
Learning and Interpreting Linear Regression

David I. Inouye

| A new function class changes **how we predict**

Previously: KNN predicts from nearby stored examples.

Now: learn a shared function that combines the input coordinates.

We begin with **house prices**, then return to **review sentiment**.

| First solve a model exactly; then learn **how to optimize**

Stage	Question
1. Regression → now	Which function, which loss, and what does the OLS solution mean?
2. Classification	How do probabilities change the model and loss?
3. Convexity	What makes these fitting problems well behaved?
4. Optimization	How can local information guide numerical updates?
5. Model limits	What can no linear fit express?

| Real Ames sales give us a **numerical prediction task**

Goal: predict a house's sale price from its recorded characteristics.

- **2,930 real sales** in Ames, Iowa, 2006–2010.
- Inputs include living area, construction year, bathrooms, and neighborhood.
- The target y is a **dollar amount**, rather than a class label.

This is **regression**: predict a numerical outcome.

| A property becomes a vector of **interpretable measurements**

Coordinate	Meaning	Example unit
Living area	Above-ground finished space	Square feet
Year built	Construction year	Calendar year
Garage capacity	Number of cars accommodated	Cars
Neighborhood indicator	Membership in one recorded neighborhood	0 or 1

Indicator encoding: turn a category into 0/1 columns.

$$A \mapsto [1, 0], \quad B \mapsto [0, 1], \quad C \mapsto [0, 0] \text{ (reference)}.$$

75 property fields → **226 numerical coordinates** after encoding.

| KNN can predict a number by **averaging nearby outcomes**

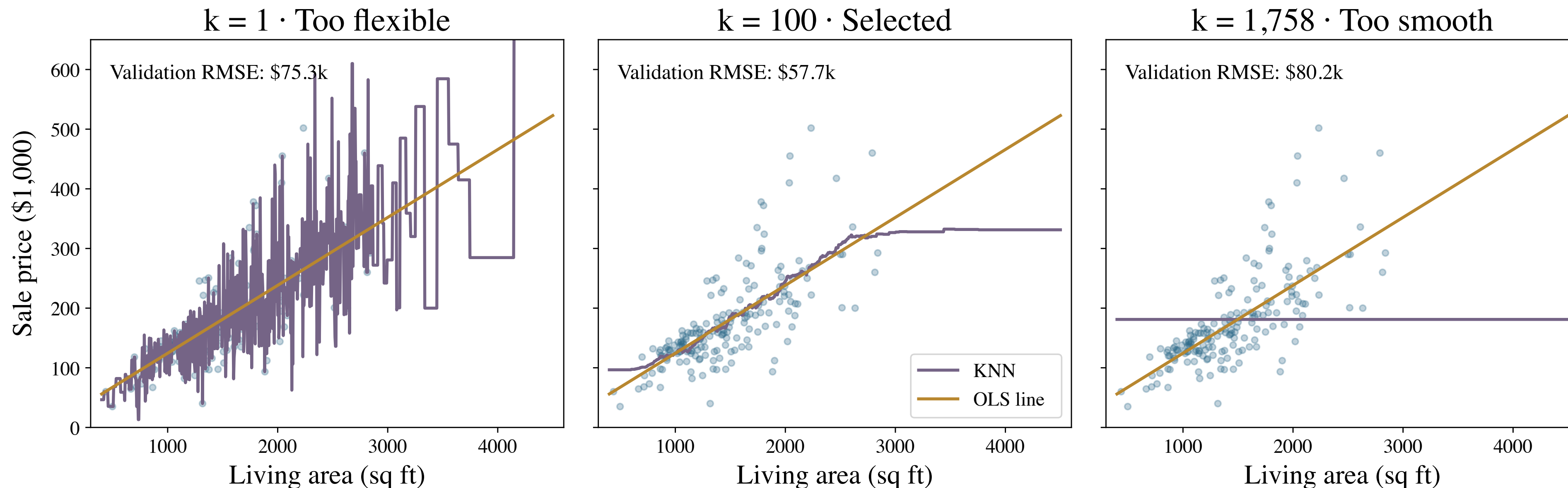
For the k nearest training inputs, indexed by $N_k(\mathbf{x})$:

$$\hat{f}_{\mathcal{D}}(\mathbf{x}) = \frac{1}{k} \sum_{i \in N_k(\mathbf{x})} y_i.$$

Three neighboring sales: \$150,000, \$180,000, and \$210,000 $\rightarrow$ **\$180,000**.

The lookup mechanism is familiar; **averaging replaces the classification vote**.

| The same sales support a local rule or a **single fitted line**



Same training sales and **living area alone**: small k follows local fluctuations; very large k loses the trend.

$k = 100$ is within **0.3% of the best validation RMSE** across all k .

| A parametric model learns **shared coefficients**

KNN regression

Retain examples; find neighbors for each query.

The stored model grows with the dataset.

Nonparametric

Linear regression

Fit coefficients; reuse the same formula for every query.

For fixed features, the number of coefficients stays fixed.

Parametric

These names describe model structure, not whether a method has settings such as k .

| A line predicts with a **slope** and an **offset**

$$\hat{y} = f_{w,b}(x) = wx + b.$$

- x : measured living area; $\hat{y}$: predicted price.
- w : change in prediction per additional square foot.
- b : where the fitted line meets the vertical axis.

