COMPLEMENTING MODELS VIA RESIDUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-based models of dynamical systems deviate from observed behavior due to simplifying assumptions, abstraction, epistemic uncertainty, unmodeled variability, couplings with other systems, and environmental influences. We introduce a dynamical state reconstruction architecture that deploys an existing physics-based model as a predictive backbone, enhanced with data-driven residual corrections learned by a self-supervised transformer. The learning protocol relies on epistemic tokenization, representing each observation as a tuple of value, observation coordinate, and epistemic state. The epistemic state distinguishes available evidence from reconstruction queries, providing an explicit learning and inference signal. The learned correction is explicitly bounded, preventing arbitrarily large deviations from the physics-based model predictions. We instantiate the approach using transformer-based state reconstruction of a permanent magnet synchronous machine from multimodal signals, including quantities observed during training but unavailable at inference time. The resulting architecture integrates transparent, interpretable physics-based models with evidence-driven deep learning to close the gap between the reference model's predictions and observed system behavior.

1 INTRODUCTION

Physics-based models of technical systems are a hallmark of engineering science and practice. For a particular engineered system, modeling supports decision-making throughout its lifecycle. In particular, models can address epistemic concerns related to system health, including condition monitoring, anomaly detection, diagnosis, and prognosis. Models developed for these purposes generally serve different objectives from those used for control. While many control applications can tolerate substantial model imprecision and uncertainty, anomaly detection, diagnosis, and prognosis place more stringent demands on model fidelity, as deviations between predicted and observed behavior directly affect our ability to characterize and reason about system health throughout its operational lifetime. Accurate, high-fidelity multiphysics models are increasingly accessible through modern engineering environments, but their computational demands generally make them unsuitable for online deployment and real-time inference. Abstraction and simplifying assumptions are therefore powerful engineering tools, often yielding tractable lumped-parameter models expressed as ordinary differential equations (ODEs). These models can be deployed in digital twins of cyber-physical systems, enabling epistemic reasoning tasks that contribute to awareness of system health. Abstraction and simplification come at a cost, however, manifested in the systematic omission or approximation of nonlinear dynamics and variation across operating conditions. There is therefore a case for learning lightweight data-driven models that reduce discrepancies between predicted and observed system behavior. Purely data-driven models, however, may comprise large numbers of parameters that do not directly correspond to physical quantities, making their collective behavior difficult to interpret, characterize, and constrain. This is particularly concerning outside the observed data distribution and in safety-critical applications, where observations from critical operating regions may be scarce or entirely unavailable.

We develop WRAPPER, a recursive acronym for *WRAPPER from Residuals Around Prior Predictors through Empirical Reconstruction*. A WRAPPER retains a prior predictor and learns from its residuals to complement its predictions. Access through the prior's predictive interface makes the architecture independent of its internal structure and of ODE realizations. We present the self-supervised learning protocol and transformer used for reconstruction; in this case study, reconstruction is grounded in the Cauchy initial value problem of an ODE model.

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.

Table 1: Comparison of structured observation representations. Notation: v denotes an observation value, v^* a value carrying an out-of-distribution token, t time, t_r relative time, Δt elapsed time since the last observation of a variable, ℓ a spatial location, j a variable or channel, with a subscript j indicating that its identity is given by the structure of the representation, m an observation mask, and e an epistemic state.

Method	Reference	Representation	Epistemic encoding
GRU-D	Che et al. (2018)	$(v, m, \Delta t)$	m: observed / missing
M-RNN	Yoon et al. (2019)	$(v, m, \Delta t)$	m: observed / missing
CDSA	Ma et al. (2019)	(t, ℓ, v)	Implicit through sparsity
mTAN	Shukla & Marlin (2021)	$(t, v)_j$	Implicit through sparsity
STraTS	Tipirneni & Reddy (2022)	(t, j, v)	Implicit through sparsity
Tripletformer	Yalavarthi et al. (2023)	(t, j, v)	Implicit through sparsity
SAITS	Du et al. (2023)	$(v, m)_j$	m: observed / missing
Signal inference	Muñoz Gutiérrez et al. (2024)	$(c, v^*)_j$	Out-of-distribution token in v^*
WRAPPER	This work	$(\mathbf{v}, \mathbf{t_r}, \mathbf{e})_\mathbf{j}$	**Explicit epistemic state** e

The main contributions are threefold. First, we introduce a WRAPPER architecture for complementing predictive models through empirical residual learning while retaining the reference model as the predictive backbone. Second, we develop a self-supervised transformer reconstruction architecture using epistemic tokenization to distinguish available evidence from reconstruction queries, together with bounded residual correction. Third, we evaluate the approach on experimental measurements from a permanent-magnet synchronous machine, using a physics-based model as the prior and reconstructing quantities measured during training but unavailable at deployment.

The paper is organized as follows. Section 2 reviews related approaches. Section 3 formulates the reconstruction problem. Section 4 introduces the WRAPPER architecture, including its epistemic tokenization, self-supervised learning protocol, transformer-based reconstruction, and bounded residual correction. Section 5 presents the case study and experimental setup, followed by the results in Section 6. Section 7 discusses the implications and limitations of the approach and concludes the paper.

2 Related Work

Tables compare WRAPPER with prior work on observation representation, self-supervised reconstruction, hybrid modeling, and residual learning for diagnosis; Appendix A provides details.

Observation representations. Table 1 compares how reconstruction methods represent values, their coordinates, and the status of available or withheld observations. Earlier methods convey that status through masks, sparse observations, or a special value token. WRAPPER assigns each token an explicit epistemic state, specifying its role as evidence or a reconstruction target independently of its value and relative temporal coordinate.

Self-supervised reconstruction. Table 2 compares how reconstruction methods select the evidence supplied to a model and the values it must recover. Existing approaches use corruption, masking, or modality assignments to define training targets. WRAPPER constructs queries to reconstruct residuals and selected inputs from evidence available at inference time.

Bridging model-based and data-driven approaches. Table 3 distinguishes hybrid approaches by where learning enters the model: within its parameters or governing dynamics, through constraints on a learned predictor, or as a correction to its output. WRAPPER retains the executable prior predictor and learns to reconstruct its healthy residual from the available context. The reconstructed residual complements the prior prediction while leaving the prior model unchanged.

Residual learning for fault diagnosis. Table 4 compares where learning enters the residual generator. The RNN and NODE approaches learn nominal dynamics, with structural constraints, prediction stability, or uncertainty shaping the resulting diagnostic residuals. WRAPPER retains an executable prior for those dynamics and uses the transformer to reconstruct the residual expected under healthy operation.

Table 2: Self-supervised reconstruction strategies. The comparison focuses on how input evidence and reconstruction targets are selected.

Approach	Reconstruction task	Input–target selection	References
Denoising autoencoders	Recover an input from a corrupted version	Corruption determines what must be recovered	Vincent et al. (2008; 2010)
Masked token modeling	Predict withheld text or visual tokens	Masks select token positions as targets	Devlin et al. (2019); Bao et al. (2022)
Masked time-series modeling	Reconstruct withheld values or series segments	Random masks determine the missing evidence	Zerveas et al. (2021); Dong et al. (2023)
Multimodal masked modeling	Infer target modalities from available modalities	Input and target modalities vary across tasks	Bachmann et al. (2024)
Structured signal inference	Predict, retrodict, or infer masked signals	Predefined masking patterns specify tasks	Muñoz Gutiérrez et al. (2024)
WRAPPER	**Reconstruct residuals and selected inputs**	**Queries specify available evidence and targets**	**This work**

3 State Reconstruction Problem

State reconstruction uses a dynamical model to estimate the evolution between successive observations. Given an ODE model F, each reconstruction starts from the measured state at t_{k-1} and solves the corresponding Cauchy initial value problem up to t_k. Because inputs are observed only at sampling times, admissible signal extensions specify their behaviour between observations, allowing numerical integration with steps smaller than the sampling interval. The state trajectory itself is determined by the model and the initial condition. The resulting prediction $\widehat{\boldsymbol{x}}_{p,k}$ defines the residual

$$\boldsymbol{r}_k = \boldsymbol{x}_k - \widehat{\boldsymbol{x}}_{p,k}.$$

Reinitialization from each measured state limits the accumulation of prediction error across intervals. Whereas learned dynamic residual generators estimate the nominal evolution from data, the present realization of WRAPPER executes the prior dynamics and learns a correction to the resulting residual. The formal construction is given in Appendix C.

4 WRAPPER Architecture

WRAPPER complements mechanistic state reconstruction through residual learning from empirical evidence. Figure 1 provides an overview of the deployed architecture, from mechanistic reconstruction and residual formation, through learned residual reconstruction, to the governed application of the reconstructed residual in complemented state estimation. The following subsections develop the three stages and explain the self-supervised residual learning method underlying the architecture.

4.1 Mechanistic State Reconstruction

The Sampling and Reconstruction components of the deployed architecture in Fig. 1 instantiate the state reconstruction problem of Section 3. Within WRAPPER, we denote the reconstruction produced by the prior mechanistic model by $\widehat{\boldsymbol{x}}_p$ to distinguish it from the complemented estimate $\widehat{\boldsymbol{x}}$. The sampled input is extended to the admissible continuous-time input $\widetilde{\boldsymbol{u}}$ and supplied, together with the sampled state, to the mechanistic reconstruction, yielding the reconstructed digital state signal $\widehat{\boldsymbol{x}}_p$. The resulting discrepancy from the observed state, $\boldsymbol{r}$, provides the empirical signal for residual learning.

4.2 Residual Learning

To distinguish the two reconstruction roles considered in this section, we refer to the mechanistic reconstruction of the system state as *state reconstruction* (SR) and to the learned reconstruction of its residuals as *residual reconstruction* (RR).

Table 3: Representative forms of hybridization between model-based and data-driven components. Notation: M denotes a model-based component, L a learned component, x its inputs, θ model parameters or internal quantities, $r = y - M(x)$ a model residual, and $\mathcal{F}_{\text{phys}}$ a family of learned models satisfying specified physical properties.

Hybridization	Role of learning	Representative form	References
Additive	Complement a model-based prediction with a learned component	$\hat{y} = M(x) + L(x)$	Haesloop & Holt (1990)
Internal estimation	Estimate unknown parameters or quantities used by the model	$\hat{y} = M(x; L_\theta(x))$	Psichogios & Ungar (1992); Huang et al. (2024)
Governing-term complement	Complement known governing dynamics with a learned term	$\dot{x} = f_M(x, u) + f_L(x, u)$	Rackauckas et al. (2020)
Model-discrepancy learning	Learn model discrepancy for subsequent model correction or compensation	$\hat{r} = L_r(\phi(x))$	Wang et al. (2017); Duraisamy et al. (2015); Tracey et al. (2015); Parish & Duraisamy (2016); Singh & Duraisamy (2016); Singh et al. (2017); Lu et al. (2024)
Physics-consistent by construction	Guarantee selected physical properties through the structure of the learned model	$L_\theta \in \mathcal{F}_{\text{phys}}$	Huang et al. (2020); Xu et al. (2021a;b)
Physics-constrained learning	Constrain a learned predictor using mechanistic knowledge	$\min_L \ \mathcal{L}_{\text{data}}(L) + \lambda \mathcal{L}_{\text{phys}}(L)$	Karniadakis et al. (2021)
Physics-supported learning	Transfer knowledge generated by a mechanistic model to a learned predictor	$M \to \mathcal{D}_M \to L$	Lu et al. (2024); Liu et al. (2024)
Compound	Combine internal learned estimation with external residual correction	$\hat{y} = M(x; L_\theta(x)) + L_r(\phi(x))$	Wang et al. (2026)
Residual learning	**Reconstruct residuals around an unchanged prior predictor**	$\hat{\mathbf{y}} = \mathbf{M}(\mathbf{x}) + \mathbf{L}(\mathbf{r})$	**This work**

Table 4: Learning in residual-based fault diagnosis. RNN: recurrent neural network; NODE: neural ordinary differential equation.

