MOTIONBENCH-XAI: BENCHMARKING WHAT SHAPES SHAPLEY EXPLANATIONS IN SPATIOTEMPORAL MACHINE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting outcomes from spatiotemporal data, such as assessing Parkinson's severity from human motion, often requires explainability in addition to accuracy. Practitioners need to know what drove the prediction, whether it be spatial (a body joint), temporal (a specific window), or both. *Shapley values* are attributions computed by treating input features as players in a cooperative game and measuring how the model output changes as coalitions of players are removed. Computing them involves choosing what counts as players, how absent players are imputed, and how the game is evaluated. How these choices affect explanation quality is poorly understood for spatiotemporal data, and real data carries no ground-truth attributions to check against. We present **MotionBench-XAI**, a benchmark built on i) five synthetic spatiotemporal datasets with exact ground-truth Shapley values, ii) three real datasets across clinical gait, electrocardiogram, and audio, iii) ten attribution methods across three player sets, and iv) three trained architectures. We show that grouping features into players, across time, space, or both, matters more than the choice of imputer. Generative imputers improve explanations only when their conditional models are accurate. On real data, the standard proxy metrics select contradictory best methods. We distill these findings into practical insights and provide an open-source pipeline.

1 INTRODUCTION

Spatiotemporal data, such as 3D motion skeletons, multi-lead electrocardiograms, and audio signals, are used in machine learning for safety-critical domains including clinical diagnostics and monitoring. In these settings, practitioners often demand explainability alongside predictive accuracy. Beyond knowing that a prediction is accurate, they wish to understand which spatial components of a sequence, and at which times, drove that prediction. Shapley-value attribution (Shapley, 1953; Lundberg & Lee, 2017) is one of the most widely deployed tools to quantify the importance of input features in a model's final prediction (Barredo Arrieta et al., 2020; Slack et al., 2020). For example, it quantifies which joints in a patient's skeleton drive a Parkinson's severity prediction. The method treats model inputs as *players* in a cooperative *game* and computes the contribution, called Shapley value, of each player to the prediction.

Computing Shapley values requires computing the model's updated prediction when it is only shown a subset of players, for many different subsets. Since standard models cannot accept missing inputs, absent features are imputed using a variety of methods. Two active lines of work adapt the classical Shapley method to structured temporal data. *Manifold-aware* variants change the imputer, restricting imputed players to the data manifold (Frye et al., 2021; Aas et al., 2021a; Olsen et al., 2022; Taufiq et al., 2023; Zhang et al., 2026). *Temporal* variants organize the players in terms of time steps, and thus report attributions on a temporal scale (Bento et al., 2021; Nayebi et al.; Villani et al., 2022; Franco de la Peña et al., 2025). Each of those Shapley variants makes a series of design choices in the process of producing an explanation. The first choice defines what counts as players (whole time windows, whole joints, or fine joint$\times$window cells). The second choice defines how masked players are replaced (zeros, averages, copies of other sequences, or samples from a generative model of the data manifold). The third choice decides how the cooperative game is played, and

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.

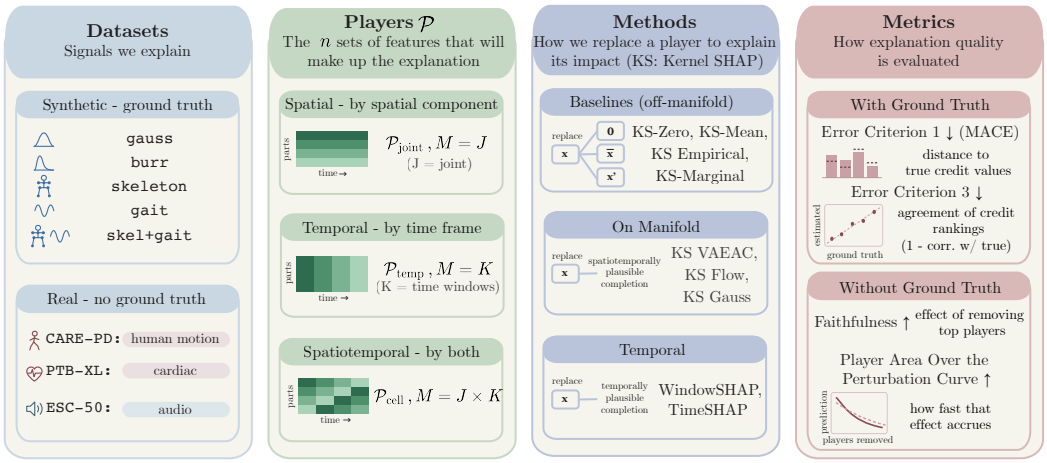


Figure 1: **MotionBench-XAI** evaluates spatiotemporal Shapley value attributions as a feature-based explainability method. *Datasets (first column):* Five synthetic spatiotemporal datasets paired with ground-truth Shapley values, and three real-world datasets. *Players $\mathcal{P}$ (second column):* Input features partitioned spatially (P_{joint}), temporally (P_{temp}), or across a spatiotemporal grid (P_{cell}). *Imputation methods (third column):* Strategies for replacing absent players via off-manifold baselines, on-manifold generative models, or temporal SHAP variants. *Metrics (fourth column):* Evaluation via exact attribution error and linear agreement error for synthetic cases, or proxy metrics (faithfulness, PlayerAOPC) for real-world cases where ground truth is absent.

what the ground truth, if it existed, would even be. The final choice is the *metric* used to grade the quality of the explanation, with or without access to that ground truth. These design choices, while critical to the outcome, are, to the best of our knowledge, never benchmarked jointly. This is especially critical for spatiotemporal data, which exhibits both feature dependence across channels and temporal coherence.

Contributions. To remedy this, we propose **MotionBench-XAI**, a benchmark to compare on and off-manifold Shapley methods on data with both feature dependence and temporal coherence (Figure 1). In both synthetic and real-world settings, MotionBench-XAI serves as an evaluation instrument to measure how hidden design choices dictate explanation quality for spatiotemporal data.

- We construct five spatiotemporal dataset families from varied distributions with closed-form marginal and conditional expectations, allowing ground-truth Shapley values per trained classifier (§4.1, §2.4).
- We perform a systematic sweep of every combination of player set (temporal, spatial, spatiotemporal), imputation family (standard, temporal, generative), ground truth, and metric, per dataset and per classifier, using one explanation pipeline (§5.1).
- We run the same pipeline on three real datasets where ground truth is not available (§4.2): clinical gait (CARE-PD; Adeli et al., 2025), 12-lead ECG (PTB-XL; Wagner et al., 2020), and environmental audio (ESC-50; Piczak, 2015) (§5.2).
- We summarize findings with direct practical consequence for practitioners using Shapley explanations in applied settings, distilled into a practical guide (Fig. 3).

§3 situates MotionBench-XAI within the literature detailing how it resolves current evaluation gaps. All code and checkpoints are released at anonymous.4open.science/r/motionbench-xai-41C7 (Appendix I).

2 Background and Methods

We introduce Shapley explanation and the notation for the four choices underlying a given explanation (Figure 1): the *players* (§2.2), the *imputation* (§2.3), the *ground truth* (§2.4), and the

metric (§2.5). MotionBench-XAI varies each choice independently. A glossary in Table 6 lists the full notation.

2.1 SHAPLEY VALUES ON SPATIOTEMPORAL DATA

A spatiotemporal instance is a tensor $\boldsymbol{x} \in \mathbb{R}^{J\times F\times T}$: J spatial units (skeleton joints, ECG leads, frequency bands), F coordinates per unit (e.g. $F{=}3$ for x, y, z positions), and T time frames. A trained classifier $f : \mathbb{R}^{J\times F\times T} \to \mathbb{R}$ maps $\boldsymbol{x}$ to a prediction, and the goal of an attribution is to divide attribution for that prediction among parts of $\boldsymbol{x}$.

Shapley attribution (Lundberg & Lee, 2017) formalizes this as a cooperative game. A *player set* $\mathcal{P} = \{1, \ldots, M\}$ partitions the entries of $\boldsymbol{x}$ into M disjoint groups (players). For a coalition of players $S \subseteq \mathcal{P}$, the *value function*, v, evaluates the model on the instance with the players outside S hidden:

$$v(S) \;=\; f(\tilde{\boldsymbol{x}}_S)\,, \tag{1}$$

where $\tilde{\boldsymbol{x}}_S$ denotes $\boldsymbol{x}$ with the entries of players in S kept and all other entries replaced by an *imputation* (defined in §2.3). We write $\bar{S} = \mathcal{P} \setminus S$ for the hidden players, $\boldsymbol{x}_S$ for the visible entries, and $\boldsymbol{x}_{\bar{S}}$ for the hidden block, so that $\tilde{\boldsymbol{x}}_S$ is assembled from $\boldsymbol{x}_S$ and an imputation $\tilde{\boldsymbol{x}}_{\bar{S}}$ of the rest. When the imputer is random, MotionBench-XAI draws R imputations and averages them into one completion, $\tilde{\boldsymbol{x}}_{\bar{S}} = \frac{1}{R}\sum_{r=1}^{R} \tilde{\boldsymbol{x}}_{\bar{S}}^{[r]}$. Hence, we evaluate the classifier only once on the averaged fill $\tilde{\boldsymbol{x}}_{\bar{S}}$. Every stochastic imputer uses the same draw count, $R{=}5$ on the synthetic suite and $R{=}1$ on the real datasets (Appendix A). Every game value is therefore reproducible. The ground truths of §2.4 replace the averaged draws with the exact conditional or marginal mean, the $R \to \infty$ limit of this average.

The Shapley value (Shapley, 1953) of player i is its average marginal contribution over all possible coalitions of players in $\mathcal{P}$,

$$\phi_i \;=\; \sum_{S\subseteq\mathcal{P}\setminus\{i\}} \frac{|S|!\,(M-|S|-1)!}{M!}\,[v(S\cup\{i\}) - v(S)]\,. \tag{2}$$

We write $\phi = (\phi_1, \ldots, \phi_M)^\top \in \mathbb{R}^M$ for the vector of Shapley values across all M players. Exact enumeration is exponential in M. All methods in this paper estimate Eq. 2 with KernelSHAP (Lundberg & Lee, 2017), a weighted least-squares regression over sampled coalitions. We write $\hat{\phi} \in \mathbb{R}^M$ for the KernelSHAP estimate. We hold this estimator fixed across all KernelSHAP methods, evaluating the same number of coalitions throughout (Appendix A). The temporal SHAP baselines run their reference implementations (§2.3), so their rows also reflect their own estimators.

2.2 WHAT CAN A PLAYER BE?

The player set $\mathcal{P}$ decides how the input features will be preemptively divided before attributions are assigned (Figure 1, second column). Figure 2 shows the three families of players MotionBench-XAI evaluates on an example skeleton sequence. In every family, a player always includes all F coordinates of the entries it covers. **Temporal players** $\mathcal{P}_{\text{temp}}$ cut the T frames into K uniform windows, and each window is one player ($M = K$): this is the player set of temporal SHAP methods. **Spatial players** $\mathcal{P}_{\text{joint}}$ make each spatial unit one player ($M = J$). **Spatiotemporal players** $\mathcal{P}_{\text{cell}}$ combines the two, one player per joint $\times$ window cell ($M = J \times K$). For example, on the CARE-PD gait data, the same instance has $K = 4$ windows, $J = 17$ joints, which gives $M = 68$ spatiotemporal players, also called cells.

2.3 WHAT CAN AN IMPUTATION BE?

Once we have our players, Eq. 1 requires masking some of these players turn by turn. To avoid having to modify the model being explained, the masked players need to be imputed by some other value. Writing x_i for the entries of a single hidden player $i \in \bar{S}$, MotionBench-XAI considers three families of methods (Figure 1, third column).

Standard (off manifold). The hidden player is replaced by a value that ignores the rest of the sequence: zeros ($x_i = 0$), the training mean ($x_i = \mathbb{E}[x_i]$), or a draw from the marginal ($x_i \sim$

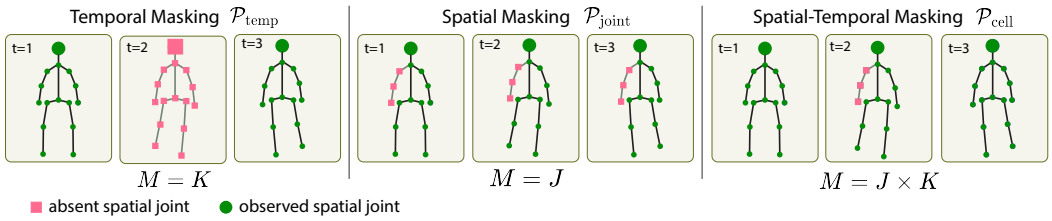


Figure 2: **What can a player be?** The three player sets, shown as masking on a skeleton sequence: a time window ($\mathcal{P}_{\text{temp}}$), a spatial joint ($\mathcal{P}_{\text{joint}}$), or a joint×window cell ($\mathcal{P}_{\text{cell}}$).

