# SHIR: EVIDENCE-GUIDED EXPLANATIONS FOR TIME-SERIES PREDICTIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Explaining a time-series prediction requires identifying which parts of the input affect it. Perturbation responses recorded on reference inputs can guide query selection, but do not establish the effects on a new input. We introduce SHIR for frozen time-series predictors, separating unit scoring from query selection through evidence-guided experimental design. A language model proposes unit-effect comparisons from recorded evidence and an explicit account of which endpoint effects remain unmeasured. Under a fixed perturbation rule, SHIR selects the query batch whose endpoints would complete the most candidate comparisons within the budget. An evidence graph enables many-to-many evidence reuse: one execution record serves multiple applicable registered comparisons while retaining its input and configuration identity. Reference construction explores inputs in separate episodes and freezes the merged graph and reference scores. For each new input, SHIR preselects one batch of at most three perturbation queries and evaluates the separately perturbed inputs in parallel. Each perturbation starts from the same original input. Valid current effects replace the corresponding reference scores, while unmeasured units retain their defaults. On forecasting tasks built from WeatherBench2 and VitalDB, SHIR reduces mean sufficiency error by approximately 11.2% and 18.7%, respectively, relative to each task's strongest baseline for this metric among seven external methods. Our code is available anonymously[1].

## 1 INTRODUCTION

Explaining a time-series forecast requires identifying which parts of the observed history influence the prediction for that input. We consider a frozen predictor whose input is organized into predefined *units*, such as temporal blocks, variable groups, or spatial regions. We score and rank these units, and the top-$k$ units form the explanation at the requested size. Perturbation and attribution methods capture temporal dependence and construct explanatory masks (Crabbé & van der Schaar, 2021; Leung et al., 2023), and shared explanation representations and amortized explainers already exploit information across inputs (Queen et al., 2023; Jethani et al., 2022). We study how perturbation responses recorded on reference inputs can guide a small set of current perturbation measurements. Two questions follow: which current effects merit a limited query budget, and how existing evidence should be organized to inform that choice. The questions are coupled, because what to measure depends on what evidence exists, where it applies, and which observations are still missing.

Ranking units and selecting perturbation queries answer different questions. A unit's score is its measured effect or its reference score, whereas query selection decides which effects to measure. Suppose the reference ranking places unit $u$ inside the selected subset and unit $v$ outside it, while the current state suggests reconsidering their order. A high reference score for $u$ does not settle whether $v$ should outrank it. Allocating queries by expected effect magnitude alone can therefore leave a comparison that matters to the current selection without the observations it needs. Our central insight is to acquire the complementary observations that complete such unresolved, selection-relevant comparisons. Each comparison concerns the relative effects of two units under the same fixed perturbation rule.

[1] https://anonymous.4open.science/r/ICLR_27-F1E8/



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 introduce **SHIR**, which realizes this insight through **evidence-guided experimental design**. Building on experimental approaches to model interpretation (Shaham et al., 2024; Zablocki et al., 2026; Marin-Llobet & Ferrando, 2026), SHIR uses variable semantics and the current input state to guide a language model in proposing unit comparisons and their expected orderings. Domain knowledge motivates these proposals, but even a correct domain relationship need not describe the frozen predictor. A shared comparison representation therefore connects the steps before and after generation. Before generation, comparison-oriented context construction organizes recorded evidence around unresolved comparisons that could affect unit selection and identifies their unmeasured endpoints. After generation, complementary query selection chooses the batch whose endpoints would complete the most candidate comparisons within the budget. Valid responses to these queries determine the numerical orderings, which check the proposals against the predictor.

Comparisons can share observations even when they originate from different proposals. For example, comparisons of $u$ with $v$ and of $v$ with $w$ share the response for $v$, so three distinct queries complete both. Exploiting this sharing requires separating why an experiment was proposed from where its result can be used. SHIR provides this separation through **many-to-many evidence reuse**, drawing on work on active evidence states (Ren et al., 2026; Chen et al., 2026). An evidence graph links registered comparisons to the execution records that supply their endpoint observations. One record can serve every applicable comparison while retaining its input and configuration identity, and it counts as a single observation in score aggregation. These links inform subsequent query planning and comparison updates.

The same exploration round serves both stages. Reference construction explores each reference input in a separate episode that uses only its own responses. It then merges the episode graphs, aggregates their records into reference scores, and freezes both. For a new input, SHIR preselects one batch of at most three perturbation queries. Valid current effects replace the corresponding reference scores, and units without a valid current effect keep their reference scores. The predictor and the shared reference remain fixed.

We evaluate SHIR on forecasting tasks built from WeatherBench2 and VitalDB. On both tasks, SHIR achieves the best reported means on all four explanation metrics against the seven external baselines, reducing mean sufficiency error by approximately 11.2% and 18.7%, respectively, relative to the best baseline for this metric. Ablation studies support the role of each stage in explanation quality: context construction and query selection, evidence sharing during reference construction, and current measurements with per-input score updates.

Our contributions are:

- **Evidence-guided experimental design:** a shared unit-comparison representation connects comparison-oriented context construction with complementary query selection, so LLM-proposed comparisons become budgeted perturbation queries whose measured effects check the proposed orderings.
- **Many-to-many evidence reuse:** an evidence graph lets one input- and configuration-specific execution record serve every compatible registered comparison without being counted more than once, in both reference construction and per-input explanation.

## 2 Related Work

### 2.1 Post-hoc Explanations for Time-Series Models

Time-series explanation adapts feature attribution to temporal dependence and structured perturbations. Removal-based methods differ in how inputs are replaced, which model behavior is examined, and how the resulting effects are summarized (Covert et al., 2021). WinIT compares removals over nested temporal intervals, TimeSHAP estimates sequence coalition attributions with temporal pruning, and WinTSR rescales perturbation effects using temporal relevance (Leung et al., 2023; Bento et al., 2021; Islam & Fox, 2025). Mask-based methods optimize explanatory masks, with some also learning replacement signals: Dynamask learns masks under dynamic perturbations, Learning Perturbations jointly learns masks and replacements, and ContraLSP combines contrastive replacement learning with sparse gates (Crabbé & van der Schaar, 2021; Enguehard, 2023; Liu et al., 2024b). TimeX++ combines compactness, distributional constraints, and prediction consistency; ORTE for-

mulates redundancy and completeness through conditional mutual information; and TIMING incorporates temporal structure into integrated gradients (Liu et al., 2024a; Yue et al., 2025; Jang et al., 2025). SHIR keeps the perturbation rule and effect definition fixed and focuses on acquiring and reusing the observations needed to update an input-specific unit ranking.

### 2.2 LLM-Guided Model Analysis and Experimental Design

External feedback supports LLM-driven testing and revision: AdaTest combines generated tests with human feedback, and CRITIC uses external tools to critique and revise LLM outputs (Ribeiro & Lundberg, 2022; Gou et al., 2024). For model interpretation, MAIA conducts iterative experiments on target models, GIFT turns local visual counterfactuals into global textual hypotheses and tests them through concept interventions, and InterpAgent refines competing feature explanations using matched controls and multiple evaluation criteria (Shaham et al., 2024; Zablocki et al., 2026; Marin-Llobet & Ferrando, 2026). For scientific hypotheses, POPPER develops sequential falsification with error control under explicit statistical assumptions (Huang et al., 2025).

Decision-oriented active learning selects observations for downstream decisions (Javdani et al., 2014), and BatchBALD accounts for dependence among queries through joint mutual information (Kirsch et al., 2019). SHIR connects LLM-proposed unit comparisons to a limited budget of perturbation queries under a fixed rule. The same comparison representation organizes the evidence supplied before generation and the missing observations acquired afterward.

### 2.3 Reusable Explanations and Structured Evidence

Explanation methods reuse representations, event summaries, and explainer parameters. TimeX learns behavior-consistent explanation representations and shared landmarks; L2GTX consolidates local temporal events into class-wise summaries by selecting representative instances under a coverage budget; and FastSHAP amortizes Shapley-value estimation into a learned explainer (Queen et al., 2023; Mekonnen et al., 2026; Jethani et al., 2022). In L2GTX, the coverage budget selects representatives of existing explanations, whereas SHIR allocates its query budget to new predictor observations that complete proposed comparisons.

Structured evidence also guides the actions of research and reasoning agents. The EviGraph of Ren et al. (2026) maintains an active research state for experiment selection and dependency repair, with restrictions on inheriting historical findings. The separate EviGraph of Chen et al. (2026) exposes unresolved information-seeking constraints and validates source-linked evidence while controlling duplication. NeSy-CSA reuses validated dependency graphs of subtasks across critical samples, combining symbolic execution with neural inference constrained by preceding evidence (Chu et al., 2026). Their graphs organize research claims, source evidence, or executable task dependencies, and they are direct precedents for operational evidence structures. SHIR reuses predictor execution records that retain their input and configuration identity, and each record's applicability to registered comparisons informs current query selection and comparison updates.

## 3 Methodology

### 3.1 Problem Setup and Overview

SHIR explains a prediction by ranking predefined input units, such as temporal blocks, variable groups, or spatial regions; the highest-ranked units form the explanation at each requested size. The predictor $f$ is frozen throughout, and each perturbation query evaluates it on an input constructed by modifying one predefined unit of the current input $x$, together with its prescribed derived-input updates. The explanation size is set separately from the number of predictor queries used to produce it.

We compare units using a fixed perturbation rule. For unit $u$, let $T_u(x)$ be the input produced by the dataset adapter, which determines the replacement values and any updates to derived inputs. With a fixed normalization scale $\sigma_c > 0$ for output component $c$, the unit's fixed-perturbation effect magnitude is

$$\boxed{a_x(u) = \max_c \left| \frac{f_c(T_u(x)) - f_c(x)}{\sigma_c} \right|.} \tag{1}$$

Here, $f_c(x)$ denotes the $c$-th explained scalar output, including the task-specific output aggregation described in Appendix B.1. The nonnegative effect $a_x(u)$ enters unit comparisons and, when measured on the current input, serves as that unit's final explanation score; the original and perturbed predictions are also retained, so signs and individual output changes remain available. The LLM chooses which units to compare, while the adapter sets their replacement values and predictor responses determine their measured effects.

SHIR first builds reusable evidence from reference inputs, then uses it to guide a small batch of measurements for each new input. Both stages share one exploration round, which proposes comparisons (§3.2), selects their required queries (§3.3), and shares the resulting observations (§3.4); §3.5 describes how each stage uses this round.

### 3.2 Comparison-Oriented Context Construction

The LLM's comparison context combines the current input description, unit semantics, and selected comparisons with their available evidence, while identifying which endpoint effects remain unmeasured. Comparisons lacking current measurements receive priority. When reference scores are available, the working ranking uses measured current effects where available and reference scores elsewhere, and comparisons crossing a requested top-$k$ cutoff receive further priority. Other admissible pairs remain eligible. A reference-input episode has no reference scores, so it starts from the input description and its own observations and assigns no full ranking to unmeasured units.

For each retrieved pair, the context presents its relevance to the current selection, matching current observations, compatible reference summaries, and the endpoints that remain unmeasured. A reference paired difference is included only when both units were observed on the same reference input. The complete current-execution table remains available independently of the displayed comparisons, so an observed endpoint counts as measured even when a comparison link is missing. Reference evidence can motivate a comparison, whereas only a current observation can fill a missing endpoint.

From this context, the LLM generates comparison proposals, each containing unit endpoints, a proposed ordering, a rationale, and evidence references; a proposal can revisit an existing comparison or suggest a new one. Proposal mapping, labeled Grounding in the implementation, translates these fields into the unit catalog and output contract while preserving the proposed meaning, and new proposals enter candidate formation only as the admissible pairs it extracts. Mapping records the proposed ordering without adjudicating it; predictor responses determine the numerical result once the required measurements are available.

