TEMPORAL HIDE&SEEK: INSTANCE-WISE TIME-FEATURE SELECTION FOR TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Instance-wise feature selection seeks to identify inputs that are informative for individual predictions. For time series, this requires identifying not only *which* variables matter, but also *when* they matter. This creates distinct challenges: the relevance of an observation can depend on its temporal context, the relevant cells can change from one window to the next, and masking individual observations can disrupt the temporal structure the predictor relies on. We introduce Temporal Hide&Seek (T-HiSe), a sequence-aware method for instance-wise selection of time–feature pairs that learns a selector and a predictor jointly and end to end. A Transformer-based Hide network reads the full input window to produce a continuous mask over every time–feature pair, and an LSTM-based Seek network predicts from the masked sequence. Masked observations are replaced with complete feature trajectories drawn from strictly earlier training windows, preserving each feature's temporal structure while avoiding future information, and an annealed sparsity penalty sharpens the masks as training progresses. On eight synthetic benchmarks with known ground-truth supports, spanning fixed and context-dependent supports, T-HiSe attains the best or tied-best support recovery on seven and averages 97.2%, with the largest gains where the relevant support switches between windows. In two real-world case studies, hourly bike-sharing demand and EEG seizure detection, T-HiSe achieves predictive performance comparable to the reported references while assigning high mask values to a relatively small part of each window. Deleting these highly ranked cells reduces predictive performance substantially more than deleting the same number of random cells.

1 INTRODUCTION

A prediction from a multivariate time series can depend on different observations in different input windows. A demand forecast may depend on the hour of day and the last few hours of usage in one window, and on a sudden change in weather in another; a seizure detector may rely on a burst of high-amplitude activity wherever it occurs in the segment. A useful instance-wise explanation should therefore highlight both the variables and the time steps associated with a particular prediction.

Three properties make this harder for time series than for independent tabular records. First, cells cannot always be judged in isolation: a spike is only informative relative to its surrounding baseline, and a switch in the predictive mechanism may be triggered by a whole subsequence rather than a single value. Second, the relevant support is often *context dependent*, so the cells that matter in one window differ from those in the next. Third, explanation methods that remove or replace inputs must do so without producing sequences the predictor cannot interpret: replacing a masked observation with a constant or an independent random draw breaks the autocorrelation of the series and can produce out-of-distribution inputs (Hooker et al., 2019).

Selector-predictor methods address instance-wise selection by training a network that chooses the relevant inputs together with a predictor that uses them (Chen et al., 2018; Yoon et al., 2019; Jethani et al., 2021). Hide&Seek (Ellinson et al., 2026) made this joint training end-to-end differentiable with continuous masks and replaced unselected features with random draws rather than fixed ablation values. It was developed for independent samples and treats each input coordinate independently, both when selecting and when replacing. In many applications we instead want to identify the important variables *and* time steps of a time series.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We propose Temporal Hide&Seek (T-HiSe) for this setting. Our contributions are:

- **Sequence-aware instance-wise selection.** A Transformer-based Hide network (Vaswani et al., 2017) reads the full window and produces a mask over every time-feature pair, so that each cell's mask can depend on its temporal context; an LSTM-based Seek network (Hochreiter & Schmidhuber, 1997) predicts from the masked sequence.
- **Temporally structured replacement from earlier windows.** Masked cells receive complete feature trajectories from training windows that end before the target window, retaining the within-feature structure of real trajectories without using observations made after the prediction time.
- **Evaluation with and without ground truth.** On eight synthetic benchmarks with exact time-feature supports, including subsequence gates, correlated decoys and long-range cells, T-HiSe attains the best or tied-best recovery on seven of eight against eight baselines. On two real-world tasks, its selections show substantial deletion and retention gaps relative to random cells and align with plausible patterns in each domain.

2 Related Work

Post-hoc attribution for time series. Gradient-based methods such as Integrated Gradients (Sundararajan et al., 2017) attribute a prediction to its inputs by accumulating gradients along a path from a baseline, and perturbation-based methods such as LIME (Ribeiro et al., 2016) fit a local surrogate to the model's responses under random perturbations. Both were designed for static inputs and treat each time-feature cell as a separate input. Several methods adapt attribution to temporal structure. TimeSHAP (Bento et al., 2021) extends Kernel SHAP to recurrent models, estimating event-, feature- and cell-level Shapley values and pruning distant events. FIT (Tonekaboni et al., 2020) and WinIT (Leung et al., 2023) measure importance through the change in prediction when past observations are replaced, WinIT accounting for the dependence between successive observations of the same feature. DynaMask (Crabbé & van der Schaar, 2021) learns a perturbation mask for each instance under area and temporal-smoothness constraints, and Enguehard (2023) learns the perturbations themselves. All of these methods explain a fixed, pretrained model, so their explanations reflect both the attribution procedure and how fully that model has captured the underlying dependencies.

Learned instance-wise selection. A second line of work trains a selector that outputs the relevant inputs for each instance. L2X (Chen et al., 2018) maximizes the mutual information between a subset of k features and the model output using a continuous relaxation of subset sampling. INVASE (Yoon et al., 2019) trains a selector jointly with a predictor and a baseline network through an actor-critic objective, allowing the number of selected features to vary. Jethani et al. (2021) showed that jointly trained selectors and predictors can encode the prediction in the selection itself, and proposed REAL-X, which trains the predictor separately. Hide&Seek (Ellinson et al., 2026) learns continuous masks end to end and replaces unselected features with marginally sampled values rather than fixed ablation constants. T-HiSe keeps end-to-end joint training but is designed for sequences: its selector reads the whole window, its predictor is recurrent, and its replacements are real feature trajectories drawn from earlier windows.

Evaluating explanations. Because real-world data rarely come with known relevant inputs, time-series explanation methods are commonly evaluated on synthetic benchmarks with controlled ground truth (Ismail et al., 2020; Tonekaboni et al., 2020) and, on real data, by how much predictions change when the highest-ranked inputs are removed or retained (Crabbé & van der Schaar, 2021). Jethani et al. (2021; 2023) further argue for quantitative support measures alongside prediction quality. We use both protocols: exact support recovery on synthetic data, and deletion/retention tests with domain checks on real data.