Model	Role of learning	Design emphasis	References
RNN	Learn nominal dynamics for residual generation	Structural constraints and fault localization; prediction stability; generalization to unseen healthy conditions	Pulido et al. (2019); Jung (2019); Mohammadi et al. (2025); Allansson et al. (2024)
NODE ensemble	Learn nominal dynamics and estimate prediction dispersion	Uncertainty-aware detection thresholds	Jung & Westny (2026)
Prior model + transformer	**Learn healthy residual corrections**	**Retain prior dynamics; reconstruct from available context**	**WRAPPER (this work)**

4.2.1 CONTEXT.

RR is conditioned on recent residual evidence and the inputs under which that evidence was observed. Using the operators defined in Appendix B, the Context component in Fig. 1 collects both

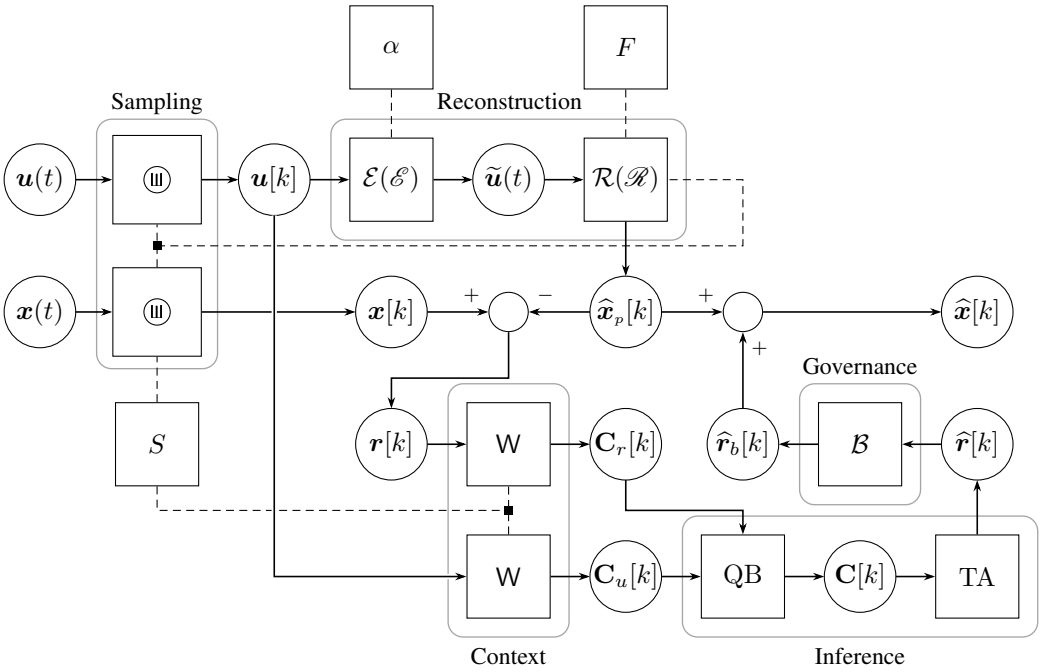


Figure 1: WRAPPER architecture for complementing mechanistic state reconstruction through residual learning. Continuous-time signals are denoted by function notation, as in $\boldsymbol{u}(t)$ and $\boldsymbol{x}(t)$, while square brackets denote discrete signals, as in $\boldsymbol{u}[k]$ and $\boldsymbol{x}[k]$. The sampling support is denoted by S. The application $\mathcal{E}(\mathscr{E})$ of the signal extension solution operator yields the admissible continuous-time input extension $\widetilde{\boldsymbol{u}}$ from $\boldsymbol{u}$, subject to the admissibility predicate α. The application $\mathcal{R}(\mathscr{R})$ of the state reconstruction solution operator yields the discrete prior-predictor reconstruction $\widehat{\boldsymbol{x}}_p$ from $\widetilde{\boldsymbol{u}}$. The corresponding residual signal is $\boldsymbol{r} = \boldsymbol{x} - \widehat{\boldsymbol{x}}_p$. The Context module forms the residual context matrix $\mathbf{C}_r = \boldsymbol{r}([*\mathsf{W}(S; t_k, L)])$ and the input context matrix $\mathbf{C}_u = \boldsymbol{u}([*\mathsf{W}(S; t_k, L)])$, where $\mathsf{W}(S; t_k, L)$ denotes the time window of context length L ending at t_k and $[*\mathsf{W}(S; t_k, L)]$ its canonical lifting. The Inference module combines $\mathbf{C}_r$ and $\mathbf{C}_u$ through the Query Builder (QB), which performs the preprocessing required to form the model-ready context $\mathbf{C}$. The transformer autoencoder (TA) maps this context to the predicted residual $\widehat{\boldsymbol{r}}$. The Governance module applies the bounding operator $\mathcal{B}$ to the predicted residual, yielding the bounded residual $\widehat{\boldsymbol{r}}_b$. The bounded residual is added to the prior-predictor reconstruction $\widehat{\boldsymbol{x}}_p$ to obtain the corrected state estimate $\widehat{\boldsymbol{x}}$.

signals over the context window $\mathsf{W}(S; t_k, L)$, yielding

$$\mathbf{C}_r[k] = \boldsymbol{r}([*\mathsf{W}(S; t_k, L)]) \in \mathbb{R}^{L \times n_x}, \qquad \mathbf{C}_u[k] = \boldsymbol{u}([*\mathsf{W}(S; t_k, L)]) \in \mathbb{R}^{L \times n_u}.$$

Here, n_x and n_u denote the numbers of state and input components, respectively; rows correspond to sampling times and columns to signal components. Together, $\mathbf{C}_r[k]$ and $\mathbf{C}_u[k]$ provide the recent residual and input evidence from which the reconstruction query is formed.

4.2.2 Query construction.

Overview. The Query Builder (QB) normalizes the residual and input contexts and applies the query mapping constructed for them to obtain the epistemic token matrix

$$\mathbf{C} = \mathcal{Q}_T(\mathcal{N}(\mathbf{C}_r, \mathbf{C}_u)) .$$

Its construction is detailed in Appendix D.

Normalization. The normalization operator $\mathcal{N}$ standardizes the residual and input contexts using componentwise statistics estimated from the training subset and held fixed thereafter. Its definition and inverse transformation of reconstructed residuals are specified in Appendix D.1.

Target specification. The reconstruction query is parameterized by a target specification $T = (T_r, T_u)$, formally defined in Appendix D.2, where T_r and T_u identify the entries of $\mathbf{C}_r$ and $\mathbf{C}_u$, respectively, to be reconstructed. A Query Controller selects target specifications through selection and admissibility policies and, for a selected and admissible T, constructs the corresponding query-construction mapping $\mathcal{Q}_T$. The complete Query Controller construction is given in Appendix D.2.

Epistemic tokenization. Given a target specification T, $\mathcal{Q}_T$ represents each observation as an epistemic token $(v, t_r, e)_j$, comprising its value, relative temporal coordinate, and epistemic state, with signal identity j retained structurally by the token sequence. For reconstruction targets, the value is withheld while the temporal coordinate and signal identity are preserved, identifying what is to be reconstructed without exposing its value. The epistemic state distinguishes reconstruction targets from available evidence independently of numerical value. The complete query construction is specified and illustrated in Appendix D.3.

Transformer input. The Query Builder applies the query-construction mapping $\mathcal{Q}_T$ to the normalized contexts, producing the epistemic token matrix

$$\mathbf{C} = \mathcal{Q}_T(\mathcal{N}(\mathbf{C}_r, \mathbf{C}_u)) .$$

The resulting token matrix $\mathbf{C}$ is supplied as input to the Transformer for reconstruction.

4.2.3 Self-supervised learning.

Learning principle. The RR model is trained using self-supervision from context episodes whose values are fully known before reconstruction targets are selected. For a context episode $(\mathbf{C}_r, \mathbf{C}_u)$, a target specification T identifies value entries whose values are withheld from the Transformer by the corresponding query-construction mapping $\mathcal{Q}_T$. The withheld values remain available to the learning procedure and provide the reconstruction targets against which the model output is evaluated. Self-supervision therefore arises by constructing reconstruction problems from observed context episodes rather than requiring externally supplied labels.

Masking known signal values to create self-supervised reconstruction targets has previously been used for signal inference Muñoz Gutiérrez et al. (2024). Here, the same self-supervised principle is realized through target specification and epistemic tokenization: targeted values are withheld from the learner while remaining known to the learning procedure.

Teaching strategy. In this work, the teaching strategy presents the learner with RR problems in support of SR. For each training episode, it selects target specifications admitted by the Query Controller, determining which known context values are epistemically withheld and reconstructed and which remain available as evidence. Targets may vary in number, location, and signal identity, while values known during training need not be retained as evidence at deployment.

Transformer reconstruction. Given the token matrix $\mathbf{C}$, the Transformer $\mathfrak{W}_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ produces a reconstructed value sequence $\widehat{\mathbf{y}} = \mathfrak{W}_{\boldsymbol{\theta}}(\mathbf{C})$. Let $\mathbf{y}$ be the corresponding unmasked values and $\mathbf{m}_T \in \{0,1\}^{Ln_c}$ select the targets specified by T. The loss is the mean squared error at those positions:

$$\mathcal{L}_T(\widehat{\mathbf{y}}, \mathbf{y}) = \frac{\|\mathbf{m}_T \odot (\widehat{\mathbf{y}} - \mathbf{y})\|_2^2}{\|\mathbf{m}_T\|_1},$$

where $\odot$ denotes elementwise multiplication.

4.3 Complemented State Reconstruction

As shown in Fig. 1, the reconstructed residual values produced by the Transformer are returned to the original residual coordinates through the inverse normalization defined in Appendix D.1, yielding $\widehat{\boldsymbol{r}}$. The contribution of $\widehat{\boldsymbol{r}}$ to the mechanistic SR $\widehat{\boldsymbol{x}}_p$ is then governed before the two are combined, yielding the complemented SR $\widehat{\boldsymbol{x}}$.

5 CASE STUDY

We evaluate WRAPPER using a mechanistic predictor of motor electrical dynamics and residual corrections learned from healthy measurements.

5.1 EXPERIMENTAL SYSTEM AND DATA

Data were collected at CEITEC, Brno University of Technology, from a three-phase interior permanent-magnet synchronous motor under field-oriented control. The motor has 21 pole pairs; measurements were sampled at $10\,\mathrm{kHz}$ across electrical angular velocities of 900–$5000\,\mathrm{rad/s}$ and speed-dependent mechanical loads within -5 to $5\,\mathrm{N\,m}$. The platform, operating conditions, and data preparation are detailed in Appendix E.

5.2 MECHANISTIC MODEL AND RECONSTRUCTION

The prior predictor F models the IPMSM's nominal electrical dynamics in the rotating dq frame Zezula et al. (2025). Its state is $\boldsymbol{x} = (i_d, i_q)^\mathsf{T}$, and its input is $\boldsymbol{u} = (u_d, u_q, \omega_e)^\mathsf{T}$, comprising the corresponding stator voltages and electrical angular speed. The dynamics are

$$\dot{i}_d = \frac{u_d - R i_d + \omega_e (L_q i_q + \lambda^q_{\mathrm{pm}})}{L_d}, \tag{1}$$

$$\dot{i}_q = \frac{u_q - R i_q - \omega_e (L_d i_d + \lambda^d_{\mathrm{pm}})}{L_q}, \tag{2}$$

where R is effective stator resistance, L_d, L_q are the axis inductances, and $\lambda^d_{\mathrm{pm}}, \lambda^q_{\mathrm{pm}}$ are the permanent-magnet flux components.

Measured torque τ is used during training to teach the Transformer to reconstruct it when unavailable; it is neither an input to F nor assumed available at deployment. To instantiate the state reconstruction problem in Section 3, each measured state anchors a one-step prediction using linearly interpolated inputs and two forward Euler half-steps. A common parameterization is identified from healthy observations across the operating envelope. Parameter identification, numerical validation, and further implementation details appear in Appendix F.

5.3 EMPIRICAL RESIDUAL RECONSTRUCTION

The empirical reconstruction model uses data collected during healthy motor operation. Each context comprises five consecutive sampling instants of the current reconstruction residuals r_d and r_q, stator voltages u_d and u_q, electrical angular speed ω_e, and torque τ. While u_d, u_q, and ω_e are assumed observable, torque represents an input whose value may be unknown during deployment, despite being measured by the dynamometer during experimentation. The epistemic representation allows residual reconstruction to be conditioned on this distinction, using the torque values when observed while explicitly representing their absence when unknown. The resulting 30 observations are represented using epistemic tokenization, as illustrated in Figure 2.

The lifted tokens are projected to a 32-dimensional latent representation and augmented with sinusoidal positional encoding before being processed by three pre-normalized Transformer blocks with four-head self-attention and $32 \to 64 \to 32$ feed-forward layers. A token-wise reconstruction head maps the resulting representations to scalar reconstructed values. A schematic of the architecture is provided in Fig. 5.

5.4 TEACHING POLICY

The teaching policy rests on the structure of measurements in engineered systems such as our electric drive: withheld values can be reconstructed from retained context. This premise aligns with the *manifold assumption* in semi-supervised learning Chapelle et al. (2006) and reconstruction from corrupted observations Vincent et al. (2010). Jung & Axelsson (2024) use the intrinsic dimension of nominal observations to select minimally redundant signals for residual construction. SimMTM Dong et al. (2023) relates masked time-series reconstruction to local manifold structure.

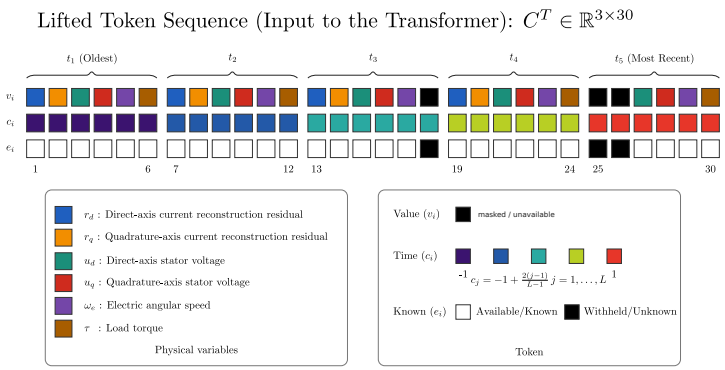


Figure 2: Lifted token sequence used by the Transformer in the case study. The transpose $\mathbf{C}^\top$ depicts tokens column-wise, with each token encoding a value v_i, relative temporal coordinate c_i, and epistemic state e_i. In this particular query, the Transformer is requested to reconstruct the residuals at t_5 ($c_5 = 1$) and the unknown torque at t_3 ($c_3 = 0$).

In this work, complete observed contexts are transformed into epistemically incomplete training queries using the cardinality-based teaching target generator defined in Appendix F.3.1. The whole residual state at the most recent context position is always targeted. Input observations are withheld according to the chosen cardinality radius, including complete masking of the mechanical-load channel. This training case prepares the WRAPPER to infer the missing load signal when load measurements are unavailable.