Let $D^{\mathrm{read}}$ contain the existing comparisons retained during context construction, and let $D^{\mathrm{new}}$ contain the admissible unordered unit pairs extracted from mapped proposals. Both sets contain pairs of distinct units under the fixed configuration. Let $M$ contain the units with valid measurements on the current input; an attempted query that yields no valid effect leaves its unit outside $M$. The pending candidate set is

$$D = \left\{ d \in D^{\mathrm{read}} \cup D^{\mathrm{new}} : d \not\subseteq M \right\}. \tag{2}$$

The union counts each unordered pair once, while expected orderings, rationales, and provenance remain in the associated proposal records. Retained existing pairs need no further LLM proposal, other existing pairs can re-enter through an admissible mapped proposal, and pairs with both current observations are excluded from the query objective. The merged set is used directly, without further candidate ranking or truncation.

The admissible unmeasured candidate endpoints form $U = (\bigcup_{d \in D} d) \setminus M$. If $|U|$ is smaller than the query budget $b = 3$, SHIR permits at most one replacement proposal set, generated and mapped using the same pre-execution information. Retained pairs are merged with admissible pairs from the replacement proposals to recompute and fix $D$ and $U$ before batch selection; Appendix A.4 specifies the detailed handling.

### 3.3 Complementary Query Selection

A comparison may require two measurements, and several comparisons may share one, so SHIR selects queries jointly. Using $D$, $M$, and $U$ from §3.2, it chooses a batch of unmeasured candidate units that completes as many pending comparisons as possible:

$$\boxed{Q^\star \in \operatorname*{arg\,max}_{Q \subseteq U,\ |Q| \le b} \sum_{d \in D} \mathbf{1}\{d \subseteq M \cup Q\}\,.} \tag{3}$$

A candidate comparison is counted when the measured units in $M$ and the endpoints selected in $Q$ together cover both of its units. The budget counts distinct unit queries, whereas the objective counts candidate comparisons whose missing endpoints the batch would supply; a numerical outcome is recorded only after both required responses are valid. Decision relevance enters earlier, through context construction and proposal generation. The completion count is a planning objective: completed comparisons may share observations, so the count measures neither independent information nor explanation fidelity.

For example, let $D = \{\{u, v\}, \{v, w\}, \{p, q\}\}$, with no current measurements and a query budget of three. Querying $\{u, v, w\}$ completes two comparisons, whereas $\{u, p, q\}$ completes only one. The first batch is preferred because the measurement of $v$ serves both comparisons, and this preference requires no advance knowledge of their response values.

For a comparison with two unmeasured endpoints, neither query alone completes it, but the pair does, so the objective is not submodular in general and standard submodular coverage guarantees do not apply. SHIR instead enumerates feasible batches in the small candidate pool to optimize Eq. (3) exactly. Adding an eligible unmeasured unit cannot reduce the completion count, so the batch uses the full budget when enough candidate endpoints exist, uses all available endpoints otherwise, and ends exploration of the input when none remain (Appendix A.4.2 gives the exact batch-size and tie rules).

All queries are selected before any response from the batch is observed. Each perturbed input is constructed separately from the same original input, and its prediction is compared with the cached prediction $f(x)$. The selector determines which endpoint observations to obtain, and the valid measured effects determine the comparison outcomes.

### 3.4 Many-to-Many Evidence Reuse

The evidence graph links registered comparisons to the executions that supply their observations. The record store determines which observations exist, and comparison links organize how they are used (Figure A1).

Let $G$ be the working evidence graph, and let $e(x, u)$ denote a valid execution record for unit $u$ on input $x$, compatible with the frozen predictor–intervention configuration and storing its predictions, response, and provenance. Records are deduplicated by execution identity; Appendix A.1 lists the matched fields. A unit belongs to $M$ exactly when such a current record exists, independently of explicit comparison links. The current records applicable to comparison $d$ are

$$\mathcal{E}_x(d) = \{e(x, w) : w \in d \cap M\}\,. \tag{4}$$

One record can serve several applicable registered comparisons, whichever proposal requested it, and aggregation counts it only once. Eq. (4) specifies applicability, and graph updates maintain the explicit links. Reference records from other inputs inform context construction and aggregation, whereas $\mathcal{E}_x(d)$ contains only records on the current input $x$.

After each batch, SHIR stores valid responses, including zero effects, refreshes $M$, and updates every affected registered comparison. When both matching current effects are available, their measured ordering or tie is recorded alongside the proposed ordering; otherwise, the comparison remains pending. The records remain valid whether the measured ordering agrees or disagrees with the proposal, and the check concerns the ordering rather than the full semantic rationale.

After the reference-input episodes finish, their graphs are merged by aligning comparison patterns while preserving each record's input and configuration identity. Reference scores use unique valid execution records under the same primary operation and configuration. Let $m_{u,c}$ be the median of the group-level absolute normalized component responses for unit $u$ and output component $c$, and let $n_u$ count reference groups with a valid observation of $u$. A group is one WeatherBench2 window or one VitalDB patient; for VitalDB, absolute normalized component responses are first reduced by their median across windows within each patient, and the patient-level values then enter the across-patient median $m_{u,c}$. The reference score is

$$s^{\mathrm{g}}(u) = \max_c \left[ \frac{n_u}{n_u + \kappa} m_{u,c} + \frac{\kappa}{n_u + \kappa} \mu_c \right], \qquad n_u > 0, \quad \kappa = 5. \tag{5}$$

Here, $\mu_c = \mathrm{median}_{v:n_v>0}\, m_{v,c}$ is the empirical shrinkage target, with equal weight for each unit having reference observations, and a unit with $n_u = 0$ receives $s^{\mathrm{g}}(u) = \max_c \mu_c$. Component-wise aggregation and shrinkage precede maximization; Appendix A.2 gives the full deduplication and group-reduction procedure. The merged graph supplies evidence for subsequent proposals, and the reference scores initialize per-input explanation.

### 3.5 Reference Construction and Per-Input Explanation

Reference construction repeats the round in §§3.2–3.4 for each reference input. Each episode starts with a separate empty graph, caches its original prediction, and uses only its own responses until its budget is exhausted or no admissible unmeasured endpoint remains. The finished graphs are merged and their records aggregated as described above, and the shared graph, reference scores, and execution settings are then frozen before test explanation.

A new input starts without current measurements. SHIR constructs its comparison context from the frozen reference, selects one batch of at most three perturbation queries as in §3.3, and evaluates the perturbed inputs in parallel. The per-input score update R0 then replaces reference scores with valid current effects and retains them elsewhere:

$$\boxed{\widehat{s}_x(u) = \begin{cases} a_x(u), & u \in M, \\ s^{\mathrm{g}}(u), & u \notin M. \end{cases}} \tag{6}$$

For $u, v \in M$, the updated score difference satisfies $\widehat{s}_x(u) - \widehat{s}_x(v) = a_x(u) - a_x(v)$. Every valid current effect updates its unit's score, including when a comparison remains incomplete or its outcome disagrees with the proposed ordering: comparisons guide query selection, and measured effects determine the updated scores.

We sort the resulting scores once and use their prefixes for all requested explanation sizes; Section 4 evaluates the fidelity of these fixed rankings. Current records and score updates remain local to the input, and the predictor and shared reference stay frozen. Appendix A details query accounting and exceptional cases, and Algorithm A1 in Appendix A.9 gives the shared exploration procedure.

Figure 1 illustrates query selection, evidence sharing, and the per-input score update on a concrete case.

## 4 Experiments

We evaluate whether SHIR can use reference perturbation evidence and three current perturbation measurements to produce faithful explanations. We first compare overall explanation quality with seven attribution and mask-based baselines, then ablate comparison-oriented context construction and query selection, evidence linking during reference construction, and per-input score updates.

### 4.1 Experimental Setup

We instantiate forecasting tasks on WeatherBench2 (Rasp et al., 2024) and VitalDB (Lee et al., 2022). On WeatherBench2, a frozen ConvGRU predicts the 2 m temperature field at 6, 12, and 24 h. We explain the cosine-latitude-weighted regional mean at each horizon using 90 units defined by temporal segments, variable groups, and spatial regions.

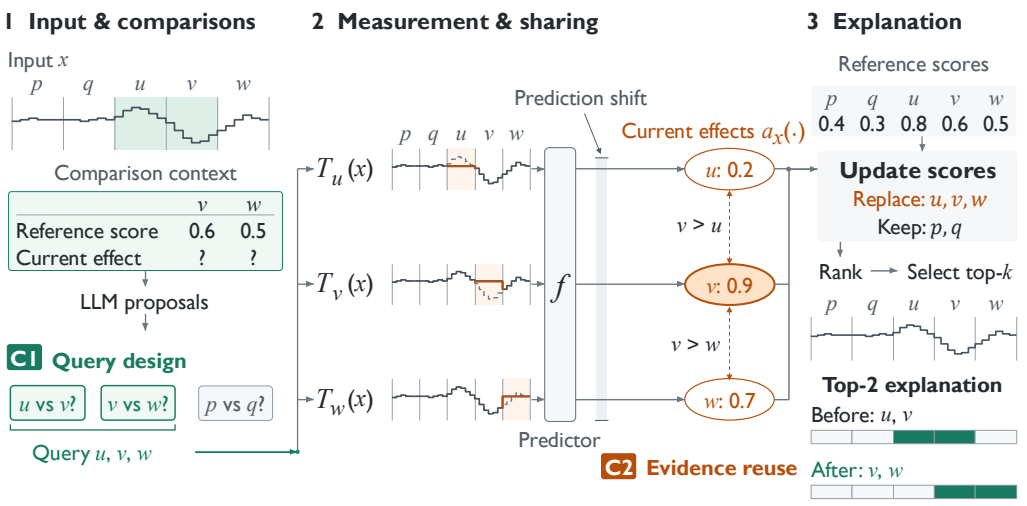



Figure 1: **Illustrative case with hand-specified values, not experimental results.** (1) The comparison context pairs per-unit reference scores with missing current effects; among candidates $u/v$, $v/w$, and $p/q$, querying $u, v, w$ completes the first two. (2) All three perturbations start from the same $x$, and Eq. (1) against the cached $f(x)$ yields three distinct effects; the single $v$ record serves both comparisons. (3) Every valid effect replaces its unit's reference score, regardless of comparison completion or agreement with the proposed ordering, while $p, q$ keep theirs; the top-2 explanation changes from $u, v$ to $v, w$. Reference-score Top-3 also selects $u, v, w$, so the case shows neither a strategy difference nor a fidelity gain. Appendix A.8 gives the full settings.

On VitalDB, a frozen causal temporal convolutional network (TCN) uses five minutes of history to predict mean arterial pressure (MAP) medians over 60-second windows ending at 1, 3, and 5 min ahead. Its 20 intervention units combine five one-minute blocks with four variable groups (MAP, HR, $ETCO_2$, and $SpO_2$), while other inputs remain contextual. Appendix B.1 gives the detailed architectures and unit catalogs.

Predictor training, reference construction, and explanation evaluation use separate data partitions, split chronologically for WeatherBench2 and by patient for VitalDB. For each task, SHIR constructs its reference from 200 fixed validation windows and evaluates explanations on 50 fixed test windows. The weather evaluation windows have nonoverlapping input-to-target intervals, and the VitalDB evaluation windows come from 50 distinct patients. Predictors are frozen before reference construction, and the reference is frozen before per-input explanation. Appendix B.1 gives the full partitions and sampling details.

**SHIR implementation details.** Proposal generation and proposal mapping use `deepseek-flash-v4.1`, with reasoning effort `max` for generation and reasoning disabled for mapping. Reference construction allows at most four rounds of up to three queries per reference input, with no further reference-construction queries after merging. Each new input uses one preselected batch of at most three perturbation queries, followed by R0. Appendix A.6.4 lists the request settings, Appendix A.7 reproduces the prompts, and Appendix A.4 covers candidate bounds, ties, and repairs.