3 TEMPORAL HIDE&SEEK

3.1 PROBLEM SETUP

We observe F features over time and write $\mathbf{x}_s \in \mathbb{R}^F$ for the feature vector at time s. To predict a target y_t, we use the window of the last τ time steps,

$$\boldsymbol{X}_t = (\mathbf{x}_{t-\tau+1}, \ldots, \mathbf{x}_t)^\top \in \mathbb{R}^{\tau \times F},$$

whose rows are time steps and whose columns are features. We call each entry of $\boldsymbol{X}_t$ a *cell* and write $\boldsymbol{X}_t^{(f)} \in \mathbb{R}^\tau$ for its f-th column, the trajectory of feature f within the window. The *support* S_t is the set of cells relevant to y_t. In the global setting it is the same for every window, relative to its end; in the instance-wise setting it can change from one window to the next.

T-HiSe trains two networks jointly. The Hide network h_ϕ outputs a mask with one value per cell,

$$\mathbf{M}_t = h_\phi(\boldsymbol{X}_t) \in [0,1]^{\tau \times F},$$

where a value near 1 keeps the observed cell and a value near 0 hides it. The Seek network s_ψ predicts y_t from the window after the hidden cells have been replaced. The goal is accurate predictions from as few kept cells as possible.

3.2 HIDE AND SEEK NETWORKS

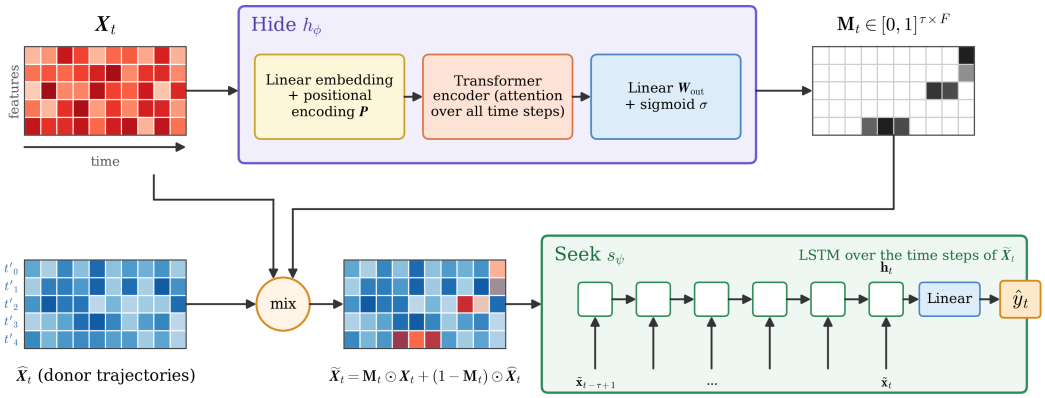

Figure 1: T-HiSe architecture. Hide produces a mask $\mathbf{M}_t$ from the input window $\boldsymbol{X}_t$. Each feature's hidden cells are filled from that feature's trajectory in another training window ($\widehat{\boldsymbol{X}}_t$), giving the mixed window $\widetilde{\boldsymbol{X}}_t$, from which Seek predicts $\hat{y}_t$. The grids show features as rows and time as columns for display.

Hide. Hide is a Transformer encoder (Vaswani et al., 2017) that treats each time step of the window as a token. The feature vectors are embedded linearly, a positional encoding $\boldsymbol{P}$ is added, and the encoder output is mapped back to one mask value per cell:

$$\mathbf{M}_t = \sigma\Big(\text{Transformer}\big(\boldsymbol{X}_t \boldsymbol{W}_{\text{in}} + \boldsymbol{P}\big)\, \boldsymbol{W}_{\text{out}}\Big),$$

where $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}_{\text{out}}$ are learned linear maps (bias terms omitted) and σ is the element-wise sigmoid. Row s of $\mathbf{M}_t$ gives the mask of every feature at time step s. Because attention spans the whole window, the mask of a cell can depend on its context, for example whether a value is large relative to its surroundings. Attention runs in both directions within the window. This is valid because the full window is available when predicting y_t, and no observation after time t is used.

Seek. Seek is an LSTM (Hochreiter & Schmidhuber, 1997) that reads the mixed window $\widetilde{\boldsymbol{X}}_t$ (Section 3.4) one row at a time and predicts from its final hidden state:

$$\boldsymbol{h}_s = \mathrm{LSTM}\big(\tilde{\mathbf{x}}_s,\, \boldsymbol{h}_{s-1}\big), \quad s = t-\tau+1, \ldots, t, \qquad \hat{y}_t = \boldsymbol{w}^\top \boldsymbol{h}_t + b,$$

where $\tilde{\mathbf{x}}_s$ is row s of $\widetilde{\boldsymbol{X}}_t$. For binary classification, $\hat{y}_t$ is a logit. Seek receives only the mixed window, never the mask itself. Layer sizes and other settings are listed in Appendix A.

Selection and the predictor. Because Hide and Seek are trained jointly, the pattern of selected cells could in principle carry information on its own (Jethani et al., 2021). T-HiSe limits this in two ways. Seek never sees the mask, and hidden cells are filled with real data rather than a fixed value that would mark them as removed. We also test every explanation after training by deleting and retaining the selected cells and comparing against random cells (Section 5).

3.3 Replacing hidden cells with real trajectories

Replacing hidden cells with a constant, or with an independent random value per cell, breaks the temporal structure of the series and gives Seek inputs unlike real data. T-HiSe instead fills them with a real trajectory: for each feature, it copies that feature's column from another randomly chosen training window. The donor window $\widehat{\boldsymbol{X}}_t$ is built column by column,

$$\widehat{\boldsymbol{X}}_t^{(f)} = \boldsymbol{X}_{t'_f}^{(f)}, \qquad t'_f \sim \mathrm{Uniform}\{\text{training windows ending before } t\}, \qquad f = 0, \ldots, F-1.$$

The hidden stretches of each feature therefore look like a genuine piece of that feature's time series. The donors t'_f are drawn independently for each feature and redrawn every epoch. Because every donor window ends before t, no value from after the prediction time can enter the input. The first training window has no earlier donor, so Seek receives its observed values. At test time, donors are drawn from the whole training set. When instances are separate short recordings rather than windows cut from one long series (as in the EEG study), there is no shared time order, and donors are simply random training segments.