5.5 Experimental Protocol

WRAPPER is evaluated by four-fold cross-validation grouped by physical recording. Each recording is tested once, with no recording shared among training, validation, and test subsets within a fold. Normalization is fitted on each fold's training subset before reconstruction pairs are compiled. A separate Transformer is trained for each fold using Adam and masked reconstruction loss; the model with the lowest validation loss is evaluated on that fold's test recordings. The split, pair construction, and training parameters are detailed in Appendix F.4.

6 Results

Table 5 reports healthy residuals for 21 recordings, each evaluated by the WRAPPER model for which it was held out. RMSE falls from 76–147 mA for the prior predictor to 45–55 mA with WRAPPER, while P99 falls from 151–228 mA to 96–126 mA. Bias and ellipse area also decrease for every recording. Figure 3 shows the corresponding changes in residual means and ellipses. The smaller ellipses retain empirical coverage under the temporal-segment evaluation (Appendix F.5). Emp95 ranges from 89.3%–93.6% for the prior to 93.0%–95.2% with WRAPPER; Emp99 ranges from 95.4%–98.6% to 97.7%–98.9%.

7 Discussion and Conclusion

WRAPPER makes two conceptual commitments. Epistemic lifting gives observations and reconstruction targets explicit roles in a query, allowing the same architecture to act on different configurations of available evidence. Residual learning then complements an executable prior predictor while leaving its mechanistic component intact and separately inspectable. A fixed governance threshold bounds the correction; confidence- and support-dependent modulation could attenuate it smoothly as empirical support weakens. This separation preserves the interpretability of the prior, although it does not make the learned correction intrinsically interpretable.

The architecture does not require a physics-based prior. With a null backbone, WRAPPER reconstructs observations directly, connecting it to data-driven residual generation. The present experi-

Table 5: Held-out healthy dq-current residuals by recording. M identifies the independently trained WRAPPER fold model, R the test recording, and n the number of paired test contexts. Each numerical pair is prior/WRAPPER. For residual vector r and its sample mean μ, Bias is $\|\mu\|_2$; RMSE and P99 are the root-mean-square and 99th percentile of $\|r\|_2$. A_{95} is the area of the fitted 95% Gaussian covariance ellipse. Emp95 and Emp99 are empirical coverage percentages for the 95% and 99% Gaussian ellipses, respectively. Each complete recording, spanning all operating points and transitions, is divided into five contiguous, approximately equal-length temporal segments. Each segment is evaluated against ellipses fitted to the other four segments of the same recording, and coverage is pooled over the five held-out segments. Olive bold marks the smaller magnitude or area, or coverage closer to its nominal percentage.

Model	Record	n	Bias (mA)	RMSE (mA)	P99 (mA)	A_{95} (10^{-3} A^2)	Emp95 (%)	Emp99 (%)
M1	R3	3560	58.9/**9.36**	85/**46**	155/**104**	33.2/**19.3**	92.1/**94.9**	98.4/**98.5**
M1	R8	4520	137/**13.5**	147/**49**	228/**107**	26.0/**20.4**	93.6/**94.2**	98.5/**98.8**
M1	R9	5000	92.9/**1.89**	111/**52**	187/**119**	32.4/**25.4**	91.6/**93.4**	97.2/**97.7**
M1	R15	8600	54/**5.92**	83/**48**	167/**107**	35.9/**21.0**	91.9/**94.6**	97.6/**98.4**
M1	R20	12200	66.1/**16.8**	97/**54**	195/**126**	43.0/**24.9**	90.3/**93.0**	96.2/**97.8**
M2	R1	3080	72.2/**6.54**	93/**47**	172/**107**	32.3/**20.8**	90.8/**95.0**	97.4/**98.4**
M2	R6	4040	84.1/**2.95**	102/**47**	190/**102**	30.4/**21.0**	91.8/**94.8**	97.5/**98.9**
M2	R10	5000	121/**1.4**	132/**47**	211/**104**	25.9/**21.1**	93.3/**94.5**	98.6/**98.7**
M2	R17	11000	53/**12.7**	90/**52**	178/**115**	44.4/**23.6**	89.3/**94.2**	95.4/**98.4**
M2	R18	11000	51.9/**7.91**	82/**52**	171/**118**	37.5/**25.2**	92.4/**93.7**	97.6/**98.0**
M3	R4	3560	99.2/**11.9**	114/**51**	201/**108**	28.8/**22.7**	92.9/**95.1**	98.0/**98.8**
M3	R7	4520	112/**10.6**	127/**51**	205/**113**	31.5/**23.1**	92.0/**94.7**	97.3/**98.4**
M3	R11	5000	45.2/**3.51**	76/**46**	151/**102**	32.3/**20.1**	91.8/**94.3**	98.1/**98.8**
M3	R16	8600	65.3/**8.6**	90/**51**	187/**113**	35.0/**23.6**	92.2/**94.3**	98.0/**98.2**
M3	R21	12200	46.4/**14.6**	79/**55**	165/**120**	37.8/**26.6**	92.6/**93.9**	97.7/**98.4**
M4	R2	3080	113/**9.58**	126/**47**	208/**101**	27.8/**20.2**	92.6/**95.2**	98.1/**98.8**
M4	R5	4040	48.5/**5.72**	78/**46**	151/**100**	33.3/**19.5**	91.6/**94.3**	97.7/**98.7**
M4	R12	5000	75.2/**9.48**	95/**47**	186/**104**	30.3/**20.0**	92.6/**94.7**	98.0/**98.7**
M4	R13	6200	50.3/**4.02**	78/**45**	160/**96**	32.2/**18.5**	93.3/**94.9**	98.2/**98.9**
M4	R14	6200	69.4/**8.77**	91/**47**	187/**101**	30.9/**19.8**	92.4/**94.7**	97.8/**98.7**
M4	R19	12200	72.8/**10.2**	100/**52**	196/**116**	41.5/**24.8**	90.7/**93.7**	96.5/**98.4**

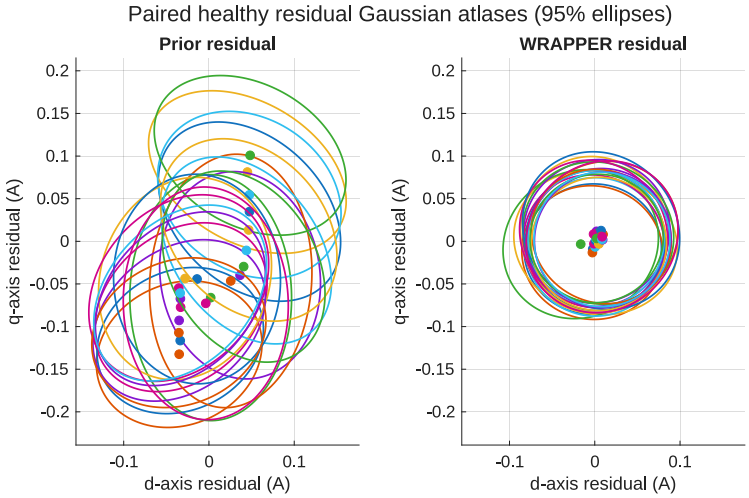


Figure 3: Healthy dq-current residuals for 21 recordings, before and after WRAPPER correction. Each dot marks a recording's sample residual mean; each ellipse encloses the nominal 95% region of a bivariate Gaussian with mean and covariance estimated from that recording's residuals.

ment instead examines the benefit of a mechanistic backbone: across held-out healthy recordings, the corrected residuals are less biased and more concentrated, with empirical ellipse coverage closer to its nominal level. These findings concern healthy residual behavior; they do not establish fault sensitivity or isolability. The present realization uses one vector-valued residual generator. Fault diagnosis generally requires a bank whose responses jointly meet detection, isolation, and robustness criteria. We are extending WRAPPER to articulated backbones and refining its epistemic descriptors to support richer meanings at learning and inference time.

AI USE STATEMENT

OpenAI ChatGPT (GPT-5.6 Sol) was used throughout this work as an interactive research and writing assistant supporting the research process. It supported scientific discussion and ideation, formulation and refinement of concepts, literature exploration, software development, analysis, visualization, and drafting and editing of the manuscript. AI-assisted methodological and conceptual suggestions were critically evaluated before incorporation. AI-assisted code was reviewed, tested, and validated against the underlying models, data, and expected numerical behavior. Literature references and factual claims were checked against their original or authoritative sources. Generative AI was not used to generate experimental data. Scientific decisions, interpretation of results, and conclusions remained the responsibility of the authors. All AI-assisted content was reviewed and edited by the authors, who take responsibility for the final content of this work, including text, claims, code, analyses, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work did not include human subjects, personal data, or experiments on living organisms. The proposed approach is intended to support the monitoring and analysis of technical systems and is not presented as a replacement for established safety mechanisms or safety-critical control and protection systems. Deployment in safety-critical applications requires rigorous validation and compliance with the corresponding engineering and safety requirements.

REPRODUCIBILITY STATEMENT

The reconstruction problem and proposed architecture are specified in Sections 3 and 4. The reference model, dataset, preprocessing, and experimental protocol are described in Section 5. Implementation code, experiment configurations, and scripts for reproducing the methodology and reported analyses are provided in the supplementary materials. The experimental dataset was provided to the authors by the original data providers and is not redistributed with this work. Researchers interested in access to the dataset should contact the original data providers directly.

AUTHOR CONTRIBUTIONS AND ACKNOWLEDGMENTS

Omitted for anonymous review.

REFERENCES

Niklas Allansson, Arman Mohammadi, Daniel E. Jung, and Mattias Krysander. Fuel Injection Fault Diagnosis Using Structural Analysis and Data-Driven Residuals. *IFAC-PapersOnLine*, 58(4): 360–365, 2024. doi: 10.1016/j.ifacol.2024.07.244.

Roman Bachmann, OÄŸuzhan Fatih Kar, David Mizrahi, Ali Garjani, Mingfei Gao, David Griffiths, Jiaming Hu, Afshin Dehghan, and Amir Zamir. 4M-21: An Any-to-Any Vision Model for Tens of Tasks and Modalities, 2024.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (eds.). *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. ISBN 978-0-262-03358-9.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. SimMTM: A simple pre-training framework for masked time-series modeling. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Wenjie Du, David Côté, and Yan Liu. SAITS: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023.

Karthikeyan Duraisamy, Ze Jia Zhang, and Anand Pratap Singh. New approaches in turbulence and transition modeling using data-driven techniques. In *53rd AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2015. doi: 10.2514/6.2015-1284.

Michael Friendly, Georges Monette, and John Fox. Elliptical insights: Understanding statistical methods through elliptical geometry. *Statistical Science*, 28(1):1–39, 2013. doi: 10.1214/12-STS402.

Dan Haesloop and Bradley R. Holt. Neural networks for process identification. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 429–434, 1990.

Daniel Z. Huang, Kailai Xu, Charbel Farhat, and Eric Darve. Learning constitutive relations from indirect observations using deep neural networks. *Journal of Computational Physics*, 416:109491, 2020. doi: 10.1016/j.jcp.2020.109491.

Kaide Huang, Beichen Ding, Chunyan Lai, and Guodong Feng. Flux linkage tracking-based permanent magnet temperature hybrid modeling and estimation for PMSMs with data-driven-based core loss compensation. *IEEE Transactions on Power Electronics*, 39(1):1410–1421, 2024. doi: 10.1109/TPEL.2023.3327956.

Daniel Jung. Isolation and localization of unknown faults using neural network-based residuals. In *Proceedings of the Annual Conference of the PHM Society*, volume 11, 2019. doi: 10.36001/phmconf.2019.v11i1.819. URL `https://papers.phmsociety.org/index.php/phmconf/article/view/819`.

Daniel Jung and David Axelsson. A study on redundancy and intrinsic dimension for data-driven fault diagnosis. In Ingo Pill, Avraham Natan, and Franz Wotawa (eds.), *35th International Conference on Principles of Diagnosis and Resilient Systems (DX 2024)*, volume 125 of *Open Access Series in Informatics (OASIcs)*, pp. 4:1–4:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi: 10.4230/OASIcs.DX.2024.4. URL `https://drops.dagstuhl.de/entities/document/10.4230/OASIcs.DX.2024.4`.

Daniel Jung and Theodor Westny. Uncertainty-aware fault diagnosis of unknown faults using ensemble-based NODE residuals. *Mechanical Systems and Signal Processing*, 242:113599, 2026. doi: 10.1016/j.ymssp.2025.113599.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

Anuj Karpatne, Gowtham Atluri, James H. Faghmous, Michael Steinbach, Arindam Banerjee, Auroop R. Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017. doi: 10.1109/TKDE.2017.2720168.

Lin Liu, Wenliang Yin, and Youguang Guo. Hybrid mechanism-data-driven iron loss modelling for permanent magnet synchronous motors considering multiphysics coupling effects. *IET Electric Power Applications*, 18(12):1833–1843, 2024. doi: 10.1049/elp2.12530.

Yuting Lu, Kaide Huang, Benfei Wang, Chunyan Lai, and Guodong Feng. Data-driven modeling and compensation strategy of PMSM considering core loss and saturation. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 12(2):1894–1905, April 2024. doi: 10.1109/JESTPE.2024.3364974.

Jiawei Ma, Zheng Shou, Alireza Zareian, Hassan Mansour, Anthony Vetro, and Shih-Fu Chang. CDSA: Cross-dimensional self-attention for multivariate, geo-tagged time series imputation. *CoRR*, abs/1905.09904, 2019.

Arman Mohammadi, Mattias Krysander, and Daniel E. Jung. Consistency-Based Diagnosis Using Data-Driven Residuals and Limited Training Data. *Control Engineering Practice*, 159:106283, 2025. doi: 10.1016/j.conengprac.2025.106283.

