A Appendix to Post-Estimation Smoothing: A Simple Baseline for Learning with Side Information

A.1 On satisfying the conditions of Theorem 1 ($\gamma + \beta < 1$)

Recall the definitions $\gamma(\varepsilon, W)$ and $\beta(\varepsilon, W; y)$:

$$\gamma(\varepsilon, W) := \mathbb{E}[\varepsilon^\top W \varepsilon] / \mathbb{E}[\|\varepsilon\|_2^2]$$
$$\beta(\varepsilon, W; y) := \mathbb{E}[\varepsilon^\top (W - I) y] / \mathbb{E}\left[\|\varepsilon\|_2^2\right] .$$

The condition $\gamma + \beta < 1$ captures a trade-off between choosing a weight matrix W which reduces the magnitude of the errors (small γ), while not affecting too much the signal in the predictions (small β). The next paragraph shows that under a reasonable assumption on the predictions, $\beta + \gamma < 1$ can always be satisfied. The paragraph after details practical considerations in picking W and checking the conditions of the theorem.

Manipulation of the definitions of gamma and beta shows that the condition $\beta + \gamma < 1$ is equivalent to the condition $\mathbb{E}[\varepsilon^\top (W - I)\hat{y}] < 0$, which is always satisfiable with some W, so long as $\mathbb{E}[\varepsilon \hat{y}^\top]$ is not the all zeros matrix. Further, when $|\mathbb{E}[\varepsilon^\top y]| < \mathbb{E}[\varepsilon^\top \varepsilon]$, $W = t \cdot I$ for any $t < 1$ will suffice so that $\beta + \gamma < 1$. The wide range of possible t is because the matrix W is combined in a convex combination with the identity matrix to form $S_c(t)$ in Eq. (2).

Lemma 1 and Example 3.1 show that an optimal smoothing matrix averages out errors in the predictions, depending on the structure in y and ε. We'd like our empirical choice of W to be close to this optimal matrix. For practical applications, we could (a) use empirical covariance matrices from training/validation data to inform our choice of W, and/or (b) for a pre-specified W we could estimate γ and β by using the training/validation data to estimate ε and y. We suspect that estimating γ and β in this way may not be practically necessary, for the following reason. If the chosen matrix W does not reduce the mean squared error for any choice of $c \in (0, 1]$, then cross validation over parameter c will result in $c = 0$, such that no smoothing occurs. Since cross-validating over c amounts to only vector (not matrix) operations, it is practical to sweep over a large number of possible c's. Thus, it could be just as fast to check if smoothing with matrix S_c (for any of the c's) reduces the MSE as to check the condition $\gamma + \beta < 1$.

A.2 Proof of Theorem 1

We now prove Theorem 1 in full generality. Recall the original theorem statement:

Theorem 1. *Given any predictor $\widehat{y}$ of y with error residuals satisfying $\mathbb{E}\left[\|\varepsilon\|_2^2\right] \neq 0$, and any weight matrix W satisfying $\gamma(\varepsilon, W) + \beta(\varepsilon, W; y) < 1$, there exists a constant $c \in (0, 1]$ such that the smoothing matrix $S_c = c \cdot W + (1 - c) \cdot I$ strictly reduces expected MSE:*

$$\mathbb{E}\left[\tfrac{1}{n}\|S_c \widehat{y} - y\|_2^2\right] < \mathbb{E}\left[\tfrac{1}{n}\|\widehat{y} - y\|_2^2\right] .$$

Proof. Let $\mu := \mathbb{E}[\varepsilon] = \mathbb{E}[\hat{y} - y]$. The squared error ($n \times$ the MSE) of using smoothing matrix $S_c = cW + (1 - c)I$ decomposes as:

$$\begin{aligned}
\|S_c \hat{y} - y\|_2^2 &= \|c(W\hat{y} - y) + (1 - c)(\hat{y} - y)\|_2^2 \\
&= \|c(W\hat{y} - y) + (1 - c)(\varepsilon)\|_2^2 \\
&= c^2 \|W\hat{y} - y\|_2^2 + (1 - c)^2 \|\varepsilon\|_2^2 + 2c(1 - c)(\varepsilon^\top W \varepsilon + \varepsilon^\top (W - I) y)
\end{aligned}$$

so that the expected reduction in MSE is given by

$$\begin{aligned}
\mathbb{E}\left[\|S_c \hat{y} - y\|_2^2\right] - \mathbb{E}\left[\|\hat{y} - y\|_2^2\right] &= c^2 \mathbb{E}\left[\|W\hat{y} - y\|_2^2\right] + \left(1 + (c^2 - 2c) + 2(c - c^2)\gamma\right) \mathbb{E}\left[\|\varepsilon\|_2^2\right] \\
&\quad + 2c(1 - c)\mathbb{E}\left[\varepsilon^\top (W - I) y)\right] - \mathbb{E}\left[\|\varepsilon\|_2^2\right] \\
&= c^2 \mathbb{E}\left[\|W\hat{y} - y\|_2^2\right] + \left((c^2 - 2c) + 2(c - c^2)(\gamma + \beta)\right) \mathbb{E}\left[\|\varepsilon\|_2^2\right]
\end{aligned}$$

