MDPI

Article

Estimating Causal Effects from Tabular Data with Assumption Gates and Automated Estimator Selection for Trustworthy Artificial Intelligence

Alessandro Berti [1]

[1] Process and Data Science Group, Department of Computer Science, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany; a.berti@pads.rwth-aachen.de

Abstract

Causal analysis can guide decisions by estimating how interventions change outcomes and making the assumptions behind those estimates explicit. We present the Causal World Foundation Model, a framework for answering causal questions from tabular data. It checks declared conditions, combines classical estimators, and calibrates uncertainty. Effect queries receive an estimate with diagnostics or an explanation for refusal. A pretrained neural router serves as a baseline for comparing estimator selection and aggregation policies. The evaluation covers treatment effects, observed regimes, and network interference, alongside a semi-synthetic test using energy sensor covariates. In confirmatory experiments, a shallow selector and a convex stack improve accuracy over the evaluated deep router. Stress tests demonstrate consistent refusal when declared conditions fail, while causal interpretation remains conditional on truthful and complete declarations. A separate audit clarifies the strengths and limitations of empirical support checks. Together, these findings support a practical approach to transparent causal artificial intelligence through flexible estimator selection and clear reporting of assumptions, uncertainty, and decision reasons.

Keywords: causal inference; trustworthy artificial intelligence; interpretable artificial intelligence; foundation models; estimator selection; identification; abstention; conformal prediction; network interference

1. Introduction

Causal effect estimates help compare treatments, pricing decisions, and industrial control actions. Their interpretation depends on assumptions about how treatment and outcome are generated, because associations alone need not describe intervention effects [1,2]. Automated causal artificial intelligence (AI) therefore needs accurate estimation and a clear account of the conditions supporting each answer.

This work addresses two practical needs: selecting estimators for datasets generated by different mechanisms and making the conditions for returning an answer inspectable. We connect these steps so that the analyst receives either an estimate with uncertainty or a specific reason why the requested analysis cannot proceed under the supplied conditions.

Existing work provides strong foundations for this integration. DoWhy separates modeling, identification, estimation, and refutation [3]. Causal model-selection methods use proxy losses to compare estimators when counterfactual errors are unavailable [4,5]. Prior-data fitted networks reuse synthetic pretraining for causal estimation [6,7,8]. The gap addressed here is the joint evaluation of estimator selection, declared assumptions, empirical support, and explicit refusal within one interface. Section 3 develops this positioning.

Received:
Accepted:
Published:

The Causal World Foundation Model (CWFM) brings these elements together for static treatment effects, observed regimes, and network spillovers. It accepts a dataset, a formal query, and an assumption ledger, then coordinates validation, classical estimation, and interval calibration. A pretrained neural router serves as one aggregation baseline alongside simpler policies. The name CWFM identifies this implementation; foundation model refers to reuse of its pretrained neural component across the implemented tasks.

The confirmatory results show the value of simple aggregation within this framework. On 1,800 effect queries, a shallow random-forest risk selector achieves mean absolute error (MAE) 0.141 and a convex stack 0.147, compared with 0.161 for the deep router. The shared gate also refuses all 1,200 stress queries that violate the declared conditions. These refusals demonstrate consistent contract enforcement. They remain conditional on truthful and complete declarations, which the gate itself cannot verify.

The contributions are:

- a typed interface connecting causal queries, declared assumptions, deterministic refusal rules, and calibrated effect estimates;
- a comparison of neural routing and simple aggregation over the same classical estimator library, with separate tuning, calibration, and test streams;
- an evaluation of contract compliance, structural decisions, presentation sensitivity, and transfer to semi-synthetic energy data, supplemented by a direct audit of the public empirical support checks.

The research questions (RQs) concern accuracy on answerable queries (RQ1), enforcement of declared conditions (RQ2), regime detection (RQ3), generalization to families excluded from training (RQ4), and the computational pathways producing the reported results (RQ5).

Section 2 introduces the background, and Section 3 reviews related work. Section 4 presents the method. Section 5 describes the experimental setup and implementation, Section 6 reports the results, and Sections 7 and 8 provide the discussion and conclusion.