**Fixed interventions.** Weather queries replace selected values with seasonal climatology computed from training data. VitalDB queries use, for each selected base variable, the median of its quality-valid values in the current five-minute input after excluding the selected one-minute block. A frozen training median serves as the fallback when too few valid values remain (Appendix A.6.2). SHIR obtains its unit effects from these operations during reference construction and per-input queries; final evaluation instead uses the validation-donor joint transformations described below.

**Baselines.** We compare against Occlusion, Integrated Gradients (Sundararajan et al., 2017), KernelSHAP (Lundberg & Lee, 2017), TimeSHAP (Bento et al., 2021), TIMING (Jang et al., 2025), Dynamask (Crabbé & van der Schaar, 2021), and ContraLSP (Liu et al., 2024b). All methods explain the same frozen predictor and use the same unit catalog, evaluation inputs, and donor-based evalu-

Table 1: **Overall explanation quality.** Results are means over 50 evaluation inputs per task from one experimental run. Comp and Suff error average four explanation sizes; CPD and CPP error evaluate the 10% endpoint. Bold marks the best mean within each task. All measurements use the joint deletion and retention protocol in §4.1.

| Method | WeatherBench2 | | | | VitalDB | | | |
|---|---|---|---|---|---|---|---|---|
| | Comp↑ | Suff error↓ | CPD @10%↑ | CPP error @10%↓ | Comp↑ | Suff error↓ | CPD @10%↑ | CPP error @10%↓ |
| Occlusion | 0.1463 | 0.0349 | 0.1456 | 0.0327 | 0.7694 | 0.2255 | 0.8014 | 0.1907 |
| Integrated Gradients | 0.1480 | 0.0350 | 0.1467 | 0.0339 | 0.7946 | 0.2035 | 0.7999 | 0.1989 |
| KernelSHAP | 0.1480 | 0.0350 | 0.1468 | 0.0350 | 0.8140 | 0.1807 | 0.8229 | 0.1708 |
| TimeSHAP | 0.1482 | 0.0349 | 0.1462 | 0.0351 | 0.8140 | 0.1807 | 0.8229 | 0.1708 |
| TIMING | 0.1459 | 0.0362 | 0.1467 | 0.0346 | 0.7662 | 0.2268 | 0.7703 | 0.2202 |
| Dynamask | 0.1380 | 0.0525 | 0.1430 | 0.0471 | 0.5025 | 0.5203 | 0.5665 | 0.4577 |
| ContraLSP | 0.1463 | 0.0418 | 0.1514 | 0.0386 | 0.7230 | 0.2589 | 0.7350 | 0.2492 |
| **SHIR** | **0.1634** | **0.0310** | **0.1596** | **0.0254** | **0.8541** | **0.1469** | **0.8647** | **0.1402** |

ation transformations. Appendix B.2 gives each method's gradient access, explanation-generation budget, and conversion of fine-grained attributions or masks into unit scores.

**Evaluation.** We assess an explanation by the prediction changes caused by jointly deleting or retaining its selected units. For each test input, five validation donor windows are chosen in advance and shared by all methods. Deletion replaces all selected units with the corresponding values from one donor, and retention replaces their complement within the intervention catalog. Each perturbed input is constructed directly from the original input. Non-intervened contextual inputs remain available; for VitalDB, observation masks are preserved and the registered affected derived factors are recomputed. Retention error therefore measures prediction preservation under this catalog-restricted transformation. Evaluation begins after the ranking is fixed and feeds nothing back to explanation generation.

We evaluate prefixes of one fixed ranking per input at 5%, 7.5%, 10%, and 15% of the unit catalog, rounded upward, giving sizes $(5, 7, 9, 14)$ on WeatherBench2 and $(1, 2, 2, 3)$ on VitalDB. Prediction displacement divides each output's absolute prediction difference by its frozen scale and averages over the three outputs and five donors. Comp (↑) and Suff error (↓) average the joint deletion effect and joint retention error, respectively, across the four explanation sizes; CPD@10% (↑) and CPP error@10% (↓) report the same joint endpoints at the 10% size. Appendix B.3 provides the complete metric definitions, donor selection, and aggregation rules.

### 4.2 Overall Explanation Quality

SHIR achieves the best mean on all four metrics in both tasks against the seven baselines (Table 1). Relative to the strongest baseline for each metric, it reduces mean sufficiency error by approximately 11.2% on WeatherBench2 and 18.7% on VitalDB, and increases mean comprehensiveness by 10.3% and 4.9%, respectively.

### 4.3 Ablation Studies

Table 2 reports ablations of per-input query design, evidence linking during reference construction, and current measurements with per-input score updates. Each ablation variant averages three repetitions on the same 50 evaluation inputs per task, whereas SHIR reuses the single-run result from Table 1; Table B2 in Appendix B.5 gives all four metrics.

#### 4.3.1 Context Construction and Query Selection

**Comparison-oriented context construction.** Flat evidence presentation retains the same raw evidence and reference/current-input labels but organizes them by unit or execution record rather than

Table 2: **Ablation results.** Comp and Suff error on the same 50 evaluation inputs per task. Ablation variants average three repetitions; SHIR reuses the single-run result from Table 1. Bold marks the best reported mean in each column.

| Variant | WeatherBench2 | | VitalDB | |
|---|---|---|---|---|
| | Comp↑ | Suff error↓ | Comp↑ | Suff error↓ |
| SHIR | 0.1634 | **0.0310** | 0.8541 | **0.1469** |
| Flat evidence presentation | 0.1558 | 0.0387 | 0.8407 | 0.1618 |
| Reference-score Top-3 | **0.1660** | 0.0362 | **0.8561** | 0.1541 |
| No automatic evidence linking | 0.1621 | 0.0321 | 0.8304 | 0.1488 |
| Graph-only | 0.1531 | 0.0406 | 0.7790 | 0.2036 |

around comparisons and missing observations. SHIR improves all four metrics on both tasks, reducing mean sufficiency error by approximately 19.9% on WeatherBench2 and 9.2% on VitalDB (Table B2).

**Complementary query selection.** From the same candidate pool, Reference-score Top-3 selects the three units with the highest reference scores in place of SHIR's joint comparison-completion objective. This choice attains slightly higher mean comprehensiveness on both tasks (Table 2). Complementary selection trades this small difference for lower mean sufficiency error, by approximately 14.4% on WeatherBench2 and 4.7% on VitalDB, and for a larger 10% deletion effect and a lower 10% retention error on both tasks (Table B2).

#### 4.3.2 Evidence Reuse and Per-Input Score Updates

**Automatic evidence linking.** No automatic evidence linking disables the automatic association of existing execution records with other applicable registered comparisons after each reference-construction query round. It retains the complete valid current-execution table, and subsequent rounds proceed from the resulting evidence state to build the shared reference. With automatic linking, SHIR attains higher mean comprehensiveness and lower mean sufficiency error on both tasks (Table 2) and improves all four metrics (Table B2).

**Current measurements.** Graph-only uses the same frozen reference ranking without current perturbation queries or score updates. Adding SHIR's three current measurements and R0 improves all four metrics on both tasks, and 10% retention error falls by approximately 38.0% on WeatherBench2 and 27.8% on VitalDB (Table B2). Together, the ablations support the complete design: accumulated evidence guides a small set of current measurements, whose valid effects update that input's explanation scores.

## 5 Conclusion

SHIR explains frozen time-series predictors using reference perturbation evidence and a small budget of current perturbation measurements. Its central insight is that unit scoring and query selection answer different questions: unit scores use reference scores or measured current effects, while queries obtain the observations needed for comparisons relevant to selecting an explanation. Evidence-guided experimental design gives comparison-oriented context construction and complementary query selection a shared comparison representation, and many-to-many evidence reuse lets one observation serve every applicable registered comparison while preserving execution identity. Valid current effects then update the input's scores without changing the shared reference. On the evaluated WeatherBench2 and VitalDB tasks, SHIR achieves better reported mean deletion and retention results than the seven external baselines, and the ablations support the roles of context construction, query selection, evidence linking, and current measurements with score updates.

## AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and editing, LaTeX formatting, and the retrieval and verification of bibliographic information. All AI-assisted material was reviewed by the authors. The authors take responsibility for the final content of the manuscript, including its text, claims, and references.

## REFERENCES

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 ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2021. doi: 10.1145/3447548.3467166. URL https://arxiv.org/abs/2012.00073v2.

Jiashun Chen, Yirong Mao, and Wenhui Que. EviGraph: Towards verifiable evidence construction for information-seeking agents. *arXiv preprint arXiv:2608.24667*, 2026. URL https://arxiv.org/abs/2608.24667v1.

Qitong Chu, Xunjie He, Chen Deng, Huaxin Pei, and Yufeng Yue. NeSy-CSA: A neuro-symbolic framework for open-ended critical scenario attribution. *arXiv preprint arXiv:2607.03847*, 2026. URL https://arxiv.org/abs/2607.03847v1.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021. URL https://www.jmlr.org/papers/v22/20-1316.html.

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. URL https://proceedings.mlr.press/v139/crabbe21a.html.

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. URL https://proceedings.mlr.press/v202/enguehard23a.html.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.11738v4.

Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candes, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25372–25437. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25n.html.

Md. Khairul Islam and Judy Fox. WinTSR: A windowed temporal saliency rescaling method for interpreting time series deep learning models. *arXiv preprint arXiv:2412.04532*, 2025. URL https://arxiv.org/abs/2412.04532v3.

Hyeongwon Jang, Changhun Kim, and Eunho Yang. TIMING: Temporality-aware integrated gradients for time series explanation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26877–26895. PMLR, 2025. URL https://proceedings.mlr.press/v267/jang25a.html.

Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, Drew Bagnell, and Siddhartha Srinivasa. Near optimal Bayesian active learning for decision making. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pp. 430–438. PMLR, 2014. URL https://proceedings.mlr.press/v33/javdani14.html.

Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2107.07436.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/95323660ed2124450caaac2c46b5ed90-Abstract.html.

Hyung-Chul Lee, Yoonsang Park, Soo Bin Yoon, Seong Mi Yang, Dongnyeok Park, and Chul-Woo Jung. VitalDB, a high-fidelity multi-parameter vital signs database in surgical patients. *Scientific Data*, 9:279, 2022. URL https://www.nature.com/articles/s41597-022-01411-5.

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. URL https://openreview.net/pdf?id=C0q9oBc3n4.

Zichuan Liu, Tianchun Wang, Jimeng Shi, Xu Zheng, Zhuomin Chen, Lei Song, Wenqian Dong, Jayantha Obeysekera, Farhad Shirani, and Dongsheng Luo. TimeX++: Learning time-series explanations with information bottleneck. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32062–32082. PMLR, 2024a. URL https://proceedings.mlr.press/v235/liu24bl.html.

Zichuan Liu, Yingying Zhang, Tianchun Wang, Zefan Wang, Dongsheng Luo, Mengnan Du, Min Wu, Yi Wang, Chunlin Chen, Lunting Fan, and Qingsong Wen. Explaining time series via contrastive and locally sparse perturbations. In *International Conference on Learning Representations*, 2024b. URL https://arxiv.org/abs/2401.08552v1.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Arnau Marin-Llobet and Javier Ferrando. Automated interpretability and feature discovery in language models with agents. *arXiv preprint arXiv:2605.01555*, 2026. URL https://arxiv.org/abs/2605.01555v1.

Ephrem Tibebe Mekonnen, Luca Longo, Lucas Rizzo, and Pierpaolo Dondio. L2GTX: From local to global time series explanations. In *4th World Conference on Explainable Artificial Intelligence*, 2026. URL https://arxiv.org/abs/2603.13065v2.