This is a quadratic in c:

$$\begin{aligned}
\mathbb{E}\left[\|S_c \hat{y} - y\|_2^2\right] - \mathbb{E}\left[\|\hat{y} - y\|_2^2\right] &= c^2 \left(\mathbb{E}\left[\|W\hat{y} - y\|_2^2\right] + (1 - 2(\gamma + \beta))\mathbb{E}\left[\|\varepsilon\|_2^2\right]\right) \\
&\quad + 2c \left((\gamma + \beta - 1)\mathbb{E}\left[\|\varepsilon\|_2^2\right]\right)
\end{aligned}$$

We first show that under the assumptions above, the above expression is convex. Afterwards, we will show that the nonzero root is strictly greater than zero, and therefore conclude that there must be a value $c \in (0, 1]$ for which the

objective is negative. We first get a handle on the coefficient of the quadratic term:

$$\begin{aligned}\mathbb{E}\left[\|W\hat{y}-y\|_2^2+(1-2(\gamma+\beta))\|\varepsilon\|_2^2\right] &= \mathbb{E}\left[\|W\hat{y}-y\|_2^2+\|\varepsilon\|_2^2-2\varepsilon^\top W\varepsilon+2\varepsilon^\top(I-W)y\right]\\ &= \mathbb{E}\left[\|W\widehat{y}\|_2^2-2y^\top W\widehat{y}+\|\widehat{y}\|_2^2-2\varepsilon^\top W\widehat{y}\right]\\ &= \mathbb{E}\left[\|(W-I)\hat{y}\|_2^2\right]\\ &\geq 0\end{aligned}$$

The coefficient on the quadratic term is nonnegative, so that the expression is convex in c. Now we show that under the conditions outlined in the theorem statement, the coefficient on the linear term is negative. Recall the condition that the matrix W acts close to the identity on y but close to the zero matrix on ε, with respect to the errors: $\gamma(\varepsilon,W)+\beta(\varepsilon,W;y)<1$. When this conditions holds, we have

$$2\left((\gamma+\beta-1)\mathbb{E}\left[\|\varepsilon\|_2^2\right]\right)<0\ .$$

Thus, the optimal c value is given as

$$c^*=\frac{(1-(\gamma+\beta))\,\mathbb{E}\left[\|\varepsilon\|_2^2\right]}{\mathbb{E}\left[\|W\hat{y}-y\|_2^2\right]+(1-2\gamma-2\beta)\mathbb{E}\left[\|\varepsilon\|_2^2\right]}\ .$$

Since c^* is always positive, by convexity and continuity of the objective function, the optimal value for c within the range $(0,1]$ is $\min(c^*,1)$.

If $c^*>1$, this implies that

$$\begin{aligned}(1-(\gamma+\beta))\,\mathbb{E}\left[\|\varepsilon\|_2^2\right] &> \mathbb{E}\left[\|W\hat{y}-y\|_2^2\right]+(1-2\gamma-2\beta)\mathbb{E}\left[\|\varepsilon\|_2^2\right]\\ (\gamma+\beta))\mathbb{E}\left[\|\varepsilon\|_2^2\right] &> \mathbb{E}\left[\|W\hat{y}-y\|_2^2\right]\ .\end{aligned}$$

If this is the case, then clipping the chosen c to be $c=1$ (denote the resulting smoothing matrix S_1) will result in expected MSE decrease

$$\begin{aligned}\mathbb{E}\left[\tfrac{1}{n}\|S_1\hat{y}-y\|_2^2\right]-\mathbb{E}\left[\tfrac{1}{n}\|\varepsilon\|_2^2\right] &= \mathbb{E}\left[\tfrac{1}{n}\|W\hat{y}-y\|_2^2\right]-\mathbb{E}\left[\tfrac{1}{n}\|\varepsilon\|_2^2\right]\\ &< -(1-\gamma-\beta)\mathbb{E}\left[\tfrac{1}{n}\|\varepsilon\|_2^2\right]\ .\end{aligned}$$