For the one-input Ames fit: **100 more sq ft** → **about \$11,400 more predicted price.**

| More inputs give a **weighted sum of contributions**

$$\hat{y} = \boldsymbol{w}^\top \boldsymbol{x} + b = \sum_{j=1}^d w_j x_j + b.$$

A coefficient w_j changes the prediction per unit of x_j , **holding the other coordinates fixed**.

With 226 input coordinates, there are **227 coefficients including the offset**.

The conventional name is “linear model”; the offset makes this formally **affine**.

| Fitted coefficients determine differences in predicted price

Rounded coefficients from the four-feature Ames model:

Feature	Coefficient per unit
Living area (sq ft)	+\$90
Construction year	+\$840
Full bathrooms	−\$11,500
Garage spaces	+\$23,600

Compare two houses: same year and bathrooms; house B has **100 more sq ft and one fewer garage space**.

$$\hat{y}_B - \hat{y}_A \approx 100(90) - 23,600 = -\$14,600.$$

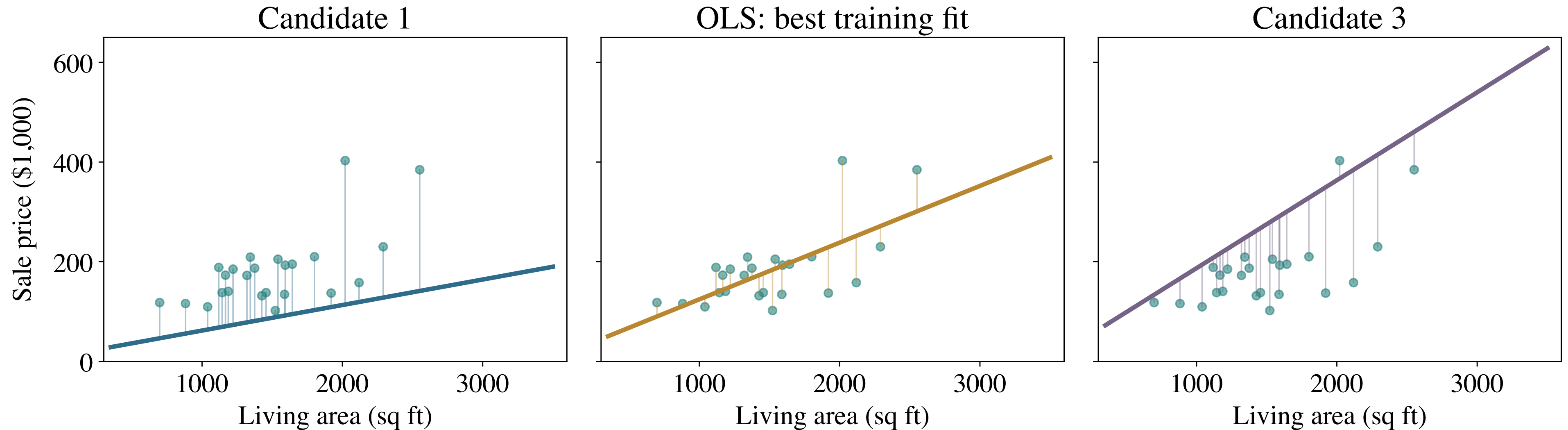
| Choosing a function leaves **which coefficients?** unresolved

Established: every (w, b) gives a prediction rule.

Next: quantify fitting error, solve for coefficients, and interpret the solution.

Our KNN procedure had no explicit training-objective minimization. Here we will explicitly minimize **squared prediction error**.

| Residuals measure vertical prediction errors



$r_i = y_i - \hat{y}_i$: observed price minus predicted price for house i .

Stack the errors: $\mathbf{r} = [r_1, \dots, r_n]^T \in \mathbb{R}^n$ — n **observations, not d input features.**

| Mean squared error makes **large residuals costly**

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2, \quad J(w, b) = \frac{1}{2} \text{MSE}.$$

Doubling a residual multiplies its squared contribution by **four**.

The factor $1/2$ simplifies derivatives; it does not change the minimizer.

For interpretation: $\text{RMSE} = \sqrt{\text{MSE}}$ is a standard-deviation-like error scale in **dollars**, rather than dollars squared.

For fitting: minimize squared error; taking the square root adds no new minimizer and complicates the derivatives.

| Can large positive and negative errors cancel the fitting problem?

A model makes two residuals: +\$50,000 and −\$50,000.

What does their zero average establish?

