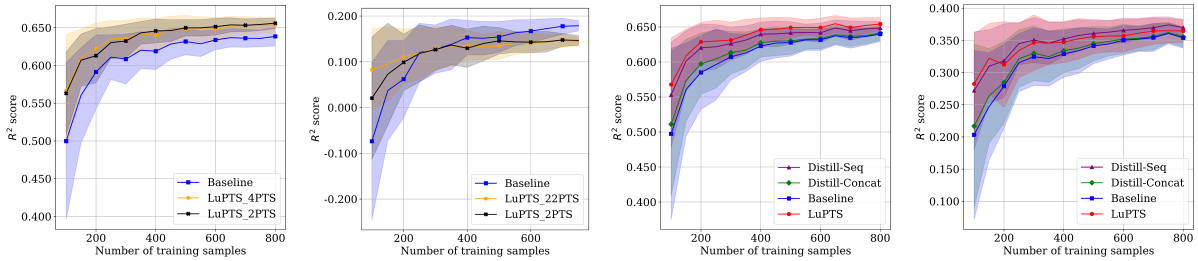
(a) 6 hour forecast (b) 24 hour forecast (c) 6 hour forecast (d) 12 hour forecast

Figure 8: **Beijing:** 8a, 8b) Changing the amount of privileged information for the LUPTS for different time horizons, where the X in LuPTS_XPTS indicates the number of privileged time points. 8c, 8d) Comparing LuPTS to the distillation-based approaches, which use the same privileged information. Metric used is R^2 (Higher is better); shaded region indicates one standard deviation across 75 iterations.



(a) 6 hour forecast (b) 24 hour forecast (c) 6 hour forecast (d) 12 hour forecast

Figure 9: **Chengdu:** 9a, 9b) Changing the amount of privileged information for the LUPTS for different time horizons, where the X in LuPTS_XPTS indicates the number of privileged time points. 9c, 9d) Comparing LuPTS to the distillation-based approaches, which use the same privileged information. Metric used is R^2 (Higher is better); shaded region indicates one standard deviation across 75 iterations.



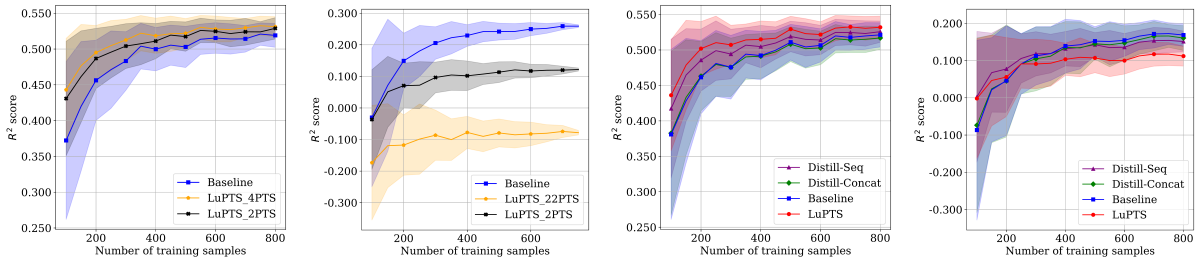
(a) 6 hour forecast (b) 24 hour forecast (c) 6 hour forecast (d) 12 hour forecast

Figure 10: **Shanghai:** 10a, 10b) Changing the amount of privileged information for the LUPTS for different time horizons, where the X in LuPTS_XPTS indicates the number of privileged time points. 10c, 10d) Comparing LuPTS to the distillation-based approaches, which use the same privileged information. Metric used is R^2 (Higher is better); shaded region indicates one standard deviation across 75 iterations.

D.4 Alzheimer's Progression Modeling

In this section, we present the entire feature set used for the Alzheimer's disease progression modeling tasks (see Table 5). All experimental results are also found in tabular form with values rounded to two decimals (see Tables 6 and 7). Lastly, we give a detailed description of the data pre-processing that was performed for the dataset (ADNIMERGE).

Pre-processing There are a significant number of missing values in the observations from the ADNI dataset. The missingness varies with the time of measurement, as does which subjects are present at certain follow-ups.