Otherwise (if $c^*\leq 1$), the resulting expected MSE decrease is upper bounded as

$$\begin{aligned}\mathbb{E}\left[\tfrac{1}{n}\|S_{c^*}\hat{y}-y\|_2^2\right]-\mathbb{E}\left[\tfrac{1}{n}\|\varepsilon\|_2^2\right] &\leq -\frac{(1-\gamma-\beta)^2\,\mathbb{E}\left[\|\varepsilon\|_2^2\right]^2}{n\left(\mathbb{E}\left[\|W\hat{y}-y\|_2^2\right]+(1-2\gamma-2\beta)\mathbb{E}\left[\|\varepsilon\|_2^2\right]\right)}\\ &= -(1-\gamma-\beta)\mathbb{E}\left[\tfrac{1}{n}\|\varepsilon\|_2^2\right]\cdot\frac{(1-\gamma-\beta))\,\mathbb{E}\left[\|\varepsilon\|_2^2\right]}{\left(\mathbb{E}\left[\|W\hat{y}-y\|_2^2\right]+(1-2\gamma-2\beta)\mathbb{E}\left[\|\varepsilon\|_2^2\right]\right)}\ .\end{aligned}$$

The optimal resulting MSE reduction from using S_c where $c=\min\{c^*,1\}$ is then bounded as

$$\begin{aligned}&\mathbb{E}\left[\tfrac{1}{n}\|S_c\hat{y}-y\|_2^2\right]-\mathbb{E}\left[\tfrac{1}{n}\|\hat{y}-y\|_2^2\right]\\ &\quad\leq -(1-\gamma-\beta)\mathbb{E}\left[\tfrac{1}{n}\|\varepsilon\|_2^2\right]\cdot\min\left\{1,\frac{(1-\gamma-\beta)\mathbb{E}\left[\|\varepsilon\|_2^2\right]}{\left(\mathbb{E}\left[\|W\hat{y}-y\|_2^2\right]+(1-2\gamma-2\beta)\mathbb{E}\left[\|\varepsilon\|_2^2\right]\right)}\right\}\\ &\quad< 0\ .\end{aligned}$$

□

A.3 Proof of Lemma 1

We now provide a proof of Lemma 1. Recall the original statement:

Lemma 1. *For a predictor $\widehat{y}$ of y with error residuals distributed as $\varepsilon(t)=\widehat{y}(t)-y(t)$, when $K_{\widehat{y}\widehat{y}}\succ 0$, the optimal linear smoothing matrix has the form*

$$\begin{aligned}S^* &= \operatorname*{arg\,min}_{S\in\mathbb{R}^{n\times n}}\mathbb{E}\left[\tfrac{1}{n}\|S\widehat{y}-y\|_2^2\right]\\ &= I-(K_{\varepsilon\varepsilon}+K_{y\varepsilon})^\top(K_{yy}+K_{y\varepsilon}+K_{\varepsilon y}+K_{\varepsilon\varepsilon})^{-1}\ .\end{aligned}$$

The expected MSE reduction of applying S^* versus using the original predictions $\widehat{y}$ is always non-negative, and is given by

$$\tfrac{1}{n}\mathbb{E}\left[\|\widehat{y}-y\|_2^2\right]-\|S^*\widehat{y}-y\|_2^2\right]=\tfrac{1}{n}\mathrm{tr}\left(K_{\widehat{y}y}^\top(K_{\widehat{y}\widehat{y}})^{-1}K_{\widehat{y}y}\right)\ .$$

Proof. Setting the matrix differential of the following convex objective to zero, any solution S^* to

$$S^* = \arg\min_{S \in \mathbb{R}^{n \times n}} \mathbb{E}\left[\frac{1}{n}\|S\widehat{y} - y\|_2^2\right]$$

satisfies

$$\frac{\partial}{\partial S}\mathbb{E}\left[(S\widehat{y} - y)^\top (S\widehat{y} - y)\right] = 2(SK_{\widehat{y}\widehat{y}} - K_{y\widehat{y}}) = 0 \ .$$

If $K_{\widehat{y}\widehat{y}}$ is positive definite (and thus invertible), the objective is strictly convex and the unique optimal solution is

$$\begin{aligned} S^* &= K_{y\widehat{y}}(K_{\widehat{y}\widehat{y}})^{-1} \\ &= I - K_{\varepsilon\widehat{y}}(K_{\widehat{y}\widehat{y}})^{-1} \\ &= I - (K_{\varepsilon\varepsilon} + K_{\varepsilon y})(K_{yy} + K_{y\varepsilon} + K_{\varepsilon y} + K_{\varepsilon\varepsilon})^{-1} \ . \end{aligned}$$

since the identity matrix I is within the set of possible estimators ($\mathbb{R}^{n \times n}$), we know that the resulting objective satisfies $\mathbb{E}\left[\|S^*\widehat{y} - y\|_2^2\right] \leq \mathbb{E}\left[\|\widehat{y} - y\|_2^2\right]$. In fact, applying properties of the trace operator (cyclic property, invariance to transposes) gives the following expression for the reduction in expected squared error:

$$\begin{aligned} \mathbb{E}\left[\|\widehat{y} - y\|_2^2 - \|S^*\widehat{y} - y\|_2^2\right] &= \operatorname{tr}\left(K_{yy} + K_{\widehat{y}\widehat{y}} - 2K_{y\widehat{y}} - K_{yy} + K_{y\widehat{y}}\left(K_{\widehat{y}\widehat{y}}\right)^{-1} K_{\widehat{y}y}\right) \\ &= \operatorname{tr}\left((K_{\widehat{y}\widehat{y}} - K_{y\widehat{y}})(K_{\widehat{y}\widehat{y}})^{-1}(K_{\widehat{y}\widehat{y}} - K_{\widehat{y}y})\right) \\ &= \operatorname{tr}\left((K_{\varepsilon\varepsilon} + K_{y\varepsilon})^\top (K_{yy} + K_{\varepsilon\varepsilon} + K_{\varepsilon y} + K_{y\varepsilon})^{-1}(K_{\varepsilon\varepsilon} + K_{y\varepsilon})\right) \ . \end{aligned}$$

Applying a matrix trace inequality for positive definite matrix A and positive semi-definite matrix B: $\operatorname{tr}\left(A^{-1}B\right) \geq \lambda_{\min}(A^{-1})\operatorname{tr}(B) = \operatorname{tr}(B)/\lambda_{\max}(A) \geq \operatorname{tr}(B)/\operatorname{tr}(A)$ gives an upper bound on the reduction:

$$\mathbb{E}\left[\|\widehat{y} - y\|_2^2 - \|S^*\widehat{y} - y\|_2^2\right] \geq \frac{\operatorname{tr}\left((K_{\varepsilon\varepsilon} + K_{y\varepsilon})(K_{\varepsilon\varepsilon} + K_{y\varepsilon})^\top\right)}{\operatorname{tr}(K_{yy} + K_{\varepsilon\varepsilon} + K_{\varepsilon y} + K_{y\varepsilon})} \ .$$

Note that $(K_{\varepsilon\varepsilon} + K_{y\varepsilon})(K_{\varepsilon\varepsilon} + K_{y\varepsilon})^\top$ and $K_{yy} + K_{\varepsilon\varepsilon} + K_{\varepsilon y} + K_{y\varepsilon} = K_{\hat{y}\hat{y}}$ are positive semi-definite by construction and positive definite by assumption, respectively. □

A.4 Linear example (continued)

Here we give a more thorough analysis of the example presented in example 3.1 in the main text. Recall the setting: the zero-mean stochastic processes $x(t)$ and $y(t)$ which are dependent on a third zero-mean hidden process $z(t)$, but with independent additive Gaussian noise $\omega(t)$, $\mu(t)$, respectively. In particular:

$$\begin{aligned} z &\sim \mathcal{N}(0, \Sigma_z) \\ x(t) &= z(t) + \omega(t), \quad \omega(t) \sim_{i.i.d.} \mathcal{N}(0, \sigma_x^2) \\ y(t) &= c \cdot z(t) + \mu(t), \quad \mu(t) \sim_{i.i.d.} \mathcal{N}(0, \sigma_y^2) \end{aligned}$$

The autocorrelation matrices show that there is shared variation due to the "hidden" process z:

$$\begin{aligned} K_{xx}[t,s] &= K_{zz}[t,s] + K_{\omega\omega}[t,s] \\ K_{yy}[t,s] &= c^2 K_{zz}[t,s] + K_{\mu\mu}[t,s] \\ K_{xy}[t,s] &= cK_{zz}[t,s] \end{aligned}$$

Consider the problem of learning a predictor for unseen samples by learning the 1-dimensional regression weight $\hat{c}$ from a sample of $\{x_i, y_i\}_{i=1}^n$ data pairs drawn from the distribution above. Then for a fresh, independently drawn sample will have predicted value $\widehat{y} = \hat{c} \cdot x$, and

$$K_{y\widehat{y}} = \mathbb{E}\left[y\widehat{y}^\top\right] = \mathbb{E}\left[y(\hat{c}x)^\top\right] = \mathbb{E}[\hat{c}]K_{xy}^\top = c\mathbb{E}[\hat{c}]K_{zz} \ .$$

Similarly,

$$K_{\widehat{y}\widehat{y}} = \mathbb{E}\left[\hat{c}x(\hat{c}x)^\top\right] = \mathbb{E}[\hat{c}^2](K_{zz} + K_{\omega\omega}) = \mathbb{E}[\hat{c}^2](K_{zz} + \sigma_x^2 I) \ .$$