Owen Queen, Tom Hartvigsen, Teddy Koker, Huan He, Theodoros Tsiligkaridis, and Marinka Zitnik. Encoding time-series explanations through self-supervised model behavior consistency. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/65ea878cb90b440e8b4cd34fe0959914-Abstract.html.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024. doi: 10.1029/2023MS004019. URL https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2023MS004019.

Zhenjiang Ren, Ruiji Li, Xujing Zhang, Ziliang Pang, Shuo Ren, and Jiajun Zhang. EviGraph: Evidence-guided autonomous research agents. *arXiv preprint arXiv:2608.04738*, 2026. URL https://arxiv.org/abs/2608.04738v2.

Marco Tulio Ribeiro and Scott Lundberg. Adaptive testing and debugging of NLP models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3253–3267. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.230. URL https://aclanthology.org/2022.acl-long.230/.

Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 44293–44321. PMLR, 2024. URL https://proceedings.mlr.press/v235/shaham24a.html.

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. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Jinghang Yue, Jing Wang, Lu Zhang, Shuo Zhang, Da Li, Zhaoyang Ma, and Youfang Lin. Optimal information retention for time-series explanations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73851–73876. PMLR, 2025. URL https://proceedings.mlr.press/v267/yue25d.html.

Éloi Zablocki, Valentin Gerard, Amaia Cardiel, Eric Gaussier, Matthieu Cord, and Eduardo Valle. GIFT: A framework towards global interpretable faithful textual explanations of vision classifiers. *Transactions on Machine Learning Research*, 2026. URL https://arxiv.org/abs/2411.15605v3.

# A Implementation and Evidence Details

## A.1 Evidence records and identity

The graph distinguishes comparison patterns, associated proposal records, executions, responses, and input-specific comparison results. A comparison pattern identifies an unordered unit pair under the fixed configuration. Proposal records retain the proposed ordering, rationale, evidence references, and proposal time. Comparison status can be derived from the available records for a particular input.

| Record | Information retained |
|---|---|
| Comparison pattern and associated proposal records | Canonical unordered unit pair and predictor/target/adapter configuration; associated proposal records retain the proposed ordering, rationale, evidence references, and proposal time. |
| Execution | Input identity, unit, primary operation, target and scale configuration, adapter version, deterministic perturbation-replacement parameters, and execution validity. |
| Response | Original and perturbed predictions, signed normalized component changes, and the nonnegative perturbation-effect magnitude used by the method. |
| Input-specific comparison status and result | Applicable execution references, availability of each current endpoint measurement, measured ordering or tie, and the proposed ordering being checked. |

Execution identity determines which observations are distinct. Equal response values from different inputs remain separate observations, whereas repeated citations of one execution refer to the same observation. Aggregation counts unique execution records, irrespective of the number of response nodes, references, or graph paths that cite them. The cost log records every predictor call, including failures and repeated queries.

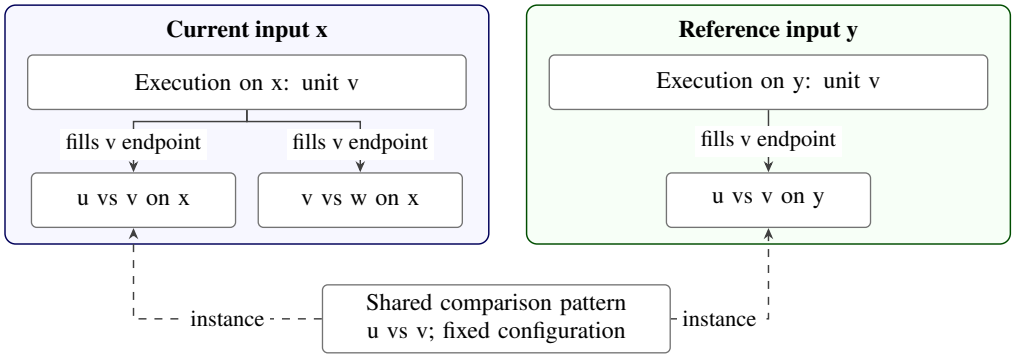



Figure A1: A pattern can be shared across inputs while its comparison states and execution records remain separate. Only the records for unit v are shown: each comparison still requires its other endpoint from the same input. Dashed edges show pattern instances; solid edges show how one recorded observation supplies an endpoint to a comparison.

The program assigns stable card IDs before proposal mapping, called `Grounding` in the implementation. Mapping translates unit endpoints, proposed orderings, and evidence references into the required format while preserving each proposal's meaning and rationale. It neither judges the rationale's truth nor changes an ordering to match observed effects. If endpoint canonicalization reverses the endpoint order, it adjusts the direction label accordingly.

The retained output schema uses `untranslatable` for cards that cannot be faithfully mapped and for later duplicates of an already admitted unordered pair. This label records admission status, not

a judgment about the rationale. Candidate formation and the batch objective count each admissible unordered pair once.

### A.2 Reference score aggregation

The reference-score aggregation in Eq. (5) uses only valid execution records from the same primary operation and configuration. Absolute normalized responses are processed separately for each output component. Within each input, repeated references to an execution are removed before taking the median for each unit and component. The canonical single-unit query normally yields one value per input and unit.

For VitalDB, absolute normalized component responses are first reduced by their median across windows within each patient, then by their median across patients. The count $n_u$ records how many reference groups have a valid observation of unit $u$; each group here is a patient. For WeatherBench2, each window is a reference group. Values are reduced across windows, and $n_u$ counts windows with a valid observation of $u$. Valid zero responses remain in the aggregation; units without reference observations receive no synthetic zero records.

Let $m_{u,c}$ denote the resulting component-wise median for a unit $u$ with at least one valid reference observation. The adopted shrinkage parameter is $\kappa = 5$:

$$a^{\mathrm{g}}_{u,c} = \frac{n_u}{n_u + 5} m_{u,c} + \frac{5}{n_u + 5} \mu_c, \qquad s^{\mathrm{g}}(u) = \max_c a^{\mathrm{g}}_{u,c}, \quad n_u > 0. \tag{A1}$$

The empirical shrinkage target is computed from the frozen reference graph, with equal weight for each unit that has a valid reference observation:

$$\mu_c = \mathrm{median}_{u:n_u>0}\, m_{u,c}. \tag{A2}$$

The target is computed from reference observations without separate tuning or weighting by $n_u$. A unit without reference observations receives component defaults $\mu_c$ and score $\max_c \mu_c$. This initialization requires the graph to contain valid primary-operation evidence.

Component-wise aggregation and shrinkage precede the maximum; taking the maximum before aggregation would yield a different reference score. Reference-group counts are distinct from execution counts and link counts. The patient-level reduction applies to reference intervention evidence. Output-scale estimation follows its own sampling and reduction procedure in §A.6.3.

Reference paired summaries use windows with valid observations of both units under the same primary operation and compatible configuration. For a canonically ordered pair $(u, v)$, the window-level difference is $\Delta_x(u, v) = a_x(u) - a_x(v)$, using the perturbation-effect magnitudes in Eq. (1). Each WeatherBench2 window forms one group. For VitalDB, paired window-level differences are reduced by their median within each patient. The reference paired difference is then the median across group-level differences, with one value per group containing a valid paired observation.

The support count records the number of reference groups with valid paired observations. Positive, negative, and tie counts describe the signs of group-level differences for the canonical endpoint order. They measure effect-magnitude orderings, not agreement with arbitrary proposal expectations. Ties follow the fixed zero-tolerance rule, and the three counts sum to the pair's support count. This count is distinct from single-unit support and graph-link counts.

A window contributes only when both endpoint measurements are valid. Missing or invalid endpoints are excluded, not assigned a zero difference. A group contributes only if it contains valid paired windows; with no qualifying group, the summary remains missing. Differences between marginal reference scores cannot substitute for paired observations.

### A.3 Comparison states and numerical handling

A current measurement requires matching input, unit, operation, target, scale, and adapter configurations, together with positive finite scales. A required original or perturbed target prediction, normalized component response, or scalar effect that is NaN or infinite makes the measurement

invalid. Invalid records are excluded from the measured-unit set $M$, numerical comparisons, reference score aggregation, and R0's current-value branch. A finite zero effect is valid and replaces the corresponding reference score.

Validity checks follow the dataset's existing preprocessing and observation-mask handling. An invalid component invalidates the measurement: components are not dropped, clipped, or replaced with zeros to obtain a score, and the prescribed maximum or mean is unchanged. Failed calls count toward query use and do not authorize an extra perturbation query during per-input explanation.

The same validity and configuration rules determine $M$ and the applicable records in Methodology Eq. (4). A record remains applicable whether the measured comparison ordering agrees or disagrees with the proposed ordering. Completing a comparison requires a matching record for each distinct endpoint; a nonempty record set alone is insufficient. Deduplication uses execution identity, not response-node identity or numerical equality.

The comparison tolerance is $\epsilon_{\mathrm{cmp}} = 0$. Once both current effects are available, the sign of their raw difference determines the ordering; an exactly zero difference gives a completed comparison with a tie. A comparison with a missing valid endpoint measurement remains pending. The tolerance is distinct from the output-scale floor $10^{-8}$; the reported configuration has no positive near-tie band.

After the per-input score update, unrounded scores are sorted in descending order, with exact ties broken by ascending Unit ID. Scores determine the ranking; graph wins, proposed orderings, and language-model judgments do not enter the sort. For any two measured units, the score difference equals the measured effect-magnitude difference, including pairs that were not registered as comparisons. This identity does not add unregistered pairs to the batch objective. Every valid measurement updates its unit's score, even if the resulting comparison disagrees with its proposed ordering or remains incomplete.

The batch objective uses the comparison set fixed before execution. New responses can update other compatible registered comparisons. Comparisons proposed after a response remain outside that batch's selection-time completion count.

### A.4 Candidate bounds and exceptional cases

#### A.4.1 Comparison context and candidate formation

Comparison-oriented context construction displays at most **six registered comparisons**. It prioritizes comparisons missing current-input evidence, then comparisons crossing a declared explanation-budget boundary when a complete working ranking is available. Other admissible comparisons remain eligible. A separately initialized reference-input episode lacks the reference scores needed for a complete working ranking, so unmeasured units have no assigned boundary positions.

Within each context-construction priority class, each unordered pair is represented by its two Unit IDs in ascending order. Canonical pairs are sorted lexicographically, and the first six comparisons in the resulting priority-ordered list are retained. This deterministic ordering preserves the evidence and boundary priorities without random sampling.

For each displayed comparison, the context contains:

| Information | Per-comparison bound |
|---|---|
| Current endpoint evidence | At most one current execution per endpoint, including all signed component responses and the scalar effect. |
| Reference endpoint evidence | One summary per endpoint, with the number of groups containing valid observations, component statistics, and reference score, where available. |
| Matched reference comparison | At most one paired summary from matched within-input observations; absence is stated rather than filled from marginal-score differences. |

The full current-input execution table is retained separately, with one entry per unique execution that yields a valid measurement. A four-round episode contains at most 12 such unit measurements. Reference evidence is supplied through summaries, rather than a full expansion of raw records. The context uses fixed limits on the numbers of displayed comparisons and evidence records, with no separate per-comparison token cap. The templates in §A.7 specify the display order. Measurement availability comes from the execution table: a missing comparison summary neither removes an endpoint record nor causes a unit with a valid current measurement to be queried again.

The proposal stage returns comparison cards containing at most **three distinct unordered unit pairs**. Each round retains the existing unresolved pairs selected during context construction, without requiring another LLM proposal. It adds new pairs extracted from faithfully mapped proposals. Pairs from both sources undergo catalog and fixed-configuration checks, normalization, and deduplication. Pairs with valid current measurements at both endpoints are excluded from the query objective, while their records are retained. Other existing pairs enter the selector only through an admissible mapped proposal. The merged set receives no additional candidate-ranking or truncation step.