A. The model has zero mean squared error.

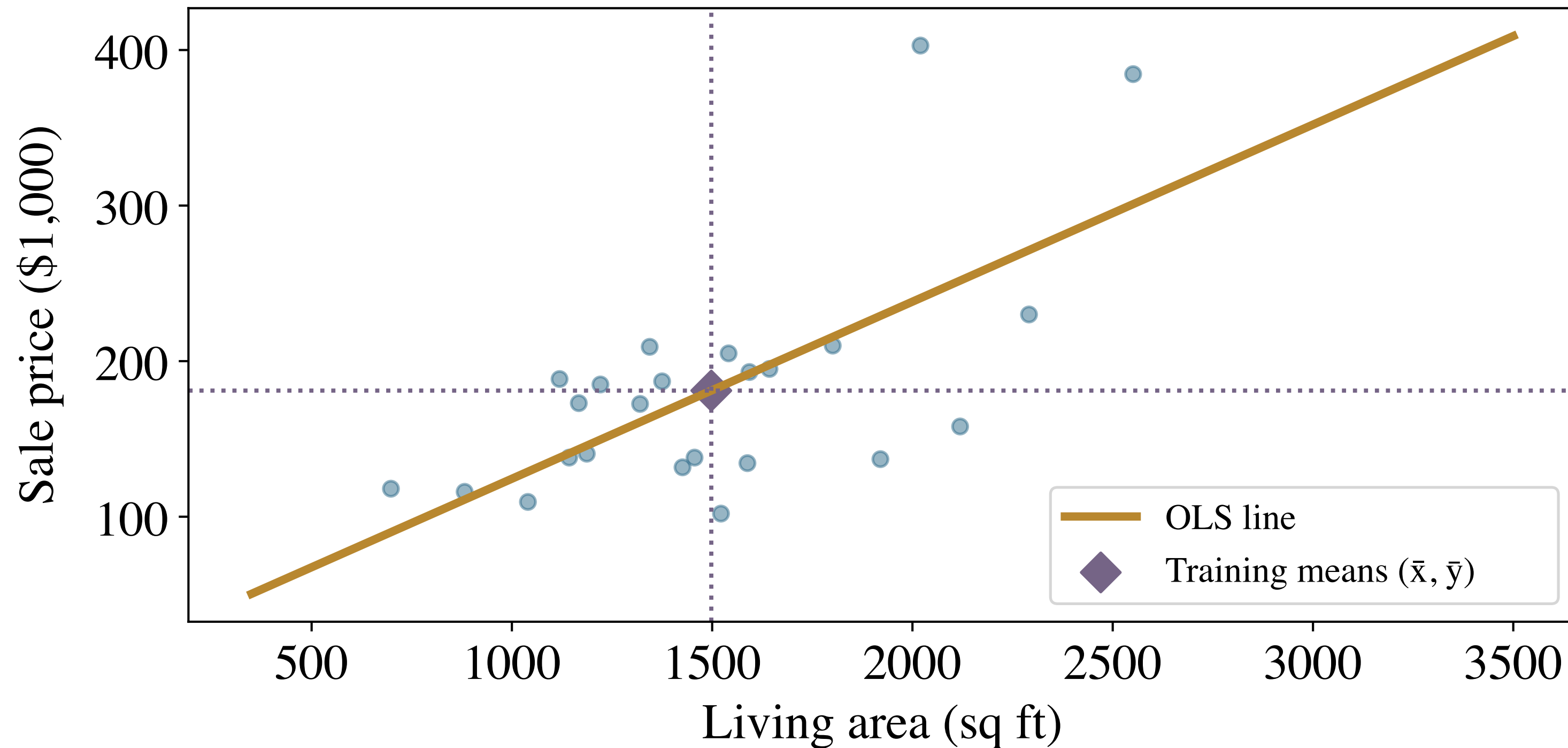
B. The signs balance, but squared error can still be large.

C. The fitted line must be the least-squares solution.

B. Squaring prevents opposite signs from hiding large mistakes.

| First solve **one input and one fitted line**

Simplify to build intuition: living area x predicts price y using $w x + b$.



At $x = \bar{x}$, choosing $b = \bar{y} - w\bar{x}$ gives $w\bar{x} + b = w\bar{x} + (\bar{y} - w\bar{x}) = \bar{y}$.

This places the line through $(\bar{x}, \bar{y})$. Why is that offset optimal?

| Optimizing the offset **balances the residuals**

$$J(w, b) = \frac{1}{2n} \sum_i (wx_i + b - y_i)^2.$$

$$\begin{aligned} \frac{\partial J}{\partial b} &= \frac{1}{n} \sum_i (wx_i + b - y_i) \\ &= w \frac{1}{n} \sum_i x_i + b - \frac{1}{n} \sum_i y_i \end{aligned}$$

$$0 = w\bar{x} + b - \bar{y}$$

$$\hat{b}(w) = \bar{y} - w\bar{x}$$

(Differentiate each squared residual)

(Distribute the sum; b repeats n times)

(Set the derivative to zero)

(Solve for b at a fixed slope)

The intercept derivative is the mean prediction error: this choice makes it zero.

| Substituting the offset **centers both inputs and outcomes**

Write $x_i^c = x_i - \bar{x}$ and $y_i^c = y_i - \bar{y}$.

$$\begin{aligned} wx_i + \hat{b}(w) - y_i &= wx_i + (\bar{y} - w\bar{x}) - y_i \\ &= w(x_i - \bar{x}) - (y_i - \bar{y}) = wx_i^c - y_i^c \end{aligned}$$

(Substitute the optimal offset)

(Collect centered quantities)

$$\frac{d}{dw} J(w, \hat{b}(w)) = \frac{1}{n} \sum_i x_i^c (wx_i^c - y_i^c)$$

(Differentiate the centered squared errors)

Now set this derivative to zero and solve for the slope.