3.4 Training objective

Seek receives the mixed window, which blends each observed cell with its donor cell according to the mask:

$$\widetilde{\boldsymbol{X}}_t = \mathbf{M}_t \odot \boldsymbol{X}_t + (\mathbf{1} - \mathbf{M}_t) \odot \widehat{\boldsymbol{X}}_t,$$

where $\odot$ is element-wise multiplication and $\mathbf{1}$ is the $\tau \times F$ matrix of ones. For a batch $\mathcal{B}$ of windows, both networks are trained end to end to minimize

$$\mathcal{L}(\phi, \psi) = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \mathcal{L}_{\text{pred}}\big(s_\psi(\widetilde{\boldsymbol{X}}_t),\, y_t\big) + \lambda_e\, \overline{\mathbf{M}}, \qquad \overline{\mathbf{M}} = \frac{1}{|\mathcal{B}|\,\tau F} \sum_{t \in \mathcal{B}} \sum_{s,f} (\mathbf{M}_t)_{s,f},$$

where $\mathcal{L}_{\text{pred}}$ is the mean squared error for regression or class-weighted binary cross-entropy for classification, and $\overline{\mathbf{M}}$ is the average mask value over all cells of all windows in the batch.

Annealed sparsity penalty. If the full penalty is applied from the start, the masks shrink towards zero before Seek has learned which cells are useful. We therefore increase the penalty weight over the epochs $e = 0, \ldots, E-1$:

$$\lambda_e = \lambda_{\max} \left(\frac{e}{E-1} \right)^q.$$

The penalty starts at zero, so Seek first learns to predict. It then grows to $\lambda_{\max}$ and pushes the masks to become sparse. A larger exponent q keeps the penalty weak for longer. The values of q, $\lambda_{\max}$ and all other hyperparameters are given in Appendix A.

4 Synthetic benchmarks

We construct eight synthetic datasets, each defined by a data-generating process with a known time-feature support S_t. To reduce the risk of designing benchmarks around the strengths of our own

method, the data-generating processes were generated by a large language model assistant independently of T-HiSe's development and locked in prior to model tuning, with no subsequent modifications. The datasets span two regimes: *global* datasets (syn1–syn2), in which the same support is relevant for every instance and which primarily evaluate recovery of a fixed support, and *instance-wise* datasets (syn3–syn8), in which a gate computed from part of the window determines which cells are relevant, so that a correct explanation must change from window to window. Two of the gates are aggregates over a subsequence rather than single cells, and syn8's gate compares the level of a feature in two adjacent blocks, so the switch depends on a local change in the series rather than on any single value.

Input process. All datasets share the same input process. With $F = 11$ features and $N = 10{,}000$ time steps, each feature follows an independent stationary AR(1) process,

$$x_{t,f} = \phi\, x_{t-1,f} + \epsilon_{t,f}, \qquad \epsilon_{t,f} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2), \qquad f = 0, \dots, F-1, \tag{1}$$

with $\phi = 0.7$ and $\sigma^2 = 1 - \phi^2$, so that each feature has unit stationary variance. Syn2 subsequently transforms some features, replacing one by a smoothed version and overwriting others with correlated decoys.

Windows and targets. Each instance is a window $\boldsymbol{X}_t$ of length $\tau = 30$, and

$$y_t = U_t + \eta_t, \qquad \eta_t \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1^2), \tag{2}$$

where U_t is a noise-free function of a small number of cells in the window. Targets are standardized using training statistics. Consecutive windows overlap in $\tau - 1$ time steps. Windows are split chronologically 80/20 into train and test for syn1–syn7; syn8 uses a chronological 70/15/15 split. Because windows overlap, in syn1–syn7 the first $\tau - 1$ test windows share raw observations with the last training windows; this concerns fewer than 2% of test windows and is identical for every method. In syn8 the unused validation block separates the training and test windows.

Ground-truth support. For the instance ending at time t, S_t is the set of pairs (s, f) whose values $x_{s,f}$ enter U_t. In syn1–syn2, S_t has the same lag structure for every t. In syn3–syn8, a gate g_t computed from feature 10 determines which branch is active, and S_t contains the gate cells together with the cells of the active branch. Table 3 in Appendix A.2 lists S_t explicitly for every dataset.

syn1 (global). Recent cells, an interaction, a saturating nonlinearity and a long-range cell:

$$U_t = 1.6\, x_{t,0} + 0.8\, x_{t,9}\, x_{t-1,3} + 1.2 \tanh\big(1.3\, x_{t-3,2}\big) + 0.9\, x_{t-7,5} + 1.0\, x_{t-22,7},$$

with $S_t = \{(t,0), (t,9), (t-1,3), (t-3,2), (t-7,5), (t-22,7)\}$.

syn2 (global, hard). Feature 2 is replaced by its exponentially smoothed version $\tilde{x}_{t,2} = 0.8\, \tilde{x}_{t-1,2} + 0.2\, x_{t,2}$, restandardized, and three decoys d are overwritten with 0.95-correlated copies of source features s, $x_{t,d} \leftarrow 0.95\, x_{t,s} + \sqrt{1 - 0.95^2}\, x_{t,d}$ for $(d, s) \in \{(1,0), (3,2), (7,6)\}$. These transformed features are the inputs seen by every method. Then

$$U_t = 1.5\, x_{t,0} + 1.2 \tanh\big(1.5\, \tilde{x}_{t-1,2}\big) + 0.9\, x_{t-2,4}\, x_{t-3,6} + 1.5 \max\big(x_{t-4,8} - 1,\ 0\big),$$

with $S_t = \{(t,0), (t-1,2), (t-2,4), (t-3,6), (t-4,8)\}$. The decoys are correlated with support features but do not enter U_t.