After bounded replacement supplementation and remerging, when needed under §A.4.3, the candidate set and its unique unmeasured endpoints are fixed for query selection and execution. The set contains retained pairs and admissible pairs from mapped proposals; other pairings of their endpoints are not added. The six displayed comparisons and three new pairs give a **derived upper bound of nine candidate pairs and 18 distinct endpoints**. Filtering and shared endpoints reduce these counts. The six-endpoint bound applies only to three *pairs from new proposals*, not to the merged pool.

### A.4.2 Batch selection and ties

The selector enumerates feasible batches to maximize the number of distinct candidate comparisons whose missing endpoints the batch would supply. It uses the full query budget when enough eligible endpoints are available, as specified in Methodology §3.3. For batches tied in objective value and chosen size, it sorts each batch's Unit IDs and selects the lexicographically smallest list. This tie rule is deterministic and uses no pending response values.

Each batch contains distinct admissible units without valid current measurements. Perturbed inputs are constructed separately from the same original input and evaluated in parallel. The normal per-input query budget is three, with bounded supplementation as described below. Per-input explanation uses this single batch, without a second adaptive query round.

### A.4.3 Bounded supplementation, repair, and failure

Before query execution, the application forms the merged candidate set in Eq. (2) from retained existing pairs and pairs extracted from mapped proposals. It counts the distinct admissible endpoints that lack valid current measurements. If fewer than three are available, it allows at most one supplemental proposal followed by proposal mapping, using the same pre-execution information. This supplementation is separate from format repair and leaves the predictor-query budget unchanged.

The supplement replaces the new proposal set with proposals containing at most three pairs; it does not append another three-pair pool. The retained existing comparisons are merged with admissible pairs from the replacement proposals, then checked and filtered again. The final merged pool determines the batch. It supplies three queries when enough eligible endpoints exist and the available number otherwise. Exploration stops if the merged pool contains no admissible endpoint without a valid current measurement. The final candidate set is fixed before query selection and execution.

For example, a retained u/v pair and a new v/w pair yield merged endpoints u, v, and w when all three are unmeasured. This pool needs no supplement, even though the new proposal alone has two endpoints. §A.7.10 gives the message and its application-level invocation condition.

Every selected perturbation query uses an admissible endpoint from the merged pool. Batches are not filled with arbitrary units, and per-input explanation adds no fourth query. Failed or invalid queries remain recorded attempts under §A.3; their effects are not set to zero, and their units are not added to the measured set.

Each LLM request allows at most three total transport attempts. Proposal-format repair and proposal-mapping/JSON repair each allow at most one repair, after which any remaining failure is recorded. The repair messages in §A.7 preserve unit choices, proposed orderings, and rationales; they do not change proposal meaning to fit known outcomes. Actual requests, repairs, supplements, and attempted predictor calls are counted separately.

### A.5 Cost and evaluation boundary

A normal per-input batch with three eligible perturbation queries adds **three perturbed predictor evaluations** when the original prediction is cached, or **four total evaluations** when it must also be computed. Batch submission preserves the number of logical queries. Comparison updates, graph linking, and R0 need no further predictor evaluations. For reduced batches, counts reflect the evaluations actually attempted, including failed calls.

Reference construction allows at most four rounds with at most three distinct unit queries per round, for at most 12 perturbation evaluations per reference-input episode. An uncached original prediction adds one evaluation. These are protocol bounds, not measured means. Additional perturbation queries after graph merging are **disabled**. Once the separately initialized episodes end, their graphs are merged and their records aggregated into reference scores. Both are then frozen.

Costs are counted separately for reference construction, LLM requests and tokens, per-input perturbation queries, and fidelity-evaluation queries. Configured token limits bound request outputs; actual consumption is recorded separately.

§B.3 specifies the separate fidelity-evaluation protocol and its boundary from explanation generation.

### A.6 Configuration record

#### A.6.1 Tasks and reference inputs

Task definitions, predictor architectures, unit catalogs, reference inputs, and data partitions are specified in §B.1. The corresponding primary interventions and output scales are defined in §§A.6.2–A.6.3.

#### A.6.2 Fixed primary interventions

**WeatherBench2: training-only seasonal climatology.** The replacement climatology uses 1979–2017 training data only. Statistics are computed separately for each variable/pressure level, grid location, and UTC slot (00, 06, 12, 18). For each day of year, the climatology is the arithmetic mean over the ±15-day seasonal window at the same UTC slot, pooled across training years with cyclic handling at the year boundary. Each selected timestamp, channel, and grid point receives its corresponding climatology; coupled channels in the variable group are handled together.

The seasonal window remains fixed. If the replacement climatology is unavailable or the operation fails the adapter's validity checks, the execution is marked invalid/failed. The adapter does not substitute a global mean or use validation or test values to supply a missing climatology.

**VitalDB: out-of-block median within the current input.** For the selected one-minute block, the adapter excludes its six bins and computes a median separately for each basic variable using quality-valid values in the remaining four minutes of the current five-minute input. At least six valid bins are required. Otherwise it uses the frozen training median for that variable and records the fallback. The out-of-block replacement baseline uses values within the observed input window, including any observations after the selected block and before the prediction cutoff.

After the basic values are replaced, MAP trend and fluctuation, HR trend, and $ETCO_2$ trend are recomputed. $SpO_2$ has no corresponding derived-feature update. The derived calculations retain their existing past-six-bin definition. Affected positions include the selected block and up to the following 50 seconds, bounded by the input cutoff. Observation/quality masks and unrelated variables remain unchanged. A single-unit perturbation query includes all registered derived-input updates, including affected positions outside the basic block.

Both primary operations use complete replacement, intensity **1.0**, during reference construction and per-input explanation. The LLM cannot alter their replacement values, intensity, or operator family. These perturbation replacement values are distinct from the validation-donor values used in final joint deletion and retention evaluation.

### A.6.3 FROZEN OUTPUT SCALES

Positive component scales are fixed as part of the predictor–target–adapter configuration before test explanation. The same values normalize reference responses, current comparisons, R0, and the common evaluation protocol. They remain fixed across candidate batches and final evaluation.

The output-scale estimation set $\mathcal{V}_{\text{cal}}$ contains all eligible original windows assigned to `validation` in the prepared sample manifest, after preprocessing and quality screening. It contains 2,905 WeatherBench2 windows from 2018–2019 and 46,999 VitalDB windows from the patient-disjoint validation partition. For each target component, the frozen predictor is evaluated on these unperturbed inputs. The scale is the interquartile range of its outputs divided by 1.349, with a lower bound of $10^{-8}$:

$$\sigma_c = \max\left\{\frac{Q_{75}(\{f_c(x) : x \in \mathcal{V}_{\text{cal}}\}) - Q_{25}(\{f_c(x) : x \in \mathcal{V}_{\text{cal}}\})}{1.349}, 10^{-8}\right\}. \tag{A3}$$

Ground-truth labels, perturbed predictions, and intervention-effect thresholds are not used to compute these scales. WeatherBench2 components are the cosine-latitude-weighted mean predictions over the output region in §B.1.1, not individual grid-point outputs. VitalDB components are the predicted MAP medians over 60-second windows ending at the specified horizons.

Both tasks use window-level quantiles, with one unperturbed prediction per window for each component. Each VitalDB window contributes separately to output-scale estimation. The patient-grouped reduction in §A.2 applies to reference intervention evidence.

**Output-scale estimation sets.**

| Dataset | Eligible validation windows |
|---|---|
| WeatherBench2 | 2,905 |
| VitalDB | 46,999 |

**Frozen output scales.** Values are displayed to six decimal places in target order. Calculations use the full-precision values given below.

| Dataset | Output key | Scale | Unit |
|---|---|---|---|
| WeatherBench2 | `lead_6h` | 7.926657 | K |
| WeatherBench2 | `lead_12h` | 7.919746 | K |
| WeatherBench2 | `lead_24h` | 7.879262 | K |
| VitalDB | `map_1m` | 13.435533 | mmHg |
| VitalDB | `map_3m` | 12.553558 | mmHg |
| VitalDB | `map_5m` | 11.774407 | mmHg |

**Full-precision scale parameters**


```
{
  "WeatherBench2": {
    "lead_6h": 7.9266566744252325,
    "lead_12h": 7.919746033395541,
    "lead_24h": 7.879262215796636
  },
  "VitalDB": {
    "map_1m": 13.435533437311783,
    "map_3m": 12.55355789716196,
    "map_5m": 11.77440668760714
  }
}
```


These output-scale estimation sets serve a different purpose from the legacy operator-threshold calibration lists retained for donor exclusion. Participation in $\mathcal{V}_{\mathrm{cal}}$ alone leaves donor eligibility unchanged; §B.3 specifies the exclusions.

### A.6.4 LLM AND REQUEST SETTINGS

Proposal generation and proposal mapping use the same model, **`deepseek-flash-v4.1`**. The implementation labels these stages `Proposal` and `Grounding`, respectively. Their decoding and request settings are listed below.

| Stage | Reasoning | Requested temperature | Max. output tokens | Output format |
|---|---|---|---|---|
| Proposal | `max` | 0.3 | 65,536 | Text |
| Mapping | Off | 0 | 4,096 | Strict JSON |

The Proposal token limit includes reasoning. Each Proposal output contains at most three textual comparison cards. Each request allows at most three total transport attempts, and proposal-format repair and mapping/JSON repair each allow at most one repair. Bounded replacement supplementation follows §A.4.3.

The temperature entries report the submitted request values, not the deployed service's effective decoding behavior.

Proposal mapping encodes unit endpoints, proposed orderings, and allowed evidence references in strict JSON while preserving each original rationale. Mapping preserves each proposal's intended ordering for later comparison with predictor responses. Assessing semantic truth falls outside this mapping step.

Section A.7 gives the system messages, user templates, stage instructions, card-ID wrapper, repair messages, and bounded supplementation message. Only the stage instruction changes between reference construction and per-input explanation. Both stages use the same effect definition and core contract.

The input description supplies dataset and instance information and the target components. Current predictions are given in physical units alongside the frozen scales. The context also includes the fixed operator, admissible unit catalogue, reference evidence where available, current execution records, registered comparisons, and allowed evidence IDs.

The instance card summarizes the current input window using unit-level attributes and their percentile positions in training-only empirical distributions. After preprocessing and quality screening, the renderer selects each unit's variables, temporal indices, and spatial region where applicable. It then computes the configured attributes and marks them as valid, partially available, or unavailable. The card combines unit semantics, current input statistics, and percentiles; it describes observed input values, not unmeasured perturbation effects or attribution scores. Predictor outputs, reference execution evidence, and current execution records are supplied in separate context fields.

### A.6.5 CORE PARAMETERS

The shared exploration mechanism uses the following numeric settings. Aggregation, comparison, candidate formation, and execution rules are specified in §§A.2–A.5.

| Parameter | Value |
|---|---|
| Queries per batch, $b$ | 3 |
| Rounds per reference input (max.) | 4 |
| Comparisons in context (max.) | 6 |
| New pairs per proposal set (max.) | 3 |
| Replacement proposal sets per round (max.) | 1 |
| Shrinkage, $\kappa$ | 5 |
| Comparison tolerance, $\epsilon_{\text{cmp}}$ | 0 |

## A.7 PROMPTS AND OUTPUT CONTRACTS

This appendix reproduces the original request templates. In the manuscript, proposal mapping refers to the stage labeled Grounding in these templates; reference construction and per-input explanation correspond to their offline and online stages. Template wording, schema fields, and runtime identifiers are retained. The prompt symbol `s_c` denotes the paper's frozen output scale $\sigma_c$.

The appropriate stage block replaces {`stage_instruction`} in the Proposal user message. Request settings are specified in §A.6.4, and the {`instance_card`} field follows the construction in that section. The other context fields contain the task description, unit catalog, current records, and applicable reference evidence.