Rearranging the zero derivative gives **the closed-form slope**

$$\begin{aligned} 0 &= \sum_i x_i^c (\widehat{w} x_i^c - y_i^c) \\ &= \sum_i [\widehat{w} (x_i^c)^2 - x_i^c y_i^c] \\ &= \widehat{w} \sum_i (x_i^c)^2 - \sum_i x_i^c y_i^c \end{aligned}$$

$$\widehat{w} \sum_i (x_i^c)^2 = \sum_i x_i^c y_i^c$$

$$\widehat{w} = \frac{\sum_i x_i^c y_i^c}{\sum_i (x_i^c)^2}$$

(Set derivative to zero; multiply by n)

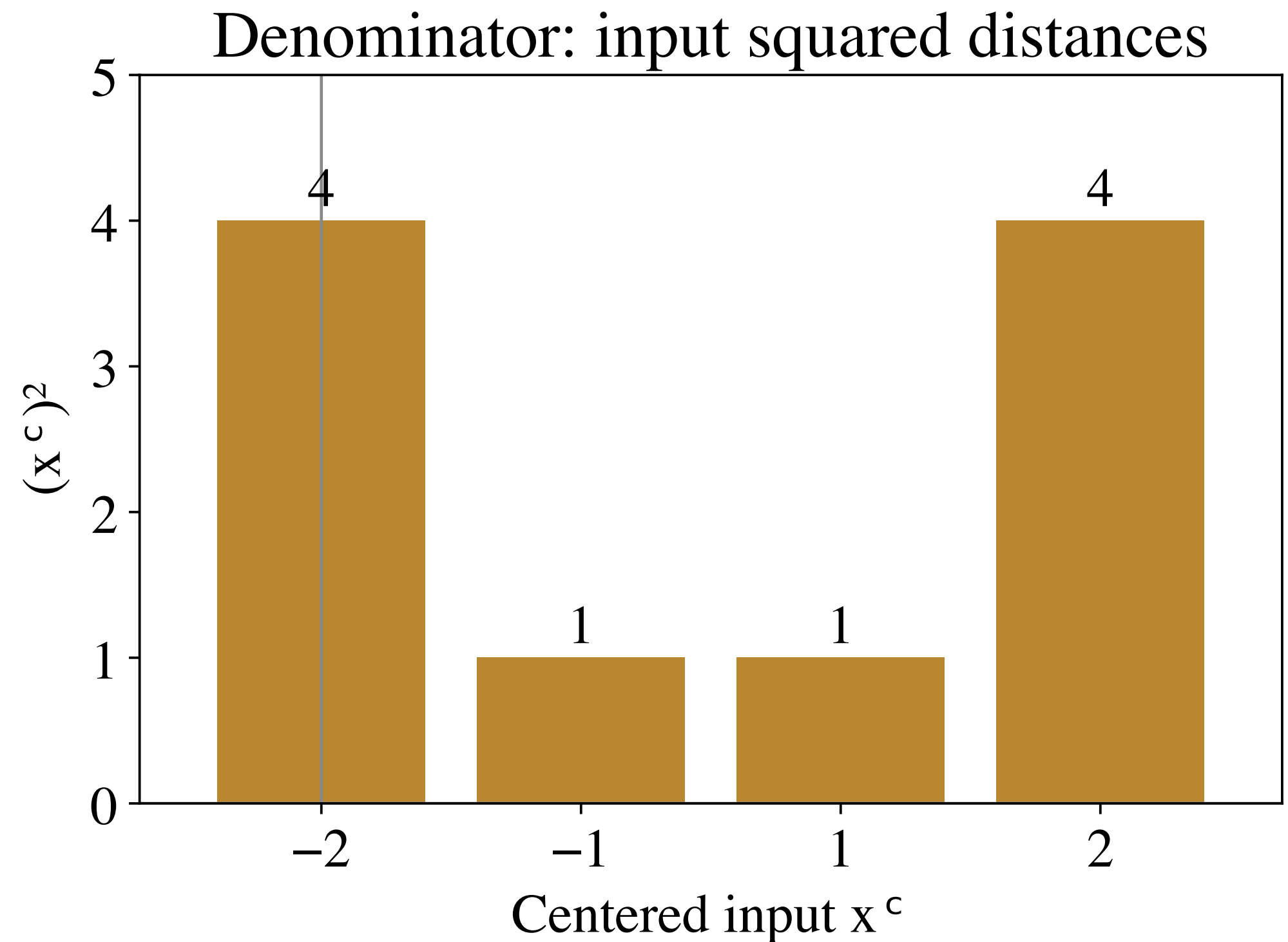
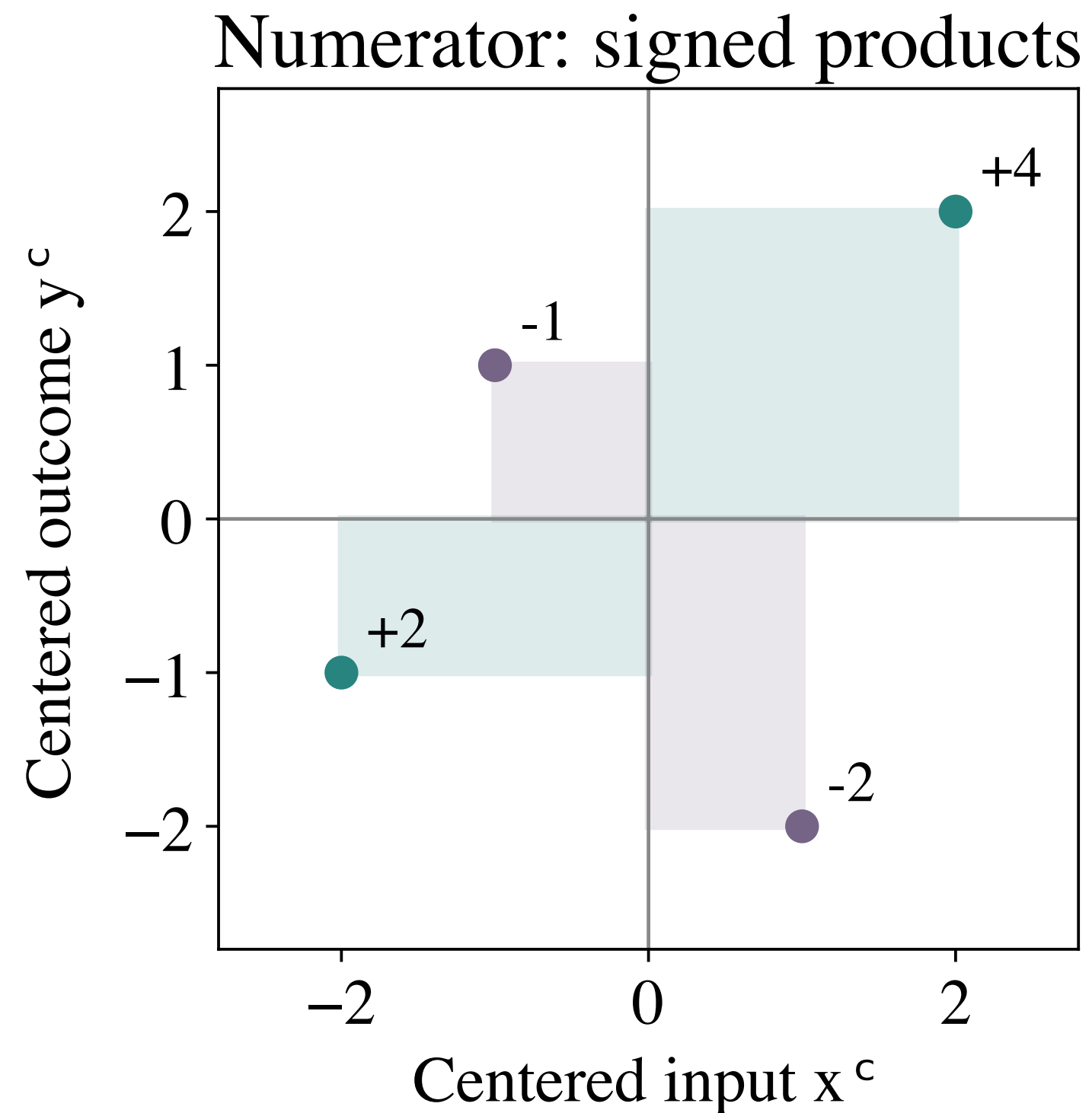
(Distribute each product)