Stan Muñoz Gutiérrez, Adil Mukhtar, and Franz Wotawa. Transformer-based signal inference for electrified vehicle powertrains. In *Digital Transformation (DX 2024)*, volume 125 of *Open Access Series in Informatics (OASIcs)*, 2024. doi: 10.4230/OASIcs.DX.2024.29.

Eric J. Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016. doi: 10.1016/j.jcp.2015.11.012.

Dimitris C. Psichogios and Lyle H. Ungar. A hybrid neural network–first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992. doi: 10.1002/aic.690381003.

Belarmino Pulido, Jesús M. Zamarreño, Alejandro Merino, and Aníbal Bregón. State space neural networks and model-decomposition methods for fault diagnosis of complex industrial systems. *Engineering Applications of Artificial Intelligence*, 79:67–86, 2019. doi: 10.1016/j.engappai.2018.12.007.

Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

Satya Narayan Shukla and Benjamin M. Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

Anand Pratap Singh and Karthik Duraisamy. Using field inversion to quantify functional errors in turbulence closures. *Physics of Fluids*, 28(4):045110, 2016. doi: 10.1063/1.4947045.

Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA Journal*, 55(7):2215–2227, 2017. doi: 10.2514/1.J055595.

Sindhu Tipirneni and Chandan K. Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Brendan D. Tracey, Karthikeyan Duraisamy, and Juan J. Alonso. A machine learning strategy to assist turbulence model development. In *53rd AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2015. doi: 10.2514/6.2015-1287.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 1096â€“1103, New York, NY, USA, 2008. Association for Computing Machinery. doi: 10.1145/1390156.1390294.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. A physics informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on DNS data. *Physical Review Fluids*, 2(3):034603, 2017. doi: 10.1103/PhysRevFluids.2.034603.

Yanbo Wang, Chuhang Lin, Zhirui Chen, Tatsuki Sasamura, and Takeshi Morita. Hybrid physics-informed and data-driven model for torque estimation—toward real-time sensorless control of ultrasonic motors. *Advanced Engineering Informatics*, 69:103877, 2026. doi: 10.1016/j.aei.2025.103877.

Kailai Xu, David Z. Huang, and Eric Darve. Learning constitutive relations using symmetric positive definite neural networks. *Journal of Computational Physics*, 428:110072, 2021a. doi: 10.1016/j.jcp.2020.110072.

Kailai Xu, Alexandre M. Tartakovsky, Jeff Burghardt, and Eric Darve. Learning viscoelasticity models from indirect data using deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, 387:114124, 2021b. doi: 10.1016/j.cma.2021.114124.

Vijaya Krishna Yalavarthi, Johannes Burchert, and Lars Schmidt-Thieme. Tripletformer for probabilistic interpolation of irregularly sampled time series. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 986–995. IEEE, 2023. doi: 10.1109/BigData59044.2023.10386325.

Jinsung Yoon, William R. Zame, and Mihaela van der Schaar. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, May 2019. doi: 10.1109/TBME.2018.2874712.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2114–2124. Association for Computing Machinery, 2021. doi: 10.1145/3447548.3467401.

Lukas Zezula, Matus Kozovsky, Ludek Buchta, and Petr Blaha. Discrete-time modeling of interturn short circuits in interior PMSMs. *IEEE Transactions on Industrial Electronics*, 2025. doi: 10.1109/TIE.2025.3591680.

A Extended Related Work

A.1 Observation Representations

We represent observations as tuples that not only contain measurements but also the information needed to locate and interpret them within a reconstruction query. Related representation strategies arise in multivariate time-series analysis, particularly in imputation for missing data and interpolation of irregularly sampled variables. Many algorithms benefit from explicitly representing observational status, temporal coordinates, and variable identity alongside observed values. Che et al. (2018) use a triple to represent samples in multivariate time series with missing observations. Their representation consists of value, observation mask, and elapsed time since the most recent available value of the variable, with the mask taking value 1 if the observation is available and 0 otherwise. Their architecture, based on gated recurrent units (GRUs), was evaluated on synthetic and medical time series across several classification tasks. Yoon et al. (2019) addressed the problem of data reconstruction via a Multi-directional Recurrent Neural Network (M-RNN) that includes an interpolation block and an imputation block, both trained simultaneously to minimize reconstruction error on observed measurements withheld as inputs. Their interpolation block represents each observation as a triple of value, observation mask, and elapsed time since the last measurement of the variable, with the mask taking value 0 if the observation is missing and 1 otherwise.

Temporal coordinates constitute another important component of observation representations. Ma et al. (2019) applied cross-dimensional self-attention (CDSA) to address the missing-value imputation problem in multivariate time series represented along three dimensions: time, geo-location, and measurement. Shukla & Marlin (2021) introduce the Multi-Time Attention Network (mTAN), representing each irregularly sampled variable as a pair of observation times and corresponding values. Learnable time embeddings within a VAE-based encoder-decoder architecture map sparse observations to a fixed-length latent representation from which values can be reconstructed at arbitrary times. Tipirneni & Reddy (2022) use a lifted token representation for a transformer-based architecture that is related to ours. Their tokens comprise three components: time, feature variable, and observation value, with continuous values mapped to learnable embeddings. They demonstrate the suitability of this representation for clinical records in a mortality prediction task and use self-supervised learning to improve predictive performance. Similarly, Yalavarthi et al. (2023) introduce Tripletformer, representing observations as triples of time, channel, and observed value. In contrast to mTAN, which uses separate time-attention models for individual channels, Tripletformer jointly encodes observations from all channels, allowing attention to learn inter- and intra-channel interactions simultaneously. Their experiments show that this joint encoding improves interpolation performance over the channel-wise approach. Tripletformer performs probabilistic interpolation, producing a predictive distribution for each queried observation. Du et al. (2023) introduce the Self-Attention-based Imputation for Time Series (SAITS) model, representing input values together

with a binary observation mask. SAITS combines a masked imputation task, in which observed values are withheld from the input and reconstructed, with an observed reconstruction task for values that remain available. In Muñoz Gutiérrez et al. (2024), a transformer autoencoder represented masked inputs using out-of-distribution values, while self-supervised learning relied on structured masking patterns defining pretext tasks for prediction, retrodiction, and signal inference in simulated electric-motor dynamics. These approaches demonstrate the utility of representing values together with observational status, temporal coordinates, and variable identity. Our representation builds on these principles while lifting masking into an explicit epistemic state that distinguishes the roles of observations within reconstruction queries at learning and inference time.

A.2 Self-Supervised Reconstruction

Reconstruction provides a natural mechanism for self-supervised learning by withholding part of the available information and using it as a learning target. Vincent et al. (2008) introduced denoising autoencoders, training an autoencoder to reconstruct an original input from a partially corrupted version. Rather than learning an identity mapping, the reconstruction task encourages the model to capture dependencies in the data that allow corrupted information to be recovered from the remaining evidence. Vincent et al. (2010) further interpret denoising geometrically under the manifold assumption, with corruption perturbing inputs away from the data manifold and reconstruction learning to map them back toward it.

Masked modeling extends this principle by explicitly selecting which information is provided as input and which information is used as a reconstruction target. It became a prominent self-supervised learning strategy with BERT Devlin et al. (2019), which predicts tokens withheld from their surrounding context and represents masked positions using the special `[MASK]` token. Bao et al. (2022) introduce BEiT, a transformer-based approach for self-supervised representation learning in which image patches are masked and the model is trained to predict visual tokens corresponding to the masked patches. Zerveas et al. (2021) apply masked self-supervised learning to multivariate time series. Their transformer encoder is pretrained by reconstructing values withheld using randomly generated masks with contiguous masked segments whose lengths follow a geometric distribution. Representations learned using this strategy improved performance over fully supervised learning across several multivariate time-series regression and classification benchmarks. Dong et al. (2023) develop SimMTM, a robust pretraining framework for time series that reconstructs an original series from multiple randomly masked perturbations whose representations are combined using cosine similarity and nonlinear regression. Bachmann et al. (2024) extend masked modeling to a massively multimodal setting with the 4M framework, representing different modalities through sequences of tokens and training a single model on arbitrary subsets of input and target modalities. Their any-to-any formulation allows the same model to perform different inference tasks according to the information made available as input and selected as output. In Muñoz Gutiérrez et al. (2024), a transformer autoencoder used structured masking patterns to define self-supervised pretext tasks for prediction, retrodiction, and signal inference in simulated electric-motor dynamics. The approach, however, does not provide a mechanism for controlling the combinatorial growth of possible masking configurations. In contrast, our self-supervised learning strategy builds on these principles but structures the withheld information according to reconstruction queries that anticipate the epistemic acts required at inference time.

A.3 Bridging Model-Based and Data-Driven Approaches

Early hybrid modeling approaches explored how neural networks could complement rather than replace models derived from prior knowledge. Haesloop & Holt (1990) demonstrated this principle by complementing linear models of chemical processes with early neural network models. They modified a feed-forward network to decompose linear and nonlinear components by introducing a linear mapping directly from the inputs to the outputs. The resulting architecture significantly improved upon linear approximations when the process deviated from the model operating point, while retaining an identifiable linear component. Psichogios & Ungar (1992) demonstrated the advantages of hybrid bioreactor modeling by combining a traditional first-principles model of a dynamical process with an artificial neural network that estimates unknown process parameters. Their results demonstrated good predictive accuracy while retaining a structured model that was easier to analyze than

an unstructured neural network; the approach, however, relies on prior knowledge of the relevant process dependencies.

Karpatne et al. (2017) systematize the integration of scientific knowledge and data-driven learning under the paradigm of theory-guided data science (TGDS). They organize TGDS into five approaches for incorporating scientific knowledge into data-driven learning, identifying *hybrid modeling* as one strategy for exploiting the complementary strengths of theory-based and data-driven models, particularly when available scientific models are incomplete or imperfect. From the TGDS perspective, model performance is characterized as being directly proportional to accuracy, simplicity, and consistency with existing scientific knowledge, with the latter criterion being the focus of their essay. Wang et al. (2017) synthesize a hybrid approach for simulating turbulent flows in which Reynolds-averaged Navier-Stokes (RANS) models approximate the flow dynamics and a random-forest algorithm, whose input is a vector of ten invariant properties of the flow, predicts model discrepancies. Their resulting hybrid, termed a *physics-informed machine learning* approach, improves the reference model and, in the square-duct case, demonstrates better Reynolds-stress predictions than the advanced Launder-Gibson Reynolds-stress transport model (RSTM) baseline across Reynolds numbers. A closely related line of research developed *field inversion and machine learning* (FIML), in which data are used to infer model-form discrepancies and machine learning is subsequently used to construct corrective model forms Duraisamy et al. (2015); Tracey et al. (2015); Parish & Duraisamy (2016); Singh & Duraisamy (2016); Singh et al. (2017). The objective is ultimately to improve the predictive capability of the underlying physics-based model by incorporating a data-driven model of discrepancies learned across benchmark problems. These approaches exploit substantial problem-specific knowledge of the governing physics, including physical constraints, invariant properties, and engineered features known to characterize relevant aspects of the flow dynamics. Our objective differs in that the learned complement need not constitute a transferable correction to the reference model, but may remain grounded in empirical evidence associated with a particular physical realization of the modeled system and its operating conditions. Such specificity is necessary for monitoring, where small departures from observed behavior provide evidence of changes in system health.

The term physics-informed machine learning (PIML) is more general than the approach followed by Wang et al. (2017). Karniadakis et al. (2021) provides an accessible overview of the field. PIML addresses the integration of physical theories and laws with evidence from measured data for modeling physical systems, particularly when available knowledge and data are insufficient for either conventional mathematical modeling or purely data-driven methods to be effective on their own. Machine-learning models are often artificial neural networks, into which physical knowledge can be incorporated using assorted techniques, including observational biases through data preparation, inductive biases captured in the architecture of the model, and learning biases whereby physical knowledge is encouraged through soft constraints, often expressed as terms in the loss function. In the latter case, particularly in physics-informed neural networks, the loss can include both an observational residual and a physics-based residual. In contrast, our approach does not seek to make the machine-learning model learn the dynamics of the physical system. Instead, the reference mathematical model remains the predictive backbone, while learning is restricted to the unexplained part of its prediction, namely its residual with respect to observations. The authors also report on hybrid methods, among which a subfamily closely relates to ours in objective and model decomposition. Rackauckas et al. (2020) developed the universal differential equation (UDE) formalism, in which learnable components are embedded within differential equations to complement known dynamics, together with numerical methods and a software framework for efficiently training the resulting hybrid models. Addressing the stability of hybrid model realizations in structural dynamics, Xu et al. (2021a) developed the symmetric positive definite neural network (SPD-NN). They showed that accurate machine-learning approximations of constitutive relations may nevertheless lead to unstable dynamics when coupled with conventional structural dynamics models, and imposed symmetry and positive definiteness by construction. Huang et al. (2020) hybridize mechanistic and learned models in solid mechanics by using data-driven components to represent constitutive relations within a numerical PDE model. The learned component is then integrated into the governing model used in the subsequent forward problem. Xu et al. (2021b) extend this approach to learn viscoelasticity models, embedding a learned constitutive relation in the computational graph of the numerical PDE solver and combining automatic differentiation with implicit-function differentiation to learn it from indirect displacement observations.