### A.7.1 PROPOSAL-GENERATION SYSTEM MESSAGE

```
You propose testable comparisons about the behavior of a frozen time-series
prediction model.

You will receive:
1. the dataset, current input and prediction target;
2. a fixed legal intervention operator and a Unit catalog;
3. scoped historical summaries, when available;
4. actual intervention responses already observed on the current input;
5. registered comparisons and their missing current-input evidence.

For a Unit u, the fixed operation produces T_u(x). Its effect is:

r_c(x, u) = (f_c(T_u(x)) - f_c(x)) / s_c
a_x(u) = max_c |r_c(x, u)|

The maximum is taken over ALL target components supplied in the task.
Your operational hypothesis compares a_x(u) with a_x(v). It does not compare
the signs of the responses, and it does not describe a joint intervention
on u and v.

Your role is to propose which Units are worth comparing, an expected relative
effect order, a brief semantic rationale, and references to supplied evidence.

The application determines the intervention, missing queries, query batch,
numerical comparison and final ranking. Do not generate replacement values,
intervention strengths, alternative operators, importance scores, confidence
numbers, or a final ranking.

Statements and rationales are unverified conjectures. They may use physical,
clinical, causal or model-mechanism reasoning and may be wrong. Predictor replay
tests only the specified relative-effect comparison, not the entire narrative.
```

```
Keep historical evidence and current-input observations distinct:
- Historical summaries are advisory defaults, not measurements of this input.
- An observed zero is valid evidence.
- An unobserved response is unknown, not zero.
- One execution may support several comparisons, but remains one observation.
- Previously observed outcomes must not be presented as new prospective
  predictions.

Return only the requested comparison cards, without a preamble or separate
analysis.
```

### A.7.2 PROPOSAL-GENERATION USER MESSAGE

```
# DATASET AND CURRENT INSTANCE

Dataset:
{dataset_description}

Current train-reference-relative instance information:
{instance_card}

# FROZEN PREDICTION TASK

Prediction target:
{target_description}

Target components:
{target_components}

Current predictions in physical units:
{prediction_by_component}

Frozen output scales:
{scale_by_component}

# FIXED INTERVENTION AND UNIT CATALOG

The only numerical operation available in this run is:
{primary_operator_description}

Its reference values, parameters and validity checks are determined by the
dataset adapter. You must not modify them.

Unit structure and coupling:
{unit_scope_and_coupling_description}

Available Unit IDs and descriptions:
{unit_catalog}

Use exact Unit IDs. Do not include display labels as part of an ID.
A Unit may contain coupled channels or derived dependencies; do not split it
into unsupported smaller interventions.

# EXECUTION STAGE

{stage_instruction}

Current round: {round_number}
Maximum rounds for this input: {maximum_rounds}
New predictor-query budget for this round: 3
Maximum number of proposed comparisons: 3

# HISTORICAL REFERENCE - ADVISORY ONLY
```

```
{historical_reference_context}

When available, this section contains historical default scores, their
reference ranking, endpoint support and genuinely matched historical
comparisons.

A difference between two historical default scores is not automatically a
matched historical difference: the two endpoints may have been observed on
different reference inputs.

A default assigned to an unsupported Unit is not an observed response.
Historical records never fill a missing current-input endpoint.

# CURRENT-INPUT EXECUTION TABLE

{current_execution_table}

This table contains only successful, compatible executions on the current
input. Each unique execution appears once, with its evidence ID, Unit ID,
signed normalized component responses and scalar effect.

Units absent from this table have not been measured on the current input.
Do not request another predictor execution for an already observed endpoint.

# REGISTERED COMPARISONS

{registered_comparison_view}

Each displayed comparison identifies:
- its two endpoints and previously stated expectations;
- compatible historical evidence, when available;
- current endpoint observations;
- which current endpoints are still missing;
- its numerical result, if both current endpoints are available;
- its relation to the declared explanation budgets, when a complete reference
  ranking is available.

A comparison that has not been explicitly organized may still be resolvable
from existing endpoint records. Lack of an organized summary does not imply
lack of observations. Such a comparison can be resolved without repeating
predictor queries.

# ALLOWED EVIDENCE REFERENCES

{allowed_evidence_ids}

Cite only IDs in this list.
Use INSTANCE_CARD for a new exploratory comparison motivated by the supplied
current-input information. Do not invent evidence IDs or observed values.

# PROPOSAL TASK

Propose up to three distinct unordered Unit pairs.

For each pair:
1. identify two different Units from the supplied catalog;
2. state the expected order of their scalar effects under the fixed operation;
3. explain briefly why this comparison is worth examining for the current input;
4. cite the relevant supplied evidence.

Prioritize unresolved comparisons that could affect a declared top-k prefix
when a complete reference ranking is available. This is a priority, not a
hard restriction: new directions and pairs currently on the same side of a
budget boundary are allowed when motivated by the input or evidence.
```

```
You may revisit a registered comparison that still lacks current evidence.
If both endpoints were already observed, do not describe their known ordering
as a new prediction or expect it to consume another predictor query.

Where possible, provide at least three distinct unqueried endpoints across
the proposed pairs. Shared endpoints are allowed when scientifically
motivated. Do not manufacture a clique or force endpoint sharing merely to
increase the number of completed comparisons.

Do not select the final query batch. The application will choose at most
three new Unit queries from the grounded endpoints.

Do not infer that a larger single-Unit effect proves joint sufficiency,
redundancy, interaction, or real-world causality.

# OUTPUT CONTRACT

Return up to three cards, numbered consecutively, using exactly these fields:

## COMPARISON <n>
Unit A:
<exact Unit ID>
Unit B:
<exact Unit ID>
Expected order:
<A_GT_B | B_GT_A | TIE>
Evidence references:
<comma-separated allowed evidence IDs>
Brief rationale:
<brief unverified rationale tied to the current input and available evidence>

A_GT_B means a_x(Unit A) > a_x(Unit B).
B_GT_A means a_x(Unit B) > a_x(Unit A).
TIE means the effects are expected to be equal under the application's fixed
numerical comparison rule. Do not introduce your own tolerance.

Do not submit the same unordered pair twice, even with reversed endpoint
order or a different rationale.

If no legal comparison can be proposed, return exactly:
NO_VALID_COMPARISON
```

#### A.7.3 REFERENCE-CONSTRUCTION STAGE INSTRUCTION (ORIGINAL OFFLINE BLOCK)

```
This is an independently initialized reference-input episode.

Use only the supplied instance information and evidence collected earlier
on THIS input. Do not use the completed global graph or evidence collected
from other reference-input episodes.

Training-data preprocessing, fixed intervention references and instance
statistics are available; cross-input model-behavior history is not.

There is no complete historical reference ranking in this cold-start episode.
Unobserved Units remain unknown. Do not invent a complete ranking or
budget-boundary positions.

Later rounds may use the responses obtained in earlier rounds of this episode.
```

#### A.7.4 PER-INPUT EXPLANATION STAGE INSTRUCTION (ORIGINAL ONLINE BLOCK)

```
This is a single-batch explanation of a new input using a frozen reference graph.
```

```
Historical scores and comparison summaries may guide your proposal, but none
of their executions count as current-input observations.

All three new queries must be selected before any response from this batch
is revealed. There will be no adaptive follow-up query round for this input.

After replay, the application uses each queried Unit's measured effect in
place of its historical default score. Unqueried Units retain their historical
defaults. You do not produce or revise the final numerical ranking.

The frozen reference graph is not updated with this test input.
```

### A.7.5 PROPOSAL-MAPPING SYSTEM MESSAGE (GROUNDING)

```
Map the supplied comparison cards into the finite Unit catalog for a frozen
time-series predictor.

Each card concerns the relative scalar effects of TWO separate single-Unit
executions under ONE fixed operator:

a_x(u) = max_c |(f_c(T_u(x)) - f_c(x)) / s_c|

The operator, target components, scales and query budget are fixed by the
application. They are not fields for you to choose.

Proposal statements and rationales are unrestricted, unverified conjectures.
Do not judge them true or false. Do not change an expected order to agree
with historical or current observations.

For each card, faithfully map:
- its two Unit endpoints;
- its expected relative-effect order;
- its references to the supplied evidence IDs.

Preserve the supplied Card ID. Return Unit A first and Unit B second.
The application will canonicalize the unordered pair and adjust the direction
if necessary.

When no faithful mapping exists, mark the card untranslatable. This means
the proposal cannot be operationalized by the supplied catalog and contract;
it does not mean its domain rationale is false.

Do not add hypotheses, invent evidence, choose a query batch, or create
operators, backgrounds, strengths, tensor indices, thresholds, importance
scores or final claims.

The original proposal remains authoritative for its semantic rationale.
A rationale echoed in JSON must not introduce a new explanation.

Return strict JSON only, with no extra keys or surrounding prose.
```

### A.7.6 PROPOSAL-MAPPING USER MESSAGE (GROUNDING)

```
# FROZEN TASK

Dataset:
{dataset_description}

Prediction target:
{target_description}

Target components:
{target_components}
```

```
Fixed primary operation:
{primary_operator_description}

Effect being compared:
a_x(u) = max_c |(f_c(T_u(x)) - f_c(x)) / s_c|

# RAW COMPARISON CARDS

{raw_comparison_cards_with_card_ids}

# ALLOWED UNIT CATALOG

{unit_catalog}

# ALLOWED EVIDENCE REFERENCES

{allowed_evidence_ids}

# MAPPING CONTRACT

1. Return exactly one item for each supplied card, in the supplied order.
   There are at most three cards.

2. Copy card_id exactly. Do not generate a new Card ID.

3. For a grounded item:
   - units must contain exactly two different allowed Unit IDs;
   - units[0] corresponds to Unit A in the original card;
   - units[1] corresponds to Unit B in the original card;
   - expected must be one of:
       first_gt_second
       second_gt_first
       tie
   - evidence_refs may contain only supplied evidence IDs;
   - rationale must faithfully preserve the original brief rationale.

4. Map A_GT_B to first_gt_second.
   Map B_GT_A to second_gt_first.
   Map TIE to tie.
   These describe absolute-effect ordering, not positive or negative effects
   on the prediction.

5. Do not replace an unsupported endpoint with a merely similar Unit.
   If the endpoints or expected comparison cannot be faithfully mapped,
   return status "untranslatable", units [], expected null and evidence_refs [].
   Preserve the original rationale.

6. If multiple cards map to the same unordered pair, retain the first faithfully
   grounded card and mark the later duplicates untranslatable. Different wording
   or reversed endpoint order does not create a new comparison.

7. If no cards were supplied, return {"comparisons": []}.

# OUTPUT FORMAT

Return one JSON object:

{
  "comparisons": [
    {
      "card_id": "<supplied Card ID>",
      "status": "grounded",
      "units": ["<Unit A ID>", "<Unit B ID>"],
      "expected": "first_gt_second",
      "rationale": "<faithful echo of the original brief rationale>",
```

```
      "evidence_refs": ["<supplied evidence ID>"]
    }
  ]
}

Allowed status values: grounded, untranslatable.
Allowed expected values for grounded items:
first_gt_second, second_gt_first, tie.
For untranslatable items, expected must be null.
```

#### A.7.7 STABLE CARD-ID WRAPPER

```
## CARD 1
Card ID: P1
Unit A: ...
Unit B: ...
Expected order: A_GT_B
Evidence references: ...
Brief rationale: ...
```

#### A.7.8 PROPOSAL FORMAT-REPAIR MESSAGE

```
The previous response failed the comparison-card format or envelope contract.

VALIDATION ERRORS:
{validation_errors}

PREVIOUS RESPONSE:
{previous_response}

Return up to three complete corrected COMPARISON cards.
Correct only the reported formatting, missing-field or envelope violations.
Preserve the substantive endpoint choices, expected orders and rationales
whenever they are present. Do not introduce a new exploration strategy.
Return the corrected cards only.
```