(Take the shared slope outside the sum)

(Move the cross-products to the other side)

(Divide by the input sum of squares)

| Each observation contributes a **signed product** and a **squared distance**



Toy centered data: numerator = $2 - 1 - 2 + 4 = 3$; denominator = $4 + 1 + 1 + 4 = 10$.

Same-sign deviations support a positive slope; opposite signs oppose it. Here $\hat{w} = 3/10$.

| The fitted slope is a **weighted average of slopes from the mean**

For observations with $x_i^c \neq 0$:

$$\widehat{w} = \sum_i \frac{(x_i^c)^2}{\underbrace{\sum_j (x_j^c)^2}_{\text{weight, summing to 1}}} \underbrace{\frac{y_i^c}{x_i^c}}_{\text{slope from the mean point}} .$$

Inputs farther from the mean have **more weight** in determining the slope.

Dividing by $\sum_i (x_i^c)^2$ normalizes the weights. It is not an unweighted average of the individual slopes.

Finally, $\hat{b} = \bar{y} - \widehat{w}\bar{x}$ positions this slope through the mean point.

| Stacking predictions turns the problem into **one matrix equation**

Put observations in rows of $X \in \mathbb{R}^{n \times d}$.

$$\widetilde{X} = [X \ \mathbf{1}], \quad \boldsymbol{\theta} = \begin{bmatrix} w \\ b \end{bmatrix}, \quad \widehat{\mathbf{y}} = \widetilde{X}\boldsymbol{\theta}.$$

$$J(\boldsymbol{\theta}) = \frac{1}{2n} \|\widetilde{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2.$$

Shape check: $(n \times (d + 1))((d + 1) \times 1) = n \times 1$ predictions.

| Write squared error as an **inner product** before expanding

$$J(\boldsymbol{\theta}) = \frac{1}{2n} \|\widetilde{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2$$