(a) 6 hour forecast (b) 24 hour forecast (c) 6 hour forecast (d) 12 hour forecast

Figure 11: **Guangzhou:** 11a, 11b) Changing the amount of privileged information for the LUPTS for different time horizons, where the X in LuPTS_XPTS indicates the number of privileged time points. 11c, 11d) Comparing LuPTS to the distillation-based approaches, which use the same privileged information. Metric used is R^2 (Higher is better); shaded region indicates one standard deviation across 75 iterations.

Table 3: Comparison of regression methods on the air quality forecasting task with a fixed sample size $n = 200$ and a prediction horizon of 6 hours. Metric used is R^2 (Higher is better); mean value for each method with standard deviation across 200 iterations. The methods with highest R^2 and lowest variance are marked in bold for each city.

Method	Beijing	Shanghai	Shenyang	Chengdu	Guangzhou
Baseline	0.64 ± 0.02	0.58 ± 0.06	0.66 ± 0.04	0.63 ± 0.04	0.45 ± 0.06
LuPTS	0.64 ± 0.03	**0.62 ± 0.03**	**0.70 ± 0.03**	0.65 ± 0.03	0.49 ± 0.04
Stat-LuPTS	0.64 ± 0.02	**0.62 ± 0.04**	0.69 ± 0.03	0.65 ± 0.03	**0.50 ± 0.04**
Distill-Seq	**0.65 ± 0.02**	**0.62 ± 0.03**	0.68 ± 0.03	**0.67 ± 0.02**	0.49 ± 0.05
Distill-Concat	**0.65 ± 0.02**	0.60 ± 0.04	0.66 ± 0.04	0.66 ± 0.03	0.46 ± 0.07
RF	0.62 ± 0.03	0.58 ± 0.07	0.53 ± 0.06	0.61 ± 0.04	0.48 ± 0.05
KNN	0.57 ± 0.04	0.51 ± 0.23	0.49 ± 0.05	0.51 ± 0.04	0.26 ± 0.09

Table 4: Comparison of regression methods on the air quality forecasting task with a fixed sample size $n = 200$ and a prediction horizon of 12 hours. Metric used is R^2 (Higher is better); mean value for each method with standard deviation across 200 iterations. The methods with highest R^2 and lowest variance are marked in bold for each city.

Method	Beijing	Shanghai	Shenyang	Chengdu	Guangzhou
Baseline	0.37 ± 0.05	0.29 ± 0.06	**0.53 ± 0.07**	0.35 ± 0.08	0.07 ± 0.11
LuPTS	0.40 ± 0.04	**0.33 ± 0.04**	0.51 ± 0.07	0.42 ± 0.04	0.05 ± 0.14
Stat-LuPTS	0.40 ± 0.04	**0.33 ± 0.04**	0.51 ± 0.07	0.42 ± 0.04	0.05 ± 0.14
Distill-Seq	**0.41 ± 0.03**	0.31 ± 0.04	0.52 ± 0.08	**0.43 ± 0.04**	0.07 ± 0.11
Distill-Concat	**0.41 ± 0.03**	0.31 ± 0.04	0.52 ± 0.08	**0.43 ± 0.04**	0.07 ± 0.11
RF	0.36 ± 0.05	0.23 ± 0.08	0.40 ± 0.07	0.38 ± 0.06	**0.14 ± 0.10**
KNN	0.30 ± 0.05	0.24 ± 0.07	0.35 ± 0.06	0.31 ± 0.06	-0.05 ± 0.12

Hence, a subset of the subjects in the study needs to be selected in order to carry out the experiments. This means that subjects without an observation of the target outcome at the follow-up at 48 months are excluded. Furthermore, it is required that the subjects with an observation of the target at this time point also are present at the intermediate follow-ups used as privileged information, which are 12 months, 24 months and 36 months after baseline. Categorical features, here considered to consist of biological sex (PTGENDER) and APOE4 gene expression, are one-hot encoded. Additionally, if any feature has more than 70% of the observations missing for the selected subjects at any of the time points in consideration, they are excluded. The features excluded as a result of this constraint are FDG, ABETA, TAU and PTAU. Finally, mean imputation is used for missing values and the data is zero-mean unit-variance standardized.