For electrical machines, learned components have been used to complement established electromagnetic and control models to account for phenomena that are unknown, poorly characterized by available models, or highly nonlinear. Huang et al. (2024) use two RBF network instances to capture highly nonlinear core-loss compensation functions and reference virtual flux. Dedicated offline experimental test protocols are used to generate suitable input-output pairs for learning these relations. For online estimation, the hybrid model combines the data-driven predictions with model-based transformations to estimate the temperature of the permanent magnets. In contrast, WRAPPER learns from the residuals of a prior predictor without requiring access to its internal model structure. Lu et al. (2024) develop a hybrid model for PMSMs, where a neural network is trained to estimate aggregate model-mismatch terms capturing the highly nonlinear effects of core loss and saturation. Their approach derives a simplified low-speed model from a nonlinear mechanistic steady-state machine model by neglecting core loss. Expressions to compute model mismatch are then derived and used to teach the feedforward network to estimate these residuals across different operating conditions from measured speed and dq-axis currents. Once trained, the network estimates can be used to compensate model mismatch in optimal control. Liu et al. (2024) develop a hybrid iron loss model for PMSMs that combines a high-fidelity mechanistic model with data-driven measurements assimilation from a laboratory PMSM. A detailed multiphysics model accounting for operating and environmental effects is used to generate a synthetic dataset for pretraining a CNN to predict iron loss. The pretrained network is subsequently fine-tuned using experimental data, allowing the data-driven model to adapt the mechanistically acquired knowledge to measurements from the physical system. Wang et al. (2026) develop a hybrid model for sensorless torque estimation in ultrasonic motors with two levels of data-driven participation. *PhysCorrNet* estimates an effective damping coefficient from selected and properly weighted engineered features. This estimate is used within a mechanistic model to compute torque. In parallel, *ResCorrNet* learns the residual between the mechanistic torque prediction and measurements, and its correction is added to the mechanistic prediction to obtain the final estimate.

A.4 Residual Learning for Fault Diagnosis

In residual-based fault diagnosis, discrepancies between observations and model predictions provide evidence of faults. Structural analysis (SA) uses dependencies in the system's dynamical equations to design residuals whose response patterns can distinguish among fault sources. For those patterns to be informative, residuals must remain sensitive to faults of interest while suppressing nominal dynamics and tolerating epistemic model uncertainty and aleatoric system variability. When these dependencies are known, but the corresponding analytical expressions are unavailable, residual learning can supply the missing relationships from data.

Pulido et al. (2019) derive Possible Conflicts and Minimal Evaluable Models to identify distinct residual generators, then constrain the connections of state-space neural networks according to those dependencies. The networks learn the remaining mappings from nominal data. Similarly, Jung (2019) uses structural redundancy and causal matchings to construct recurrent neural residual generators for detecting and localizing faults without requiring examples of each fault during training. Signal selection, dynamic realization, and generalization also shape diagnostic residuals. Jung & Axelsson (2024) use the intrinsic dimension of nominal observations to select signal subsets. Mohammadi et al. (2025) construct residual models that integrate predicted state derivatives, train them over extended open-loop horizons to improve prediction stability, and show how equation form affects structural residual construction. Allansson et al. (2024) find that a neural residual model outperforms a linear state-space model within its training regime but generalizes less well to an unseen healthy operating condition, illustrating the risk of mistaking model mismatch for fault evidence. Jung & Westny (2026) integrate neural-ODE state derivatives to predict nominal behaviour and use ensemble dispersion to set uncertainty-aware fault-detection thresholds. In this work, WRAPPER combines a physics-based residual with a correction learned by a transformer from healthy data. Subtracting the learned correction from the original residual brings the diagnostic residual toward zero under healthy operation.

B DEFINITIONS

Definition 1 (ODE Formal Model)**.** *An* ODE formal model *is a tuple*

$$F = (\mathbb{X}, \mathbb{U}, \mathbb{Y}, f, h),$$

where $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ *is an open state space,* $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ *is an input space, and* $\mathbb{Y} \subseteq \mathbb{R}^{n_y}$ *is an observation space.*

The evolution map

$$f : \mathbb{R} \times \mathbb{X} \times \mathbb{U} \to \mathbb{R}^{n_x}$$

is continuous and locally Lipschitz continuous in its state argument, and the observation map

$$h : \mathbb{R} \times \mathbb{X} \times \mathbb{U} \to \mathbb{Y}$$

is continuous.

An input trajectory of F *is a continuous function*

$$\boldsymbol{u} : \mathbb{R} \to \mathbb{U}.$$

Definition 2 (Cauchy Problem)**.** *Let*

$$F = (\mathbb{X}, \mathbb{U}, \mathbb{Y}, f, h)$$

be an ODE formal model, let $\boldsymbol{u}$ *be an input trajectory of* F*, let* $t_a \in \mathbb{R}$*, and let* $\boldsymbol{x}_a \in \mathbb{X}$*.*

The Cauchy problem

$$\mathscr{C} = (f, \boldsymbol{u}, t_a, \boldsymbol{x}_a)$$

consists of finding a continuously differentiable trajectory

$$\gamma : J \to \mathbb{X},$$

defined on its maximal interval of existence $J \subseteq \mathbb{R}$ *containing* t_a*, such that*

$$\dot{\gamma}(t) = f(t, \gamma(t), \boldsymbol{u}(t)), \qquad \gamma(t_a) = \boldsymbol{x}_a.$$

Definition 3 (Solution Operator)**.** *Let* $\mathfrak{C}$ *denote the class of Cauchy problems admitting a unique solution. The* solution operator $\mathcal{S}$ *maps each* $\mathscr{C} \in \mathfrak{C}$ *to its unique solution trajectory* γ*, that is,*

$$\mathcal{S}(\mathscr{C}) = \gamma.$$

Definition 4 (Signal)**.** *Let* $\mathbb{T} \subseteq \mathbb{R}$ *and let* $\mathbb{X}$ *be a set. A* signal *over* $\mathbb{X}$ *with time domain* $\mathbb{T}$ *is a mapping*

$$s : \mathbb{T} \to \mathbb{X},$$

with value at time $t \in \mathbb{T}$ *denoted by* $s(t)$ *or* s_t*.*

Definition 5 (Multivariate Signal)**.** *Let* $n \in \mathbb{N}$*,* $n > 1$*, and let*

$$\mathbb{X} = X_1 \times \cdots \times X_n.$$

A signal

$$\boldsymbol{s} : \mathbb{T} \to \mathbb{X}$$

is a multivariate signal *if it can be written as*

$$\boldsymbol{s} = (s_1, \ldots, s_n),$$

where

$$s_i : \mathbb{T} \to X_i, \qquad i = 1, \ldots, n,$$

are its component signals.

Definition 6 (Sampling Support)**.** *Let* $\mathbb{T} \subseteq \mathbb{R}$ *be a time domain and let*

$$\sigma : \mathbb{T} \to \{\top, \bot\}$$

be a sampling predicate. The sampling support *of* $\mathbb{T}$ *under* σ *is*

$$\mathbb{T}[\![\sigma]\!]_{\downarrow} = \{\, t \in \mathbb{T} \mid \sigma(t) = \top \,\}.$$

Definition 7 (Signal Sampling)**.** *Let*

$$s : \mathbb{T} \to X$$

be a signal and let

$$S = \mathbb{T}[\![\sigma]\!]_{\downarrow}$$

be a sampling support. The sampling *of s on S is denoted by*

$$s \circledcirc S : S \to X,$$

and is defined by

$$s \circledcirc S := s|_S.$$

Definition 8 (Boundary-Complete Sampling Support)**.** *Let*

$$S = \mathbb{T}[\![\sigma]\!]_{\downarrow}$$

be a sampling support on a bounded closed time domain

$$\mathbb{T} = [t_a, t_b] \subseteq \mathbb{R}.$$

The sampling support S is boundary-complete *on $\mathbb{T}$ if*

$$\sigma(t_a) = \sigma(t_b) = \top.$$

Equivalently,

$$t_a, t_b \in S.$$

Definition 9 (Canonical Lifting)**.** *Let*

$$S = \mathbb{T}[\![\sigma]\!]_{\downarrow}$$

be a finite sampling support. Writing

$$S = \{t_1, \dots, t_n\}, \qquad t_1 < \cdots < t_n,$$

the canonical lifting *of S is*

$$[*S] = \begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix} \in \mathbb{R}^n.$$

Definition 10 (Vectorial Function Application)**.** *Let*

$$f : A \to \mathbb{R}^n$$

and let

$$\boldsymbol{a} = [a_1 \quad \cdots \quad a_m]^{\mathsf{T}} \in A^m.$$

The vectorial application *of f to $\boldsymbol{a}$ is defined as*

$$f(\boldsymbol{a}) := \begin{bmatrix} f(a_1)^{\mathsf{T}} \\ \vdots \\ f(a_m)^{\mathsf{T}} \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

Definition 11 (Locally Finite Sampling Support)**.** *Let*

$$S = \mathbb{T}[\![\sigma]\!]_{\downarrow}$$

be a sampling support. The support S is locally finite *if*

$$S \cap I$$

is finite for every bounded interval $I \subseteq \mathbb{R}$.

Definition 12 (Time Window Operator)**.** *Let S be a locally finite sampling support and let $t_k \in S$. For a* context length *$L \in \mathbb{N}$, provided that t_k has at least $L - 1$ predecessors in S, the* time window operator W *is defined by*

$$\mathsf{W}(S; t_k, L) := \{t_{k-L+1}, \dots, t_k\}.$$

Definition 13 (Discrete Signal)**.** *Let*

$$S = \mathbb{T}[\![\sigma]\!]_{\downarrow}$$

be a locally finite sampling support. A discrete signal *over* $\mathbb{X}$ *is a signal*

$$s : S \to \mathbb{X}.$$

Equivalently, if

$$s : \mathbb{T} \to \mathbb{X}$$

is a signal, then its restriction to a locally finite sampling support $S = \mathbb{T}[\![\sigma]\!]_{\downarrow}$*,*

$$s|_S : S \to \mathbb{X},$$

is a discrete signal.

Definition 14 (Signal Extension Class)**.** *Let*

$$s : S \to \mathbb{X}$$

be a discrete signal with $S \subseteq \mathbb{T}$*. The* signal extension class *of* s *over* $\mathbb{T}$ *is*

$$\mathfrak{E}_{\mathbb{T}}(s) = \{\tilde{s} : \mathbb{T} \to \mathbb{X} \mid \tilde{s}|_S = s\}.$$

An element of $\mathfrak{E}_{\mathbb{T}}(s)$ *is called a* signal extension *of* s*.*

Definition 15 (Admissible Extension)**.** *Let*

$$s : S \to \mathbb{X}$$

be a discrete signal, and let

$$\alpha : \mathfrak{E}_{\mathbb{T}}(s) \to \{\top, \bot\}$$

be an admissibility predicate. The admissible extension class *of* s *over* $\mathbb{T}$ *is*

$$\mathfrak{E}_{\mathbb{T}}(s)\,[\![\alpha]\!] = \{\tilde{s} \in \mathfrak{E}_{\mathbb{T}}(s) \mid \alpha(\tilde{s}) = \top\}.$$

An element of $\mathfrak{E}_{\mathbb{T}}(s)\,[\![\alpha]\!]$ *is called an* admissible extension *of* s*.*

Definition 16 (Signal Extension Problem)**.** *Let* $s : S \to X$ *be a signal with* $S \subseteq T$*, and let*

$$\alpha : \mathfrak{E}_T(s) \to \{\top, \bot\}$$

be an admissibility predicate. A signal extension problem *is the tuple*

$$\mathscr{E} = \big(\mathfrak{E}_T(s), \alpha\big),$$

and consists of determining

$$\mathfrak{E}_T(s)\,[\![\alpha]\!].$$

An element of $\mathfrak{E}_T(s)\,[\![\alpha]\!]$ *is a* solution *of* $\mathscr{E}$*. The problem admits a solution if*

$$\mathfrak{E}_T(s)\,[\![\alpha]\!] \neq \varnothing,$$

and admits a unique solution if

$$|\mathfrak{E}_T(s)\,[\![\alpha]\!]| = 1.$$

Definition 17 (Signal Extension Solution Operator)**.** *Let*

$$\mathfrak{E}_{\mathbb{T}}(s)\,[\![\alpha]\!] = \{\tilde{s}\}.$$

The signal extension solution operator $\mathcal{E}$ *maps each uniquely solvable signal extension problem* $\mathscr{E}$ *to its unique admissible extension, that is,*

$$\mathcal{E}(\mathscr{E}) = \tilde{s}.$$

Definition 18 (Support Floor and Ceiling)**.** *Let*

$$S = \mathbb{T}[\![\sigma]\!]_{\downarrow}$$

be a locally finite, boundary-complete sampling support on the bounded closed time domain

$$\mathbb{T} = [t_a, t_b].$$

For every $t \in \mathbb{T}$*, the* support floor *and* support ceiling *are defined by*

$$\lfloor t \rfloor_S = \max\{\,\tau \in S \mid \tau \leq t\,\}, \qquad \lceil t \rceil_S = \min\{\,\tau \in S \mid t \leq \tau\,\},$$

respectively.