2. Background

2.1. Structural Causal Models and Identification

A structural causal model (SCM) specifies how each variable depends on its direct causes and an exogenous disturbance [1]. Disturbances may be dependent when the model permits unobserved common causes. An intervention replaces a variable's assignment mechanism by a fixed value. For a binary treatment, the average treatment effect (ATE) is

$$\tau = \mathbb{E}[Y \mid do(A = 1)] - \mathbb{E}[Y \mid do(A = 0)]. \tag{1}$$

Here, τ is the population average treatment effect, Y is the outcome, A is a treatment taking values zero and one, $do(A = a)$ denotes intervention setting A to a, and $\mathbb{E}$ is expectation over the target population under that intervention. This estimand differs from a comparison of observed treatment groups [2].

The *estimand* is the population quantity being requested, here the average treatment effect. *Identification* asks whether the observed distribution and stated assumptions determine that quantity. *Estimation* then approximates it from a finite sample. This order matters: an accurate fit to observed outcomes does not by itself supply missing causal assumptions. A sufficient pre-treatment adjustment set accounts for relevant common causes of treatment and outcome. Consistency links the observed outcome to the treatment actually received, and positivity requires the compared treatments to be possible in the relevant covariate groups [1,2].

For illustration, consider a treatment study in which disease severity X affects both treatment assignment and outcome. Assumed intervention mean outcomes of 8 when treatment is assigned and 6 when it is withheld give $\tau = 2$ outcome units in Equation 1. These values illustrate the estimand; they are not observed treatment-group means. Severity belongs to the adjustment model, rather than the notation of the marginal effect itself.

Under the stated adjustment conditions, g-computation fits an outcome model using treatment and severity. It predicts an outcome for each observed covariate profile under treatment and again without treatment, then averages the prediction differences [2]. The profiles being compared remain the same, which separates the treatment contrast from differences in the composition of the observed groups. Additional identifying information is needed when the same observed distribution is compatible with different intervention effects.

2.2. Network Interference and Support

Interference occurs when one unit's treatment changes another unit's outcome [9]. An exposure mapping summarizes relevant neighboring treatments, for example the treated fraction of a unit's neighbors. A spillover contrast compares two exposure levels while holding the unit's own treatment fixed. Identification depends on the assignment design and the correctness of the declared mapping [10,11].

Thus an untreated unit can still have high exposure if many of its neighbors are treated. The unit's own treatment and its exposure describe different inputs to the outcome mechanism. CWFM's network query changes the exposure level while preserving own treatment for each unit, then averages the resulting contrasts over the observed units.

Positivity is a population condition requiring the relevant treatment or exposure levels to be possible for the target units. Empirical support diagnostics instead assess the finite sample [12]. A failed diagnostic can motivate collecting more data, changing the estimand, or adopting explicit extrapolation assumptions. CWFM's implemented policy withholds the requested estimate. Passing the policy does not prove population positivity.

2.3. Regimes and Mechanism Changes

A regime is a region in which a conditional outcome mechanism is stable. Regime determination asks whether that mechanism changes across a covariate threshold. Model-based recursive partitioning provides a related framework [13]; heterogeneous treatment-effect partitioning targets a different quantity [14]. A smooth nonlinear mechanism may be approximated by several linear pieces without containing a true discontinuous change. Improved predictive fit alone therefore does not establish a regime boundary [15,16].

Here, the boundary divides observations according to a measured covariate. It need not represent a change over time. The requested output is a structural decision about the conditional outcome model. This differs from estimating the effect of intervening on the covariate used to define the regions.

2.4. Amortized Inference and Conformal Calibration

Amortized inference trains a shared model across synthetic problems and reuses its parameters for new datasets [17]. Prior-data fitted networks (PFNs) implement this approach under a synthetic training prior. CWFM applies it to estimator selection; its classical estimators are still fitted to each submitted dataset.

Split-conformal calibration uses errors on separate calibration examples to determine interval radii [18,19]. Under exchangeability of calibration and future examples and the appropriate finite-sample quantile rule, it provides marginal coverage. Here, an example is an entire causal episode with a known simulated effect. The resulting guarantee concerns exchangeable episodes, not repeated sampling from an arbitrary fixed real-world process.

3. Related Work

3.1. Formal Identification and Causal-Analysis Software

Structural causal models and the potential-outcomes framework give causal questions a precise meaning [1,2]. Their central tool is the theory of *identification*: rules that determine whether a causal question can be answered from the available data under stated assumptions. When point identification fails, sensitivity analysis quantifies how conclusions change as hidden confounding grows [20], partial identification replaces a point estimate with an identified interval [21], and dedicated frameworks cover interference between units [9].