#### A.7.9 PROPOSAL-MAPPING REPAIR MESSAGE (GROUNDING)

```
The previous response failed JSON, catalog or reference validation.

VALIDATION ERRORS:
{validation_errors}

PREVIOUS RESPONSE:
{previous_response}

Preserve every supplied Card ID and the original proposal meaning.
Correct the reported JSON, endpoint mapping, direction mapping or reference
errors. Do not change an expectation because you believe it is false.

If a card cannot be faithfully mapped, mark it untranslatable using the
specified empty-field representation.

Return one corrected JSON object only.
```

#### A.7.10 BOUNDED SUPPLEMENTATION MESSAGE

The application counts distinct admissible unmeasured endpoints across the merged candidate set, including retained old comparisons. It issues the following message only when that count is fewer than three. This merged-set condition governs invocation of the original message, whose wording refers to the grounded proposal.

```
# ONE BOUNDED SUPPLEMENT
```

```
The grounded proposal provided fewer than three distinct new legal endpoints.

Using the SAME pre-replay information, return a replacement proposal of up to
three distinct Unit pairs. Include at least three unqueried legal endpoints
if possible. You may retain valid earlier choices.

Do not infer unknown responses, alter the operation, or expand the query budget.
The replacement proposal must satisfy the original output contract.

PREVIOUS GROUNDED PROPOSAL:
{previous_grounded_proposal}
```

### A.8 FIGURE EXAMPLE SETTINGS

The case uses hand-specified waveforms, reference scores, and effects to illustrate SHIR's calculation. These values are neither experimental results nor real-patient observations, and were not obtained by running the synthetic trace through the trained predictor.

Figure 1 uses the VitalDB setting in §B.1: five minutes of history in thirty ten-second bins, with twenty units from four variable groups (MAP, HR, $ETCO_2$, $SpO_2$) and five one-minute blocks. Only the five MAP units are visualized. Their chronological mapping is identical in every waveform:

| Unit | Relative interval | Ten-second bins, zero-based |
|---|---|---|
| p | [-5, -4) min | 0–5 |
| q | [-4, -3) min | 6–11 |
| u | [-3, -2) min | 12–17 |
| v | [-2, -1) min | 18–23 |
| w | [-1, 0) min | 24–29 |

In chronological unit order $p, q, u, v, w$, the illustrative reference scores are $0.4, 0.3, 0.8, 0.6, 0.5$. All fifteen unshown units retain reference scores no greater than $0.1$. The table labeled "Comparison context" places the defaults $s^{\mathrm{g}}(v) = 0.6$ and $s^{\mathrm{g}}(w) = 0.5$ above missing current effects (?, ?). These per-unit defaults inform the reference ranking and subsequent score update; they are not paired response differences or current measurements. The displayed score range does not impose an upper bound on normalized effects.

With no current observations, the candidates are $u/v$, $v/w$, and $p/q$. The three-query batch $\{u, v, w\}$ completes two comparisons, whereas querying $p, q$ and one further unit completes only $p/q$. Both completed comparisons use the same execution record for $v$, together with their respective other endpoint. The two numerical comparison results thus use three distinct execution records.

Each of $T_u(x)$, $T_v(x)$, and $T_w(x)$ is constructed separately from the same original input $x$. The fixed primary operation in §A.6.2 excludes the selected block when computing the current-input median and applies the stated validity and fallback rules. Its illustrative replacement value is 80 mmHg for each shown query. Only the selected MAP block is replaced at the base-variable level. Affected derived inputs are recomputed, while other base groups, observation masks and outside-catalog context are preserved. The waveforms show only the MAP projection of the full input.

The frozen predictor forecasts MAP medians over sixty-second windows ending at the +1, +3 and +5 min horizons. Each perturbed prediction is compared separately with the same cached $f(x)$. The current effects shown in Figure 1 are computed separately for the three perturbations using the absolute normalized component changes and maximum in Eq. (1). The three paths share the original prediction and calculation rule and produce three separate current effects and execution records. The original prediction is cached or computed separately. As specified in §A.5, the three-query budget counts perturbed evaluations. Current records remain local to the input, leaving the frozen reference unchanged.

The assigned effects are $a_x(u) = 0.2$, $a_x(v) = 0.9$, and $a_x(w) = 0.7$. They appear once each inside the execution records as $u$: 0.2, $v$: 0.9, and $w$: 0.7. "Update scores" implements Eq. (6) by replacing the defaults in the compact "Reference scores" band for $u, v, w$ with their current effects and retaining the defaults for $p, q$ and the fifteen unshown units.

The resulting ranking is $v, w, p, q, u$, ahead of all unshown units. Rank and Select top-$k$ operate on these scores. The input waveform and Before band highlight $u, v$; the After band highlights $v, w$ on the same original input. The $v > u$ and $v > w$ cards report completed comparisons; score updates apply to valid effects regardless of comparison completion or agreement with a proposed ordering. The queried set contains three units, while each explanation contains two. All waveforms, the score band, and the aligned selection bands preserve the chronological order $p, q, u, v, w$.

Under the stated reference scores, Reference-score Top-3 also selects $\{u, v, w\}$, so the case does not contrast the strategies' query choices. The hand-specified values illustrate comparison completion, evidence reuse, and the per-input score update rather than a measured fidelity gain.

### A.9 Shared exploration round

Algorithm A1 summarizes the round shared by reference construction and per-input explanation. The working graph $G$ contains the complete current execution records and their comparison relations. During per-input explanation, its frozen reference is read-only; new records and comparison states remain local to the current input.

**Algorithm A1** A shared exploration round

**Input:** $x$; working graph $G$; cached $f(x)$; reference scores when available; fixed predictor/intervention configuration and query budget $b = 3$.
**Output:** updated working graph $G$, containing valid current effects.

$M \leftarrow$ units with valid compatible current records in $G$.
Construct the comparison context and retain pairs $D^{\text{read}}$ as specified in §3.2.
Obtain admissible pairs $D^{\text{new}}$ through proposal generation followed by proposal mapping.
$D \leftarrow \{d \in D^{\text{read}} \cup D^{\text{new}} : d \nsubseteq M\}$; $U \leftarrow (\bigcup_{d \in D} d) \setminus M$.
**if** $|U| < b$ **then**
    Obtain one bounded replacement set through proposal generation and proposal mapping (Appendix A.4.3).
    Replace $D^{\text{new}}$ and recompute $D, U$, retaining the same $D^{\text{read}}$.
**end if**
Fix $D$ and $U$ for the current batch; preserve proposal records and register final comparisons in $G$.
**if** $U = \varnothing$ **then return** $G$.
$Q^\star \leftarrow$ a solution of Eq. (3), using the batch-size and tie rules in A.4.2.
**for each** $u \in Q^\star$ **in parallel do**
    Evaluate $f(T_u(x))$ and retain the attempted execution and response.
**end for**
Compute valid effects by Eq. (1), store each valid $e(x, u)$ once, and refresh $M$.
Associate the new records with every applicable registered comparison; update measured orderings or missing-endpoint states using Eq. (4).
**return** $G$.

The batch is fixed before its responses are observed, and every $T_u(x)$ is constructed from the same original input. Proposal and mapping repairs follow §A.4.3, validity and failure handling follow §A.3, and all actual calls are counted under §A.5.

A finite zero effect is a valid measurement. A failed query remains a failure rather than a zero effect and does not authorize an extra perturbation query. Records determine $M$ independently of explicit comparison links, and shared updates cover every applicable registered comparison. After the per-input query batch, R0 in Eq. (6) uses every valid current effect, whether the associated comparisons remain incomplete, agree with their proposed orderings, or disagree.

# B EXPERIMENTAL PROTOCOL

## B.1 TASK CONSTRUCTION AND DATA SEPARATION

**Model and task parameters.**

| Parameter | WeatherBench2 | VitalDB |
|---|---|---|
| Predictor | ConvGRU | Causal TCN |
| Depth | 2 layers | 4 blocks |
| Hidden channels | 32 | 64 |
| Input shape | $12 \times 9 \times 41 \times 61$ | $30 \times 40$ |
| Sampling interval | 6 h | 10 s |
| Forecast horizons | 6, 12, 24 h | 1, 3, 5 min |
| Intervention units | 90 | 20 |

WeatherBench2 inputs extend from $t-66$ h to $t$. The predictor forecasts 2 m temperature fields; each explanation target is the cosine-latitude-weighted mean over the output region in §B.1.1. VitalDB inputs cover five minutes. Each prediction target is the MAP median over a 60-second window ending at its forecast horizon. The temporal, variable, and spatial unit catalogs are specified in §§B.1.1–B.1.2.

**Data partitions and fixed sample sets.**

| Split or sample set | WeatherBench2 | VitalDB |
|---|---|---|
| Predictor training | 1979–2017 | 1,116 patients |
| Validation | 2018–2019 | 239 patients |
| Test | 2020–2021 | 239 patients |
| Reference construction | 200 validation windows | 200 validation windows |
| Explanation evaluation | 50 test windows | 50 test windows |

Weather samples retain coupled wind components in the same variable group. Each sample's complete interval, from its earliest input to its farthest prediction target, stays within one split. The evaluation windows have nonoverlapping complete intervals; prediction origins are at least 96 h apart on the six-hour grid.

The VitalDB base-variable groups are mean arterial pressure (MAP), heart rate (HR), end-tidal carbon dioxide ($ETCO_2$), and peripheral oxygen saturation ($SpO_2$). Medication, ventilation, and other contextual inputs remain outside the intervention catalog. All cases and windows from a patient are assigned to the same partition by subject identifier. The evaluation windows each come from a different test patient, and construction and evaluation patients are disjoint.

Training data are used to fit predictor weights and preprocessing statistics. Validation data are used for model selection, output-scale estimation, and reference construction. Predictors and preprocessing are frozen before reference construction; the shared graph and reference scores are frozen before per-input explanation. Method development used validation data and development panels explicitly marked as previously seen. The reported evaluation was performed after the method and its configuration had been frozen.

Output-scale estimation uses all eligible prepared validation windows as specified in §A.6.3; reference construction uses the fixed 200-window subset for each task.

### B.1.1 WeatherBench2 input and unit catalog

The WeatherBench2 input has shape $12 \times 9 \times 41 \times 61$. Its unit catalog is the Cartesian product of three temporal segments, six variable groups covering all nine channels, and five disjoint spatial regions on a 1.5° grid. A unit identifier is formed as `wb2::<temporal_segment>::<variable_group>::<spatial_region>`, yielding 90 units. Coupled wind components remain in the same variable group.

The forecast target covers grid centers from $15°$ to $45°$N and $100.5°$ to $145.5°$E, inclusive, on the 1.5° grid. This yields a $21 \times 31$ output grid at each of the +6, +12, and +24 h horizons, or a $3 \times 21 \times 31$ prediction field. The explained scalar at each horizon is the cosine-latitude-weighted mean over this output region. The output region is defined separately from the input unit partition named `target_core`.