syn3 (binary switch). With $g_t = x_{t-1,10}$,

$$U_t = \begin{cases} 1.8\, x_{t,0} + 1.1 \tanh\big(x_{t-2,1}\big) & g_t < 0, \\ 1.6\, x_{t,2} - 1.2 \sin\big(0.8\, x_{t-1,3}\big) & g_t \geq 0, \end{cases}$$

and S_t contains $(t-1, 10)$ and the two cells of the active branch.

syn4 (three-way switch). With $g_t = x_{t-1,10}$ and standard-normal terciles $\pm c$, $c \approx 0.4307$,

$$U_t = \begin{cases} 1.6\, x_{t,0} + 1.0 \tanh\big(1.5\, x_{t-2,2}\big) & g_t < -c, \\ 1.5 \sin\big(1.2\, x_{t,4}\big) + 1.0\, x_{t-3,5} & -c \leq g_t < c, \\ -1.4\, x_{t-1,7} + 1.1 \tanh\big(x_{t-3,9}\big) & g_t \geq c. \end{cases}$$

syn5 (recent or long-range branch). With $g_t = x_{t,10}$,

$$U_t = \begin{cases} 1.7\, x_{t,1} + 1.1 \tanh\big(1.3\, x_{t-1,3}\big) & g_t \geq 0, \\ 1.5\, x_{t-1,4} + 0.8\, x_{t-20,6} & g_t < 0. \end{cases}$$

syn6 (gate with a direct effect). With $g_t = x_{t-6,10}$,

$$U_t = 1.5\, g_t + \begin{cases} 1.7\, x_{t,1} + 1.1\, x_{t-1,5} & g_t < 0, \\ -1.6\, x_{t,8} + 1.2\, x_{t-2,9} & g_t \geq 0. \end{cases}$$

syn7 (subsequence gate). With $g_t = \frac{1}{4}\sum_{k=3}^{6} x_{t-k,10}$,

$$U_t = \begin{cases} 1.7\, x_{t,2} + 1.0 \tanh\big(1.2\, x_{t-2,4}\big) & g_t \geq 0, \\ -1.5\, x_{t-1,6} + 1.3\, x_{t-9,7} & g_t < 0, \end{cases}$$

and S_t contains the four gate cells $\{(t-k, 10)\}_{k=3}^{6}$ plus the active branch.

syn8 (adjacent-block gate). The gate compares the mean of feature 10 over two three-step blocks,

$$g_t = \frac{1}{3}\sum_{k=2}^{4} x_{t-k,10} \;-\; \frac{1}{3}\sum_{k=7}^{9} x_{t-k,10}, \qquad U_t = 1.2\, g_t + \begin{cases} 1.6 \cdot \dfrac{1}{3}\displaystyle\sum_{k=0}^{2} x_{t-k,2}, & g_t \geq 0, \\ -1.6 \cdot \dfrac{1}{3}\displaystyle\sum_{k=0}^{2} x_{t-k,6}, & g_t < 0, \end{cases}$$

and S_t contains the six gate cells $(t-k, 10)$, $k \in \{2, 3, 4, 7, 8, 9\}$, plus the three cells of the selected trajectory.

4.1 Baselines and evaluation

We compare T-HiSe with eight baselines spanning post-hoc attribution and learned selection: Integrated Gradients (Sundararajan et al., 2017), LIME (Ribeiro et al., 2016), TimeSHAP (Bento et al., 2021), WinIT (Leung et al., 2023), DynaMask (Crabbé & van der Schaar, 2021), an adapted L2X (Chen et al., 2018) (L2X-adapted), INVASE (Yoon et al., 2019), and the original, non-temporal Hide&Seek (Ellinson et al., 2026) applied to each window flattened into a single τF-dimensional vector. The post-hoc methods and L2X-adapted explain an LSTM predictor with the same width as T-HiSe's Seek network, so they measure how well an attribution procedure recovers the support from a fixed, well-trained predictor; INVASE, Hide&Seek and T-HiSe train their own predictors and learn the selection jointly. All methods are scored on the same test windows against the same ground truth. For methods formulated with classification losses, we use mean squared error in the regression experiments. Further adaptations, all hyperparameters, and the number of explained windows per method are listed in Appendix A.4.

All methods are scored with the same top-k support-recovery metric. For each explained instance t, the $k_t = |S_t|$ highest-scoring cells under a method's mask or attribution are compared with S_t, and we report top-k recall, the fraction of S_t found among these cells (reported as TPR), averaged over instances and then over five training seeds (mean $\pm$ standard deviation). Because $k_t = |S_t|$, top-k recall equals precision and F1.

4.2 RESULTS

Table 1: Support recovery on the synthetic benchmarks. Each entry is the percentage of truly relevant cells that a method ranks among its top k_t cells, where $k_t = |S_t|$ is the size of the true support of that window; 100% means every relevant cell was found. Values are the mean ± standard deviation over five training seeds. Because every method selects exactly k_t cells, this top-k recall (TPR) also equals precision and F1. Avg. is the mean over the eight datasets. The best mean in each dataset column is bolded; ties are all bolded.