Then the optimal smoothing matrix from Lemma 1 is

$$S^* = K_{y\widehat{y}} \left(K_{\widehat{y}\widehat{y}}\right)^{-1} = \frac{c\mathbb{E}[\hat{c}]}{\mathbb{E}[\hat{c}^2]} K_{zz} \left(K_{zz} + \sigma_x^2 I\right)^{-1} \ .$$

The model defined above can be described as an "errors in variables" model, if we consider z as the true regresssor and x as an error-imbued observation of it. Under such a model, total least squares provides a consistent estimator of c (see below), and thus it is the estimator that we analyze in the main text. However, we are concerned first and foremost with recovering y without postprocessing, the ordinary least squares estimator might be a preferable solution. We first expand upon the exposition from the main paper of the example under the total least squares estimator, then follow with a discussion of using the ordinary least squares estimator in this context.

Total least squares (TLS) estimator. To compute the forms of the auto-correlation matrices above for the TLS estimator, we make use of the following fact found, for example, in Huffel (1991); Schneeweiss (1976):

- For the errors in variables model described above, the asymptotic distribution of the TLS estimator is normal, with mean c, and variance approaching 0 as $n \to \infty$.

Which gives us the approximations approximations $\mathbb{E}\left[\hat{c}_{tls}\right] \approx c$, and $\mathbb{E}[\hat{c}]^2 \approx \mathbb{E}[\hat{c}^2]$. The calculations in the main text are thus written out more expositionally as:
Expected unsmoothed performance:

$$\begin{aligned}
\mathbb{E}\left[\tfrac{1}{n}\|\widehat{y} - y\|_2^2\right] &= \tfrac{1}{n}\mathrm{tr}\left(K_{yy} - 2K_{y\widehat{y}} + K_{\widehat{y}\widehat{y}}\right) \\
&= \tfrac{1}{n}\mathrm{tr}\left(c^2 K_{zz} + \sigma_y^2 I - 2c\mathbb{E}[\hat{c}]K_{zz} + \mathbb{E}[\widehat{c}^2](K_{zz} + \sigma_x^2 I)\right) \\
&\approx \sigma_y^2 + c^2\sigma_x^2 \ .
\end{aligned}$$

From Lemma 1, the expected smoothed performance using S^* is:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n}\|S^*\widehat{y} - y\|_2^2\right] &= \frac{1}{n}\mathrm{tr}\left(c^2 K_{zz} + \sigma_y^2 I - c^2\frac{\mathbb{E}[\hat{c}]^2}{\mathbb{E}[\hat{c}^2]} K_{zz}^2 \left(K_{zz} + \sigma_x^2 I\right)^{-1}\right) \\
&\approx \frac{c^2}{n}\mathrm{tr}\left(K_{zz}\left(I - K_{zz}\left(K_{zz} + \sigma_x^2 I\right)^{-1}\right)\right) + \sigma_y^2 \\
&= c^2\sigma_x^2\left(1 - \frac{1}{n}\mathrm{tr}\left(\left(\sigma_x^{-2}K_{zz} + I\right)^{-1}\right)\right) + \sigma_y^2 \\
&\geq \sigma_y^2 \ .
\end{aligned}$$

Using the second to last line above, the expected decrease in MSE achieved from applying the optimal linear smoothing matrix to the asymptotic total least squares estimator is then

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n}\|\widehat{y} - y\|_2^2\right] - \mathbb{E}\left[\frac{1}{n}\|S^*\widehat{y} - y\|_2^2\right] &\approx \frac{c^2\sigma_x^2}{n}\mathrm{tr}\left(\left(\sigma_x^{-2}K_{zz} + I\right)^{-1}\right) \\
&\geq \frac{c^2\sigma_x^2}{n}\sum_n \frac{1}{1 + \sigma_x^2 \cdot \lambda_{\max}(K_{zz})} = c^2\frac{\sigma_x^2}{1 + \sigma_x^2 \cdot \lambda_{\max}(K_{zz})}
\end{aligned}$$

where $\lambda_{\max}(\cdot)$ denote the maximum eigenvalue of a matrix.