**Variable groups.**

| Variable-group ID | Included channels | Meaning |
|---|---|---|
| `surface_temperature` | `t2m` | 2 m temperature |
| `surface_pressure` | `msl` | Mean sea-level pressure |
| `surface_wind` | `u10`, `v10` | Two 10 m wind components |
| `midlevel_circulation` | `z500` | 500 hPa geopotential |
| `lower_thermodynamics` | `t850`, `q850` | 850 hPa temperature and specific humidity |
| `lower_wind` | `u850`, `v850` | Two 850 hPa wind components |

**Temporal segments.** Indices are zero-based and half-open, with one frame every six hours.

| Temporal-segment ID | Input indices | Times relative to prediction cutoff t |
|---|---|---|
| `old` | `[0,4)` | −66, −60, −54, −48 h |
| `middle` | `[4,8)` | −42, −36, −30, −24 h |
| `recent` | `[8,12)` | −18, −12, −6, 0 h |

**Spatial regions.** Ranges specify grid centers and include both endpoints. The five regions partition the $41 \times 61$ input grid.

| Spatial-region ID | Latitude centers | Longitude centers |
|---|---|---|
| `south_belt` | 0–13.5°N | 75–165°E |
| `north_belt` | 45–60°N | 75–165°E |
| `upstream_west` | 15–43.5°N | 75–99°E |
| `target_core` | 15–43.5°N | 100.5–144°E |
| `downstream_east` | 15–43.5°N | 145.5–165°E |

### B.1.2 VitalDB input and unit catalog

The VitalDB predictor receives 20 numerical factors and 20 corresponding observation-mask channels at each of 30 ten-second bins, giving a $30 \times 40$ input. The intervention catalog covers only `MAP_family`, `HR_family`, `ETCO2_family`, and `SpO2`, crossed with five one-minute blocks. It therefore contains 20 units rather than one unit for every numerical factor. Medication, ventilation, BIS, and waveform features remain contextual inputs, and observation masks are preserved.

**Numerical input factors.**

| Category | Factors |
|---|---|
| 12 direct factors | ART_MBP, HR, SpO2, ETCO2, BIS, MAC, Propofol_CE, Remifentanil_CE, PEEP, MAWP, RR_CO2, TV |
| 8 derived factors | MAP_slope, MAP_volatility, pulse_pressure, HR_slope, ETCO2_slope, ECG_log_RMSSD, PPG_amplitude_log_ratio, PPG_amplitude_slope |

**Intervention groups and registered derived updates.**

| Intervention group | Base variable | Derived quantities recomputed after modification |
|---|---|---|
| MAP_family | ART_MBP | MAP_slope, MAP_volatility |
| HR_family | HR | HR_slope |
| ETCO2_family | ETCO2 | ETCO2_slope |
| SpO2 | SpO2 | None |

**Temporal blocks.** Each one-minute block contains six ten-second bins; indices are zero-based and half-open.

| Block | Input indices | Relative interval |
|---|---|---|
| block_1 | [0,6) | $[-5, -4)$ min |
| block_2 | [6,12) | $[-4, -3)$ min |
| block_3 | [12,18) | $[-3, -2)$ min |
| block_4 | [18,24) | $[-2, -1)$ min |
| block_5 | [24,30) | $[-1, 0)$ min |

Derived updates follow §A.6.2, using the original past-six-bin windows. Their affected range can extend up to 50 seconds after a modified block without crossing the input cutoff.

### B.2 Baseline settings and unit-score conversion

Table B1: **Baseline settings.** Counts specify each method's configuration. Integration steps, optimization steps, sampling budgets, and unit queries are distinct; elapsed time and total predictor calls require separate measurement.

| Method | Settings |
|---|---|
| Occlusion | queries/unit=1; units=90 (WeatherBench2), 20 (VitalDB) |
| Integrated Gradients | integration steps=64 |
| KernelSHAP | samples=512; l1_reg=0 |
| TimeSHAP | samples=512; prefix pruning=off |
| TIMING | time-conditioned gradient steps=64 |
| Dynamask | optimization steps=500; mask retention rate=0.1 |
| ContraLSP | optimization steps=100; training support samples=3 |

Integrated Gradients, TIMING, Dynamask, and ContraLSP use predictor gradients; Occlusion, KernelSHAP, and TimeSHAP do not. Occlusion and Integrated Gradients require no additional fitting. TIMING performs gradient integration without a separate fitting stage.

KernelSHAP fits a weighted regression for each input with its official solver. TimeSHAP uses per-input regression for grouped-cell attribution. Dynamask optimizes an explanation mask for each input. ContraLSP fits local mask, trend, and perturbation networks using the current input and three fixed training support samples.

Occlusion, KernelSHAP, and grouped TimeSHAP operate directly on the prescribed units. For Integrated Gradients and TIMING, we first sum the signed fine-grained contributions within each unit for each output. We then divide each component contribution by its output scale and take the absolute value. The maximum across the three outputs gives the unit score. For Dynamask and ContraLSP, the unit score is the mean of the explanation mask within that unit. Each method's ranking is fixed for all four explanation sizes.

### B.3 Donors, joint interventions, and metric aggregation

We evaluate each ranking after it has been fixed, using joint interventions with validation donors. These evaluation responses remain outside proposal generation, query selection, and R0. SHIR uses the fixed primary interventions in §A.6.2 to obtain its unit effects.

For each evaluation input, we select five validation donor windows in advance with a fixed seed and without replacement. All methods use these same donors. Weather donors match the input's UTC time slot and lie within approximately fifteen calendar days of its seasonal date. VitalDB donors satisfy the preprocessing quality criteria and come from patients outside the test partition. Selection does not use explanation scores. It excludes the recorded sample identifiers used for reference construction, legacy operator-threshold calibration, and earlier construction-strategy experiments.

The legacy exclusion lists contain 385 distinct WeatherBench2 windows and 400 distinct VitalDB windows from Q95 effect-threshold calibration under an earlier operator protocol. Each list is the deduplicated union of two operator-specific lists of 200 valid inputs. These exclusions differ from the full validation collection used for output-scale estimation in §A.6.3. Use in output-scale estimation alone does not exclude a donor window, so donors may overlap $\mathcal{V}_{\mathrm{cal}}$. The legacy lists define donor exclusions; current comparison decisions use the fixed primary operation and zero tolerance in §A.3.

Evaluation, reference-evidence calculations, and current-effect calculations share the positive frozen output scales $\sigma_c$ defined in §A.6.3. We compute prediction displacement by averaging the absolute normalized changes across the three explained outputs:

$$\delta_f(x, x') = \frac{1}{3} \sum_{c=1}^{3} \frac{|f_c(x') - f_c(x)|}{\sigma_c}. \tag{B1}$$

For input $x_i$, let $S_k(x_i)$ be the top $k$ units of its fixed ranking and $d_{i,j}$ its $j$-th donor. The donor-averaged deletion effect and retention error are

$$\begin{aligned} \mathrm{Del}_i(k) &= \frac{1}{5} \sum_{j=1}^{5} \delta_f(x_i, T_{\mathrm{delete}}(x_i, S_k(x_i), d_{i,j}))\,, \\ \mathrm{KeepErr}_i(k) &= \frac{1}{5} \sum_{j=1}^{5} \delta_f(x_i, T_{\mathrm{keep}}(x_i, S_k(x_i), d_{i,j}))\,. \end{aligned} \tag{B2}$$

For deletion, one donor jointly replaces every unit in $S_k$. For retention, it jointly replaces the complement of $S_k$ within the intervention catalog. Each transformation starts from the original input and retains the non-intervened contextual inputs. VitalDB also preserves observation masks and recomputes the registered affected derived inputs after changing base variables.

We compute the absolute normalized difference separately for each donor and output, then average over outputs, donors, and evaluation inputs. Equation (B1) uses the mean across the three outputs for evaluation. Methodology Eq. (1) uses their maximum to define a unit's effect under SHIR's primary operation.

The four explanation fractions are 5%, 7.5%, 10%, and 15%, with unit counts rounded upward. This gives $(5, 7, 9, 14)$ on WeatherBench2 and $(1, 2, 2, 3)$ on VitalDB. For $N = 50$ evaluation inputs and the four sizes $k_1, \ldots, k_4$, the summaries across explanation sizes are

$$\begin{aligned} \mathrm{Comp} &= \frac{1}{4N} \sum_{i=1}^{N} \sum_{m=1}^{4} \mathrm{Del}_i(k_m), \\ \mathrm{SuffErr} &= \frac{1}{4N} \sum_{i=1}^{N} \sum_{m=1}^{4} \mathrm{KeepErr}_i(k_m). \end{aligned} \tag{B3}$$

Each entry receives one-quarter of the weight across explanation sizes, including both VitalDB entries with $k = 2$. At the 10% explanation size, we report

$$\begin{aligned} \mathrm{CPD@10\%} &= \frac{1}{N} \sum_{i=1}^{N} \mathrm{Del}_i(k_{10\%}), \\ \mathrm{CPPerr@10\%} &= \frac{1}{N} \sum_{i=1}^{N} \mathrm{KeepErr}_i(k_{10\%}). \end{aligned} \tag{B4}$$

Both CPD@10% and CPP error@10% are evaluated at $k_{10\%} = 9$ on WeatherBench2 and $k_{10\%} = 2$ on VitalDB. Higher Comp and CPD indicate larger deletion effects; lower Suff error and CPP error indicate better preservation of the original prediction. These joint-endpoint definitions are separate from the path-accumulated CPD/CPP measures described by TIMING. All four metrics report normalized prediction displacement.

### B.4 Component interventions and query accounting

**Context Construction and Query Selection.** Flat evidence presentation and Reference-score Top-3 change only per-input query design, retaining SHIR's frozen reference. Flat presentation changes how the LLM receives the evidence while preserving the same raw evidence and reference/current labels. This change in context can lead to different proposals. Reference-score Top-3 (the variant previously labeled Historical-score Top-3 in the result records) holds the candidate pool fixed and selects the three unit queries by their reference scores.

**Evidence Reuse and Per-Input Score Updates.** Each reference-input episode is initialized separately. After each query round, the automatic-linking ablation disables automatic association of existing execution records with other applicable registered comparisons. The complete valid current-execution table in Appendix A.4.1 remains available, so already measured endpoints can still be identified.

The next round uses this modified local evidence state. The other algorithmic rules remain unchanged, while later proposals, queried units, and collected evidence can change. After the episodes finish, their graphs are merged and their recorded effects are aggregated into reference scores for per-input explanation.

Graph-only returns the reference ranking without current perturbation queries or per-input score updates. On its normal per-input explanation path, SHIR makes three current perturbation queries and applies every valid response in R0. The comparison therefore includes both the three queries and the corresponding per-input score update.

§A.5 gives the accounting rules for per-input perturbation queries, cached predictions, reference construction, LLM requests, and fidelity-evaluation calls.

### B.5 FULL ABLATION RESULTS

Table B2: **Full ablation results.** All rows use the same 50 evaluation inputs per task. SHIR reuses the single-run result in Table 1, while each ablation variant averages three repetitions. Flat presentation and Reference-score Top-3 change per-input query design; automatic linking is disabled after each reference-construction query round; Graph-only omits current queries and score updates. Bold marks the best mean for each task and metric.

| Dataset | Variant | Faithfulness metrics | | | |
|---|---|---|---|---|---|
| | | Comp ↑ | Suff error↓ | CPD @10%↑ | CPP error @10%↓ |
| **WeatherBench2** | SHIR | 0.1634 | **0.0310** | **0.1596** | **0.0254** |
| | Flat evidence presentation | 0.1558 | 0.0387 | 0.1473 | 0.0318 |
| | Reference-score Top-3 | **0.1660** | 0.0362 | 0.1574 | 0.0287 |
| | No automatic evidence linking | 0.1621 | 0.0321 | 0.1546 | 0.0302 |
| | Graph-only | 0.1531 | 0.0406 | 0.1540 | 0.0410 |
| **VitalDB** | SHIR | 0.8541 | **0.1469** | **0.8647** | **0.1402** |
| | Flat evidence presentation | 0.8407 | 0.1618 | 0.8324 | 0.1576 |
| | Reference-score Top-3 | **0.8561** | 0.1541 | 0.8575 | 0.1436 |
| | No automatic evidence linking | 0.8304 | 0.1488 | 0.8317 | 0.1677 |
| | Graph-only | 0.7790 | 0.2036 | 0.7868 | 0.1943 |