(Squared-residual form)

$$= \frac{1}{2n} (\widetilde{X}\boldsymbol{\theta} - \mathbf{y})^\top (\widetilde{X}\boldsymbol{\theta} - \mathbf{y})$$

(A squared norm is an inner product)

$$= \frac{1}{2n} (\boldsymbol{\theta}^\top \widetilde{X}^\top - \mathbf{y}^\top) (\widetilde{X}\boldsymbol{\theta} - \mathbf{y})$$

(Transpose the difference)

| The expansion contains **two equal scalar cross terms**

$$2nJ(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \widetilde{X}^\top \widetilde{X} \boldsymbol{\theta} - \boldsymbol{\theta}^\top \widetilde{X}^\top \mathbf{y} \\ - \mathbf{y}^\top \widetilde{X} \boldsymbol{\theta} + \mathbf{y}^\top \mathbf{y}$$

(Multiply out all four terms)

$$\mathbf{y}^\top \widetilde{X} \boldsymbol{\theta} = (\mathbf{y}^\top \widetilde{X} \boldsymbol{\theta})^\top = \boldsymbol{\theta}^\top \widetilde{X}^\top \mathbf{y}$$

(A scalar equals its transpose)

$$2nJ(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \widetilde{X}^\top \widetilde{X} \boldsymbol{\theta} - 2\boldsymbol{\theta}^\top \widetilde{X}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y}$$

(Combine the cross terms)

| Differentiate each term, then set the gradient to zero

The matrix $\widetilde{X}^\top \widetilde{X}$ is symmetric.

Term in $2nJ$	Gradient with respect to θ
$\theta^\top \widetilde{X}^\top \widetilde{X} \theta$	$2\widetilde{X}^\top \widetilde{X} \theta$
$-2\theta^\top \widetilde{X}^\top y$	$-2\widetilde{X}^\top y$
$y^\top y$	0: independent of the coefficients

$$\begin{aligned} \nabla J(\theta) &= \frac{1}{2n} (2\widetilde{X}^\top \widetilde{X} \theta - 2\widetilde{X}^\top y + 0) && \text{(Combine the term derivatives)} \\ &= \frac{1}{n} (\widetilde{X}^\top \widetilde{X} \theta - \widetilde{X}^\top y) && \text{(Cancel the factor of two)} \\ 0 &= \frac{1}{n} (\widetilde{X}^\top \widetilde{X} \hat{\theta} - \widetilde{X}^\top y) && \text{(Set the gradient to zero)} \\ \widetilde{X}^\top \widetilde{X} \hat{\theta} &= \widetilde{X}^\top y && \text{(Multiply by n; rearrange)} \end{aligned}$$

| Independent columns give the **OLS inverse formula**

If $\widetilde{X}$ has full column rank:

$$\hat{\theta} = (\widetilde{X}^T \widetilde{X})^{-1} \widetilde{X}^T y.$$

- $\widetilde{X}^T y$ measures feature–outcome alignment.
- $\widetilde{X}^T \widetilde{X}$ records feature sizes and overlap.
- Its inverse accounts for that overlap when assigning coefficients.

In code, use a **least-squares solver**, rather than explicitly forming the inverse.

| At the solution, residuals are **orthogonal to every feature column**

Let $\mathbf{r} = \mathbf{y} - \widetilde{\mathbf{X}}\hat{\boldsymbol{\theta}} \in \mathbb{R}^n$: **one residual per observation, not per feature**.

Each column of $\widetilde{\mathbf{X}} \in \mathbb{R}^{n \times (d+1)}$ also has n entries: one feature measured across all observations.

$$\widetilde{\mathbf{X}}^\top \mathbf{r} = \widetilde{\mathbf{X}}^\top (\mathbf{y} - \widetilde{\mathbf{X}}\hat{\boldsymbol{\theta}})$$

(Substitute the residual vector)

$$= \widetilde{\mathbf{X}}^\top \mathbf{y} - \widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}}\hat{\boldsymbol{\theta}}$$

(Distribute the matrix product)