Method	SYN1	SYN2	SYN3	SYN4	SYN5	SYN6	SYN7	SYN8	Avg.
T-HiSe (ours)	**100.0 ±0.0**	**100.0 ±0.0**	**99.4 ±0.2**	**98.5 ±0.2**	**91.6 ±10.6**	**99.6 ±0.1**	90.2 ±3.3	**97.9 ±0.8**	**97.2**
Hide&Seek (Ellinson et al., 2026)	96.7 ±7.5	64.0 ±21.9	35.7 ±1.8	54.0 ±17.2	67.5 ±12.3	66.8 ±13.3	**90.6 ±4.5**	85.3 ±8.5	70.1
L2X-adapted (Chen et al., 2018)	86.7 ±18.3	80.0 ±0.0	94.1 ±0.6	34.4 ±19.0	67.1 ±12.1	85.6 ±5.4	68.7 ±5.8	71.0 ±24.7	73.4
INVASE (Yoon et al., 2019)	80.0 ±13.9	92.0 ±17.9	63.1 ±1.5	43.1 ±6.6	66.6 ±0.6	65.1 ±1.7	72.5 ±4.0	78.7 ±3.4	70.1
DynaMask (Crabbé & van der Schaar, 2021)	**100.0 ±0.0**	84.0 ±8.9	45.0 ±1.2	32.9 ±1.6	31.9 ±1.6	42.1 ±5.1	33.3 ±0.9	57.9 ±1.1	53.4
Integrated Gradients (Sundararajan et al., 2017)	**100.0 ±0.0**	**100.0 ±0.0**	79.3 ±5.8	49.7 ±4.1	62.7 ±4.9	76.8 ±8.4	36.6 ±5.3	67.7 ±2.9	71.6
LIME (Ribeiro et al., 2016)	**100.0 ±0.0**	**100.0 ±0.0**	72.6 ±1.9	55.0 ±4.2	60.2 ±3.5	73.6 ±3.6	40.1 ±1.5	66.0 ±2.0	71.0
TimeSHAP (Bento et al., 2021)	**100.0 ±0.0**	**100.0 ±0.0**	75.8 ±1.7	58.3 ±3.4	54.8 ±2.3	72.7 ±9.5	40.1 ±1.0	67.9 ±1.9	71.2
WinIT (Leung et al., 2023)	63.3 ±18.3	80.0 ±0.0	72.2 ±1.4	53.6 ±2.9	62.3 ±3.3	39.8 ±3.3	42.1 ±2.0	62.6 ±0.9	59.5

Findings. T-HiSe achieves the best or tied-best mean TPR on seven of the eight datasets and averages 97.2%, 23.8 points above the next-best method (L2X-adapted, 73.4%). The global datasets serve as a sanity check that every method can recover a fixed support, and several post-hoc methods are also perfect there. The separation appears on the instance-wise datasets, where the explanation must change with the gate: on syn3–syn6, T-HiSe's margin over the next-best method is 5.3 (syn3), 40.2 (syn4), 24.1 (syn5) and 14.0 (syn6) percentage points. Syn4, with a three-way switch, shows the largest gap: T-HiSe recovers 98.5% of the relevant cells, whereas no baseline exceeds 58.3%. T-HiSe is also markedly more stable across training seeds: its standard deviation is below one point on six of the eight datasets, whereas the learned-selection baselines vary by up to 25 points.

The subsequence gate (syn7). Syn7 is the only benchmark on which T-HiSe's mean TPR is lower than that of a baseline. Original Hide&Seek scores 90.6% and T-HiSe 90.2%; the difference is within the seed variation (standard deviations of 4.5 and 3.3 points), and both remain far ahead of every post-hoc method (at most 42.1%). Inspecting the masks shows that T-HiSe keeps the active branch cells but only part of the gate subsequence: because the gate averages four autocorrelated cells, a subset already carries most of the information needed to choose the branch, and the sparsity penalty favours keeping that subset. A smaller $\lambda_{\max}$ lowers the cost of each additional gate cell, and this remedy works: with $\lambda_{\max} = 2$, T-HiSe reaches $94.0 \pm 1.2\%$ on syn7, above its result in Table 1 and with a third of the seed variation, whereas original Hide&Seek at the same reduced penalty falls to $87.3 \pm 3.5\%$ (Appendix A.6). Table 1 nevertheless keeps one $\lambda_{\max}$ for all datasets.

5 REAL-WORLD CASE STUDIES

We apply T-HiSe to two real tasks in which both the inputs and the target are real and there is no ground-truth support: regression on hourly bike-sharing demand and classification of EEG seizures.

Without ground truth, an explanation can still be tested directly: if the selected cells are the ones the trained predictor relies on, then removing them should reduce predictive performance, while retaining them may preserve much of it. Following DynaMask (Crabbé & van der Schaar, 2021), we therefore *delete* the k highest-mask cells (replacing them with donor values while the rest of the window stays real) and, separately, *retain* only those k cells (replacing all others), for k between 2% and 40% of the window. Each test is repeated with the same number of randomly chosen cells, which provides the reference: informative rankings should produce larger gaps than random selection in both tests. We also examine whether the selections align with plausible domain patterns.

Both studies share a protocol. No target window shares raw observations with another split; standardization and all reference models use training data only, and validation and test replacements use donor trajectories sampled from the training split. T-HiSe uses the architecture of Section 3.2, with $\lambda_{\max} = 1$ to apply weaker sparsity pressure than in the synthetic experiments (Appendix A.7). The deletion and retention tests apply the same binary masks and the same donor draws to the mask-based and random selections, so the comparison isolates the effect of which cells are selected.

5.1 Hourly bike-sharing demand

Data and task. The UCI Bike Sharing dataset (Fanaee-T & Gama, 2014) records hourly rentals of the Capital Bikeshare system in Washington, D.C., in 2011–2012, together with weather and calendar information. We reindex it to a complete hourly grid of 17,544 hours; the 165 hours absent from the data (no recorded rentals, including system outages) are assigned zero rentals and the previous hour's weather. The target is the next-hour change in log rentals, $y_t = \log(1 + c_{t+1}) - \log(1 + c_t)$, predicted from a 30-hour window of 12 features: log total, casual and registered rentals, temperature, apparent temperature, humidity, wind speed, weather situation (clear, mist, rain/snow), working-day and holiday indicators, and the hour of day encoded as a sine and cosine. The split is chronological, with 12,230 training, 1,721 validation (from 27 May 2012) and 3,503 test windows (from 8 August 2012).

Prediction. T-HiSe reaches a test R^2 of 0.725 (Table 2), identical to a ridge regression that uses the entire flattened window, and more than double the 0.356 obtained by repeating the change observed 24 hours earlier; predicting no change gives $R^2 = 0$. T-HiSe attains this accuracy while concentrating each prediction on 28.5 of the 360 cells in a window (mask above 0.5), i.e. about 8% of the input: it matches a full-window model while highlighting a small set of cells the trained model emphasizes for its forecasts.