Ordinary Least Squares (OLS) estimator Due to the noise process in x, OLS will produce a biased estimator $\hat{c}$:

$$\begin{aligned}
\hat{c}_{ols} &= (x^\top x)^{-1}x^\top y \\
&= ((z+\omega)^\top(z+\omega))^{-1}(z+\omega)^\top(cz + \mu)
\end{aligned}$$

μ is uncorrelated with z and ω, so that

$$\mathbb{E}\left[\hat{c}_{ols}\right] = c\left(1 - \mathbb{E}\left[\frac{w^\top w + z^\top\omega}{(z+\omega)^\top(z+\omega)}\right]\right) \ .$$

As $n \to \infty$,

$$\mathbb{E}\left[\hat{c}_{ols}\right] \to c\left(1 - \frac{\sigma_x^2}{\sigma_x^2 + \frac{1}{n}\mathrm{tr}\left(K_{zz}\right)}\right) \ .$$

We see that the noise associated with x biases the estimated regression coefficient to be shallower; this is a well known phenomenon in the errors-in-variables model termed attenuation bias. This bias limits the amount to which P-ES can denoise the estimations, as shown in Fig. 4(B). In comparison to Fig. 1(B), we see that the unsmoothed OLS estimator exhibits the same qualitative behavior over the parameter selections as the unsmoothed TLS estimator. Moreover, the same pattern of the smoothed estimates (with performance floor around σ_y^2) is maintained in Fig. 4(B), although this is trend is less fitting for larger σ_x^2 (corresponding to larger magnitude of bias in $\hat{c}_{ols}$).

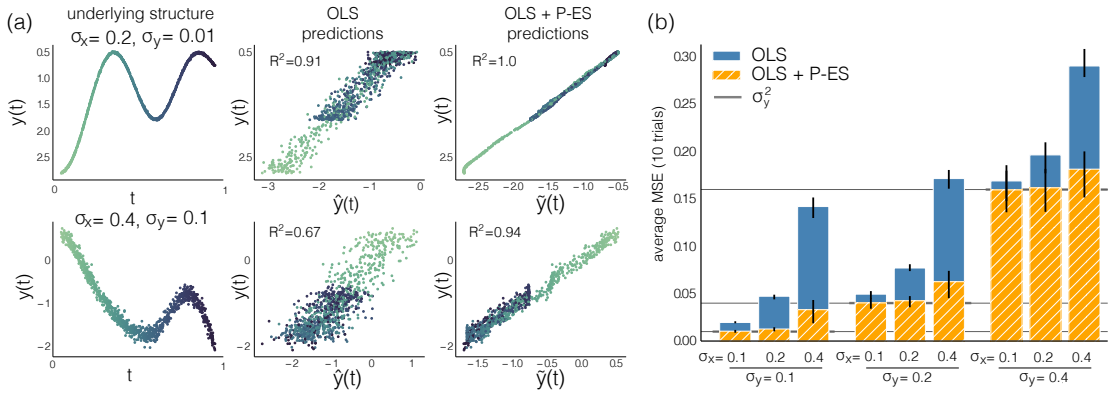

Figure 4: Figure for exact same run of simulations but using ordinary least squares (OLS) estimator instead of total least squares, as in Fig. 1.

B Experiment Details

All experiments were run on a machine with 48 cores, each of them an Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz, and 256G RAM. All experimental code is written in python, and the relevant libraries used are listed below. Our code is available at `www.github.com/estherrolf/p-es`. Instructions for downloading and using the intermediate video predictions from Kanazawa et al. (2019) are detailed there. The housing data is provided by Zillow through the Zillow Transaction and Assessment Dataset (ZTRAX). More information on accessing the data can be found at `http://www.zillow.com/ztrax`. (The results and opinions are those of the authors of this work and do not reflect the position of Zillow Group).

B.1 Video experiments

Metrics. For consistency, we use the same metrics as reported in Kanazawa et al. (2019), and the same code to calculate these metrics. All metrics are defined per video, and averaged over all videos. A description of each metric is given here; See Kanazawa et al. (2019) and `https://github.com/cbsudux/Human-Pose-Estimation-101` (Babu, 2019) for further explanation:

- Percentage key points (PCK): percentage of 2D key points that fall within $\alpha \cdot \max\{h, w\}$ of the labeled key point, where h and w parameters of a per-frame tight bounding box around the entire person; here $\alpha = 0.05$.
- Mean per joint position error (MPJPE): Mean euclidean distance of predicted to ground truth joint, averaged over joints in the human pose model (calculated after aligning root joints), measured in millimeters.
- Mean per joint position error after Procrustes alignment (PA-MPJPE): MPJPE after alignment to the ground truth by Procrustes alignment method, measured in millimeters.
- Acceleration Error (Accel Err): defined in Kanazawa et al. (2019) as "the average difference between ground truth 3D acceleration and predicted 3D acceleration of each joint in mm/s^2."
- Acceleration (Accel) For 2D datasets, measures "acceleration in mm/s^2" (Kanazawa et al., 2019). Note that this metric is only useful in conjunction with other metrics, as a baseline constant predictor would achieve 0 acceleration. However, for predictions that also do well on PCK, lower acceleration is more meaningful.