Identification is also an algorithmic problem. Shpitser and Pearl give a complete graphical procedure for identifying interventional distributions in semi-Markovian causal models [22]. DoWhy and its graphical causal model extension (DoWhy-GCM) organize graph specification, causal estimation, model evaluation, and a broader range of causal queries [3], while Deep End-to-End Causal Inference (DECI) combines discovery and effect inference [23].

CWFM uses a finite set of sufficient identification rules for randomized treatment effects, observational effects under a declared adjustment set, and network spillovers under a declared exposure mapping. This contract validator integrates readily with empirical support checks and estimator selection. General derivation of identifying functionals remains the role of graphical identification algorithms.

Causal structure can be learned through constraint-based search [24], continuous graph optimization [25], invariant prediction [15], heterogeneous data [16], and nonlinear time series [26]. These algorithms automate defined analysis steps. The analyst supplies the causal scope and configures the assumptions and procedures. In CWFM, fitting and aggregation are automated once the query and assumptions are declared.

3.2. Amortized Causal-Effect Estimation

Prior-data fitted networks reuse a transformer trained on synthetic problems, approximating Bayesian inference under their training prior [17,27]. Several causal variants are closely related to this work. CausalFM covers back-door, front-door, and instrumental-variable settings [6]. Do-PFN predicts interventional outcome distributions from observations without requiring a graph at inference [7]. CausalPFN estimates average and heterogeneous treatment effects under ignorability in its prior [8]. These published models illustrate amortized estimation of effects and interventional distributions under specified training priors.

Partial-graph-conditioned models accept causal knowledge as an input [28]. For temporal data, amortized causal discovery learns shared dynamics across datasets with different graphs [29]. Longitudinal effect estimation is addressed by the Causal Transformer, which models treatment histories and time-varying confounders [30]. These methods target different tasks; the latter is trained for the dataset under study.

The causal-effect models encode identification assumptions in their priors or interfaces and primarily evaluate effect or distribution estimation. CWFM adds a joint evaluation of estimation and a query-level contract and refusal interface. Direct causal PFNs predict effects; CWFM's neural baseline predicts risks for classical estimators fitted to each dataset. For causal-structure prediction, amortized variational inference for causal discovery (AVICI) learns to infer graphs from simulated observational or interventional datasets [31]. Tables 1 and 7 compare the evaluated interfaces and targets; they do not imply that other systems cannot support additional capabilities.

3.3. Validation, Selection, and Aggregation of Causal Estimators

Causal model selection compares estimators without observing their counterfactual errors. For a given unit, the dataset records the outcome under the received treatment, but not the outcome that the same unit would have had under the alternative. Ordinary outcome-prediction errors therefore do not directly reveal treatment-effect errors [2].

Proposed criteria use influence functions [4], pseudo-outcomes and the R-learner objective [32], counterfactual cross-validation [33], and selective learning for doubly robust functionals [34]. Empirical studies assess how reliably these criteria rank candidates [35,36]. Aggregation offers another approach: causal Q-aggregation combines candidate treatment-effect predictions using a doubly robust loss and provides model-selection guarantees under its assumptions [5]. AutoCATE automates the wider pipeline for conditional average treatment effect (CATE) estimation [37].

CWFM builds on these methods by learning estimator risks across synthetic episodes with known effects. The neural baseline reuses this risk predictor on new datasets, conditioned on the query and fitted candidate estimates. AutoCATE instead optimizes a richer conditional-effect pipeline within each dataset. The common interface in CWFM also allows fixed, shallow, and convex aggregation policies to be evaluated over the same library. This separation makes the choice of aggregation policy explicit and supports direct comparison in Section 6.1.

3.4. Reliability under Non-Identification, Weak Support, and Prior Shift

Positivity and empirical support.

Positivity is a population condition; empirical diagnostics assess support in a finite sample [12]. Trimming [38], overlap weighting [39], extrapolation, and partial identification provide different responses to limited overlap. Trimming and weighting can change the target population. CWFM retains the requested target and withholds its estimate when the implemented support criterion fails.

Sensitivity analysis and partial identification.