$P(x_i)$, copied from another sequence in the evaluation pool). These define the methods KS-Zero and KS-Mean (Lundberg & Lee, 2017), KS-Marginal (Aas et al., 2021a), and KS-Empirical (Aas et al., 2021a; Redelmeier et al., 2020). KS-Empirical refines KS-Marginal by copying values from the sequences most similar to the explained one, weighted by a kernel. These imputations are cheap and model-agnostic but off manifold (Frye et al., 2021). On human skeleton data, for example, they can produce anatomically impossible motion.

Temporal (on the temporal manifold). WindowSHAP (Nayebi et al.) and TimeSHAP (Bento et al., 2021) keep the standard zero fill. They distinguish themselves by cleverly picking temporal players to begin with: fixed windows (WindowSHAP-S), adaptively split windows (WindowSHAP-D), or per-event scoring with pruning (TimeSHAP). We evaluate them at temporal players only, aggregating per-timestep attributions into the K uniform windows.

Generative (on manifold). This family uses a conditional model to impute the hidden player from the visible ones, $x_i \sim P(x_i \mid \boldsymbol{x}_S)$. KS-VAEAC (Olsen et al., 2022; Ivanov et al., 2019) and KS-Flow use learned conditional models; their formulations are given in Appendix H. We build KS-Flow for this benchmark so that the generative family contains two learned imputers of different strength. It mirrors KS-VAEAC except that it replaces the VAEAC generative model with a flow matching model (Lipman et al., 2023). KS-Gauss (Aas et al., 2021a) is the closed-form Gaussian conditional, with no learned components. These imputers stay near the data manifold but are learned or assumption-bound, hence imperfect.

2.4 What can a ground truth be?

In cooperative game theory, a game is defined as a rule that assigns a score to every coalition of players. In our setting, that rule is Eq. 1. The score of a coalition is the model's prediction on the input with the remaining players imputed, so the imputation is part of the rule. When we change how the hidden players are filled, the same coalition receives a different score. We are then playing a different game, and Eq. 2 assigns different attributions to the same input. In this work, two games stand out. If we fill the hidden players with their marginal expectation $\mathbb{E}[\boldsymbol{x}_{\bar{S}}]$, we play the *marginal game*. If we instead fill them with their conditional expectation $\mathbb{E}[\boldsymbol{x}_{\bar{S}} \mid \boldsymbol{x}_S]$, we play the *conditional game* (Janzing et al., 2020; Chen et al., 2020). The standard and temporal imputers of §2.3 approximate the marginal game. The generative imputers approximate the conditional game. A *ground truth Shapley value* solves a game exactly, using the expectation itself instead of an estimate of it.

On synthetic data drawn from a well-chosen distribution, both expectations are available in closed form for every coalition. We can therefore compute both ground truths exactly (see App. B). For player sets with at most 12 players we enumerate every coalition, so the ground-truth Shapley values carry no error at all. For larger player sets we solve the same exact games on large sampled coalition designs and measure the remaining target error directly Importantly, since the two exact imputations differ, the two ground truths assign different attributions. MotionBench-XAI grades each imputer method against the ground truth of its own game, which we call *game-matched* grading. Where possible, we also report the standard convention of grading everything against $\phi^{\star-\text{cond}}$. On real data, neither expectation is knowable.

Table 1: **Coverage of MotionBench-XAI's evaluation dimensions.** Synthetic ground truth provides known informative features or numerical attribution targets. Conditional imputers fill missing inputs using the observed inputs and are evaluated within explanation methods. Multiple player sets means comparing different groupings of input features. ✓ indicates a reported component and – an unreported one.

Study	Synthetic ground truth	Temporal data	Conditional imputers	Multiple player sets
BAM (Yang & Kim, 2019)	✓	–	–	–
CLEVR-XAI (Arras et al., 2022)	✓	–	–	–
Olsen et al. (2024)	✓	–	✓	–
XTSC-Bench (Höllig et al., 2023)	✓	✓	–	–
xaitimesynth (Baer, 2026)	✓	✓	–	–
SHAPformer (Hertel et al., 2026)	✓	✓	–	✓
TimeSHAP (Bento et al., 2021)	–	✓	–	✓
ShaTS (Franco de la Peña et al., 2025)	–	✓	–	✓
GroupSegment-SHAP (Kim & Park, 2026)	✓	✓	–	✓
MotionBench-XAI (ours)	✓	✓	✓	✓

2.5 What metric do we use to score the quality of an explanation?

With ground truth (synthetic data). Following Aas et al. (2021a); Olsen et al. (2022), MotionBench-XAI reports two error criteria (EC, following Olsen et al. (2022)):

$$\mathrm{EC1} = \tfrac{1}{M} \left\| \hat{\phi} - \phi^{\star} \right\|_1 \qquad \text{and} \qquad \mathrm{EC3} = 1 - r\left(\hat{\phi},\, \phi^{\star}\right), \tag{3}$$

where $\phi^{\star}$ is the game-matched ground truth and r is the Pearson correlation per sequence. EC1 is the mean absolute attribution error. EC3 is its scale-invariant counterpart that measures how well the player attributions' order and weight match the ground truth. Lower is better for both.

Without ground truth (real data). MotionBench-XAI uses the two standard deletion-based tests. *Faithfulness correlation* (Bhatt et al., 2020) correlates the summed attributions of hidden players with the model's performance drop across coalitions. *PlayerAOPC* (Samek et al., 2017) removes players in decreasing-attribution order and records the model's response. Higher is better for both. For both metrics, $v(S)$ is computed with each method's own imputer, on the same coalitions. EC3 and faithfulness can be compared across player sets. However, EC1 and PlayerAOPC scale with the number of players, so their values compare only within one player set.

3 Related Work

Prior work addresses dependence-aware Shapley attribution, temporal player definitions, and explanation evaluation. MotionBench-XAI studies how these choices interact (Table 1).

Dependence-aware and temporal Shapley attribution. Marginal-background KernelSHAP (Lundberg & Lee, 2017) fills missing features with background values without conditioning on observed features. With dependent features, this can produce off-manifold inputs (Frye et al., 2021). Conditional Shapley methods include empirical and copula-based conditioning (Aas et al., 2021a;b) and conditional inference trees (Redelmeier et al., 2020). Variational approaches (Olsen et al., 2022) use VAEAC (Ivanov et al., 2019) to sample missing features. ManifoldShap restricts interventional evaluations to a specified domain of validity (Taufiq et al., 2023). Related work studies transport-based paths for Aumann–Shapley attribution (Zhang et al., 2026).

For sequences, TimeSHAP provides feature, timestep and cell attributions (Bento et al., 2021), while WindowSHAP groups adjacent time points within each variable (Nayebi et al.). Villani et al. (2022) develop autoregressive surrogates and time-consistent attributions. ShaTS studies temporal and feature groupings (Franco de la Peña et al., 2025), using background substitution. GroupSegment-SHAP combines dependence-based variable groups with temporal segments (Kim & Park, 2026) and mainly uses mean replacement. Grouping alone does not provide conditional completion.

Marginal and conditional Shapley games answer different explanatory questions (Janzing et al., 2020; Chen et al., 2020; Sundararajan & Najmi, 2020; Kumar et al., 2020). Our synthetic targets

Table 2: **The five synthetic datasets.** All are tensors $\boldsymbol{x} \in \mathbb{R}^{J\times 3\times T}$. The columns give each dataset's spatial coupling, temporal correlation, and marginal distribution, with the full construction in Appendix B. K is the number of temporal windows and $M = J \times K$ the number of joint×window cells. Each dataset has N=200 evaluation sequences.

	Dataset	Spatial coupling Σ_J	Temporal Σ_T	Marginal	J	T	K	M
	`gauss`	low	AR(1)	Gaussian	5	16	4	20
	`burr`	low	AR(1)	Burr	5	20	5	25
	`skeleton`	high (H3.6M)	AR(1)	Gaussian	17	16	4	68
	`gait`	uncoupled	periodic	Gaussian	17	16	4	68
	`skel+gait`	high (H3.6M)	periodic	Gaussian	17	16	4	68

use marginal or conditional mean fills (§2.4). Predicting on these fills generally differs from averaging predictions over imputations. Amortized estimators (Jethani et al., 2022) and improved regression-based estimation (Covert & Lee, 2021) address computational efficiency.

Explanation evaluation. Fixed-model deletion tests can confound feature importance with the distribution shift caused by replacing inputs. ROAR addresses this by retraining on modified data (Hooker et al., 2019). Bhatt et al. (2020) study faithfulness and sensitivity, while Quantus provides evaluation metrics and illustrates how their settings affect rankings (Hedström et al., 2023).

Benchmarks and spatiotemporal applications. Synthetic time-series benchmarks identify informative regions (Ismail et al., 2020; Höllig et al., 2023; Baer, 2026). These masks differ from numerical Shapley values (Table 1). Olsen et al. (2022; 2024) evaluate conditional Shapley estimates in tabular simulations with known distributions. SHAPformer computes numerical Shapley values for a synthetic data-generating process (Hertel et al., 2026), which need not match a fitted model's attributions.

Applications to human action and motion use DeepExplainer with background samples (Auletta et al., 2023; Tempel et al., 2025a). ShapGCN assesses explanations by perturbing learned edge weights (Tempel et al., 2025b), while ECG analysis uses layer-wise relevance propagation (Strodthoff et al., 2021).

4 Datasets

4.1 Synthetic datasets: designed distributions with oracles

We generate five synthetic datasets with computable marginal and conditional imputations (Appendix B). Our five synthetic datasets (Table 2) vary along three specific axes:

1. Marginal family. `gauss` draws Gaussian coordinates. `burr` pushes the same latent sequence through a symmetric heavy-tailed Burr copula (tail index 4), testing whether conclusions survive non-Gaussian marginals.

2. Spatial structure. `skeleton` uses a joint-correlation matrix inspired by the Human3.6M skeleton (Ionescu et al., 2014) (J=17 joints), giving the strong cross-joint coupling of real motion capture. `gauss` and `burr` use weak coupling (J=5).

3. Temporal structure. `gauss`, `burr`, and `skeleton` use autoregressive temporal correlation. `gait` and `skel+gait` use a gait-inspired *periodic* temporal correlation.

Each synthetic dataset defines a three-class classification task. To label a sequence, we average the coordinates of its first joint over all frames, which gives one label per sequence. We then split the sequences into three equally sized classes according to this label. For each dataset we train three classifier architectures. An MLP sees only the mean of each temporal window, while a 1-D convolutional network (LeCun et al., 1998) and a Transformer (Vaswani et al., 2017) see the raw frames. Training details are in Appendix E.

Table 3: **Synthetic results, temporal players** (K uniform windows). EC3 = one minus the Pearson correlation between estimated and ground-truth credits (lower is better), mean ± spread across the classifier architectures that perform sufficiently well (Appendix F). Each method is graded against the exact ground truth of its own imputer family (game-matched). Gray text indicates periodic datasets (Result 1); they are excluded from the per-column best. EC1 is reported in Appendix C.

	Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers	KS-Marginal	0.36 ±0.04	0.41 ±0.09	0.28 ±0.04	**0.21** ±0.08	**0.16** ±0.07
	WindowSHAP-S	0.29 ±0.06	0.50 ±0.13	0.30 ±0.01	0.44 ±0.04	0.54 ±0.13
	WindowSHAP-D	0.59 ±0.09	0.71 ±0.19	0.54 ±0.12	0.86 ±0.14	0.80 ±0.18
	TimeSHAP	0.35 ±0.10	0.37 ±0.08	0.37 ±0.15	0.43 ±0.27	0.50 ±0.16
	KS-Empirical	0.51 ±0.03	0.47 ±0.09	0.48 ±0.03	0.31 ±0.05	0.28 ±0.09
Generative imputers	KS-VAEAC	**0.12** ±0.03	**0.16** ±0.05	**0.10** ±0.02	0.76 ±0.08	0.87 ±0.04
	KS-Flow	0.41 ±0.07	0.38 ±0.09	0.29 ±0.03	0.78 ±0.10	0.77 ±0.12
	KS-Gauss (closed form)	0.24 ±0.04	0.28 ±0.08	0.23 ±0.03	0.69 ±0.09	0.61 ±0.15
	KS-Oracle (reference)	0.21 ±0.05	0.20 ±0.06	0.19 ±0.04	0.22 ±0.06	0.22 ±0.03

4.2 Real-world datasets

MotionBench-XAI includes three public datasets spanning spatiotemporal modalities and application domains of particular interest for explanation. **CARE-PD** (Adeli et al., 2025): 3-D skeletons of people walking (J=17). The goal is to predict the patient's Parkinson's severity score. We consider pre-trained architectures (MotionBert (Zhu et al., 2023) and MotionAGFormer (Mehraban et al., 2024)), reported as models A and B, over three leave-one-subject-out folds. **PTB-XL** (Wagner et al., 2020): recordings of a 12-lead electrocardiogram. The model (a per-fold 1-D ResNet) predicts normal cardiac health or myocardial infarction. **ESC-50** (Piczak, 2015): 5-second environmental audio clips over 50 classes. We explain the Audio Spectrogram Transformer (AST) (Gong et al., 2021). Further details are provided in Appendix G.