What is selected. The mask concentrates on the hour of day (mean mask 0.21 and 0.17 for the cosine and sine), registered-user rentals (0.14), total rentals (0.13) and the working-day indicator (0.11); temperature, humidity, wind and holidays receive at most 0.03. On past rentals the mask decays with lag: 0.99 at lag 1, 0.95 at lag 2, 0.68 at lag 3, 0.21 at lag 6 and essentially zero beyond lag 12. This pattern is consistent with commuting schedules among registered users and the importance of recent usage for hourly demand. The rentals 24 hours earlier are not selected (mask < 0.01) even though daily seasonality is strong: the hour-of-day and working-day features already encode the daily cycle, and T-HiSe selects one of the two equivalent sources of that information rather than both. The explanation also adapts to the weather context: the weather situation in the last three hours receives a higher mask after recent rain or snow than in dry conditions (0.50 vs. 0.42).

Faithfulness. Deleting only the top 2% of cells by mask (7 of 360) drops test R^2 from 0.725 to -0.67, worse than predicting no change at all, whereas deleting 2% random cells leaves it at 0.67, almost unchanged. Conversely, keeping only the top 20% of cells by mask gives $R^2 = 0.74$, at least as good as the full model, whereas keeping 20% random cells gives -0.78. At every fraction tested, mask-based selection is far better than random selection in both tests.

5.2 EEG seizure detection

Data and task. The Epilepsy dataset (Andrzejak et al., 2001) is a widely used real-world benchmark for EEG seizure detection. It contains 11,500 one-second single-channel EEG segments of 178 samples from 500 recordings, in five classes; following common practice, we classify seizure activity against the four other classes. We split *by recording*, so that segments of one recording never

Table 2: Faithfulness on the two real-world tasks (test split): R^2 for bike sharing, AUROC for EEG. *Delete* replaces the given fraction of cells with the highest mask by donor values (lower is better); *Retain* keeps only those cells and replaces the rest (higher is better). "Mask" ranks cells by T-HiSe's mask; "random" uses the same number of random cells. Full model: $R^2 = 0.725$ (ridge on the full window: 0.725), 28.5 of 360 cells above 0.5; AUROC $= 0.976$ (log-amplitude reference: 0.981), 33.8 of 178 steps above 0.5.

	Bike sharing (R^2)				EEG seizure (AUROC)			
	Delete		Retain		Delete		Retain	
Fraction	mask	random	mask	random	mask	random	mask	random
2%	-0.672	0.674	-0.479	-0.985	0.953	0.975	0.723	0.561
5%	-0.973	0.571	-0.205	-0.955	0.915	0.971	0.811	0.626
10%	-1.109	0.433	0.525	-0.892	0.840	0.963	0.915	0.686
20%	-1.035	0.168	0.735	-0.783	0.711	0.947	0.968	0.759
40%	-1.029	-0.332	0.743	-0.566	0.632	0.902	0.973	0.848

appear in two splits: 350 recordings (8,050 segments) for training, 50 (1,150) for validation and 100 (2,300) for testing, with seizure shares of 0.34 and 0.15 in validation and test. Seek outputs a logit, the prediction loss is class-weighted binary cross-entropy, and donors are whole training segments.

Prediction. T-HiSe reaches a test AUROC of 0.976 and accuracy of 0.923. A logistic model on the log amplitude (standard deviation) of each segment reaches AUROC 0.981, confirming that seizure segments are characterized by much larger amplitude; we use this well-established marker as the domain reference for what the masks should select. T-HiSe matches this reference to within 0.005 AUROC while, in addition, locating where in each segment the evidence lies. The mask keeps on average 33.8 of 178 time steps (mean mask 0.18), with similar counts for seizure (37.3) and non-seizure (33.2) segments. The class is therefore expressed by which steps are selected rather than by how many.

What is selected. In seizure segments the mask follows the signal amplitude: the within-segment correlation between mask and $|x|$ is 0.28, and the mean mask on the time steps in the top 10% of $|x|$ values is 0.41, more than twice the 0.19 on the remaining steps. In non-seizure segments the correlation is essentially zero (0.01), consistent with the absence of a discriminative burst. The selections are also temporally coherent: 94.7% of selected steps are adjacent to another selected step, yielding many contiguous stretches rather than only isolated samples, a pattern compatible with examining EEG activity over short intervals.

Faithfulness. Deleting the top 20% of time steps by mask reduces test AUROC from 0.976 to 0.711, whereas deleting 20% random steps leaves it at 0.947; at 40% the values are 0.632 and 0.902. Retaining only the top 10% of steps gives AUROC 0.915 against 0.686 for random steps, and the top 20% gives 0.968, within 0.01 of the full model, against 0.759. As for bike sharing, mask-based selection beats random selection at every fraction in both tests.

6 Conclusion

T-HiSe learns instance-wise time-feature masks from full input windows, jointly with a recurrent predictor, and replaces masked cells with real feature trajectories sampled only from earlier training windows. On eight synthetic benchmarks it attains the best or tied-best support recovery on seven and averages 97.2%, 23.8 points above the next-best method, with its largest gains where the relevant cells switch between windows. On hourly bike-sharing demand and EEG seizure detection it matches strong predictive references while assigning high mask values to a relatively small part of each window. The ranked cells align with plausible task-specific patterns, and deleting them harms the trained predictor more than deleting random cells at the tested fractions. Future work includes conditional replacements, benchmarks with variable-position temporal patterns, and comparisons with other explainers on real data.

AI USE STATEMENT

The data-generating processes of the synthetic datasets, including their functional forms and ground-truth supports, were generated by a large language model assistant (Claude, Anthropic) prior to model tuning and locked in without subsequent modification, to reduce the risk of designing the benchmarks around the strengths of T-HiSe. The assistant also helped write and debug experiment code, including data pipelines, baseline implementations and evaluation scripts; locate and summarize related literature and check bibliographic details; and edit the manuscript for clarity and consistency. It served as a sounding board for diagnosing results and planning additional checks. The authors made the final methodological and reporting decisions, reviewed AI-assisted code and text, and take full responsibility for the paper.

REPRODUCIBILITY STATEMENT

The T-HiSe implementation, the generators for all eight synthetic datasets (Section 4), the scripts for every baseline, and the two case-study pipelines, which download the public data, reproduce the splits and write every reported number, are provided in the anonymized supplementary material. All hyperparameters and split sizes are listed in Appendix A.