When point identification fails, useful outputs can include an identified set [21] or a sensitivity curve indexed by confounding strength [20]. Quantile-balancing sensitivity analysis estimates sharp treatment-effect bounds under a specified marginal sensitivity model [40]. CWFM currently reports a refusal for the requested point estimate; sensitivity and partial-identification outputs offer natural extensions.

Prior-induced bias in causal PFNs.

Recent work studies corrections for prior-induced confounding bias in causal PFNs [41]. Broader empirical work shows that selecting heterogeneous-effect estimators requires careful evaluation of model-selection criteria [36]. CWFM combines classical estimators fitted to the submitted dataset, while its neural selection policy retains the assumptions of its synthetic training prior. Section 6.6 examines this dependence through development diagnostics.

Conformal calibration and its target.

Conformal prediction provides marginal coverage under exchangeability of calibration and future examples [18,19]. Causal applications include counterfactual outcomes and individual treatment effects [42] and calibration of heterogeneous-effect predictors [43]. Their guarantees depend on the stated identification, sampling, and estimation conditions. CWFM calibrates scalar estimation errors across simulator episodes. Its coverage target is therefore marginal over exchangeable episodes, as detailed in Section 4.7.

Four meanings of refusal.

A causal system can withhold an answer because the supplied assumptions do not identify the effect, the observed sample has insufficient support, a learned rule predicts high error [44], or the input differs from the training distribution. CWFM implements the first two through deterministic rules. Uncertainty-aware treatment-effect models can flag unreliable predictions under limited overlap or covariate shift [45]. Self-compatibility checks assess causal discovery across subsets of variables without requiring ground-truth graphs [46].

Causal reasoning and refusal evaluation.

CLadder evaluates language-model reasoning on associational, interventional, and counterfactual questions with formally generated answers [47]. CausalT5k evaluates refusal and failure modes across causal rungs [48]. These benchmarks concern language models. CWFM evaluates an estimator pipeline, recording explicit decision statuses alongside numerical errors. Separating identification from estimation also follows the graphical identification framework [22].

3.5. Task-Specific Foundations: Interference and Regime Determination

Network interference.

Design-based frameworks connect treatment assignment, exposure mappings, and causal estimands under interference [9,10]. Further work studies misspecified mappings [11] and learned representations of network-mediated effects [49]. CWFM uses a supplied network and exposure mapping, then screens whether the requested contrast is represented in the observations. The validity of the network and mapping remains an analysis assumption.

Regime determination.

CWFM's regime task concerns changes in the conditional outcome mechanism. Model-based recursive partitioning [13] motivates the one-split baseline; causal trees instead target heterogeneous treatment effects [14]. Nonstationary-data methods [16] and invariant prediction [15] also use mechanism stability. Our evaluation includes stationary nonlinear controls because several fitted linear regions can approximate a smooth mechanism without establishing a structural change.

Hybrid learning pipelines also support engineering inspection. Zhang et al. integrate image-processing and learning components to detect bridge-foundation damage from remotely operated vehicle imagery [50]. Their work illustrates component integration in a predictive application. CWFM addresses the additional identification requirements of causal effect estimation.

3.6. Positioning and Research Gap

CWFM integrates formal causal queries, declared-assumption rules, empirical support checks, classical estimation, and interval calibration. The contribution is an inspectable analysis pipeline and an evaluation that separates estimator accuracy from refusal behavior. The shared library supports a direct comparison of neural and simple aggregation policies. General graphical identification and detection of undeclared confounding remain outside the implemented scope. Table 1 summarizes the interfaces of representative systems.

Table 1. Interface positioning of representative causal systems.

System	Target	Assumptions and selection	Non-answer interface
Identification algorithm [22]	Interventional distributions	Complete graphical procedure	Proves non-identifiability
DoWhy [3]	Effect estimands	User graph and estimator choice	Refutation diagnostics
DECI [23]	Graph and effects	Learned graph within model class	Not reported
AutoCATE [37]	Conditional effects	Per-dataset pipeline selection	Not reported
Causal aggregation [5]	Treatment effects	Candidate assumptions; fitted weights	Not reported
CausalFM [6]	Treatment effects	Settings encoded in prior	Not reported
Do-PFN [7]	Interventional outcomes	Synthetic pretraining prior	Not reported
CausalPFN [8]	Average and conditional effects	Ignorability in prior	Not reported
Causal conformal [42]	Intervals	Inherits base-analysis conditions	Not reported
CWFM	Scalar effects	Declared rules; support screen; estimator selection	Typed refusal with reasons

Not reported means the cited publication does not evaluate the capability as a first-class interface. It does not imply that the capability cannot be added. PFN: prior-data fitted network; DECI: Deep End-to-End Causal Inference.