5 Results

Protocol. MotionBench-XAI evaluates every method with identical amounts of coalitions at each player set, ensuring identical amounts of classifier evaluations per explanation (Appendix A). Every run uses up to N=200 sequences per fold. Synthetic numbers are averaged over the participating classifier architectures, and ± is the spread across architectures. Real-data numbers are averaged over the three data folds, and ± is the spread across folds. A classifier participates if and only if it performs at least 0.05 above chance and has a train–test gap of at most 0.30. This criterion excludes the CNN on the two periodic synthetic datasets (overfitting). Per-classifier accuracies are in Appendix F. Synthetic tables (Tables 3–4) report EC3 under game-matched grading (method is scored against the ground truth of its own imputer family). EC1, the mean absolute credit error, and the cross-graded convention are in Appendix C.

5.1 Synthetic suite: ground-truth grading

Tables 3–4 score every method on every dataset at the three player sets. The KS-Oracle row runs KernelSHAP with the exact conditional imputer under the same protocol as every other method. The ground truth fills hidden players with the exact conditional *mean*, while KS-Oracle, like every stochastic imputer, averages R=5 random draws of the hidden players. Its nonzero error therefore measures the cost of the shared protocol itself. Namely, it accounts for the finite draws and, at player sets too large to enumerate every coalition, the shared sampled coalition design (Appendix B). Thus the row serves as a reference: error above it reflects the quality of a method's imputer.

Result 1: Choosing a good player set is the first step towards achieving meaningful explanations. On periodic data like `gait` and `skel+gait` (Table 3), temporal players are unsurprisingly meaningless: they contain the same information and thus receive uniform attribution values. On the other hand, spatial players provide useful insights. For example, on `skel+gait`,

Table 4: **Synthetic results, spatial and spatiotemporal players.** EC3, mean ± spread across the accuracy-passing classifier architectures, game-matched grading. Gray represents periodic data (Result 1) and is excluded from per-column best.

(a) *Spatial players (one player per joint; J=5 or 17)*

	Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers	KS-Marginal	0.32 ±0.03	0.20 ±0.08	0.30 ±0.02	**0.17** ±0.15	0.16 ±0.09
	KS-Empirical	0.40 ±0.02	0.28 ±0.05	0.41 ±0.03	0.26 ±0.13	0.27 ±0.09
Generative imputers	KS-VAEAC	**0.19** ±0.06	**0.13** ±0.02	**0.16** ±0.06	0.19 ±0.15	0.22 ±0.05
	KS-Flow	0.29 ±0.06	0.23 ±0.06	0.35 ±0.07	0.17 ±0.14	0.19 ±0.07
	KS-Gauss (closed form)	0.21 ±0.07	0.17 ±0.04	0.25 ±0.08	0.22 ±0.15	**0.14** ±0.07
	KS-Oracle (reference)	0.18 ±0.07	0.10 ±0.04	0.19 ±0.07	0.16 ±0.14	0.08 ±0.04

(b) *Spatiotemporal players (joint×window cells; M=20–68)*

	Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers	KS-Marginal	0.19 ±0.04	0.22 ±0.05	0.24 ±0.01	**0.16** ±0.12	**0.14** ±0.09
	KS-Empirical	0.35 ±0.00	0.33 ±0.02	0.39 ±0.00	0.26 ±0.11	0.23 ±0.10
Generative imputers	KS-VAEAC	0.20 ±0.05	**0.22** ±0.02	**0.24** ±0.03	0.48 ±0.09	0.57 ±0.01
	KS-Flow	0.26 ±0.05	0.29 ±0.05	0.32 ±0.03	0.40 ±0.11	0.53 ±0.01
	KS-Gauss (closed form)	**0.14** ±0.03	0.22 ±0.03	0.24 ±0.03	0.38 ±0.13	0.36 ±0.07
	KS-Oracle (reference)	0.13 ±0.03	0.18 ±0.02	0.20 ±0.02	0.21 ±0.08	0.22 ±0.05

KS-VAEAC's EC3 falls from 0.87 for temporal players to 0.22 for spatial players (Table 4a). This is a clearcut example of how correctly choosing players is key in determining explanation usefulness.

Result 2: Generative imputers are accurate so long as their conditional models are accurate. Under game-matched grading, KS-VAEAC reaches the lowest EC3 among the displayed methods on every non-periodic dataset at temporal and spatial players (Tables 3 and 4a). KS-Flow trails it by 0.10–0.29 on the same columns. At joint×window cells, the learned-imputer advantage over KS-Marginal disappears on `skeleton` and `burr`: differences are at most 0.005, below the target variability reported in Appendix D. KS-Gauss has the lowest error on `gauss` cells and is competitive on `skeleton` and `burr` (Table 4b). In summary, the closed-form conditional imputers are just as good as the learned imputers so long as they correctly describe the data.

Result 3: Fine granularity has a measurable evaluation cost. At joint×window cells, the grading targets are computed on sampled coalition designs and carry approximation error. We measure this error by recomputing each target using a new set of coalitions. For `skeleton` cells (M=68), increasing the target budget from 8,192 to 32,768 coalitions reduces conditional-target disagreement from EC3 ≈ 0.12 to ≈ 0.03 (Appendix D). As such, fine-grained method comparisons must be interpreted alongside this target uncertainty.

Result 4: Grading everything against one ground truth mixes up two different questions. Prior oracle benchmarks grade every method against the conditional ground truth (Aas et al., 2021a; Olsen et al., 2022), including methods that estimate the marginal game. On our centered data, KS-Zero implements the exact marginal imputation. Its temporal EC3 on `gauss` is 0.00 under game-matched grading but 0.25 when graded against the conditional target (Appendix C). This error reflects the different targets rather than an inaccurate marginal explanation. The takeaway is the importance of stating the intended game first, then grading against that game's ground truth.

5.2 Real-world suite: no oracle

Table 5 evaluates the KernelSHAP family on the three real datasets with the two standard no-oracle metrics, across temporal and spatial player sets. Temporal SHAP variants build in their own temporal players, and KS-Gauss requires a fitted Gaussian model of the full tensor. Hence, both are compared only on the synthetic suite. Joint×window results for CARE-PD and ESC-50 are in Appendix G.

Table 5: **Real-world results.** Panel (a): faithfulness correlation between summed hidden credits and the prediction drop. Panel (b): PlayerAOPC, the mean drop along the deletion path. Mean ± spread across data folds; bold: best per column and panel. AOPC magnitudes are comparable within a column.

(a) *Faithfulness correlation (↑)*

	Temporal players				*Spatial players*			
Method	CARE-PD model A, windows	CARE-PD model B, windows	PTB-XL windows	ESC-50 windows	PTB-XL per lead	CARE-PD model A, joints	CARE-PD model B, joints	ESC-50 freq. bands
KS-Zero	0.86 ±0.02	0.46 ±0.05	0.88 ±0.01	0.88 ±0.01	**0.66** ±0.03	**0.91** ±0.03	0.45 ±0.02	0.78 ±0.01
KS-Mean	0.83 ±0.12	0.82 ±0.08	0.88 ±0.01	0.88 ±0.01	0.66 ±0.03	0.76 ±0.07	0.42 ±0.09	0.79 ±0.01
KS-Marginal	**0.96** ±0.02	**0.95** ±0.02	**0.91** ±0.01	**0.92** ±0.00	0.52 ±0.01	0.81 ±0.02	**0.56** ±0.11	0.84 ±0.01
KS-VAEAC	0.69 ±0.05	0.73 ±0.08	0.85 ±0.04	0.91 ±0.00	0.55 ±0.04	0.03 ±0.35	0.30 ±0.06	**0.87** ±0.01
KS-Flow	0.66 ±0.05	0.68 ±0.01	0.63 ±0.02	0.90 ±0.01	0.64 ±0.01	0.29 ±0.17	0.45 ±0.09	0.86 ±0.01

(b) *PlayerAOPC (↑)*

	Temporal players				*Spatial players*			
Method	CARE-PD model A, windows	CARE-PD model B, windows	PTB-XL windows	ESC-50 windows	PTB-XL per lead	CARE-PD model A, joints	CARE-PD model B, joints	ESC-50 freq. bands
KS-Zero	-0.14 ±0.26	0.19 ±0.06	0.19 ±0.03	0.77 ±0.00	0.47 ±0.04	-0.08 ±0.23	**0.43** ±0.13	0.54 ±0.01
KS-Mean	**0.29** ±0.09	**0.35** ±0.11	0.19 ±0.03	0.78 ±0.00	0.47 ±0.04	**0.18** ±0.05	0.33 ±0.09	0.55 ±0.01
KS-Marginal	0.12 ±0.05	0.16 ±0.07	0.18 ±0.01	**0.88** ±0.01	0.37 ±0.03	0.08 ±0.08	0.18 ±0.02	**0.79** ±0.02
KS-VAEAC	0.09 ±0.04	0.20 ±0.02	**0.59** ±0.03	0.82 ±0.00	0.45 ±0.06	0.03 ±0.04	0.08 ±0.01	0.69 ±0.01
KS-Flow	0.11 ±0.02	0.17 ±0.01	0.27 ±0.04	0.85 ±0.01	**0.58** ±0.05	0.05 ±0.02	0.13 ±0.01	0.75 ±0.02

Setting up a Shapley analysis: three questions to answer.

1. **What is known about the data's distribution?** If the distribution is known analytically, use the exact conditional imputer. If it can be learned from data, train a generative imputer and check that it can reconstruct a hidden player well. If neither holds, use a standard imputer (marginal game).
2. **What is the goal of the explanation?** Match the players to the question. Temporal explanations cells for temporal players, and so on. More players means better precision (assuming the estimator is good) but also more model evaluations at explanation time.
3. **How will the explanation be scored?** With a ground truth, grade each method against the game its imputer plays, recording what that ground truth represents. Without one, report faithfulness and PlayerAOPC together. Only compare PlayerAOPC values within one player set.

Figure 3: **Practitioner decision guide.** Setup procedure, further described in Appendix K.

Result 5: Without a ground truth, the two standard metrics disagree, and the best method changes with the dataset. On CARE-PD temporal players, faithfulness assigns the highest score to KS-Marginal (0.96 and 0.95 on models A and B; Table 5a), while PlayerAOPC ranks KS-Mean highest (0.29 and 0.35; Table 5b). Yet, on ESC-50, KS-Marginal scores highest on both metrics for temporal players. No single method is best across datasets and player sets (Appendix G). This underscores the importance of comparing possible explanation designs going beyond choice of metric. In view of doing this in practice, we propose a practitioner decision guide in Fig. 3.

6 Limitations and Conclusions

MotionBench-XAI exposes and compares the design choices behind game-theoretic explanations for spatiotemporal machine learning. The benchmark grades ten attribution methods against exact ground truths on five designed datasets, runs the same pipeline on three real ones, and distills the outcome into useful insights. We conclude with two limitations that offer promising avenues for future work. First, the oracle construction is limited to Gaussian-field and copula families: extending exact ground truths to richer designed manifolds, such as articulated dynamics and discrete events, would widen the benchmark's reach. Second, the suite's dimensions are moderate ($J \leq 17$, $M \leq 68$) and its real-data conclusions are specific to the evaluated checkpoints: scaling to finer granularities and more architectures could better characterize real-world explanation scenarios. Overall, we look forward to MotionBench-XAI powering more intentional and practical use of Shapley explanation in applied spatiotemporal settings.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: designing and giving feedback on experimental methodology, implementing the benchmark methods and evaluation pipeline (including the generators that sample the synthetic datasets), cleaning and reformatting the real-data caches, assisting in the writing of the proofs of Proposition 1, and assisting in the interpretation of results. Generative AI did not author dataset contents directly: every synthetic sequence is drawn from the released, seeded generators, and all real data come from the cited public datasets. Translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools for creating and editing software code, editing paper text for readability and accuracy, and identifying and positioning related literature. We have reviewed all AI-assisted work and code. Moreover, AI-assisted code is covered by the released test suite. The bulk of the final submission's main text was human written while the appendices and supporting details were a combination of human and AI writing. All figures were created by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All data are public research datasets used under their licenses: CARE-PD (anonymized clinical gait motion capture), PTB-XL (de-identified ECG), ESC-50 (environmental audio), each cited in §4.2. No new human data were collected. We also emphasize that the purpose of the benchmark is to help evaluate and compare explanation methods. It is not intended to produce any clinical recommendations.

REPRODUCIBILITY STATEMENT

The code release contains the data generators, trainers, attribution and grading pipelines, the canonical result files behind every table, and SHA-256-verified reference checkpoints; Appendix I gives the complete account. All random streams behind reported numbers are pinned.

REFERENCES

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298:103502, 2021a.

Kjersti Aas, Thomas Nagler, Martin Jullum, and Anders Løland. Explaining predictive models using Shapley values and non-parametric vine copulas. *Dependence Modeling*, 9(1):62–81, 2021b.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, pp. 9505–9515, 2018.