Parameter tuning. We started with a grid search of $\sigma \in [0.5, 1, 2, 4]$ and $c \in [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$ and then interpolated best values once to obtain this final set. Specifically, this meant including $\sigma = 3$ and $c \in [0.5, 0.7, 0.9]$.

B.2 Predicting house price from attributes

Metrics. R^2, or coefficient of determination is a metric which reports the percent of squared deviation in the independent labels which is explained by the predictions. Formally it is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \text{avg}(y))^2}.$$

It is possible that this score can be negative; in this case we clip negative R^2 values at zero in computing averages and ranges (in our experiments, this only occurs for spatial extrapolation when locations are considered as features). We used the implementation of R^2 available via `sklearn.metrics.r2_score`.

Dataset. The Zillow Transaction and Assessment Database (ZTRAX) (Zillow, 2018) contains home sales of many different types; we restricted our dataset to single family homes. Only the most recent sale for a property id and location was considered, and after that only home sales occur after the year 2010 (dated by the column 'contract year'). Any observation for which any of the 12 features considered (listed below) were missing was dropped. The resulting dataset contains $608,959$ homes sales spread across the United States.

Features included were: year built (from 2010), number of stories, number of rooms, number of bedrooms, number of baths, number of partial baths, size (sqft), whether there is heating, whether there is air conditioning, the contract year, the contract month, and whether the home was new; location was encoded as the latitude and longitude of the home, and target label is the most recent sale price of the home.

Hyperparameters searched for generating Fig. 2 are given in Table 4. The ten random trials for the experiments in Fig. 2 where done as follows. For each trial we drew 60,000 data points from the total dataset, with replacement between trials. Then for each random draw, we allocated 20,000 points to the training, validation and test sets, such that no points were overlapping within in each trial. In each trial, hyperparameters were chosen to maximize validation performance for that single trial, and then the optimal hyperparameters defined the model that we applied to the holdout set.

For the timing and methodological comparisons in Table 3, we followed a similar procedure, but with only 10,000 points for training, validation and test sets, so that the total run times were reasonable and we could run enough trials to get a notion of variability in results. The parameters considered in this experiment and the total number of parameter configurations swept over for each algorithm, are given in Table 5.

The spatial extrapolation experiments followed the same sampling protocol as above for each trial, with the exception that training and validation sets were drawn from a pool of observations which lay above 37° in latitude ($376,615$ total observations), and the holdout sets were drawn from the remaining $232,344$ observations. For this experiment we considered hyperparameters $\texttt{max\_depth} \in [5, 10]$, $\texttt{num\_estimators} \in [100, 200]$, $\sigma \in \texttt{logspace}(-4, 2, \texttt{base} = 10, \texttt{num} = 9)$ and $c \in \texttt{linspace}(0, 1.0, \texttt{num} = 11)$.

We used the existing `sklearn` implementation of `GaussianProcessRegressor` for GPR[2], `xgboost.XGBRegressor` for xgboost[3], and our own implementation for HEM and LapRLS which pre-computes the Gram matrix for efficiency (all code available at www.github.com/estherrolf/p-es). We also used our own implementation of the random features algorithm of Rahimi and Recht (2009), so that each random feature is generated as a transformation of the original features x:

$$\cos(w^\top x + b); \quad w \sim \mathcal{N}(0, \sigma^2_{\texttt{RF}}) \quad b \sim \text{unif}(0, 2\pi) \ .$$

Table 4: Hyperparameters considered in runs for Figure 2.

method	hyperparameters considered
Ridge Regression	$\lambda_{\texttt{RR}} \in \texttt{logspace}(-6, 4, \texttt{base} = 10, \texttt{num} = 5)$
Random Features	number of random features $\in [100, 200]$
	$\sigma_{\texttt{RF}} \in \texttt{logspace}(-8, -4, \texttt{base} = 10, \texttt{num} = 3)$
	$\lambda_{\texttt{RR}} \in \texttt{logspace}(-6, -4, \texttt{base} = 10, \texttt{num} = 3)$
XGB	$\texttt{max\_depths} \in [2, 5, 10]$
	$\texttt{num\_estimators} \in [100, 200]$
PES	$\sigma \in \texttt{logspace}(-4, 0, \texttt{base} = 10, \texttt{num} = 5)$
	$c \in \texttt{linspace}(0, 1.0, \texttt{num} = 11)$

B.3 Performance for different data set sizes

Here we study the accuracy increases from P-ES as a function of the amount of training data. We varying train/validation and holdout set sizes in $[1000, 2000, 4000, 8000, 12000, 16000, 20000]$ and otherwise following the experimental setup in Figure 2). The resulting accuracy increases (from the unsmoothed predictions) for 10 random trials are shown in Figure 5.