4. Proposed Method

CWFM turns an observed dataset, a formal query, and an assumption ledger into a documented analysis result. An accepted effect query returns a point estimate and interval; a refused query returns the failed condition and supporting diagnostics. The following subsections describe how validation, candidate fitting, and aggregation produce these outputs.

4.1. Design Principles and Overview

Figures 1 and 2 introduce the design. Separate structural and assumption checks address different sources of error, while the estimator library supports several aggregation policies. This organization makes the basis for each output visible to the analyst.

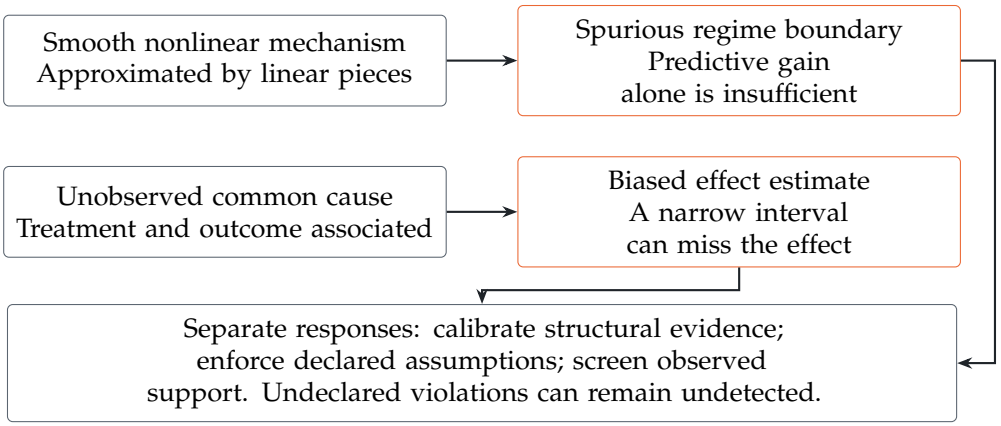

Figure 1. Two failure modes motivating separate structural and assumption checks.

A smooth mechanism can produce apparent regime boundaries when approximated by local linear fits. Hidden confounding can instead bias an effect estimate without producing an obvious predictive failure. Only disclosed violations of the implemented assumptions necessarily trigger the contract gate.

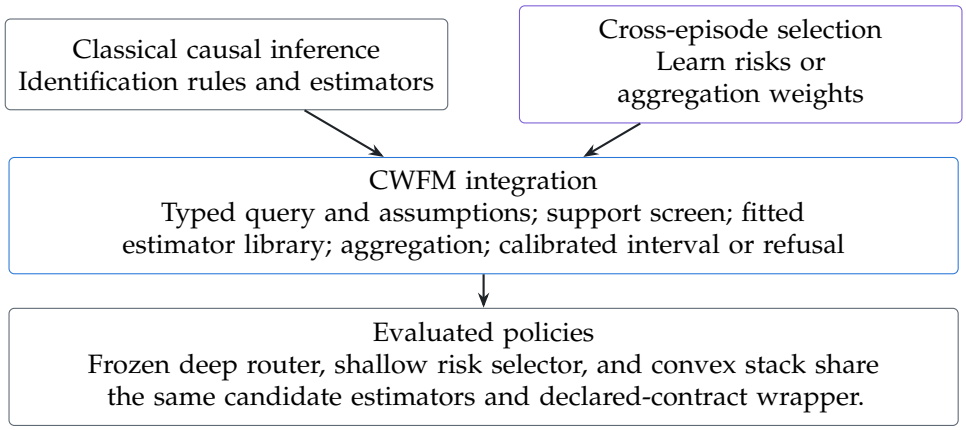

Figure 2. Integration of causal estimation and automated estimator selection.

The common interface connects established causal estimators with automated selection. Neural pretraining, shallow risk prediction, and convex aggregation can therefore be compared over the same fitted candidates. This modular design allows improvements in aggregation to be assessed without changing the causal query or declared conditions.