Vida Adeli, Ivan Klabučar, Javad Rajabi, Benjamin Filtjens, Soroush Mehraban, Diwei Wang, Trung Hieu Hoang, Minh Do, Hyewon Seo, Candice MULLER, Daniel Coelho, Claudia de Oliveira, Pieter Ginis, Moran Gilat, Alice Nieuwboer, Joke Spildooren, J. Mckay, Hyeokhyen Kwon, Gari Clifford, Christine Esper, Stewart Factor, Imari Genias, Amirhossein Dadashzadeh, Leia Shum, Alan Whone, Majid Mirmehdi, Andrea Iaboni, and Babak Taati. Care-pd: A multi-site anonymized clinical dataset for parkinson's disease gait assessment. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4391. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/bedc73979a95be7727af0c9a99c675ce-Paper-Datasets_and_Benchmarks_Track.pdf.

Leila Arras, Ahmed Osman, and Wojciech Samek. CLEVR-XAI: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022.

Fabrizia Auletta, Rachel W. Kallen, Mario di Bernardo, and Michael J. Richardson. Predicting and understanding human action decisions during skillful joint-action using supervised machine learning and explainable-ai. *Scientific Reports*, 13(1):4992, Mar 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-31807-1. URL https://doi.org/10.1038/s41598-023-31807-1.

Gregor Baer. xaitimesynth: A python package for evaluating attribution methods for time series with synthetic ground truth. *arXiv preprint arXiv:2603.06781*, 2026.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

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

Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3016–3022, 2020.

Hugh Chen, Joseph D. Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

Ian Covert and Su-In Lee. Improving KernelSHAP: Practical Shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3457–3465, 2021.

Manuel Franco de la Peña, Ángel Luis Perales Gómez, and Lorenzo Fernández Maimó. ShaTS: A shapley-based explainability method for time series artificial intelligence models applied to anomaly detection in industrial internet of things. *arXiv preprint arXiv:2506.01450*, 2025.

Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. In *International Conference on Learning Representations*, 2021.

Yuan Gong, Yu-An Chung, and James Glass. AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, pp. 571–575, 2021. doi: 10.21437/Interspeech.2021-698.

Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. Quantus: An explainable AI toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34):1–11, 2023.

Matthias Hertel, Sebastian Pütz, Ralf Mikut, Veit Hagenmeyer, and Benjamin Schäfer. Explainable time-series forecasting with sampling-free shap for transformers. *Nature Communications*, 17(1): 4723, 2026.

Jacqueline Höllig, Steffen Thoma, and Florian Grimm. Xtsc-bench: quantitative benchmarking for explainers on time series classification. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pp. 1126–1131. IEEE, 2023.

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

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.

Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. In *International Conference on Learning Representations*, 2019.

Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable AI: A causal problem. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2907–2916, 2020.

Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time Shapley value estimation. In *International Conference on Learning Representations (ICLR)*, 2022.

Jinwoong Kim and Sangjin Park. GroupSegment-SHAP: Shapley value explanations with group-segment players for multivariate time series. *arXiv preprint arXiv:2601.06114*, 2026.

I. Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with Shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 2022.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Angel Martínez-González, Michael Villamizar, and Jean-Marc Odobez. Pose transformers (POTR): Human motion prediction with non-autoregressive transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 2276–2284, 2021.

Soroush Mehraban, Vida Adeli, and Babak Taati. MotionAGFormer: Enhancing 3D human pose estimation with a transformer-GCNFormer network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6920–6930, 2024.

Amin Nayebi, Sindhu Tipirneni, Chandan K Reddy, Brandon Foreman, and Vignesh Subbian. Windowshap: An efficient framework for explaining time-series classifiers based on shapley values. *URL: http://arxiv. org/abs/2211.06507. doi*, 10.

Lars Henry Berge Olsen, Ingrid Kristine Glad, Martin Jullum, and Kjersti Aas. Using Shapley values and variational autoencoders to explain predictive models with dependent mixed features. *Journal of Machine Learning Research*, 23(213):1–51, 2022.

Lars Henry Berge Olsen, Ingrid Kristine Glad, Martin Jullum, and Kjersti Aas. A comparative study of methods for estimating model-agnostic shapley value explanations. *Data Mining and Knowledge Discovery*, 38(4):1782–1829, Jul 2024. ISSN 1573-756X. doi: 10.1007/s10618-024-01016-z. URL https://doi.org/10.1007/s10618-024-01016-z.

Karol J. Piczak. ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1015–1018, 2015.

Annabelle Redelmeier, Martin Jullum, and Kjersti Aas. Explaining predictive models with mixed features using Shapley values and conditional inference trees. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE)*, pp. 117–137, 2020.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2017.

Lloyd S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28): 307–317, 1953.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186, 2020.

Nils Strodthoff, Patrick Wagner, Tobias Schaeffter, and Wojciech Samek. Deep learning for ECG analysis: Benchmarks and insights from PTB-XL. *IEEE Journal of Biomedical and Health Informatics*, 25(5):1519–1528, 2021.

Mukund Sundararajan and Amir Najmi. The many Shapley values for model explanation. In *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9269–9278, 2020.

Muhammad Faaiz Taufiq, Patrick Blöbaum, and Lenon Minorics. Manifold restricted interventional Shapley values. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 5079–5106, 2023.

Felix Tempel, Daniel Groos, Espen Alexander F. Ihlen, Lars Adde, and Inga Strümke. Choose your explanation: A comparison of SHAP and GradCAM in human activity recognition. *arXiv preprint arXiv:2412.16003*, 2025a.

Felix Tempel, Espen Alexander F Ihlen, Lars Adde, and Inga Strümke. Explaining human activity recognition with shap: validating insights with perturbation and quantitative measures. *Computers in Biology and Medicine*, 188:109838, 2025b.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

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

Mattia Villani, Joshua Lockhart, and Daniele Magazzeni. Feature importance for time series data: Improving KernelSHAP. In *Proceedings of the ACM International Conference on AI in Finance (ICAIF) Workshop on Explainable AI in Finance*, 2022.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I. Lunze, Wojciech Samek, and Tobias Schaeffter. PTB-XL, a large publicly available electrocardiography dataset. *Scientific Data*, 7(1):154, 2020.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585, 2017.

Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701*, 2019.

Cenwei Zhang, Lin Zhu, Manxi Lin, and Lei You. From baselines to transport geodesics: Axiomatic attribution via optimal generative flows. *arXiv preprint arXiv:2603.05093*, 2026.

Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. MotionBERT: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15085–15099, 2023.

Table 6: **Notation.** The four benchmark axes and their symbols.

Symbol	Meaning
Data	
$\boldsymbol{x} \in \mathbb{R}^{J\times F\times T}$	instance: J spatial units, F coords, T frames
f	trained classifier being explained
Players	
$\mathcal{P}$, M	player set (a partition of $\boldsymbol{x}$), number of players
$S \subseteq \mathcal{P}$	coalition of visible players
Imputation and game	
$\tilde{\boldsymbol{x}}_S$	$\boldsymbol{x}$ with players outside S imputed
$v(S) = f(\tilde{\boldsymbol{x}}_S)$	value of coalition S
ϕ_i, $\hat{\phi}$	Shapley value of player i; estimated attribution
Ground truths and metrics	
$\phi^{\star-\mathrm{mar}}$, $\phi^{\star-\mathrm{cond}}$	exact Shapley values of the marginal / conditional game
EC1, EC3	error in credit sizes / credit rankings (synthetic)
Faith., PlayerAOPC	deletion-based scores (real data, no oracle)

A Coalition Sampling Protocol

Every attribution in this paper (every method row and every grading target) is produced by one estimator run on one shared coalition design. This appendix specifies both. The tables can then be read with the guarantee the main text relies on: no difference between two methods at the same player set can come from the estimation layer.

The estimator. Given a design $\mathcal{Z}$ of coalitions, always containing the empty and full coalitions, every method solves the KernelSHAP weighted least-squares problem

$$\left(\hat{\phi}_0, \hat{\phi}\right) \;=\; \arg\min_{\phi_0,\,\phi} \sum_{S\in\mathcal{Z}} \pi_M(S)\left[v(S) - \phi_0 - \textstyle\sum_{i\in S}\phi_i\right]^2, \qquad \pi_M(S) \;=\; \frac{M-1}{\binom{M}{|S|}\,|S|\,\left(M-|S|\right)}, \tag{4}$$

where π_M is the Shapley kernel, M the player count, and ϕ_0 the intercept (the value of the empty coalition). The empty and full coalitions enter as constraints, implemented as boundary rows with pinning weight 10^6, so the credits plus intercept sum exactly to the full-coalition value (the efficiency identity). The normal matrix carries a ridge of 10^{-8} for numerical stability.

Every value $v(S) = f(\tilde{\boldsymbol{x}}_S)$ uses the same completion semantics: the hidden entries of $\tilde{\boldsymbol{x}}_S$ are the *average* of the imputer's draws, and the classifier is evaluated once on that average (f-of-mean semantics, Appendix B). The zero and mean imputations are deterministic. Every stochastic imputer averages the same $R{=}5$ draws per coalition: the imputers that copy from other sequences (KS-Marginal, KS-Empirical), the conditional models (KS-Gauss, KS-VAEAC, KS-Flow), and the KS-Oracle reference, which draws from the exact conditional law. On the real datasets every imputer uses a single completion. KS-Marginal and KS-Empirical copy from the $N{=}200$ evaluation pool, and KS-Empirical's kernel weighting excludes the explained sequence from its own pool of source sequences.

Exact designs ($M \le 12$). Whenever $M \le 12$, the design enumerates all 2^M coalitions and equation 4 recovers the Shapley values of the (finite-draw) game exactly. This covers every temporal track ($K \in \{4, 5\}$ windows), the spatial tracks of the synthetic datasets with $J{=}5$ joints, the PTB-XL per-lead track ($M{=}12$, 4096 coalitions), and the ESC-50 frequency-band track ($M{=}4$).

Sampled designs ($M > 12$). Above 12 players the design is built in two stages. First, complete size pairs $(s, M{-}s)$ are enumerated in kernel-mass order, for as long as the whole pair fits inside the budget B. The total kernel mass of size s is proportional to $(M{-}1)/[s(M{-}s)]$, so the extreme sizes come first. Every enumerated row carries its exact kernel weight $\pi_M(S)$.

Second, the remaining budget is filled by sampling coalitions without replacement from the leftover sizes, with size probability proportional to kernel mass. Each sampled row carries an importance-corrected uniform weight (the total leftover kernel mass divided by the number of

sampled rows). The sampled block is thus a stand-in for the sizes it represents. All weights are computed in log space (no binomial-coefficient overflow at M=68), rescaled so the largest non-boundary weight is 1, with boundary rows pinned at 10^6.

Shared designs and budgets. In the sampled regime, the design for each (dataset, player set, budget) is drawn once, from seed 7919, and reused verbatim by every method. Methods differ only in the imputer that defines $v(S)$, never in which coalitions they see. (Exact designs need no seed: every method sees the full enumeration.)

The two grading targets are built by the same design generator at their own, higher budget. A method's distance to its target therefore includes the pure design-budget gap, which Appendix D measures directly. Method budgets are B=1024 on the sampled synthetic tracks at $M \in \{17, 20, 25\}$ and B=2048 at M=68 and on every sampled real-data track ($M \in \{16, 17, 48, 68\}$). Grading targets are computed from the same design generator at high budget (B=8192, raised to B=32,768 at M=68), and exactly (full enumeration) wherever $M \leq 12$. The residual noise of the sampled targets is measured directly in Appendix D. Every cell evaluates N=200 sequences per dataset and per fold on the real data (CARE-PD's middle fold holds 180), three folds pooled.

Metrics on sampled designs. On the real data, faithfulness is the Pearson correlation over all B+2 design rows, including the two boundary rows, mirroring the enumerated protocol. The PlayerAOPC deletion-path coalitions (players removed in decreasing-credit order) are generally not design rows and are evaluated explicitly. WindowSHAP and TimeSHAP construct their temporal players internally (§2.3) and are graded on the same targets as every other temporal-player method.

B Oracle Ground Truths: Statements and Proofs

The synthetic suite is designed so that both estimands of §2.4 (the marginal imputation $\mathbb{E}[x_i]$ and the conditional imputation $\mathbb{E}[x_i \mid \boldsymbol{x}_{\text{vis}}]$) are available in closed form. Every dataset shares one generative structure: a latent Gaussian field $\boldsymbol{z}$ with Kronecker covariance $\text{Cov}(\text{vec}\,\boldsymbol{z}) = \Sigma_J \otimes I_F \otimes \Sigma_T$, mapped to observations either directly (Gaussian marginals) or through a heavy-tailed copula. Because the designer chooses Σ_J, Σ_T, and the marginal family, both expectations have closed forms. This makes both ground truths exactly computable for any trained classifier. This appendix states and proves that claim, then connects it to the grading-target construction used throughout the paper.