Definition 19 (Bracketing Interval)**.** *Let $S \subseteq \mathbb{T}$ be a locally finite, boundary-complete sampling support. For $t \in \mathbb{T}$, the* bracketing interval *of t is*

$$\mathcal{B}_S(t) = [\lfloor t \rfloor_S , \lceil t \rceil_S].$$

Example 20 (Linear Interpolation)**.** *Let*

$$s : S \to \mathbb{R}$$

be a discrete signal with locally finite, boundary-complete sampling support $S \subseteq \mathbb{T}$, and let

$$\alpha : \mathfrak{E}_{\mathbb{T}}(s) \to \{\top, \bot\}$$

require an extension to be continuous and affine on every nondegenerate bracketing interval. Then

$$\mathfrak{E}_{\mathbb{T}}(s)\,[\![\alpha]\!] = \{\tilde{s}\},$$

where

$$\tilde{s}(t) = \begin{cases} s(t), & t \in S, \\ \dfrac{\lceil t \rceil_S - t}{\lceil t \rceil_S - \lfloor t \rfloor_S}\, s(\lfloor t \rfloor_S) + \dfrac{t - \lfloor t \rfloor_S}{\lceil t \rceil_S - \lfloor t \rfloor_S}\, s(\lceil t \rceil_S), & t \in \mathbb{T} \setminus S. \end{cases}$$

Hence, for the signal extension problem

$$\mathscr{E} = \big(\mathfrak{E}_{\mathbb{T}}(s)\,, \alpha\big),$$

the signal extension solution operator gives

$$\mathcal{E}(\mathscr{E}) = \tilde{s}.$$

C State Reconstruction Problem

We consider an ODE formal model

$$F = (\mathbb{X}, \mathbb{U}, \mathbb{Y}, f, h),$$

with the formal definitions of ODE models, signals, sampling, signal extensions, and Cauchy problems given in Appendix B. Let

$$\boldsymbol{x} : \mathbb{T} \to \mathbb{X}, \qquad \boldsymbol{u} : \mathbb{T} \to \mathbb{U}$$

be state and input signals, respectively, and let

$$S = \mathbb{T}[\![\sigma]\!]_{\downarrow} = \{t_0, \ldots, t_N\}, \qquad t_0 < \cdots < t_N,$$

be a finite sampling support. Let $\widetilde{\boldsymbol{u}}$ be an admissible extension of the input signal sampled on S.

Definition 21 (State Reconstruction Problem)**.** *A* state reconstruction problem *is the tuple*

$$\mathscr{R} = (F, S, \widetilde{\boldsymbol{u}}),$$

where F is an ODE formal model, S is a finite sampling support, and

$$\widetilde{\boldsymbol{u}} \in \mathfrak{E}_{\mathbb{T}}(\boldsymbol{u} \circledcirc S)\,[\![\alpha]\!].$$

For each $k = 1, \ldots, N$, let

$$\boldsymbol{x}_{k-1} = \big(\boldsymbol{x} \circledcirc S\big)(t_{k-1}).$$

The sampled state $\boldsymbol{x}_{k-1}$ anchors the Cauchy problem

$$\mathscr{C}_{k-1} = (f, \widetilde{\boldsymbol{u}}, t_{k-1}, \boldsymbol{x}_{k-1}).$$

If

$$\gamma_{k-1} = \mathcal{S}(\mathscr{C}_{k-1})$$

is its solution, then the model-based state reconstruction at t_k is

$$\widehat{\boldsymbol{x}}_k = \gamma_{k-1}(t_k) = \mathcal{S}(\mathscr{C}_{k-1})(t_k).$$

Definition 22 (State Reconstruction Solution Operator)**.** *Let $\mathfrak{R}$ denote the class of state reconstruction problems for which the induced Cauchy problems admit unique solutions over their corresponding sampling intervals. The* state reconstruction solution operator *$\mathcal{R}$ maps each*

$$\mathscr{R} \in \mathfrak{R}$$

to the reconstructed state signal

$$\widehat{\boldsymbol{x}} : S \setminus \{t_0\} \to \mathbb{X},$$

defined by

$$\widehat{\boldsymbol{x}}(t_k) = \mathcal{S}(\mathscr{C}_{k-1})(t_k), \qquad k = 1, \ldots, N.$$

That is,

$$\mathcal{R}(\mathscr{R}) = \widehat{\boldsymbol{x}}.$$

Each reconstruction is independently anchored by the sampled state rather than by the preceding reconstructed state. The resulting sequence therefore characterizes local model-consistent reconstruction between observations rather than free-running simulation.

In health-monitoring applications, the reference model characterizes the expected evolution of the monitored system under the observed operating conditions. The corresponding state residual at discrete time t_k is

$$\boldsymbol{r}_k = \boldsymbol{x}_k - \widehat{\boldsymbol{x}}_k,$$

which measures the departure of the observed state from the state reconstructed by the reference model over the corresponding sampling interval. Re-anchoring each reconstruction at the sampled state prevents discrepancies from accumulating across intervals and associates each residual with the local evolution from the available evidence.

D Query Builder

The Query Builder (QB) constructs the model-ready reconstruction query from the residual and input contexts. Its construction comprises normalization of the numerical evidence followed by the epistemic query construction described below.

D.1 Normalization

Normalization is fitted exclusively from the training subset of each fold and subsequently held fixed. For each residual and input component, let $\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r$ and $\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u$ denote the componentwise sample means and standard deviations estimated over the corresponding training contexts. The standardization operators are

$$\mathcal{N}_r(\mathbf{C}_r) = (\mathbf{C}_r - \boldsymbol{\mu}_r) \oslash \boldsymbol{\sigma}_r, \qquad \mathcal{N}_u(\mathbf{C}_u) = (\mathbf{C}_u - \boldsymbol{\mu}_u) \oslash \boldsymbol{\sigma}_u,$$

where subtraction and division are applied componentwise across the rows of each context matrix. We write

$$\mathcal{N}(\mathbf{C}_r, \mathbf{C}_u) = \big(\mathcal{N}_r(\mathbf{C}_r), \mathcal{N}_u(\mathbf{C}_u)\big).$$

The statistics defining $\mathcal{N}$ are reused unchanged for the validation and test subsets of the corresponding fold and during inference.

The Transformer reconstructs the residual in standardized coordinates; we denote this output by $\widehat{\boldsymbol{r}}^{\,\mathrm{n}}$. For each residual component $i = 1, \ldots, n_x$, the inverse transformation is

$$\widehat{r}_i = \mathcal{N}_{r,i}^{-1}(\widehat{r}_i^{\,\mathrm{n}}) = \mu_{r,i} + \sigma_{r,i}\widehat{r}_i^{\,\mathrm{n}}.$$

Collecting the components yields the reconstructed residual $\widehat{\boldsymbol{r}}$ in its original coordinates before it is passed to Governance. This numerical transformation and its inverse are abstracted from Fig. 1, where $\widehat{\boldsymbol{r}}$ denotes the reconstructed residual in its original coordinates.

D.2 TARGET SPECIFICATION

Let

$$\mathbf{C}_r \in \mathbb{R}^{L\times n_x}, \qquad \mathbf{C}_u \in \mathbb{R}^{L\times n_u}$$

be residual and input context matrices. Their respective address spaces are

$$\mathbb{A}_r = \mathbb{N}_L \times \mathbb{N}_{n_x}, \qquad \mathbb{A}_u = \mathbb{N}_L \times \mathbb{N}_{n_u}.$$

Definition 23 (Target specification)**.** *A* target specification *for* $(\mathbf{C}_r, \mathbf{C}_u)$ *is a pair*

$$T = (T_r, T_u), \qquad T_r \subseteq \mathbb{A}_r, \qquad T_u \subseteq \mathbb{A}_u.$$

An address $(\ell, i) \in T_r$ *identifies the entry* $(\mathbf{C}_r)_{\ell i}$ *as a reconstruction target, while an address* $(\ell, j) \in T_u$ *identifies the entry* $(\mathbf{C}_u)_{\ell j}$ *as a reconstruction target.*

Example 24. *Let* $L = 3$*,* $n_x = 2$*, and* $n_u = 1$*, with context matrices*

$$\mathbf{C}_r = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \\ r_{31} & r_{32} \end{bmatrix}, \qquad \mathbf{C}_u = \begin{bmatrix} u_{11} \\ u_{21} \\ u_{31} \end{bmatrix}.$$

For the target specification

$$T_r = \{(2,1),(3,2)\}, \qquad T_u = \{(1,1)\},$$

the entries r_{21} *and* r_{32} *of* $\mathbf{C}_r$ *and* u_{11} *of* $\mathbf{C}_u$ *are reconstruction targets. Equivalently, the target locations can be displayed as*

$$\mathbf{C}_r = \begin{bmatrix} r_{11} & r_{12} \\ \boxed{r_{21}} & r_{22} \\ r_{31} & \boxed{r_{32}} \end{bmatrix}, \qquad \mathbf{C}_u = \begin{bmatrix} \boxed{u_{11}} \\ u_{21} \\ u_{31} \end{bmatrix}.$$

D.3 EPISTEMIC TOKENIZATION

For a context of length $L > 1$, the relative temporal coordinate of row $\ell \in \mathbb{N}_L$ is

$$t_r(\ell) = -1 + 2\frac{\ell - 1}{L - 1},$$

which maps the ordered context positions uniformly to $[-1, 1]$, independently of their absolute observation times. This deterministic coordinate encoding is distinct from the statistical normalization of observation values. For $L = 5$, as used in this work,

$$t_r(1), \ldots, t_r(5) = -1, -\frac{1}{2}, 0, \frac{1}{2}, 1.$$

The present realization uses the binary epistemic state space

$$\mathcal{E} = \{\mathsf{target}, \mathsf{available}\},$$

encoded as

$$\epsilon(\mathsf{target}) = 0, \qquad \epsilon(\mathsf{available}) = 1.$$

Definition 25 (Epistemic token)**.** *Let* $T = (T_r, T_u)$ *be a target specification. For* $s \in \{r, u\}$*, let* $c^s_{\ell j}$ *denote the normalized context value at address* $(\ell, j) \in \mathbb{A}_s$*. Given* T*, the corresponding epistemic tuple is*

$$\tau^s_{\ell j} = (v^s_{\ell j}, t_r(\ell), e^s_{\ell j})_j,$$

where

$$v^s_{\ell j} = \begin{cases} 0, & (\ell, j) \in T_s, \\ c^s_{\ell j}, & (\ell, j) \notin T_s, \end{cases} \qquad e^s_{\ell j} = \begin{cases} 0, & (\ell, j) \in T_s, \\ 1, & (\ell, j) \notin T_s. \end{cases}$$

The subscript j *denotes signal identity, retained structurally by the token sequence.*

The corresponding epistemic token is the numerical encoding

$$\mathbf{z}^s_{\ell j} = \eta(\tau^s_{\ell j}) = \begin{bmatrix} v^s_{\ell j} \\ t_r(\ell) \\ e^s_{\ell j} \end{bmatrix} \in \mathbb{R}^3.$$

Targeting withholds the observation value while preserving its relative temporal coordinate and signal identity. The explicit epistemic state distinguishes a withheld target from an available observation whose normalized value is also zero.

In this work, temporal coordinates and signal identities are assumed known. Richer reconstruction domains may allow temporal coordinates to be queried, potentially requiring explicit signal identity and constraints ensuring that reconstruction queries remain well posed; such extensions are outside the scope of this work.

Definition 26 (Query construction)**.** *Let* $(\mathbf{C}_r^{\mathrm{n}}, \mathbf{C}_u^{\mathrm{n}}) = \mathcal{N}(\mathbf{C}_r, \mathbf{C}_u)$ *and* $n_c = n_x + n_u$*. For each* $\ell \in \mathbb{N}_L$*, let*

$$\mathbf{Z}_\ell = \begin{bmatrix} (\mathbf{z}^r_{\ell 1})^\top \\ \vdots \\ (\mathbf{z}^r_{\ell n_x})^\top \\ (\mathbf{z}^u_{\ell 1})^\top \\ \vdots \\ (\mathbf{z}^u_{\ell n_u})^\top \end{bmatrix} \in \mathbb{R}^{n_c \times 3}.$$

The query-construction operator $\mathcal{Q}_T$ *is*

$$\mathcal{Q}_T(\mathbf{C}_r^{\mathrm{n}}, \mathbf{C}_u^{\mathrm{n}}) = \begin{bmatrix} \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_L \end{bmatrix} = \mathbf{C} \in \mathbb{R}^{L n_c \times 3}.$$

Definition 27 (Target policies)**.** *Let*

$$\mathbb{A}_r = \mathbb{N}_L \times \mathbb{N}_{n_x}, \qquad \mathbb{A}_u = \mathbb{N}_L \times \mathbb{N}_{n_u},$$

and let

$$\mathfrak{T} = \mathcal{P}(\mathbb{A}_r) \times \mathcal{P}(\mathbb{A}_u)$$

be the space of target specifications.

A target-selection policy *is a predicate*

$$\pi_s : \left(\mathbb{R}^{L \times n_x} \times \mathbb{R}^{L \times n_u}\right) \times \mathfrak{T} \to \{\top, \bot\},$$

where

$$\pi_s((\mathbf{C}_r, \mathbf{C}_u), T) = \top$$

states that T *is selected for the contexts* $(\mathbf{C}_r, \mathbf{C}_u)$*. A* target-admissibility policy *is a predicate*

$$\pi_\alpha : \left(\mathbb{R}^{L \times n_x} \times \mathbb{R}^{L \times n_u}\right) \times \mathfrak{T} \to \{\top, \bot\},$$

where

$$\pi_\alpha((\mathbf{C}_r, \mathbf{C}_u), T) = \top$$

states that T *is admissible for the contexts* $(\mathbf{C}_r, \mathbf{C}_u)$*.*

The corresponding target policy *is the pair*

$$\pi = (\pi_s, \pi_\alpha).$$

For the target specifications considered in this work, admissibility requires at least one target and at least one context value to remain available. Thus, for $T = (T_r, T_u)$, we use

$$\pi_\alpha((\mathbf{C}_r, \mathbf{C}_u), T) = \left[T_r \neq \varnothing \vee T_u \neq \varnothing\right] \wedge \left[T_r \neq \mathbb{A}_r \vee T_u \neq \mathbb{A}_u\right].$$

Definition 28 (Query Controller)**.** *Let*

$$\mathbf{C}_r \in \mathbb{R}^{L \times n_x}, \qquad \mathbf{C}_u \in \mathbb{R}^{L \times n_u},$$

and let $\pi = (\pi_s, \pi_\alpha)$ *be a target policy. A* Query Controller *is the tuple*

$$Q_C = (\mathbf{C}_r, \mathbf{C}_u, \pi).$$

Table 6: Main physical characteristics of the experimental IPMSM.