While the performance of P-ES for the random features regressor is variable (due largely to the variability of the original predictor), we see for the other two predictors a trend that as the data sizes increase, the advantage to smoothing is not decreasing. With more training data, the underlying predictors capture better signal and there are more nearby predictions which with to smooth any given point, both of which are advantages for P-ES.

[2]Documentation available at: `https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html`

[3]Documentation available at: `https://xgboost.readthedocs.io/en/latest/python/python_api.html`

Table 5: Hyperparameters considered in runs for Table 3.

method	hyperparameters considered	total number of hyperparameters
smoothing	$\sigma \in \texttt{logpspace}(-2, 0, \texttt{base} = 10, \texttt{num} = 9)$	9
XGB	$\texttt{max\_depths} \in [2, 5, 10]$, $\texttt{num\_estimators} \in [100, 200]$	6 (3×2)
+ shrinkage	$\delta \in \texttt{linspace}(0, 1, \texttt{num} = 11)$	66 (11×6)
+ P-ES	$\sigma \in \texttt{logspace}(-4, 0, \texttt{base} = 10, \texttt{num} = 5)$ $c \in \texttt{linspace}(0, 1, \texttt{num} = 11)$	330 $((5 \times 11) \times 6)$
LapRLS	$\lambda_{\texttt{ridge}} \in \texttt{logpspace}(-2, 4, \texttt{base} = 10, \texttt{num} = 5)$, $\lambda_{\texttt{lap}} \in \texttt{logpspace}(-4, 2, \texttt{base} = 10, \texttt{num} = 5)$	25 (5×5)
GPR	$\alpha \in \texttt{logspace}(-6, 0, \texttt{num} = 3, \texttt{base} = 10)$ $\sigma_{\texttt{const}} \in \texttt{logspace}(-2, 2, \texttt{num} = 4, \texttt{base} = 10)$ $\sigma_{\texttt{gpr}} \in \texttt{logspace}(-2, 2, \texttt{num} = 4, \texttt{base} = 10)$	48 $(3 \times 4 \times 4)$
HEM	$\sigma \in \texttt{logspace}(-4, 0, \texttt{base} = 10, \texttt{num} = 5)$ $\eta \in \texttt{linspace}(0.01, 1, \texttt{num} = 6)$	180 $((5 \times 6) \times 6)$

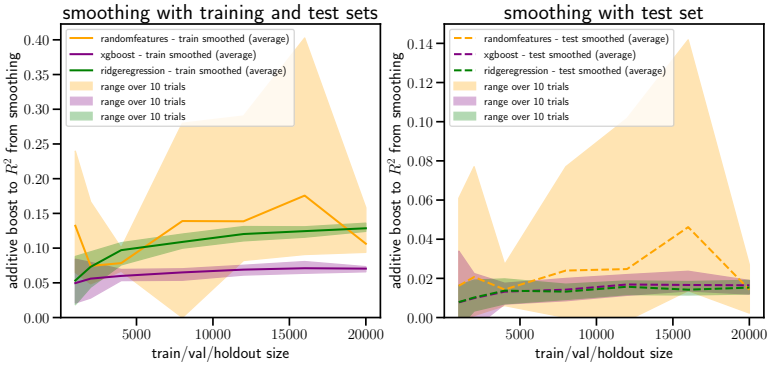

Figure 5: Additive MSE increase due to smoothing for different training set sizes.

B.4 Extrapolation experiments

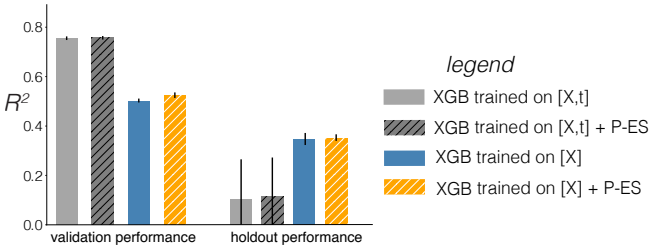

Figure 6: All comparisons for geographic extrapolation experiment.

In Figure 3 of the main text, we compared two different methods for incorporating latitude and longitude in house price predictions. Figure 6 shows the same plot, with the addition of smoothing on the predictions that included t as features. The aim of this experiment is to show that P-ES is robust to distribution shifts from e.g. extrapolation (not that it increases performance necessarily). Validation performance using [X,t] (left solid grey) is much higher than just using [X] (left blue), which might mislead a practitioner to include t as a feature when in fact the holdout performance is much worse (right solid grey blue vs. right blue). In contrast, validation P-ES performance on [X] (left solid blue) is worse than including t as a feature (left dashed grey), but exhibits no holdout degradation (left blue vs. right dashed grey).