Proposition 1 (Exact ground truths on the synthetic families)**.** *Let $\boldsymbol{z}$ be the latent Kronecker Gaussian field of a synthetic dataset, $\text{Cov}(\text{vec}\,\boldsymbol{z}) = \Sigma_J \otimes I_F \otimes \Sigma_T$, and let the observation $\boldsymbol{x}$ be either $\boldsymbol{z}$ itself (Gaussian marginals) or its coordinatewise copula transform $x = \text{sign}(z)\, Q_{\text{Burr}}(2\Phi(|z|)-1;\, 2, 2)$ (Burr marginals). Fix any trained classifier f and any player set $\mathcal{P}$ with M players. Then:*

- *(i) the exact marginal imputation is $\mathbb{E}[x_{jft}] = 0$ for every coordinate;*
- *(ii) for every coalition S, the exact conditional imputation $\mathbb{E}[\boldsymbol{x}_{\text{hid}} \mid \boldsymbol{x}_{\text{obs}}]$ is available in closed form: a linear map of the observed coordinates in the Gaussian case, and a one-dimensional integral per hidden coordinate in the Burr case, computable to any prescribed accuracy;*
- *(iii) consequently the two games $v^{\star-\text{std}}(S) = f(\boldsymbol{x}$ with hidden entries set to $0)$ and $v^{\star-\text{man}}(S) = f(\boldsymbol{x}$ with hidden entries set to $\mathbb{E}[\boldsymbol{x}_{\text{hid}} \mid \boldsymbol{x}_{\text{obs}}])$ are deterministic and exactly evaluable, and for $M \leq 12$ the ground truths $\phi^{\star-\text{mar}}$ and $\phi^{\star-\text{cond}}$ follow from exact enumeration of equation 2.*

Proof. (i) The latent field is centered, so in the Gaussian case $\mathbb{E}[x_{jft}] = 0$ directly. The symmetric Burr transform is an odd function of z applied to a symmetric variable, so each transformed marginal is symmetric about zero and its mean is again exactly zero. The exact marginal imputation is therefore the zero imputation on every family.

(ii) Since $\Sigma_F = I_F$, the F coordinate slices are independent and each carries the covariance $\Sigma = \Sigma_J \otimes \Sigma_T$ over the (j, t) grid. For a coalition S, let o and h index the observed and hidden (j, t)

positions. By standard multivariate Gaussian conditioning,

$$\boldsymbol{z}_h \mid \boldsymbol{z}_o \;\sim\; \mathcal{N}\big(\Sigma_{ho}\Sigma_{oo}^{-1}\boldsymbol{z}_o,\; \Sigma_{hh} - \Sigma_{ho}\Sigma_{oo}^{-1}\Sigma_{oh}\big),$$

where the Kronecker structure enters only through the efficient construction of the sub-blocks. In the Gaussian case $\boldsymbol{x} = \boldsymbol{z}$ and the conditional mean $W_S\,\boldsymbol{x}_{\text{obs}}$ with $W_S = \Sigma_{ho}\Sigma_{oo}^{-1}$ is the exact imputation. W_S is computed once per coalition and applied to all N sequences. In the Burr case the copula transform is a coordinatewise strictly monotone bijection with analytic inverse, so the observed coordinates map exactly to latent coordinates and conditioning happens in the latent field as above. Each hidden observation coordinate is then the image $g(z) = \operatorname{sign}(z)\, Q_{\text{Burr}}(2\Phi(|z|)-1)$ of a univariate normal $z \sim \mathcal{N}(\mu, \sigma^2)$, with μ, σ from the display above. Its conditional mean is the one-dimensional integral $\mathbb{E}[x \mid \boldsymbol{x}_{\text{obs}}] = \int g(\mu + \sigma t)\, \varphi(t)\, dt$, evaluated by the quadrature below.

(iii) With both imputations deterministic, each game value is one classifier evaluation, and equation 2 is a finite sum over all 2^M coalitions, feasible for $M \leq 12$. □

The Burr quadrature. The integrand $g(\mu+\sigma t)$ has a square-root cusp where its argument crosses the copula median $u = 1/2$, i.e. at $t_0 = -\mu/\sigma$. This caps any global quadrature rule at an algebraic convergence rate. We therefore split the integral at t_0 and substitute $t = t_0 \pm y^2$ on each side. This removes the cusp and restores the spectral accuracy of Gauss–Legendre quadrature. The domain is truncated at $|t| = 13$, where the excluded normal mass is negligible, and each side uses 64 nodes (128 integrand evaluations per hidden coordinate). The computed imputations are stable under node doubling and agree with large-sample Monte-Carlo conditional means. The released construction additionally passes a unit-test battery (unconditional covariance recovery, closed-form conditional moments, the Shapley efficiency identity, the dummy-player axiom under independence, Monte-Carlo agreement, and the Burr copula round-trip).

From exact imputations to grading targets. The value semantics of the benchmark is f-of-mean: a method's $v(S)$ evaluates the classifier at the *average* of its completions (Appendix A). As the number of draws grows, that average converges to the exact conditional (or marginal) mean. The deterministic games of Proposition 1 are therefore the infinite-draw limits of the games the methods actually play. The grading targets simply plug the limit in directly, pushing the exact imputations through the trained classifier, so the targets carry no Monte-Carlo noise of their own. For $M \leq 12$ the targets are exact Shapley vectors. Above that threshold, the same deterministic value functions are evaluated on the shared high-budget designs and solved by equation 4, with the resulting grading error measured in Appendix D.

The KS-Oracle *method* row is distinct from the target: it samples $R{=}5$ completions from the exact conditional law under the same budget as every other stochastic imputer. Its residual against the deterministic target is therefore finite-draw completion noise plus estimation, and the main tables use it as a reference. The reference is not a floor. The targets are the infinite-draw means, so a conditional model whose mean is close to the true one and whose draws are less dispersed carries less completion noise than exact conditional draws, and can score below the reference row.

C EC1 Tables and the Cross-Graded Convention

Division of labor between the metrics. EC3 is a per-sequence correlation, hence scale-free and comparable across datasets and player sets. The main text reports it exclusively. EC1 measures the absolute size error of the credits and inherits each dataset's and game's attribution scale, so it is comparable within a column but not across columns. Tables 7, 8, and 9 are the EC1 companions to main-text Tables 3–4, under the same game-matched grading and the same classifier accuracy criterion.

Two rows need a reading note. Because every synthetic family is centered, the exact marginal imputation *is* the zero imputation (Proposition 1). On exact-enumeration tracks, KS-Zero's value function therefore coincides with the standard target's, and its EC1 is zero to numerical precision. KS-Mean's empirical-mean imputation nearly coincides with it. These rows restate the grading target rather than estimate it, which is why the main tables leave them out of the synthetic ranking.

On sampled tracks the same rows are no longer exactly zero (e.g. up to 0.012 at cells, Table 9). The residual is the pure design gap between the method budget and the high-budget target, a useful

Table 7: **EC1, temporal players** (companion to Table 3): mean absolute attribution error against the exact game-matched ground truth, mean ± spread across the classifier architectures that pass the accuracy criterion (Appendix F). EC1 carries the attribution scale of each dataset, so values compare within a column, not across columns. Gray as in the main table.

Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers, graded against the marginal ground truth $\phi^{\star-\mathrm{mar}}$					
KS-Zero	**0.000** ±0.000	**0.000** ±0.000	**0.000** ±0.000	**0.000** ±0.000	**0.000** ±0.000
KS-Mean	0.016 ±0.008	0.017 ±0.011	0.009 ±0.004	0.026 ±0.025	0.019 ±0.001
KS-Marginal	0.115 ±0.052	0.105 ±0.064	0.096 ±0.044	0.085 ±0.081	0.102 ±0.049
WindowSHAP-S	0.074 ±0.023	0.073 ±0.037	0.073 ±0.025	0.085 ±0.079	0.161 ±0.010
WindowSHAP-D	0.099 ±0.034	0.082 ±0.045	0.091 ±0.031	0.111 ±0.104	0.190 ±0.028
TimeSHAP	0.113 ±0.055	0.093 ±0.057	0.106 ±0.063	0.168 ±0.164	0.219 ±0.109
KS-Empirical	0.135 ±0.052	0.116 ±0.066	0.119 ±0.048	0.100 ±0.095	0.124 ±0.054
Generative imputers, graded against the conditional ground truth $\phi^{\star-\mathrm{cond}}$					
KS-VAEAC	0.060 ±0.028	0.074 ±0.036	0.048 ±0.021	0.102 ±0.097	0.157 ±0.053
KS-Flow	0.128 ±0.059	0.104 ±0.062	0.098 ±0.044	0.101 ±0.096	0.130 ±0.061
KS-Gauss (closed form)	0.092 ±0.041	0.089 ±0.051	0.085 ±0.038	0.093 ±0.088	0.100 ±0.057
KS-Oracle (reference)	0.087 ±0.039	0.080 ±0.045	0.076 ±0.035	0.049 ±0.047	0.056 ±0.028

Table 8: **EC1, spatial players** (companion to Table 4), mean ± spread across the classifier architectures that pass the accuracy criterion (Appendix F), game-matched grading.

Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers, graded against the marginal ground truth $\phi^{\star-\mathrm{mar}}$					
KS-Zero	**0.000** ±0.000	**0.001** ±0.001	**0.000** ±0.000	**0.002** ±0.001	**0.002** ±0.001
KS-Mean	0.003 ±0.002	0.002 ±0.001	0.002 ±0.001	0.006 ±0.006	0.004 ±0.001
KS-Marginal	0.093 ±0.042	0.030 ±0.018	0.099 ±0.045	0.024 ±0.023	0.027 ±0.017
KS-Empirical	0.127 ±0.053	0.042 ±0.023	0.132 ±0.058	0.034 ±0.032	0.042 ±0.024
Generative imputers, graded against the conditional ground truth $\phi^{\star-\mathrm{cond}}$					
KS-VAEAC	0.077 ±0.037	0.036 ±0.019	0.064 ±0.030	0.027 ±0.026	0.044 ±0.021
KS-Flow	0.116 ±0.053	0.046 ±0.025	0.121 ±0.057	0.024 ±0.023	0.038 ±0.020
KS-Gauss (closed form)	0.088 ±0.044	0.040 ±0.022	0.095 ±0.046	0.031 ±0.029	0.035 ±0.020
KS-Oracle (reference)	0.081 ±0.041	0.030 ±0.018	0.082 ±0.041	0.023 ±0.022	0.025 ±0.015

lower bound on what any method can show there. Conversely, the KS-Oracle row's nonzero EC1 (0.076–0.087 on the non-periodic temporal columns) is its own R=5 completion noise against the deterministic target. The EC1 ordering otherwise mirrors the EC3 story.

Periodic data and ground-truth credit variation. We can measure the periodic near-null of Result 1 on the ground truth itself. The per-sequence spread of the conditional ground-truth credits drops to 0.05–0.06 on the two periodic datasets, a third of its level on the other three (0.12–0.15), while the marginal ground truth keeps its full spread (0.14–0.23). With this little true signal to recover, the residual error of any imperfect imputer dominates EC3: the exact conditional imputer still reaches 0.22, while every imperfect conditional imputer sits at 0.61 or above. Under spatial players the KS-Oracle reference itself drops from 0.22 to 0.08 on `skel+gait`: hiding a joint across all frames restores the credit variation that temporal windows suppress.

KS-Empirical never improves on KS-Marginal. Across all 15 dataset × player-set combinations, KS-Empirical ranks behind the simpler KS-Marginal, in the main-text EC3 tables and in the EC1 tables above. KS-Empirical approximates a conditional imputation by copying values from the most similar other sequences in the evaluation pool, weighted by a kernel (§2.3). The copied values still come from other sequences, so they carry the same off-manifold error as marginal sampling, and the similarity weighting adds estimation variance on top.

Table 9: **EC1, spatiotemporal players** (companion to Table 4), mean ± spread across the classifier architectures that pass the accuracy criterion (Appendix F), game-matched grading. Gray as in the main tables.

Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers, graded against the marginal ground truth $\phi^{\star-\mathrm{mar}}$					
KS-Zero	**0.008** ±0.005	**0.005** ±0.004	**0.012** ±0.007	**0.006** ±0.006	**0.007** ±0.006
KS-Mean	0.010 ±0.006	0.006 ±0.004	0.013 ±0.008	0.008 ±0.008	0.008 ±0.005
KS-Marginal	0.028 ±0.014	0.016 ±0.010	0.033 ±0.016	0.013 ±0.012	0.015 ±0.009
KS-Empirical	0.045 ±0.017	0.021 ±0.012	0.046 ±0.019	0.017 ±0.016	0.023 ±0.010
Generative imputers, graded against the conditional ground truth $\phi^{\star-\mathrm{cond}}$					
KS-VAEAC	0.032 ±0.015	0.019 ±0.010	0.034 ±0.015	0.016 ±0.015	0.026 ±0.008
KS-Flow	0.039 ±0.017	0.021 ±0.012	0.041 ±0.018	0.015 ±0.014	0.022 ±0.009
KS-Gauss (closed form)	0.029 ±0.012	0.019 ±0.010	0.036 ±0.015	0.015 ±0.014	0.019 ±0.009
KS-Oracle (reference)	0.026 ±0.012	0.017 ±0.009	0.032 ±0.013	0.012 ±0.011	0.014 ±0.008

Table 10: **Cross-graded EC3, temporal players**: every method graded against the on-manifold (conditional) ground truth, the convention of prior oracle benchmarks. Comparing the standard-imputer block with its game-matched values in Table 3 isolates the ground-truth divergence $\Delta(\phi^{\star-\mathrm{mar}}, \phi^{\star-\mathrm{cond}})$ from estimator error. Gray columns: on periodic data the conditional ground truth is nearly uniform across windows, so no per-column best is marked.

Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers, graded against the conditional ground truth					
KS-Zero	0.25 ±0.03	0.28 ±0.09	0.23 ±0.04	0.87 ±0.11	0.87 ±0.13
KS-Mean	0.26 ±0.02	0.29 ±0.09	0.23 ±0.03	0.86 ±0.10	0.86 ±0.12
KS-Marginal	0.45 ±0.06	0.52 ±0.09	0.38 ±0.03	0.86 ±0.08	0.84 ±0.10
WindowSHAP-S	0.46 ±0.04	0.65 ±0.11	0.43 ±0.04	0.90 ±0.03	0.91 ±0.03
WindowSHAP-D	0.68 ±0.06	0.77 ±0.15	0.61 ±0.09	0.94 ±0.06	0.98 ±0.02
TimeSHAP	0.47 ±0.06	0.48 ±0.08	0.48 ±0.12	0.91 ±0.05	0.93 ±0.03
KS-Empirical	0.53 ±0.03	0.56 ±0.09	0.51 ±0.05	0.84 ±0.05	0.87 ±0.09
Generative imputers, graded against the conditional ground truth (as in the main tables)					
KS-VAEAC	**0.12** ±0.03	**0.16** ±0.05	**0.10** ±0.02	0.76 ±0.08	0.87 ±0.04
KS-Flow	0.41 ±0.07	0.38 ±0.09	0.29 ±0.03	0.78 ±0.10	0.77 ±0.12
KS-Gauss (closed form)	0.24 ±0.04	0.28 ±0.08	0.23 ±0.03	0.69 ±0.09	0.61 ±0.15
KS-Oracle (reference)	0.21 ±0.05	0.20 ±0.06	0.19 ±0.04	0.22 ±0.06	0.22 ±0.03

What cross-grading isolates. Under *game-matched* grading, each method is scored against the ground truth of its own imputer family, so its error is estimation error within its own game. Under the *cross-graded* convention of prior oracle benchmarks, every method is scored against the on-manifold ground truth $\phi^{\star-\mathrm{cond}}$. A standard imputer's cross-graded error is then its estimation error *plus* the divergence $\Delta(\phi^{\star-\mathrm{mar}}, \phi^{\star-\mathrm{cond}})$ between the two ground truths. The two conventions bracket the divergence: for a method whose game-matched error is near zero (KS-Zero, above), the cross-graded number is essentially the divergence itself. Tables 10–12 report the cross-graded EC3 at the three player sets. The generative-imputer rows are unchanged from the main tables, since their grading target is already $\phi^{\star-\mathrm{cond}}$.

At temporal players, KS-VAEAC is the best cross-graded method on all three non-periodic datasets (0.12/0.16/0.10). KS-Zero's definitionally-exact standard answers rank second (0.25/0.28/0.23): within 0.04–0.08 of the exact conditional imputer itself (0.21/0.20/0.19) and ahead of every remaining conditional estimator. A perfect answer to the *standard* question outranks every genuine estimate of the *conditional* one but KS-VAEAC's. That is the quantitative content of main-text Result 4: cross-graded error is dominated by ground-truth divergence, not estimation quality.

The divergence grows with granularity. At the finest player set the same comparison blows up (Table 12). On `skel+gait` cells (M=68), the standard imputers score 0.66–0.72 cross-graded. That compares with game-matched errors of 0.14–0.23 for the estimating standard methods (Table 4b) and near-zero EC1 for the definitional ones (Table 9). On `gait` cells the pattern repeats

Table 11: **Cross-graded EC3, spatial players**: all methods against the on-manifold ground truth (exact conditional target for $M \leq 12$, hi-budget sampled target above).

Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers, graded against the conditional ground truth					
KS-Zero	0.35 ±0.19	0.31 ±0.16	0.35 ±0.19	**0.00** ±0.00	0.24 ±0.19
KS-Mean	0.35 ±0.19	0.31 ±0.16	0.36 ±0.19	0.05 ±0.05	0.24 ±0.18
KS-Marginal	0.37 ±0.05	0.27 ±0.09	0.39 ±0.06	0.17 ±0.15	0.22 ±0.13
KS-Empirical	0.41 ±0.02	0.29 ±0.04	0.43 ±0.04	0.26 ±0.13	0.28 ±0.08
Generative imputers, graded against the conditional ground truth (as in the main tables)					
KS-VAEAC	**0.19** ±0.06	**0.13** ±0.02	**0.16** ±0.06	0.19 ±0.15	0.22 ±0.05
KS-Flow	0.29 ±0.06	0.23 ±0.06	0.35 ±0.07	0.17 ±0.14	0.19 ±0.07
KS-Gauss (closed form)	0.21 ±0.07	0.17 ±0.04	0.25 ±0.08	0.22 ±0.15	**0.14** ±0.07
KS-Oracle (reference)	0.18 ±0.07	0.10 ±0.04	0.19 ±0.07	0.16 ±0.14	0.08 ±0.04

Table 12: **Cross-graded EC3, spatiotemporal players**: all methods against the on-manifold ground truth. At M=68 (skel+gait cells) the estimating standard imputers' cross-graded error is three to five times their game-matched error (Table 4): at fine granularity the two ground truths assign very different credit across cells.

Method	gauss	skeleton	burr	gait	skel+gait
Standard imputers, graded against the conditional ground truth					
KS-Zero	0.31 ±0.10	0.31 ±0.10	0.33 ±0.08	0.55 ±0.05	0.72 ±0.03
KS-Mean	0.31 ±0.10	0.31 ±0.10	0.33 ±0.07	0.55 ±0.04	0.72 ±0.03
KS-Marginal	0.34 ±0.06	0.35 ±0.05	0.39 ±0.04	0.59 ±0.06	0.71 ±0.04
KS-Empirical	0.39 ±0.01	0.38 ±0.02	0.45 ±0.02	0.58 ±0.07	0.66 ±0.02
Generative imputers, graded against the conditional ground truth (as in the main tables)					
KS-VAEAC	0.20 ±0.05	**0.22** ±0.02	**0.24** ±0.03	0.48 ±0.09	0.57 ±0.01
KS-Flow	0.26 ±0.05	0.29 ±0.05	0.32 ±0.03	0.40 ±0.11	0.53 ±0.01
KS-Gauss (closed form)	**0.14** ±0.03	0.22 ±0.03	0.24 ±0.03	**0.38** ±0.13	**0.36** ±0.07
KS-Oracle (reference)	0.13 ±0.03	0.18 ±0.02	0.20 ±0.02	0.21 ±0.08	0.22 ±0.05

(0.55–0.59 cross-graded vs. 0.16–0.26 game-matched). At fine granularity the two ground truths rank cells very differently, so the choice of game is most consequential exactly where explanations are most detailed.

Per-classifier breakdown. Table 13 disaggregates the temporal game-matched EC3 by classifier architecture. The ± spread in every aggregated table is the spread across these rows. Errors are generally lowest under the MLP and highest under the Transformer (e.g. KS-Gauss 0.20/0.23/0.29 across MLP/CNN/Transformer on gauss). The MLP's input features are exactly the K window means, which aligns the classifier's own geometry with the temporal players. The periodic near-null affects every admitted architecture individually (KS-VAEAC 0.84 and 0.69 on gait MLP/Transformer), so the gray cells of the main tables are not an artifact of averaging. The two † rows are the excluded CNNs (Appendix F), shown for completeness. Per-classifier values for the spatial and cell tracks ship with the canonical result files.

D Grading Error Floors

Above M=12 the grading targets are themselves solved on sampled designs (Appendix A), so the grading itself carries an approximation error whose size must be measured, not assumed. We measure it directly. For each sampled target we recompute an *independent replicate* target at the same budget from a disjoint coalition seed. We then report the error metric between the replicate and the stored target, averaged over the classifier architectures.

The floor is the resolution limit of the benchmark at that cell. A difference between two methods (or between a method and the target) smaller than the floor is not readable, and we flag such

Table 13: **Per-classifier EC3, temporal players** (game-matched). The two † rows fail the accuracy criterion (Appendix F) and are excluded from every aggregated table; they are shown here for completeness. Per-classifier values for the spatial and cell player sets ship with the canonical result files in the code release.

Dataset	Classifier	KS-Zero	KS-Mean	KS-Marginal	WindowSHAP-S	WindowSHAP-D	TimeSHAP	KS-Empirical	KS-VAEAC	KS-Flow	KS-Gauss	KS-Oracle
gauss	mlp	0.00	0.02	0.30	0.37	0.72	0.23	0.47	0.07	0.31	0.20	0.15
	cnn	0.00	0.01	0.36	0.23	0.56	0.36	0.51	0.14	0.45	0.23	0.21
	transformer	0.00	0.02	0.41	0.27	0.50	0.47	0.55	0.14	0.47	0.29	0.27
skeleton	mlp	0.00	0.02	0.29	0.68	0.98	0.26	0.35	0.10	0.26	0.19	0.13
	cnn	0.00	0.05	0.44	0.39	0.55	0.37	0.51	0.15	0.38	0.28	0.20
	transformer	0.00	0.03	0.50	0.43	0.60	0.46	0.56	0.23	0.49	0.38	0.28
burr	mlp	0.00	0.00	0.22	0.32	0.71	0.18	0.44	0.08	0.26	0.19	0.15
	cnn	0.00	0.01	0.31	0.28	0.46	0.38	0.49	0.09	0.28	0.23	0.19
	transformer	0.00	0.03	0.31	0.30	0.45	0.55	0.51	0.13	0.32	0.28	0.24
gait	mlp	0.00	0.00	0.14	0.48	1.00	0.16	0.26	0.84	0.69	0.60	0.16
	cnn†	0.00	0.01	0.20	0.93	0.85	0.70	0.32	0.53	0.56	0.48	0.21
	transformer	0.00	0.09	0.29	0.40	0.71	0.70	0.37	0.69	0.88	0.77	0.28
skel+gait	mlp	0.00	0.03	0.09	0.67	0.98	0.34	0.19	0.91	0.65	0.47	0.18
	cnn†	0.00	0.01	0.32	0.82	0.89	0.67	0.38	0.77	0.68	0.53	0.29
	transformer	0.00	0.02	0.23	0.40	0.62	0.67	0.37	0.83	0.89	0.76	0.25

Table 14: **Grading error floors** for the sampled high-budget targets: EC between the stored grading target and an independent replicate (same budget, disjoint coalition seed), averaged over the gate-passing classifier architectures. Method-vs-target differences below these levels are not interpretable. Exact-enumeration cells ($M \leq 12$) have zero grading error by construction.

Player set	Dataset	M	Budget	Floor EC3 (cond / marg)	Floor EC1 (cond / marg)
cells	gauss	20	8,192	0.016 / 0.004	0.0068 / 0.0030
cells	skeleton	68	8,192	0.118 / 0.020	0.0128 / 0.0052
cells	skeleton	68	32,768	0.026 / 0.003	0.0053 / 0.0019
cells	burr	25	8,192	0.020 / 0.013	0.0082 / 0.0048
cells	gait	68	8,192	0.075 / 0.053	0.0065 / 0.0060
cells	gait	68	32,768	0.014 / 0.022	0.0028 / 0.0023
cells	skel+gait	68	8,192	0.146 / 0.054	0.0114 / 0.0065
cells	skel+gait	68	32,768	0.032 / 0.027	0.0047 / 0.0026
spatial	skeleton	17	8,192	0.006 / 0.003	0.0064 / 0.0009
spatial	gait	17	8,192	0.001 / 0.001	0.0010 / 0.0010
spatial	skel+gait	17	8,192	0.005 / 0.001	0.0057 / 0.0013
Cross-budget agreement at M=68 (cond target, B=8,192 vs B=32,768): EC3 0.065					

comparisons rather than interpret them. Exact-enumeration cells ($M \leq 12$) have zero grading error by construction, so every temporal-player number in the paper is graded exactly.

Table 14 reports the floors. Two regularities matter for reading the synthetic tables. First, the floors grow with the number of players and are largest for the conditional game at the finest cells. This is why the M=68 targets are computed at the raised budget B=32,768 rather than B=8192 (main-text Result 3). Second, the floors are the reading rule for the finest-granularity tables: a gap between two rows that does not clear the floor of its track is a tie. More specifically, differences below the floor cannot be distinguished from the target's own sampling variability.

The cross-budget agreement row provides an independent check that the two target budgets at M=68 identify the same game. Method error and evaluation error are thereby kept separate throughout.

E Training and Experiment Details

Synthetic data generation. Each dataset is a deterministic function of a fixed seed and the parametric family of §4.1. Every dataset materialises N=200 evaluation sequences. Classifier training draws 2000 training and 500 validation sequences from the same distribution under disjoint seeds, so train and test never overlap. All five pillar datasets share one label rule, the *proxy* rule of §4.1: class labels are tertiles of the grand mean of the first spatial component, applied on each dataset's own draw. The rule is easy to state, uses no window structure, and is deliberately hard for classifiers to shortcut, which produces the wide competence range visible in Appendix F.