Quantity	Value
Motor type	Interior PMSM
Phases	3
Pole pairs	21
Stator slots	36
Rotor length	$100\,\mathrm{mm}$
Rotor diameter	$81\,\mathrm{mm}$
Permanent magnets	N38
Magnet dimensions	$7 \times 1.5 \times 24\,\mathrm{mm}$
Magnet pockets	42

A target specification $T \in \mathfrak{T}$ is selected and admissible when

$$\pi_s((\mathbf{C}_r, \mathbf{C}_u), T) = \top \qquad \textit{and} \qquad \pi_\alpha((\mathbf{C}_r, \mathbf{C}_u), T) = \top.$$

For each selected and admissible target specification T, the Query Controller instantiates the corresponding query-construction mapping $\mathcal{Q}_T$.

For each $s \in \{r, u\}$ and $(\ell, j) \in \mathbb{A}_s$, the mapping $\mathcal{Q}_T$ is constructed from the component mapping

$$c^s_{\ell j} \longmapsto \tau^s_{\ell j} = (v^s_{\ell j}, t_r(\ell), e^s_{\ell j})_j \overset{\eta}{\longmapsto} \mathbf{z}^s_{\ell j},$$

where

$$v^s_{\ell j} = \begin{cases} 0, & (\ell, j) \in T_s, \\ c^s_{\ell j}, & (\ell, j) \notin T_s, \end{cases} \qquad e^s_{\ell j} = \begin{cases} 0, & (\ell, j) \in T_s, \\ 1, & (\ell, j) \notin T_s. \end{cases}$$

The encoded tokens $\mathbf{z}^s_{\ell j}$ are assigned to output rows in temporal-major order, thereby defining $\mathcal{Q}_T$.

Continuing Example 24, suppose for simplicity that normalization leaves the displayed context values unchanged. Since $L = 3$, the relative temporal coordinates are -1, 0, and 1. For its target specification, the Query Controller produces the corresponding mapping $\mathcal{Q}_T$, whose application can be displayed as

$$\left(\begin{bmatrix} r_{11} & r_{12} \\ \boxed{r_{21}} & r_{22} \\ r_{31} & \boxed{r_{32}} \end{bmatrix}, \begin{bmatrix} \boxed{u_{11}} \\ u_{21} \\ u_{31} \end{bmatrix} \right) \xrightarrow{\mathcal{Q}_T} \begin{bmatrix} v^r_{11} & t_r(1) & e^r_{11} \\ v^r_{12} & t_r(1) & e^r_{12} \\ \boxed{v^u_{11}} & t_r(1) & \boxed{e^u_{11}} \\ \boxed{v^r_{21}} & t_r(2) & \boxed{e^r_{21}} \\ v^r_{22} & t_r(2) & e^r_{22} \\ v^u_{21} & t_r(2) & e^u_{21} \\ v^r_{31} & t_r(3) & e^r_{31} \\ \boxed{v^r_{32}} & t_r(3) & \boxed{e^r_{32}} \\ v^u_{31} & t_r(3) & e^u_{31} \end{bmatrix} = \underbrace{\begin{bmatrix} r_{11} & -1 & 1 \\ r_{12} & -1 & 1 \\ \boxed{0} & -1 & \boxed{0} \\ \boxed{0} & 0 & \boxed{0} \\ r_{22} & 0 & 1 \\ u_{21} & 0 & 1 \\ r_{31} & 1 & 1 \\ \boxed{0} & 1 & \boxed{0} \\ u_{31} & 1 & 1 \end{bmatrix}}_{\mathbf{C}}.$$

E Experimental System and Data

E.1 Experimental Platform

The experimental platform was developed at CEITEC, Brno University of Technology, around a three-phase interior permanent-magnet synchronous motor (IPMSM) operated under field-oriented control; the platform and motor are described in detail by Zezula et al. (2025). The main physical characteristics of the motor are summarized in Table 6.

The motor is coupled to a dynamometer that provides controlled mechanical loading and shaft-torque measurement. The drive and measurement platform provides the electrical and mechanical quantities required to characterize the motor response over the prescribed operating conditions.

Table 7: Experimental quantities relevant to the case study.

Quantity	Symbol	Use in this work
Phase currents	i_a, i_b, i_c	Electrical measurement
dq-axis currents	i_d, i_q	Model state
dq-axis voltages	u_d, u_q	Model input
Electrical angular velocity	ω_e	Model input
Electrical rotor position	θ_e	Reference-frame transformation
Shaft torque	T	Operating condition

Table 8: Healthy operating conditions used in the case study.

ω_e (rad/s)	Mechanical speed (rpm)	Torque (N m)
900	409	± 5.0
1000	455	± 4.5
1200	546	± 4.0
1400	637	± 3.5
1600	728	± 3.0
2000	909	± 2.5
2500	1137	± 2.0
3500	1592	± 1.5
4500	2046	± 1.0
5000	2274	$0, \pm 0.5$

E.2 Measurements

The quantities relevant to the present case study are summarized in Table 7. Phase currents are measured using external Hall-effect current sensors, while shaft torque is measured at the dynamometer. Rotor-position information is used to obtain the electrical position and angular velocity and to express the electrical quantities in the synchronously rotating dq reference frame. Measurements are acquired at $10\,\mathrm{kHz}$.

E.3 Healthy Operating Protocol

Only healthy motor operation is considered in this work. Each recording follows a speed sequence that rises in successive steps to a prescribed maximum and then descends in successive steps, under the specified mechanical load. The constant-speed intervals have equal duration, so recordings with higher maximum speeds contain more steps and last longer. The sequence captures both steady and transitional behavior across the operating envelope. Table 8 summarizes the healthy speed–load conditions.

The electrical angular velocities in Table 8 are commanded operating points; the corresponding mechanical speeds follow from the 21 pole pairs. The available mechanical load decreases with increasing speed, giving a speed-dependent operating envelope rather than the Cartesian product of the listed speeds and torques. Figure 4 shows the measured dq-axis currents during a representative healthy recording.

F Mechanistic Model and Reconstruction

The prior predictor is based on the continuous-time dq-frame PMSM model described by Zezula et al. (2025),

$$\begin{aligned} u_d &= R i_d + L_d \frac{di_d}{dt} - \omega_e \left(L_q i_q + \lambda^q_{\mathrm{pm}} \right), \\ u_q &= R i_q + L_q \frac{di_q}{dt} + \omega_e \left(L_d i_d + \lambda^d_{\mathrm{pm}} \right). \end{aligned} \tag{3}$$

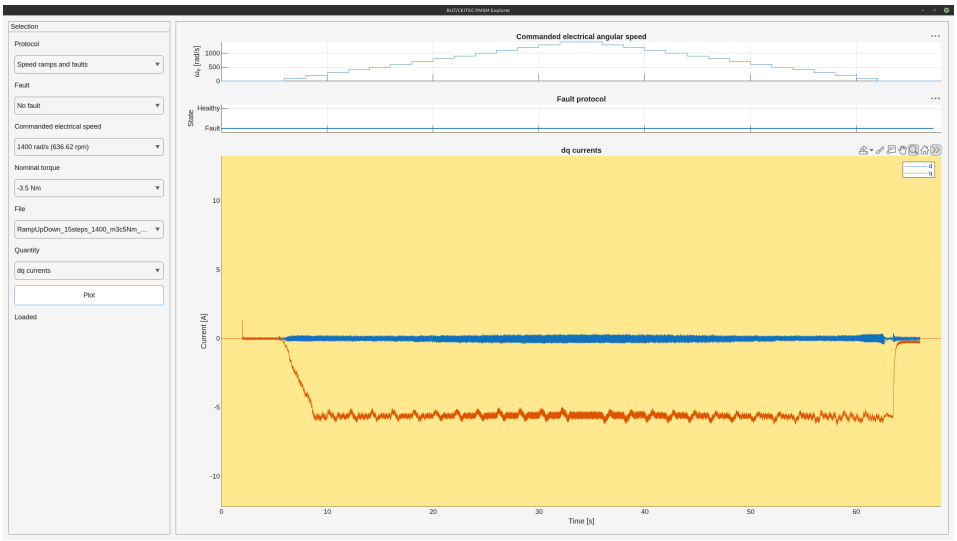


Figure 4: Measured dq-axis currents during a representative healthy speed sequence.

Equivalently,

$$\dot{\boldsymbol{x}} = \mathbf{A}(\omega_e)\boldsymbol{x} + \mathbf{B}\mathbf{u} + \mathbf{g}(\omega_e), \tag{4}$$

where

$$\mathbf{u} = \begin{bmatrix} u_d & u_q \end{bmatrix}^{\mathsf{T}},$$

and

$$\mathbf{A}(\omega_e) = \begin{bmatrix} -\frac{R}{L_d} & \frac{\omega_e L_q}{L_d} \\ -\frac{\omega_e L_d}{L_q} & -\frac{R}{L_q} \end{bmatrix}, \tag{5}$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix}. \tag{6}$$

with

$$\mathbf{g}(\omega_e) = \begin{bmatrix} \frac{\omega_e \lambda_{\mathrm{pm}}^{q}}{L_d} \\ -\frac{\omega_e \lambda_{\mathrm{pm}}^{d}}{L_q} \end{bmatrix}. \tag{7}$$

F.1 Discrete Reconstruction

The continuous-time model defines the prior reconstruction over each observation interval, initialized from the measured state as described in Section 3. For efficient evaluation over the experimental data, we approximate this local Cauchy problem by fixed-step Forward-Euler integration with linearly interpolated inputs.

The integration step was selected empirically by comparing one-step reconstructions against a high-resolution adaptive ODE solution. A $10\,\mu$s Forward-Euler step produced reconstruction errors essentially identical to the high-resolution reference, indicating that the remaining discrepancy was dominated by model mismatch rather than numerical integration. Increasing the step to $50\,\mu$s preserved the same reconstruction accuracy, corresponding to two substeps per $100\,\mu$s observation interval. We therefore use the coarsest tested discretization that did not measurably degrade the one-step reconstruction.

Let T_s denote the observation interval and

$$h = \frac{T_s}{2}.$$

Table 9: Identified parameters of the common healthy PMSM model.

Parameter	Value	Unit
R	1.4756	Ω
L_d	2.6494	mH
L_q	2.9062	mH
$\lambda_{\mathrm{pm}}^{d}$	0.016388	Wb
$\lambda_{\mathrm{pm}}^{q}$	0 (fixed)	Wb

The measured voltage and electrical angular velocity are linearly interpolated at the midpoint,

$$\mathbf{u}_{k-\frac{1}{2}} = \frac{\mathbf{u}_{k-1} + \mathbf{u}_k}{2}, \qquad \omega_{e,k-\frac{1}{2}} = \frac{\omega_{e,k-1} + \omega_{e,k}}{2}.$$

Starting from the measured state $\boldsymbol{x}_{k-1}$, two Forward-Euler half-steps approximate the continuous-time reconstruction:

$$\boldsymbol{x}_{k-\frac{1}{2}} = \boldsymbol{x}_{k-1} + h\left(\mathbf{A}_{k-1}\boldsymbol{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{g}_{k-1}\right), \tag{8}$$

$$\widehat{\boldsymbol{x}}_{p,k} = \boldsymbol{x}_{k-\frac{1}{2}} + h\left(\mathbf{A}_{k-\frac{1}{2}}\boldsymbol{x}_{k-\frac{1}{2}} + \mathbf{B}\mathbf{u}_{k-\frac{1}{2}} + \mathbf{g}_{k-\frac{1}{2}}\right), \tag{9}$$

where

$$\mathbf{A}_{k-1} = \mathbf{A}(\omega_{e,k-1}), \quad \mathbf{A}_{k-\frac{1}{2}} = \mathbf{A}(\omega_{e,k-\frac{1}{2}}), \quad \mathbf{g}_{k-1} = \mathbf{g}(\omega_{e,k-1}), \quad \mathbf{g}_{k-\frac{1}{2}} = \mathbf{g}(\omega_{e,k-\frac{1}{2}}).$$

F.2 Parameter Identification

A single parameterization of the mechanistic model is identified from the healthy experiments across the operating envelope. Each recording contains a sequence of steady operating stages connected by speed transitions. The identification data are drawn from all 21 healthy recordings using protocol-guided windows that represent both steady operating conditions and transitions. For computational tractability, 500-sample windows are used for steady operation and 100-sample windows for transitions. The steady windows are drawn from the central, speed-consistent portion of each operating stage, while the transition windows capture the changes between successive commanded speeds.

Parameter estimation is performed by nonlinear grey-box system identification using the `nlgreyest` function of the MATLAB System Identification Toolbox. Each selected window constitutes an independent identification experiment initialized from its first measured dq-current state, which is held fixed during estimation. The continuous PMSM structure of Eq. equation 3, together with the discrete reconstruction described above, defines the model structure. With the d-axis aligned with the permanent-magnet flux, $\lambda_{\mathrm{pm}}^{q} = 0$, while R, L_d, L_q, and $\lambda_{\mathrm{pm}}^{d}$ are estimated jointly across the selected healthy operating conditions. The optimization is limited to 50 iterations.

The resulting common parameterization is reported in Table 9.

F.3 Transformer Architecture

F.3.1 Teaching Target Generation

Let

$$\mathbb{I}_u = \mathbb{N}_{n_u}$$

index the input variables represented in $\mathbf{C}_u$.