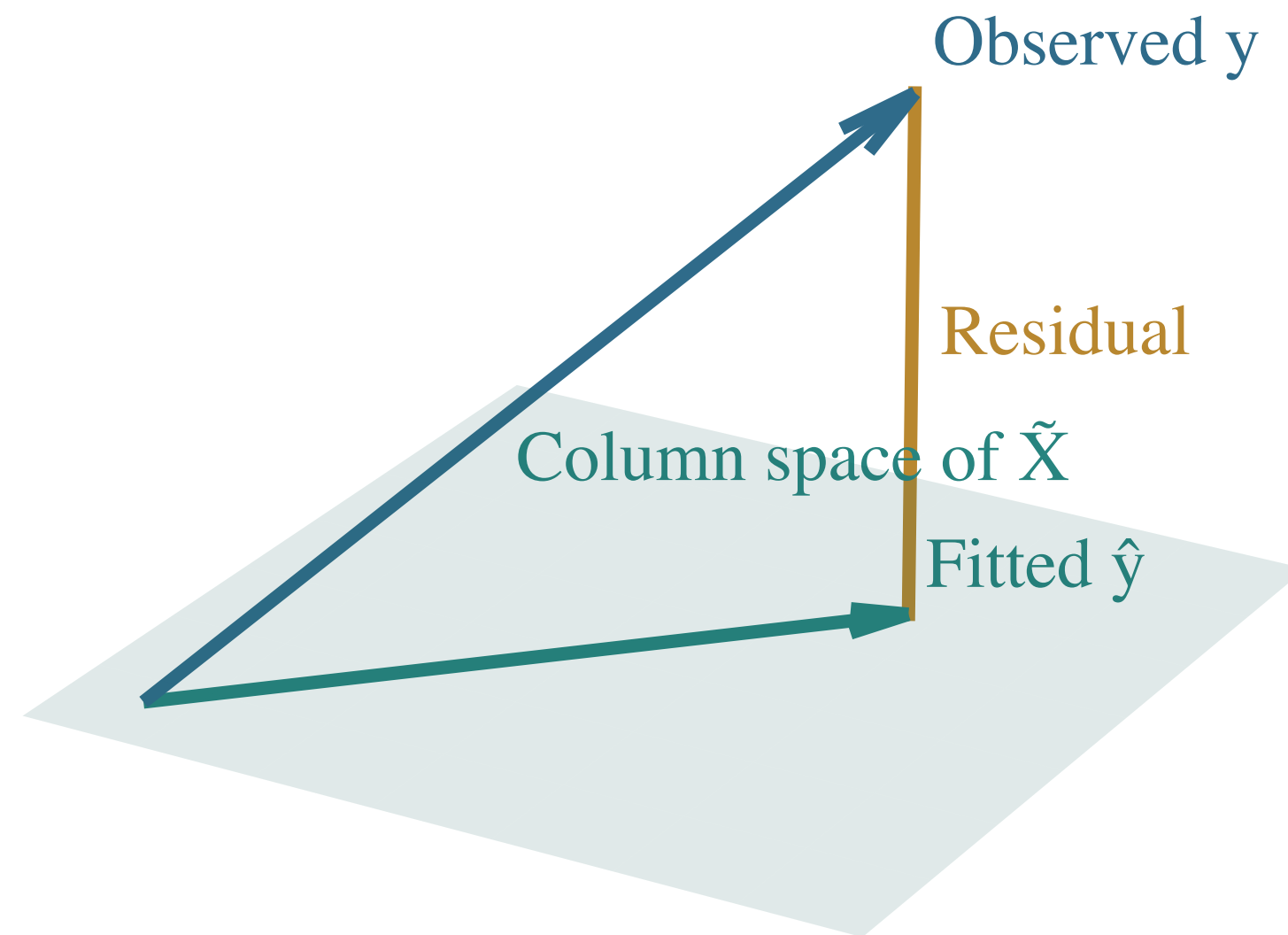
$$= \mathbf{0}$$

(Use the normal equations)

Each feature column has **zero dot product with the residual vector**: no residual component lies along a direction the model can adjust.

The column of ones also gives $\sum_i r_i = 0$: overpredictions and underpredictions balance.

| OLS projects the outcome vector onto **possible fitted vectors**



The plane represents $\text{col}(\tilde{X})$; the residual is perpendicular to it.

| Orthogonality proves that **no other fit has lower error**

- $\mathbf{r} = [r_1, \dots, r_n]^\top \in \mathbb{R}^n$: **one residual per observation**, $r_i = y_i - \hat{y}_i$.
- $\mathbf{r}$ has n **coordinates for observations**, not d coordinates for features like $\mathbf{x}_i$.
- **Recall OLS orthogonality**: $\widetilde{X}^\top \mathbf{r} = \mathbf{0}$; $\mathbf{r}$ is perpendicular to each n -**dimensional feature column**.

Any other coefficient vector is $\hat{\boldsymbol{\theta}} + \boldsymbol{\delta}$; its prediction change $\widetilde{X}\boldsymbol{\delta} \in \mathbb{R}^n$.

$$\begin{aligned}\|\mathbf{y} - \widetilde{X}(\hat{\boldsymbol{\theta}} + \boldsymbol{\delta})\|^2 &= \|\mathbf{r} - \widetilde{X}\boldsymbol{\delta}\|^2 \\ &= \|\mathbf{r}\|^2 + \|\widetilde{X}\boldsymbol{\delta}\|^2 \\ &\geq \|\mathbf{r}\|^2\end{aligned}$$

(Write the new residual)

$$(\mathbf{r}^\top (\widetilde{X}\boldsymbol{\delta}) = 0)$$

(A squared norm is nonnegative)

The bound is attained: choose $\boldsymbol{\delta} = \mathbf{0}$, so $\|\widetilde{X}\boldsymbol{\delta}\|^2 = 0$.

Thus $\hat{\boldsymbol{\theta}}$ achieves the smallest possible squared error, $\|\mathbf{r}\|^2$: it is a **global minimum**.

| Does zero training residual alignment imply perfect prediction?

The OLS solution satisfies $\widetilde{X}^T r = 0$.

Which conclusion follows?

- A. Every residual is zero.
- B. No linear coefficient adjustment reduces the training squared error.
- C. The model will have zero error on future houses.

B. A nonzero residual can be perpendicular to the entire prediction subspace.

Redundant features can give the same fit with different coefficients

Example: predict a venue's booking price from its room capacities (zero offset).