Synthetic classifiers. For each dataset we train three architectures. The *MLP* is a 2-layer network (hidden width 64, ReLU, BatchNorm on the input layer) consuming the K window-mean features. The *CNN* is a 3-layer 1-D convolutional network (filters $32\to32\to64$, kernel size 5, padding 2, ReLU, BatchNorm, global average pooling) on raw frames reshaped to $(J\cdot F, T)$. The *Transformer* is encoder-only (d_{model}=32, 4 heads, 2 layers, feed-forward 128, dropout 0.1, sinusoidal positional encoding, mean-pool readout) on raw frames reshaped to $(T, J\cdot F)$.

Training uses Adam with cross-entropy loss for 200 epochs with early stopping (patience 20, best-validation checkpoint kept). Batch size is 64 (32 for the Transformer). Learning rates are 5×10^{-3} (MLP), 10^{-3} (CNN), and 5×10^{-4} (Transformer), and weight decays 10^{-3} (MLP) and 10^{-4} (CNN, Transformer). Runs whose best validation accuracy falls below 0.50 are retried with the learning rate divided by five and the next seed. Whether a trained classifier's explanations enter the aggregated tables is decided by the uniform accuracy criterion of Appendix F.

CARE-PD classifiers. We explain the UPDRS-gait severity classifiers released with the CARE-PD benchmark (Adeli et al., 2025), one checkpoint per leave-one-subject-out fold of the BMCLab split. The classifiers are MotionBERT (Zhu et al., 2023) (DSTformer backbone, T=80, 3-D H36M coordinates with crop-scale normalization), MotionAGFormer (Mehraban et al., 2024) (attention–graph hybrid, clips zero-padded $80\to81$ frames), and POTR (Martínez-González et al., 2021) (GCN encoder with a 4-layer Transformer, d_{model}=128, root-centered z-scored coordinates). The last is excluded by the accuracy criterion (Appendix F). Coalition imputations are applied in raw world coordinates before each backbone's own input normalization. The explained output on all real datasets is the model's probability of its own predicted class on the full sequence: $v(S)$ is the softmax probability of that class after imputation.

PTB-XL classifier. One 1-D residual network (Wang et al., 2017; Strodthoff et al., 2021) is trained from scratch per fold: three residual stages of filter widths $64\to128\to128$ with kernel sizes $\{8, 5, 3\}$, batch normalization, global average pooling, and a linear head. The input is $(12, 1000)$ from 10 s recordings at 100 Hz. The class set is {Normal, Myocardial Infarction}. Training uses Adam (learning rate 10^{-3}, weight decay 10^{-4}), 30 epochs, batch size 64, and gradient clipping at norm 1. The learning-rate schedule is `ReduceLROnPlateau` (factor 0.5, patience 5, floor 10^{-6}). Training also uses a weighted random sampler for class imbalance. Folds use the public `strat_fold` column (1–10). Fold 2 matches the canonical benchmark split of Strodthoff et al. (2021), and folds 1 and 3 rotate the held-out partition:

Fold index	Train (`strat_fold`)	Val	Test
1	$\{1,\dots,7\}$	$\{9\}$	$\{8\}$
2	$\{1,\dots,8\}$	$\{9\}$	$\{10\}$
3	$\{1,\dots,7,9\}$	$\{8\}$	$\{10\}$

ESC-50 classifiers. One Audio Spectrogram Transformer (Gong et al., 2021) is fine-tuned per fold from the AudioSet-pretrained base `MIT/ast-finetuned-audioset-10-10-0.4593` with a fresh 50-class head. The fold-f model is trained only on fold f's ESC-50 training split (1600 clips) and evaluated on the held-out fold (400 clips). The best epoch is chosen on the held-out fold. This choice is inconsequential: all runs plateau before the final epoch, and across the three folds the selected epoch improves held-out accuracy by at most 0.005 over simply taking the final epoch.

Training uses AdamW (learning rate 3×10^{-5}, weight decay 0.01), 5 epochs, batch size 16, a 5% linear warmup followed by a cosine schedule to zero, gradient clipping at norm 1, and seed 42+fold. Inputs are log-mel features in the AST convention (128 mel bins $\times$ 1024 frames), identical to the attribution-time forward path. Per-fold held-out accuracies are in Table 15.

F The Classifier Accuracy Criterion

A benchmark of explanations is only as meaningful as the classifiers being explained. Every explained classifier must therefore pass one uniform gate before its results enter any aggregated table: **(G1)** held-out accuracy at least 0.05 above chance, and **(G2)** a train–test accuracy gap of at most 0.30. G1 removes classifiers that never learned the task. G2 removes classifiers whose

Table 15: **The classifier accuracy criterion.** A classifier's explanations enter the aggregated tables only if (G1) held-out accuracy $\geq$ chance $+ 0.05$ and (G2) train–test gap ≤ 0.30. Synthetic: accuracies on the training / validation / evaluation splits. Real: held-out accuracy pooled over the three data folds (per-fold values in parentheses); the training column reports accuracy on the pooled training material of the cross-validation folds. Take-away: one rule yields every exclusion. POTR fails G1, and the `gait` and `skel+gait` CNNs fail G2; every other classifier is admitted.

Dataset	Classifier	Train	Val.	Test	Gap	Admitted
Synthetic (chance $= 1/3$*; accuracy bar* 0.383*)*						
`gauss`	mlp	0.644	0.662	0.590	+0.054	✓
`gauss`	cnn	0.951	0.892	0.870	+0.081	✓
`gauss`	transformer	0.990	0.950	0.930	+0.059	✓
`skeleton`	mlp	0.533	0.542	0.430	+0.103	✓
`skeleton`	cnn	0.961	0.878	0.865	+0.096	✓
`skeleton`	transformer	0.996	0.962	0.945	+0.051	✓
`burr`	mlp	0.643	0.644	0.590	+0.053	✓
`burr`	cnn	0.953	0.876	0.815	+0.138	✓
`burr`	transformer	0.994	0.960	0.945	+0.049	✓
`gait`	mlp	0.412	0.420	0.395	+0.017	✓
`gait`	cnn	0.940	0.400	0.365	+0.575	✗ (G2)
`gait`	transformer	0.993	0.846	0.845	+0.148	✓
`skel+gait`	mlp	0.512	0.544	0.515	-0.003	✓
`skel+gait`	cnn	1.000	0.520	0.500	+0.500	✗ (G2)
`skel+gait`	transformer	0.969	0.898	0.825	+0.143	✓
Real (chance: CARE-PD $1/3$*, PTB-XL* $1/2$*, ESC-50* $1/50$*)*						
CARE-PD	MotionBERT	0.974	—	0.698 (0.24/1.00/0.97)	+0.276	✓
CARE-PD	MotionAGFormer	0.983	—	0.726 (0.31/1.00/0.97)	+0.258	✓
CARE-PD	POTR	—	—	0.374	—	✗ (G1)
PTB-XL	ResNet1d	0.968	—	0.952 (0.95/0.95/0.94)	+0.016	✓
ESC-50	AST (3 fold models)	0.999	—	0.965 (0.95/0.98/0.96)	+0.034	✓

training-set fit does not transfer, for which any explanation would describe memorisation rather than the data-generating structure. Table 15 reports the full accuracy accounting for all fifteen synthetic classifiers and the three real-data classifier families.

One rule yields every exclusion in the paper. POTR on CARE-PD fails G1 (pooled held-out accuracy 0.374 against the three-class bar of 0.383). The CNNs on the two periodic synthetic datasets fail G2 (`gait` $+0.575$, `skel+gait` $+0.500$). Their near-perfect training accuracy collapses to near-chance held out, the signature of memorising periodic templates. Every other classifier passes, including both CARE-PD backbones (gaps $+0.276$ and $+0.258$), the PTB-XL ResNets ($+0.016$), and the per-fold ESC-50 ASTs ($+0.034$).

The gate is an aggregation rule for the synthetic suite, not a data-deletion rule. The two excluded CNNs' per-cell results are still computed and still ship in the canonical result files (and appear grayed in Table 13). They are only barred from the aggregated tables and from ranking claims. POTR fails the accuracy condition, so no attributions are computed for it (explaining a classifier that has not learned the task grades noise) and only its gate record ships. Synthetic aggregates on `gait` and `skel+gait` therefore average the MLP and Transformer. All other synthetic aggregates average three architectures. CARE-PD results are reported for the two passing backbones as models A and B.

G Real-Data Protocol and Additional Tables

Table 16 collects the full real-data protocol. Each dataset contributes three tracks (temporal windows, spatial units, and their intersection cells) under the shared estimation layer of Appendix A: exact enumeration wherever $M \leq 12$, and shared sampled designs at $B{=}2048$ above. Each track evaluates $N{=}200$ sequences per fold ($200/180/200$ on CARE-PD's subject-wise folds), three folds pooled, with the $\pm$ spread across folds. All classifiers are per-fold models that pass the gate of Appendix F. POTR is excluded there and appears in no result table.

Table 16: **Real-data protocol.** Tasks, classifiers, and the three player-set tracks per dataset. All tracks: 3 folds, 200 sequences per fold (200/180/200 on CARE-PD), folds pooled. Sampled designs share the coalition protocol of Appendix A (seed 7919). CARE-PD folds are leave-one-subject-out. ESC-50 uses folds 1–3 of its 5 stratified folds.

Dataset	Task	Classifiers (one per fold)	Track	M	Coalitions
CARE-PD	UPDRS-gait severity (3-class) 17 H36M joints, T=80	MotionBERT (A), MotionAGFormer (B); POTR excluded (Appendix F)	windows (K=4) joints joint×window cells	4 17 68	exact (2^4) sampled, B=2048 sampled, B=2048
PTB-XL	Normal vs. MI (binary) 12 leads, T=1000 (100 Hz)	ResNet1d	windows (K=4) leads lead×window cells	4 12 48	exact (2^4) exact (2^{12}) sampled, B=2048
ESC-50	environmental audio (50-class) 128 mel bins, T=1024	AST fine-tune	windows (K=4) freq. bands (mel quartiles) band×window cells	4 4 16	exact (2^4) exact (2^4) sampled, B=2048

Table 17: **Real-world results, spatiotemporal players** (joint×window / lead×window / band×window cells). Faithfulness (↑) and PlayerAOPC (↑), pooled over the three data folds, ± spread across folds; bold: best per column and metric. Protocol identical to Table 5 (coalition budget 2048, N=200 per fold).

	Faithfulness				*PlayerAOPC*			
Method	CARE-PD A M=68	CARE-PD B M=68	PTB-XL M=48	ESC-50 M=16	CARE-PD A M=68	CARE-PD B M=68	PTB-XL M=48	ESC-50 M=16
KS-Zero	**0.89** ±0.03	0.66 ±0.23	0.74 ±0.03	0.77 ±0.01	-0.11 ±0.25	**0.39** ±0.13	**0.39** ±0.04	0.72 ±0.00
KS-Mean	0.82 ±0.05	**0.68** ±0.15	0.74 ±0.03	0.78 ±0.01	**0.19** ±0.04	0.33 ±0.09	0.39 ±0.04	0.72 ±0.00
KS-Marginal	0.79 ±0.03	0.50 ±0.03	0.57 ±0.02	0.87 ±0.01	0.08 ±0.08	0.19 ±0.01	0.32 ±0.02	**0.85** ±0.01
KS-VAEAC	0.39 ±0.24	0.54 ±0.10	**0.78** ±0.03	**0.88** ±0.01	0.02 ±0.04	0.08 ±0.01	0.30 ±0.05	0.80 ±0.01
KS-Flow	0.42 ±0.16	0.52 ±0.07	0.61 ±0.05	0.86 ±0.01	0.05 ±0.03	0.12 ±0.01	0.26 ±0.03	0.81 ±0.01

Best methods per dataset and player set. On CARE-PD, standard imputers score highest at every granularity and under both metrics (windows, joints, and cells; Table 5a and Table 17). On ESC-50, KS-Marginal scores highest on both metrics at windows (0.92 faithfulness, 0.88 PlayerAOPC), but on the frequency-band and band×window players the two metrics separate: KS-VAEAC is best on faithfulness (0.87, 0.88) and KS-Marginal on PlayerAOPC (0.79, 0.85). On PTB-XL no method is best everywhere. KS-Marginal has the highest faithfulness at windows. Per lead, KS-Zero and KS-Mean have the highest faithfulness (0.66) with KS-Flow close behind (0.64), and KS-Flow has the clearly highest PlayerAOPC (0.58 vs. 0.47).

Spatiotemporal cells on real data. Table 17 extends the main-text real-data tables to the finest player set: joint×window cells on CARE-PD (M=68), lead×window cells on PTB-XL (M=48), and band×window cells on ESC-50 (M=16).