Definition 29 (Teaching target generator)**.** *A* teaching target generator *is a mapping*

$$\Gamma : \mathcal{P}(\mathbb{I}_u) \to \mathcal{P}(\mathfrak{T})$$

that associates a set of potentially unknown input variables with a set of candidate target specifications for teaching.

To construct the input-output pairs used for learning, each target specification T must be associated with the corresponding target values from the complete training context.

Definition 30 (Teaching target)**.** *Let $T = (T_r, T_u)$ be a target specification and let*

$$(\mathbf{C}_r^{\mathrm{n}}, \mathbf{C}_u^{\mathrm{n}}) = \mathcal{N}(\mathbf{C}_r, \mathbf{C}_u), \qquad n_c = n_x + n_u,$$

where $c_{\ell j}^s$ denotes entry (ℓ, j) of $\mathbf{C}_s^{\mathrm{n}}$ for $s \in \{r, u\}$.

For each $s \in \{r, u\}$ and $(\ell, j) \in \mathbb{A}_s$, let

$$y_{\ell j}^s = \begin{cases} c_{\ell j}^s, & (\ell, j) \in T_s, \\ 0, & (\ell, j) \notin T_s. \end{cases}$$

For each $\ell \in \mathbb{N}_L$, let

$$\mathbf{y}_\ell = \begin{bmatrix} y_{\ell 1}^r \\ \vdots \\ y_{\ell n_x}^r \\ y_{\ell 1}^u \\ \vdots \\ y_{\ell n_u}^u \end{bmatrix} \in \mathbb{R}^{n_c}.$$

The teaching target is

$$\mathbf{y}_T = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_L \end{bmatrix} \in \mathbb{R}^{L n_c}.$$

In this work, we use a cardinality-based teaching target generator. Let

$$\mathbb{I}_u = \mathbb{I}_u^{\mathrm{obs}} \sqcup \mathbb{I}_u^{\mathrm{unk}},$$

where $\mathbb{I}_u^{\mathrm{obs}}$ contains the inputs assumed observable and $\mathbb{I}_u^{\mathrm{unk}}$ those that may be unknown.

Definition 31 (Cardinality-based teaching target generator)**.** *For $T_u \subseteq \mathbb{A}_u$, let*

$$T_u^{(j)} = T_u \cap (\mathbb{N}_L \times \{j\})$$

denote the targets associated with input j. Given $1 \le \rho < L$, T_u is a teaching target set *if*

$$|T_u^{(j)}| = 0, \qquad j \in \mathbb{I}_u^{\mathrm{obs}},$$

and

$$1 \le |T_u^{(j)}| \le \rho \quad \textit{or} \quad |T_u^{(j)}| = L, \qquad j \in \mathbb{I}_u^{\mathrm{unk}}.$$

Let

$$T_r^\star = \{L\} \times \mathbb{N}_{n_x}$$

target the whole residual state at the most recent context position. The cardinality-based teaching target generator *is*

$$\Gamma_\rho(\mathbb{I}_u^{\mathrm{unk}}) = \{(T_r^\star, T_u) \in \mathfrak{T} : T_u \textit{ is a teaching target set}\}.$$

F.4 Training and Evaluation Protocol

Training and evaluation use grouped four-fold cross-validation. All contexts from a physical recording belong to the same training, validation, or test subset within a fold; the grouping variable is the recording index. The 21 recordings are assigned to four outer folds while balancing context counts. For each fold, the assigned recordings form the test set, approximately 20% of the remaining recording groups are selected for validation, and the rest form the training set. Validation-group combinations are kept distinct across folds when feasible. Partitioning uses reference seed 1 and the `twister` random-number generator.

The resulting subsets contain the following numbers of contexts:

Fold	Training	Validation	Test
1	85 400	19 320	33 880
2	85 160	19 320	34 120
3	94 040	10 680	33 880
4	88 800	13 080	36 720

Algorithm 1 Compilation of self-supervised reconstruction pairs

Require: Complete contexts $\mathcal{D}_s$, fold normalizer $\mathcal{N}$, teaching target generator Γ, potentially unknown inputs $\mathbb{I}_u^{\mathrm{unk}}$
Ensure: Reconstruction pairs $\mathcal{P}_s$

$\mathcal{P}_s \leftarrow \varnothing$
for all $(\mathbf{C}_r, \mathbf{C}_u) \in \mathcal{D}_s$ **do**
  $(\mathbf{C}_r^{\mathrm{n}}, \mathbf{C}_u^{\mathrm{n}}) \leftarrow \mathcal{N}(\mathbf{C}_r, \mathbf{C}_u)$
  for all $T \in \Gamma(\mathbb{I}_u^{\mathrm{unk}})$ **do**
    $\mathbf{C}_T \leftarrow \mathcal{Q}_T(\mathbf{C}_r^{\mathrm{n}}, \mathbf{C}_u^{\mathrm{n}})$
    Form $\mathbf{y}_T$ from $(\mathbf{C}_r^{\mathrm{n}}, \mathbf{C}_u^{\mathrm{n}})$ and T
    $\mathcal{P}_s \leftarrow \mathcal{P}_s \cup \{(\mathbf{C}_T, \mathbf{y}_T)\}$
  end for
end for
return $\mathcal{P}_s$

Subset sizes differ because recordings contain different numbers of operating points and transitions, yielding different numbers of contexts after preprocessing. Within each fold, normalization statistics are fitted on the training subset and applied to the training, validation, and test subsets.

Algorithm 1 compiles reconstruction pairs independently for each subset $s \in \{\mathrm{train}, \mathrm{validation}, \mathrm{test}\}$, using the teaching target generator defined in Section F.3.1. In this case, $L = 5$, torque is the only potentially unknown input, and the combinatorial radius is $\rho = 1$.

Each complete context generates six teaching configurations: five withholding torque at one context position and one withholding it throughout the context. Every configuration also targets the complete residual state at the most recent position. The resulting numbers of pairs are

Fold	Training pairs	Validation pairs	Test pairs
1	512 400	115 920	203 280
2	510 960	115 920	204 720
3	564 240	64 080	203 280
4	532 800	78 480	220 320

Pairs are shuffled deterministically before training.

Following Section D.3, each pair is serialized in temporal-major order as 30 tokens with three features per token. All four Transformers use embedding dimension 32, four attention heads, three encoder stages, feed-forward dimension 64, and dropout probability 0.1; initialization uses seed 1.

Each fold is trained independently for two epochs using Adam, mini-batches of 256 pairs, an initial learning rate of 10^{-3}, and gradient clipping at threshold 1. The masked mean-squared reconstruction loss evaluates only target positions. Validation loss is evaluated every 250 iterations and at the end of each epoch; the model with the lowest validation loss is retained and evaluated on that fold's test subset. Two epochs correspond to approximately 4 000–4 400 optimization iterations per fold. All stochastic stages use seeded random-number generation, and the four folds are trained sequentially.

F.5 Residual Distribution Metrics

Table 5 compares the prior and WRAPPER on paired test contexts from each held-out healthy recording. $M1$–$M4$ identify the independently trained fold models, $R1$–$R21$ identify recordings, and n is the number of paired contexts in the indicated test recording. Each numerical entry reports the prior value followed by the WRAPPER value.

Let $r_i \in \mathbb{R}^2$ be a dq-current residual and let $\mu = n^{-1} \sum_{i=1}^{n} r_i$. We report the bias magnitude $\|\mu\|_2$, the residual RMSE

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \|r_i\|_2^2},$$

and P99, the empirical 99th percentile of the magnitudes $\|r_i\|_2$. These three quantities are expressed in milliamperes.

The ellipse metrics describe the joint distribution of the d- and q-axis residuals. For an estimated mean $\hat{\mu}$ and covariance $\hat{\Sigma}$, the Gaussian ellipse at nominal coverage p contains residuals satisfying

$$(r - \hat{\mu})^{\mathsf{T}} \hat{\Sigma}^{-1} (r - \hat{\mu}) \leq \chi_2^2(p),$$

where $\chi_2^2(p)$ is the p-quantile of a chi-squared distribution with two degrees of freedom. The area of the Gaussian covariance ellipse Friendly et al. (2013) follows from its semi-axis lengths, where λ_1 and λ_2 are the eigenvalues of $\hat{\Sigma}$:

$$\begin{aligned} A_p &= \pi \sqrt{\chi_2^2(p)\lambda_1}\, \sqrt{\chi_2^2(p)\lambda_2} \\ &= \pi\, \chi_2^2(p) \sqrt{\lambda_1 \lambda_2} \\ &= \pi\, \chi_2^2(p) \sqrt{\det \hat{\Sigma}}. \end{aligned}$$

Here, fitting a Gaussian ellipse means estimating its center and covariance from the specified residual subset, then applying the nominal 95% or 99% chi-squared threshold. The ellipses in Figure 3 are fitted to each complete recording for visualization. Emp95 and Emp99 instead assess temporally held-out coverage: each complete recording is partitioned into five contiguous, approximately equal-length segments. For each segment, ellipse parameters are estimated from the other four segments of that recording, separately for prior and WRAPPER residuals. The reported coverage is the number of held-out residuals inside their corresponding ellipses divided by the total number of residuals across the five held-out segments. Thus, every residual contributes once to the coverage calculation.

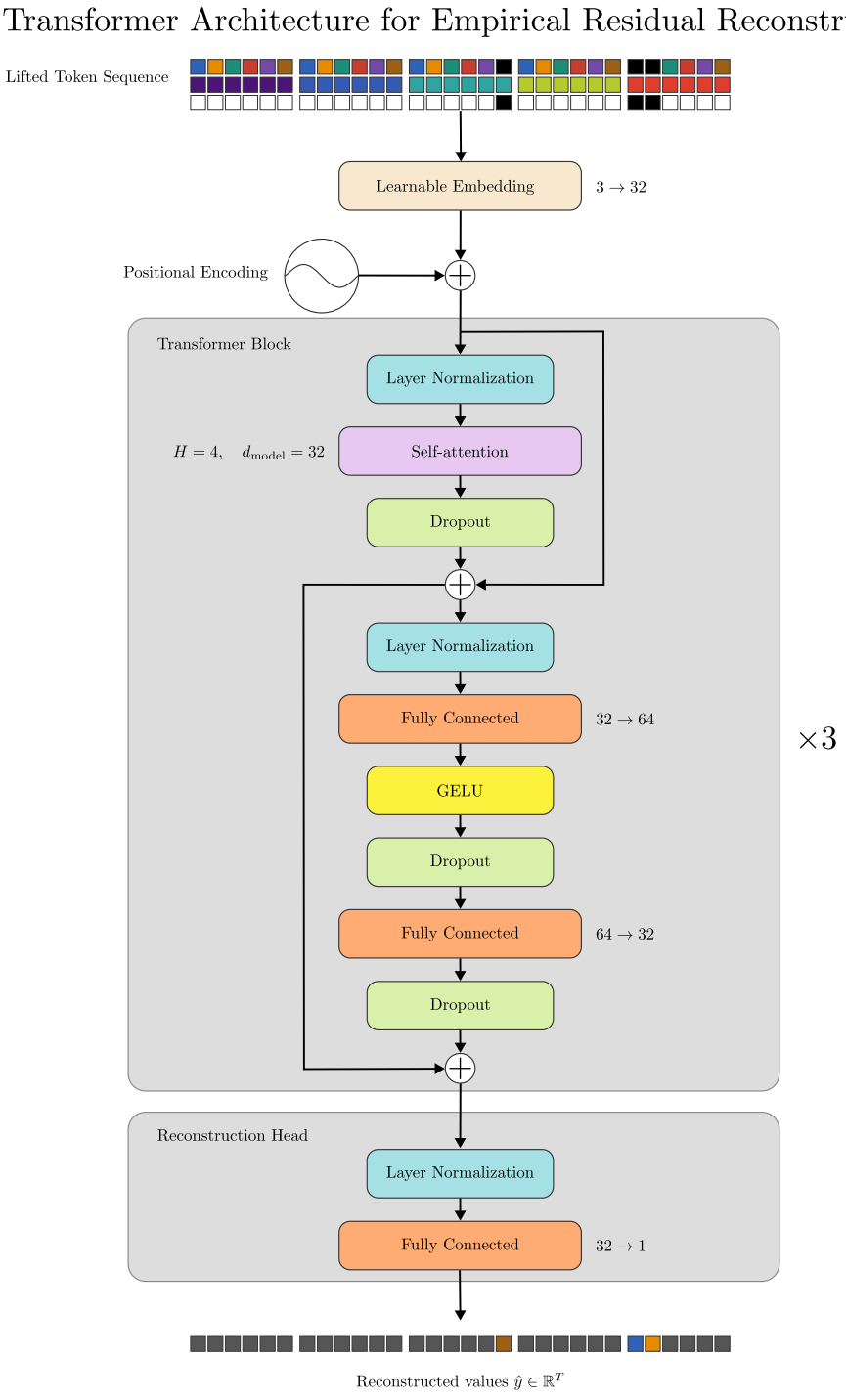


Figure 5: Transformer architecture used for empirical residual reconstruction. Lifted tokens are mapped to a 32-dimensional representation by a learnable embedding and augmented with sinusoidal positional encoding. Three pre-normalized Transformer blocks apply four-head self-attention and $32 \to 64 \to 32$ feed-forward transformations with skip connections. A reconstruction head consisting of layer normalization and a $32 \to 1$ fully connected projection produces one reconstructed value per token. Highlighted outputs constitute $\widehat{\mathbf{y}}_T$; gray outputs are not selected by T.