Table 5: Features used for the ADNI experiments

Feature tags		
AGE	PTGENDER	PTEDUCAT
APOE4	FDG	AV45
ABETA	TAU	PTAU
CDRSB	ADAS11	ADAS13
ADASQ4	MMSE	RAVLT_immediate
RAVLT_learning	RAVLT_forgetting	RAVLT_perc_forgetting
LDELTOTAL	TRABSCOR	FAQ
MOCA	EcogPtMem	EcogPtLang
EcogPtVisspat	EcogPtPlan	EcogPtOrgan
EcogPtDivatt	EcogPtTotal	EcogSPMem
EcogSPLang	EcogSPVisspat	EcogSPPlan
EcogSPOrgan	EcogSPDivatt	EcogSPTotal
Ventricles	Hippocampus	WholeBrain
Entorhinal	Fusiform	MidTemp
ICV		

Table 6: MMSE prediction experiment results, average R^2 score with one standard deviation in parenthesis from 100 iterations. **Left**: One privileged time point used. **Right**: three privileged time points used.

Samples	Baseline	LuPTS
80	-0.02 (0.34)	0.14 (0.24)
100	0.12 (0.29)	0.27 (0.17)
120	0.27 (0.14)	0.36 (0.1)
140	0.36 (0.08)	0.42 (0.06)
160	0.39 (0.08)	0.44 (0.05)
180	0.42 (0.06)	0.46 (0.05)
200	0.44 (0.05)	0.48 (0.04)
220	0.45 (0.05)	0.49 (0.04)

Samples	Baseline	LuPTS	Stat-LuPTS
80	-0.02 (0.34)	0.25 (0.16)	0.39 (0.09)
100	0.12 (0.29)	0.34 (0.11)	0.43 (0.06)
120	0.27 (0.14)	0.41 (0.06)	0.46 (0.05)
140	0.36 (0.08)	0.44 (0.05)	0.47 (0.04)
160	0.39 (0.08)	0.46 (0.04)	0.47 (0.04)
180	0.42 (0.06)	0.48 (0.05)	0.49 (0.04)
200	0.44 (0.05)	0.48 (0.04)	0.49 (0.03)
220	0.45 (0.05)	0.49 (0.03)	0.5 (0.03)

Table 7: AD prediction experiment results, average AUC with one standard deviation in parenthesis from 100 iterations **Left**: one privileged time point used. **Right**: three privileged time points used.

Samples	Baseline	LuPTS
80	0.86 (0.04)	0.85 (0.04)
100	0.86 (0.06)	0.87 (0.04)
120	0.88 (0.03)	0.9 (0.03)
140	0.89 (0.03)	0.9 (0.02)
160	0.9 (0.02)	0.91 (0.02)
180	0.9 (0.02)	0.92 (0.02)
200	0.91 (0.02)	0.92 (0.02)
220	0.91 (0.02)	0.92 (0.02)

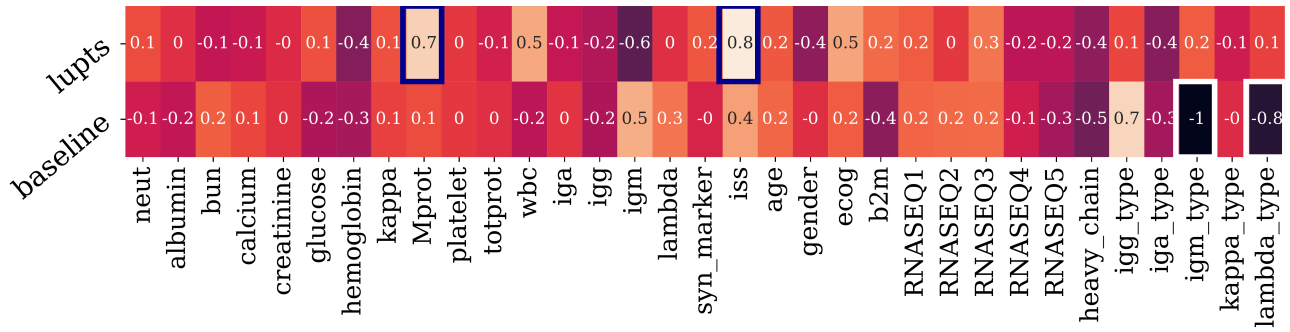
Samples	Baseline	LuPTS	Stat-LuPTS
80	0.86 (0.04)	0.87 (0.04)	0.9 (0.03)
100	0.86 (0.06)	0.89 (0.03)	0.91 (0.02)
120	0.88 (0.03)	0.91 (0.02)	0.92 (0.02)
140	0.89 (0.03)	0.91 (0.02)	0.92 (0.02)
160	0.9 (0.02)	0.92 (0.02)	0.93 (0.02)
180	0.9 (0.02)	0.92 (0.02)	0.93 (0.02)
200	0.91 (0.02)	0.93 (0.02)	0.93 (0.02)
220	0.91 (0.02)	0.93 (0.02)	0.93 (0.02)