REFERENCES

Ralph G. Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001. doi: 10.1103/PhysRevE.64.061907.

João Bento, Pedro Saleiro, André F. Cruz, Mário A. T. Figueiredo, and Pedro Bizarro. TimeSHAP: Explaining recurrent models through sequence perturbations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2565–2573, 2021.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 883–892. PMLR, 2018.

Jonathan Crabbé and Mihaela van der Schaar. Explaining time series predictions with dynamic masks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2166–2177. PMLR, 2021.

Tal Ellinson, Hadi Mohasel Afshar, and Sally Cripps. Hide&Seek: Learning to explain in an end-to-end differentiable network. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. arXiv:2608.16689.

Joseph Enguehard. Learning perturbations to explain time series predictions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9329–9342. PMLR, 2023.

Hadi Fanaee-T and João Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2–3):113–127, 2014. doi: 10.1007/s13748-013-0040-3.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, pp. 9734–9745, 2019.

Aya Abdelsalam Ismail, Mohamed Gunady, Héctor Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6441–6452, 2020.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1459–1467. PMLR, 2021.

Neil Jethani, Adriel Saporta, and Rajesh Ranganath. Don't be fooled: Label leakage in explanation methods and the importance of their quantitative evaluation. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 8925–8953. PMLR, 2023.

Kin Kwan Leung, Clayton Rooke, Jonathan Smith, Saba Zuberi, and Maksims Volkovs. Temporal dependencies in feature importance for time series prediction. In *International Conference on Learning Representations*, 2023.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017.

Sana Tonekaboni, Shalmali Joshi, Kieran Campbell, David Duvenaud, and Anna Goldenberg. What went wrong and when? Instance-wise feature importance for time-series black-box models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2019.

A Hyperparameters

A.1 Common synthetic settings

All synthetic experiments use $N = 10{,}000$ time steps, $F = 11$ AR(1) features with $\phi = 0.7$, windows of length $\tau = 30$ and target noise standard deviation 0.1. Inputs are standardized per feature and targets globally, using training statistics only. Syn1–syn7 use a chronological 80/20 train/test split and syn8 a chronological 70/15/15 split; for syn8 the validation portion is not used by any method, and all methods are scored on the same test windows. Every method is run with the five training seeds 444352–444356.

A.2 Ground-truth supports

Table 3 lists the ground-truth support S_t of every synthetic dataset. For the instance-wise datasets, S_t is the union of the gate cells and the cells of the branch that is active in that window; in syn6 the gate cell also enters U_t directly and is counted once.

Table 3: Ground-truth supports S_t of the synthetic datasets. Cells are written (s, f) for time step s and feature f; $|S_t|$ is the number of cells scored per window.

Dataset	Gate cells	Other relevant cells (by active branch)	$\lvert S_t \rvert$
syn1	–	$(t,0), (t,9), (t-1,3), (t-3,2), (t-7,5), (t-22,7)$	6
syn2	–	$(t,0), (t-1,2), (t-2,4), (t-3,6), (t-4,8)$	5
syn3	$(t-1,10)$	$g_t < 0$: $(t,0), (t-2,1)$; $g_t \geq 0$: $(t,2), (t-1,3)$	3
syn4	$(t-1,10)$	$g_t < -c$: $(t,0), (t-2,2)$; $-c \leq g_t < c$: $(t,4), (t-3,5)$; $g_t \geq c$: $(t-1,7), (t-3,9)$	3
syn5	$(t,10)$	$g_t \geq 0$: $(t,1), (t-1,3)$; $g_t < 0$: $(t-1,4), (t-20,6)$	3
syn6	$(t-6,10)$	$g_t < 0$: $(t,1), (t-1,5)$; $g_t \geq 0$: $(t,8), (t-2,9)$	3
syn7	$(t-k,10)$, $k = 3, \ldots, 6$	$g_t \geq 0$: $(t,2), (t-2,4)$; $g_t < 0$: $(t-1,6), (t-9,7)$	6
syn8	$(t-k,10)$, $k \in \{2,3,4,7,8,9\}$	$g_t \geq 0$: $(t-k,2)$, $k = 0,1,2$; $g_t < 0$: $(t-k,6)$, $k = 0,1,2$	9

A.3 T-HiSe on the synthetic benchmarks

We use the annealing exponent $q = 2$ (quadratic annealing) in every experiment. At the midpoint of training the sparsity weight reaches a quarter of its final value, which gives Seek time to learn from the selected cells before the pressure towards sparsity becomes strong, while still leaving enough epochs at a high weight for the masks to become compact. The same exponent is used in the original Hide&Seek (Ellinson et al., 2026), whose sensitivity analysis of the annealing exponent (their Appendix D.7) found a linear schedule clearly worse than a quadratic one.

Table 4: T-HiSe hyperparameters on syn1–syn8. The same values are used for every dataset.

Component	Setting
Hide	Transformer encoder, 1 pre-norm layer, width 32, 4 heads, GELU feed-forward of width 128, dropout 0, sinusoidal absolute positions, final layer norm, sigmoid mask head
Seek	LSTM, 1 layer, width 32, linear output head
Replacement	training: per-feature donor trajectory from a uniform earlier window; test: per-feature donor trajectory from a uniform training window
Sparsity	$\lambda_{\max} = 9$ on the mean mask, annealed as $\lambda_e = \lambda_{\max}\,(e/(E-1))^q$ with exponent $q = 2$
Optimization	Adam, learning rate 10^{-3}, batch size 256, $E = 1000$ epochs
Hide attention	bidirectional within the input window
Seeds	444352, 444353, 444354, 444355, 444356

A.4 Baselines on the synthetic benchmarks

All baselines use the same data, splits, scaling, seeds and top-k metric as T-HiSe. The post-hoc explainers (Integrated Gradients, LIME, TimeSHAP, WinIT, DynaMask) and L2X-adapted explain the same black-box predictor, trained once per dataset and seed: a single-layer LSTM of width 32 with a linear head, trained for 1000 epochs with Adam (learning rate 10^{-3}) and batch size 256 on mean squared error. Table 5 lists the method-specific settings. Where a method explains a subset of test windows, the number is given in the last column; all others explain every test window.