Feature	Seats	Model A coefficient	Model B coefficient
Room 1: x_1	20	1	0
Room 2: x_2	30	2	1
Total: $x_3 = x_1 + x_2$	50	0	1

$$\hat{y}_A = 1(20) + 2(30) + 0(50) = 80, \quad \hat{y}_B = 0(20) + 1(30) + 1(50) = 80.$$

Same for every venue: $x_2 + x_3 = x_1 + 2x_2$ whenever $x_3 = x_1 + x_2$.

The data cannot distinguish these coefficient choices.

The inverse formula requires **independent columns**; redundant columns make $\widetilde{X}^T \widetilde{X}$ singular.

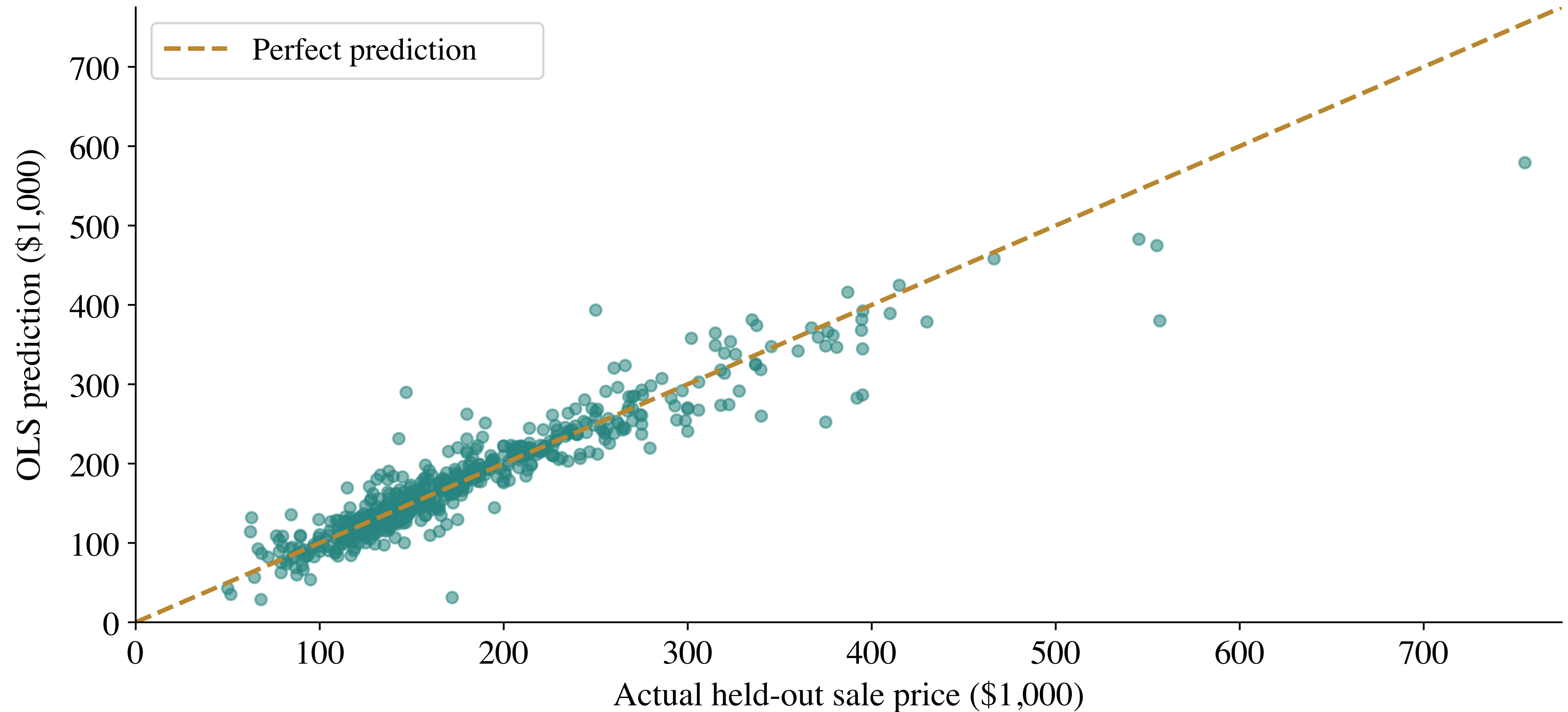
| On real Ames sales, **ordinary least squares is competitive**

Same split: **1,758 train / 586 validation / 586 test**; 226 encoded coordinates.

Model	Test RMSE ↓
Predict the training mean	\$78,500
Validation-selected KNN	\$32,800
Ordinary least squares	\$26,300
Random forest	\$25,700
Boosted trees	\$25,200

A compact linear rule is useful here even alongside flexible nonlinear methods.

| Held-out predictions reveal both useful structure and remaining errors



Each dot is one of **586 test sales**. The diagonal means prediction equals outcome.

| Solving the training problem leaves **the prediction claim to evaluate**

We can now explain: the function, the squared-error objective, the exact solution, and its geometry.

An exact fit to the objective does not make the model exact about the world.

Next: keep the linear score, but change from house prices to class probabilities.