D.5 Multiple Myeloma Progression Modeling

We elaborate on the specific features used for the multiple myeloma prediction tasks, as well as the preprocessing done on those features. The data is available via the Multiple Myeloma Research Foundation (MMRF) Researcher Gateway: `https://research.themmrf.org/`.

Features Patient biomarkers are real-valued numbers whose values evolve over time. They include: absolute neutrophil count (x10^9/l), albumin (g/l), blood urea nitrogen (mmol/l), calcium (mmol/l), serum creatinine

Figure 12: **Heatmap of Feature Weights for Early/Late Progression Task**: We display weights of the outcome logistic regression model for LuPTS and baseline OLS estimators. The x-axis shows each feature. Blue and white bounding boxes are around most important features for the LuPTS and OLS estimators, respectively.



(umol/l), glucose (mmol/l), hemoglobin (mmol/l), serum kappa (mg/dl), serum m protein (g/dl), platelet count x10^9/l, total protein (g/dl), white blood count x10^9/l, serum IgA (g/l), serum IgG (g/l), serum IgM (g/l), serum lambda (mg/dl).

We also have access to a set of static features, which we assume are available at each time step. These include demographics, risk metrics, and genomic data. With respect to the genomic features, RNA-sequencing data of CD38+ bone marrow cells are available for 769 patients. Samples from patients were collected at initiation into the study. For these patients, we use the Scanpy package in Python to identify the top 200 most variable genes, limiting the downstream analyses to these genes (Wolf et al., 2018). Subsequently, we use principal component analysis (PCA) to further reduce the dimensionality of the RNA-seq data. The projection of each patient's RNA-seq data onto the first 40 principal components serves as the final genetic features in the model.

Other static features include gender, age, and revised ISS stage, a common risk stratification score used in myeloma (Palumbo et al., 2015). Finally, binary variables detailing the patient's myeloma subtype, including whether or not they have heavy chain myeloma and presence/absence of various monoclonal proteins, are part of this set of features as well.

Pre-processing We utilize the same preprocessing strategy used by Hussain et al. (2021). For the longitudinal patient biomarkers, we first clip the values to five times the median values to account for outliers or errors in the data. The biomarkers are then normalized by subtracting the maximum value of the biomarker's healthy range from the unnormalized value. Subsequently, the value is multiplied by a biomarker-dependent scaling factor that ensures that it lies within the range, $[-8, 8]$. Aside from PCA done on the genomic data, we do zero mean, unit variance standardization on all the static features.

The data has significant missingness, with around $\sim 66\%$ of the values missing. For static features aside from the genetic features, we use mean imputation. For the genetic features, a patient's missing values are imputed with the average genetic PCA features of their five nearest neighbors, which are determined using the Minkowski distance calculated on the ISS stage, age, and other demographic features. The missing values in the longitudinal biomarkers are forward-fill imputed from the previous 2-month time point.

Evaluation We do repeated (50 repeats) 2-fold cross validation with different training and test splits across multiple training set sizes. For the early/late progression task, we exclude patients who are not eligible for autologous stem cell transplant (ASCT), resulting in 314 late progressors, 103 early progressors, and 84 right-censored patients. The privileged information for this task consists of labs taken at four equally spaced time points across the patient's first line. For the treatment response task, we restrict to patients who were given a second line of therapy, resulting in 149 PD patients, 181 non-PD patients, and 48 right-censored patients. We use two privileged time points in this case, which correspond to the end of the first line and the end of the second line, respectively. Censored patients are not included in computing AUCs. All labels have been checked with an oncologist for reliability.