The main-text findings persist at this granularity. CARE-PD keeps its highest-scoring standard imputers under both metrics (Result 5), and its two backbones continue to disagree on which standard imputer scores highest. The metric disagreement of Result 5 is at its starkest on model A, where KS-Zero has the best faithfulness (0.89) and a negative PlayerAOPC. ESC-50 reproduces the split of Result 5 at cells: KS-VAEAC leads faithfulness (0.88) while KS-Marginal leads PlayerAOPC (0.85). PTB-XL remains the contested case and reproduces the metric split at cells: KS-VAEAC leads faithfulness (0.78) while KS-Zero and KS-Mean lead PlayerAOPC (0.39), with KS-Marginal behind on both.

Reading across Table 5, and 17, no method family wins PTB-XL under both metrics at any granularity. This is the strongest real-data illustration that a no-oracle metric is itself a declared choice (Result 5). As everywhere, AOPC magnitudes shrink with player granularity and are only comparable within one player set (§2.5).

H Imputer Details and Capability Checks

KS-Gauss (closed-form Gaussian conditional). Following the parametric approach of Aas et al. (2021a), KS-Gauss fits one joint Gaussian to the training pool with Ledoit–Wolf-shrunk covariance.

For each coalition, it fills the hidden coordinates from the closed-form conditional distribution given the visible ones (R=5 draws, averaged). It has no learned components. Its only assumption is Gaussian dependence, which is exactly satisfied on the Gaussian-marginal synthetic families and deliberately violated on `burr`.

KS-VAEAC. Following the arbitrary-conditioning framework of Ivanov et al. (2019); Olsen et al. (2022), the VAEAC imputer is a per-frame-token transformer with three subnetworks (full encoder, prior encoder, decoder). Each is a 2-layer transformer trunk with d_{model}=256, 8 heads, feed-forward 512, dropout 0.1, and sinusoidal frame positional encoding. Each frame is one token (full encoder sees $[\boldsymbol{x}, \mathbf{m}]$, prior encoder $[\boldsymbol{x} \odot \mathbf{m}, \mathbf{m}]$, decoder $[\mathbf{z}, \boldsymbol{x} \odot \mathbf{m}, \mathbf{m}]$ for mask $\mathbf{m}$). The per-frame latent dimension is 64, and a Gaussian output head carries a single learnable global $\log \sigma$.

The loss is negative log-likelihood on hidden entries plus $\mathrm{KL}(q \,\|\, p)$, optimized with Adam (learning rate 10^{-3}, gradient clip 1.0, batch 16) under randomly mixed masks (temporal-window, spatial-unit, and element-wise). Training runs for 80 epochs on the synthetic families (the motion-capture imputers train through the external CARE-PD repository's configurations). Every released checkpoint records its full training configuration. At inference the latent is drawn from the prior encoder, decoded, and the observed entries are restored exactly. In the notation of §2.3, the imputer draws a latent z from the amortized conditional prior p_ψ and decodes it with g_θ:

$$\tilde{\boldsymbol{x}}_{\bar{S}}^{\text{VAEAC}} = g_\theta(z, \boldsymbol{x}_S), \qquad z \sim p_\psi(z \mid \boldsymbol{x}_S). \tag{5}$$

KS-Flow. KS-Flow is a conditional optimal-transport flow-matching imputer (Lipman et al., 2023; Tong et al., 2024). The velocity network is an encoder-only transformer (d=256, 4 heads, 4 layers, GELU) with frame-level tokenisation. It is trained to minimize $\mathbb{E}\,\|v_\theta(\boldsymbol{x}_t, t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\|^2$ along the interpolant $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$ with $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, I)$. At inference, 20 midpoint-Euler ODE steps are interleaved with RePaint harmonisation (Lugmayr et al., 2022): at each step the observed coordinates are snapped to their interpolant values, so the trajectory conditions on the visible data. In the notation of §2.3, the imputer transports noise to the manifold along the learned velocity field v_η:

$$\tilde{\boldsymbol{x}}_{\bar{S}}^{\text{FLOW}} = \boldsymbol{x}_{\bar{S}}^{(1)}, \qquad \tfrac{d\boldsymbol{x}_{\bar{S}}^{(t)}}{dt} = v_\eta\big(\boldsymbol{x}_{\bar{S}}^{(t)}, t, \boldsymbol{x}_S\big), \qquad \boldsymbol{x}_{\bar{S}}^{(0)} \sim \mathcal{N}(0, I). \tag{6}$$

Real-data checkpoints. One VAEAC and one flow imputer are trained per real dataset on that dataset's training pool and reused across all of its tracks and folds. The PTB-XL pair uses a smaller configuration recorded in its released checkpoints: d_{model}=128, 4 heads, 4 layers, feed-forward 256, VAEAC latent 32, and flow with 20 ODE steps. Every released checkpoint stores its full architecture dictionary, and the loaders instantiate from it.

The capability check. A generative imputer is a *model* of the conditional distribution, and how good a model it is depends on what is being hidden. We therefore characterize every learned imputer with a hide-one-reconstruct check, run per player-set mask family. The check hides one player (a spatial unit, a temporal window, or a cell), imputes its conditional mean from the visible rest (8 draws averaged), and correlates the imputation with the hidden ground truth. Scores are computed over 3 folds and 24 sequences per fold.

Two controls calibrate the score. The *donor* control copies the hidden entries from the most-correlated other unit of the same sequence: the recovery available from raw redundancy alone, with no model. The *unconditional* control runs the same imputer with everything hidden: the recovery available from dataset stereotypy alone, with no conditioning. An imputer demonstrably conditions on the visible context exactly when its recovery clears both controls. Table 18 reports the full matrix.

PTB-XL: conditioning proven cross-lead, absent cross-window. The twelve leads observe one cardiac source, and the imputers exploit it. A hidden lead is recovered at 0.95–0.96 correlation, against a donor control of -0.01 and an unconditional control of 0.00. That is genuine cross-lead conditioning, not redundancy or stereotypy. Cell masks behave the same way (0.98–0.99), since a hidden lead–window cell is constrained by both its lead's other windows and the other leads.

A hidden *whole window*, by contrast, removes that time span from every lead at once, leaving nothing contemporaneous to condition on. Recovery drops to 0.05–0.10: ECG imputation

Table 18: **Imputer capability check**: hide-one-recover correlation between the imputed conditional mean and the hidden ground truth, per player-set mask family (mean ± spread over masks; 3 folds, 24 sequences per fold, 8 draws). *Donor*: copy the hidden entries from the most-correlated other unit of the same sequence – the recovery available from raw redundancy alone. *Unconditional*: imputer run with everything hidden. A learned imputer conditions on the visible context exactly when its recovery clears both controls.

Dataset	Imputer	Spatial	Temporal	Cell
CARE-PD	VAEAC	1.00 ±0.00	0.96 ±0.12	1.00 ±0.00
	Flow	1.00 ±0.00	1.00 ±0.01	1.00 ±0.00
	Donor control	0.97	0.98	0.98
	Unconditional	0.99 (all families)		
PTB-XL	VAEAC	0.96 ±0.09	0.05 ±0.10	0.98 ±0.07
	Flow	0.95 ±0.12	0.10 ±0.15	0.99 ±0.04
	Donor control	-0.01	-0.01	-0.01
	Unconditional	0.00 (all families)		
ESC-50	VAEAC	0.87 ±0.24	0.31 ±0.35	0.36 ±0.39
	Flow	0.88 ±0.17	0.29 ±0.33	0.37 ±0.39
	Donor control	0.62	0.16	0.12
	Unconditional	0.75 (all families)		

interpolates across leads, not across time. The same imputer is thus a trustworthy conditional model under spatial and cell players and not under temporal players. The near-perfect lead recovery also explains the flat per-lead attributions of §5.2: deleting a lead removes almost no information, so low per-lead credit is the correct conditional answer.

CARE-PD: at the redundancy ceiling. Gait is so stereotyped and cross-joint-redundant that every number in the CARE-PD block saturates. The imputers recover hidden players at 0.96–1.00, but the donor control already reaches 0.97–0.98 and the unconditional control 0.99. Recovery itself is excellent (the imputations are realistic), but the check cannot separate conditioning from dataset stereotypy where stereotypy alone suffices. On such data the two are observationally equivalent, and the controls say so explicitly.

ESC-50: conditioning proven cross-band, partial elsewhere. On spatial (frequency-band) masks the imputers clear both controls (0.87–0.88 against donor 0.62 and unconditional 0.75): hidden bands are genuinely reconstructed from the visible spectrum. On temporal and cell masks, recovery (0.29–0.37) clears the donor control (0.16/0.12) but not the unconditional one: what is recovered there reflects dataset-level spectrogram structure more than clip-specific content.

Read as a whole, the check is a capability map, not a verdict. Before any attribution is computed, it tells a practitioner which mask families (and hence which player sets) a given generative imputer is a trustworthy conditional model for. It also supplies the controls needed to say so on a new dataset. This is the measurement behind the decision guide's "generative imputer *with* a capability check" branch (Appendix K).

Practical interpretation. If the data distribution is analytically known, the exact conditional imputer, or its closed-form Gaussian approximation, is the natural choice: KS-Gauss stays competitive even when its Gaussian assumption is violated (Result 2). Where the check finds near-perfect reconstruction, as for the hidden ECG lead above, hiding a player removes almost no information, so near-uniform credit across players is the correct explanation rather than a failure. Where no conditional model passes the check, standard imputers are the sound choice, interpreted as answers to the marginal game: an exact answer to the marginal question is more useful than an inaccurate answer to the conditional one (Result 4).

I Reproducibility

The release contains the complete pipeline: the synthetic data generators, classifier and imputer trainers, and the attribution and grading pipelines (coalition designs, imputers, WLS solver, exact

and high-budget targets, capability check). It also contains every configuration file behind every reported cell. The canonical per-cell result files ship with the code, with a table-by-table map from each paper table to its file and rerun entry point, so every reported number is verifiable in minutes, without a GPU.

Reference checkpoints are hosted with a SHA-256 manifest, verified on download, and split by the source datasets' licenses. Weights whose source license forbids redistributing derivatives retrain from the documented recipes. Each retraining path was validated by regenerating the shipped reference tracks bit-exactly. The packaged estimator, oracle imputations, and player partitions are pinned by the released test suite, and the oracle construction carries the unit-test battery of Appendix B. Per-cell results for the gate-excluded synthetic CNNs are included (Appendix F), so the criterion is reproducible and revisable. The random streams behind every reported number are pinned in the configurations and entry points: data generation, training, sampled coalition designs (seed 7919), imputer draws, and evaluation subsets.

The CARE-PD backbones and imputers fine-tune through the external CARE-PD repository. The WindowSHAP baselines wrap a third-party package. Display names map to configuration names (`gauss` → `gaussian_k4`, `burr` → `burr_m5`, `skeleton` → `skeleton_structured`, `gait` → `gait_periodic`, `skel+gait` → `skeleton_gait_combined`). KS-Gauss appears in the canonical files as `ks_shapr`, and WindowSHAP-S/D as `windowshap_stationary`/`windowshap_dynamic`.

Experiments ran on a shared SLURM-managed GPU cluster. Every training, attribution, and grading job fits on a single GPU, and the benchmark imposes no multi-node requirements.

J Additional Discussion of Explanation Evaluation

Existing evaluation techniques have shortcomings that are general and others specific to spatiotemporal data. First, general evaluation practice can certify explanations for the wrong reasons. Saliency maps can be insensitive to model and data randomization (Adebayo et al., 2018). Deletion-style evaluations conflate attribution quality with the distribution shift they induce (Hooker et al., 2019), and faithfulness metrics inherit related instabilities (Bhatt et al., 2020; Hedström et al., 2023). Our metric-disagreement finding (Result 5) extends this lineage to imputer-based Shapley evaluation on real spatiotemporal data.

K Setting up a Shapley Analysis in Practice

We summarize MotionBench-XAI's findings into a decision guide (Figure 3) organized by the three questions a practitioner should ask before computing Shapley attributions on spatiotemporal data. Each question selects one of the choices the benchmark measures, and each answer is a choice to state alongside the explanation.

What is known about the data's distribution? This question selects the imputer. If the distribution can be learned, a generative imputer is appropriate, but it should not be trusted blindly. Run the capability check first: hide one player, ask the imputer to reconstruct it from the rest, and compare the reconstruction against simply copying the most similar visible unit. The comparison against copying matters, because redundant data makes any imputer look capable. Appendix H gives the full check and its interpretation.

What is the goal of the explanation? This question selects the players, and the players should match the question, not the attribution method. To ask which phase of a movement matters, use temporal windows. To ask which anatomical unit matters, use spatial players. To localize in both space and time, use joint×window cells, and budget for them: cells need more classifier evaluations, and the grading error floor of Result 3 should be measured and reported next to the results. Two warnings transfer directly from the synthetic suite. On periodic data, temporal windows can make the conditional game unanswerable (Result 1). And existing temporal SHAP methods hard-code time windows as their players, so changing the question means changing the method (§2.3).

How will the explanation be scored? With a ground truth, grade each method against the ground truth of its own game, and state which game that is (Result 4). Without one, report faithfulness *and* PlayerAOPC together, and expect them to disagree (Result 5). The best method under one metric is a statement about how that metric is built, not about explanation quality in general. And AOPC values should never be compared across player granularities (§2.5).