Table 5: Baseline hyperparameters on syn1–syn8.

Method	Settings	Windows explained
Integrated Gradients	50 interpolation steps from an all-zero baseline (the training mean after standardization); cells ranked by attribution magnitude	all
LIME	500 perturbations per window, default kernel width, ridge surrogate with $\alpha = 1.0$	200
TimeSHAP	event-level Shapley values with 200 perturbations, recency pruning with $\eta = 0.05$, feature-level Shapley values within retained events with 200 perturbations, combined into cell scores; marginal-sampling counterfactuals in place of the original average-event baseline	200
WinIT	marginal-sampling variant: importance from replacing windows of 5 past observations of a feature, 5 counterfactual samples per window, decomposed into per-step contributions; replaces WinIT's learned conditional generator	200
DynaMask	preservation game with fade-to-moving-average perturbation; masks for keep ratios $\{0.01, 0.02, 0.03, 0.05, 0.075, 0.10\}$, 100 optimization epochs, SGD with learning rate 0.1 and momentum 0.9, size regularization 0.5 dilated to 100, no temporal regularization; per window, the smallest-area mask whose prediction error is below 0.1	all
L2X-adapted	replaces the learned variational approximator of original L2X with the frozen black-box predictor, so that the selector is trained to preserve the explained predictor's own output, as the post-hoc methods do: the black box's prediction on the masked input (unselected cells set to zero) is matched to its prediction on the full input; selector width 100, 1000 epochs, Adam 10^{-3}, batch 256, Gumbel-Softmax temperature annealed from 5.0 to 0.3, k equal to the support size (the maximum per-instance size where it varies)	all
INVASE	selector MLP on the flattened window (two ReLU layers of width 200) with predictor and baseline LSTMs (width 32) trained jointly on the labels with the actor-critic objective; unselected cells set to zero; 1000 epochs, batch 256, learning rates 10^{-4} (selector) and 10^{-3} (predictor, baseline), sparsity weight 5 on the mean selection probability	all
Hide&Seek	original method on the flattened τF window: Hide and Seek MLPs with two ReLU hidden layers of width 32, sigmoid mask; each cell replaced by an independent draw from its training marginal, resampled every epoch; 500 full-batch epochs, Adam 10^{-3}; quadratic annealing to $\lambda_{\max} = 5$ (syn1–syn7) or 9 (syn8) on the mean mask	all

A.5 Case studies

Both case studies use the T-HiSe architecture of Table 4 with $\lambda_{\max} = 1$, annealing exponent $q = 2$, $E = 300$ epochs, Adam (10^{-3}) and batch size 256. This lower sparsity weight applies less pressure to shrink the masks than the value used on the synthetic benchmarks; the trade-off is discussed in Appendix A.7.

Bike sharing. Window $\tau = 30$ hours, 12 features (Section 5.1); the test set is the last 20% of windows (3,503 windows) and the validation set the last 10% of windows before it (1,721 windows), with a gap of τ windows between splits so that no hour appears in two splits, leaving 12,230 training windows. Training donors are drawn from earlier training windows as described in Section 3.3; validation and test donors are drawn from the training windows. The ridge reference uses $\alpha = 10$ on the standardized, flattened training windows.

EEG. Segments of 178 samples, one channel, standardized with the training mean and standard deviation; split by recording into 350/50/100 recordings (8,050/1,150/2,300 segments). Donors are whole training segments drawn uniformly. The positive class is weighted by the ratio of negative to positive training segments in the binary cross-entropy.

A.6 SYN7 WITH A SMALLER SPARSITY WEIGHT

Table 6 compares T-HiSe and original Hide&Seek on syn7 at the sparsity weight used in Table 1 and at the reduced value $\lambda_{\max} = 2$. All other settings, the data, the seeds and the top-k metric are those of Section 4.1.

Table 6: Support recovery (TPR, %) on syn7, mean $\pm$ standard deviation over five training seeds.

Method	$\lambda_{\max}$ as in Table 1	$\lambda_{\max} = 2$
T-HiSe (ours)	90.2 ± 3.3 $(\lambda_{\max} = 9)$	$\mathbf{94.0 \pm 1.2}$
Hide&Seek (Ellinson et al., 2026)	90.6 ± 4.5 $(\lambda_{\max} = 5)$	87.3 ± 3.5

Lowering the penalty helps T-HiSe and hurts original Hide&Seek: T-HiSe gains 3.8 points and its seed variation falls from 3.3 to 1.2 points, while original Hide&Seek loses 3.3 points. This is consistent with the explanation in Section 4.2: with a cheaper price per selected cell, T-HiSe can afford to keep more of the four-step gate subsequence.

A.7 IMPLICATIONS OF THE SPARSITY WEIGHT

The final sparsity weight $\lambda_{\max}$ sets the trade-off between predictive accuracy and the size of the explanation. A smaller $\lambda_{\max}$ places less pressure on mask values to shrink, while a larger $\lambda_{\max}$ places more pressure on them and tends to produce more selective masks. Because the penalty acts on the mean mask, its scale is comparable across tasks with different window sizes, and the value that yields the most useful explanation depends on how the predictive information is distributed over the window. On the synthetic benchmarks we use a single value, $\lambda_{\max} = 9$, for all eight datasets, so T-HiSe's results in Table 1 involve no per-dataset tuning of the penalty. For both case studies we use $\lambda_{\max} = 1$, applying weaker sparsity pressure while retaining the predictive performance and deletion/retention results reported in Section 5.

Selecting the sparsity weight. The choice of $\lambda_{\max}$ should balance predictive performance, compactness and the informativeness of the selected cells. A stronger penalty can yield a smaller explanation, but may also remove useful predictive information; a weaker penalty allows more cells to receive substantial mask values. When choosing this weight on a new task, compare predictive performance with an unpenalized or full-window reference and evaluate deletion and retention against random selections, using validation data and keeping the other training settings fixed. Combining these checks with predictive accuracy identifies the settings whose selected cells provide the most useful explanation. The test set should be reserved for reporting the final selected configuration.