The method assigns a distinct role to each component: gates enforce declared conditions, classical estimators fit the data, aggregation combines compatible estimates, and calibration supplies uncertainty intervals. These responsibilities remain separate, so an aggregation policy cannot override a refusal.

Figure 3 presents the public workflow. Validated inputs proceed to candidate fitting and the frozen neural baseline. The router uses both the encoded episode and candidate summaries to produce aggregation weights. Stored calibration then supplies an interval for the combined estimate. Sections 4.8 and 5 give the scope and evaluation protocol.

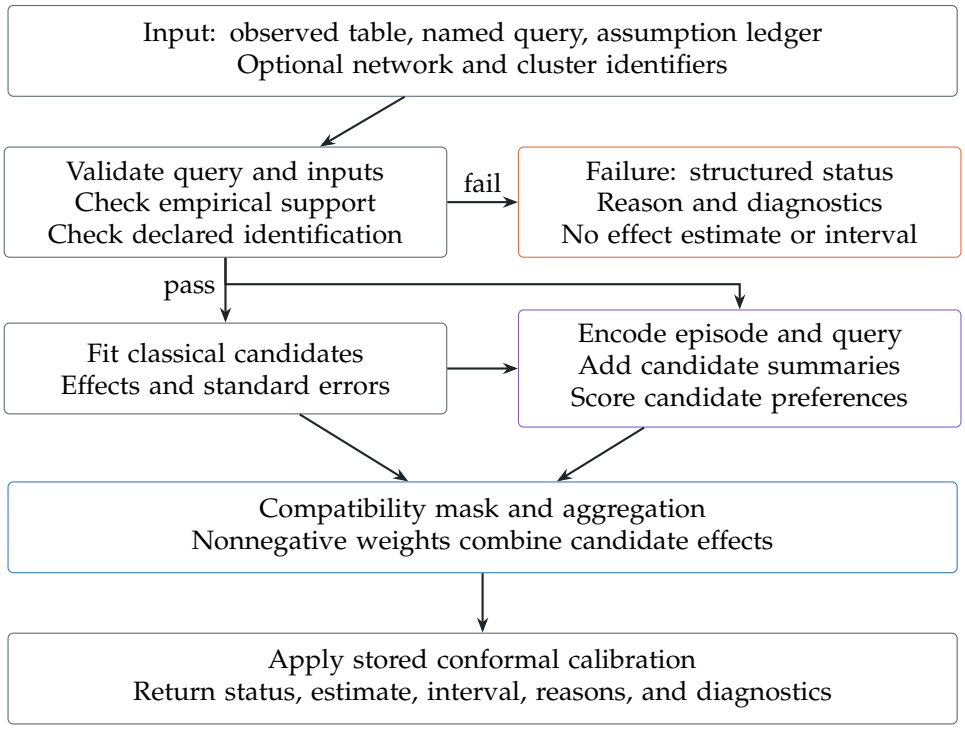

Figure 3. Workflow of the public CWFM analysis pipeline.

The public runtime checks input validity, empirical support, and identification declarations in that order. Candidate effects and standard errors then enter the risk router alongside the encoded episode. The weighted classical estimate determines the reported effect at the evaluated checkpoint. The result retains the candidate weights and diagnostics for inspection.

4.2. Causal Episodes and the Formal Query Language

A causal episode contains observed values, missingness indicators, variable types and roles, optional relations between units, and declared design information. Roles identify treatment, outcome, exposure, and covariates. A relation matrix represents the observed network for interference queries. The ledger separately records the analyst's causal assertions.

A formal query names the variables and requested estimand. Separate query classes cover a binary static ATE, observed-regime determination, and a network exposure contrast averaged over observed own-treatment values. The submitted episode defines the target population. Each class fixes the required inputs and whether the output is an effect estimate or a structural decision.

The query and declared assumptions define the analysis *contract*. We write the input-output mapping as

$$\mathcal{F}(E, Q) \;\rightarrow\; \big(S,\ \widehat{\tau},\ C_\alpha,\ R,\ D\big), \tag{2}$$

Here, $\mathcal{F}$ is the analysis function, E is the episode, including observed inputs and the assumption ledger, Q is the query, and S is the output status. The scalar $\widehat{\tau}$ is the effect estimate and C_α is its interval at nominal coverage $1-\alpha$, where α is the miscoverage level. Both are absent for refused queries; regime queries instead return a structural decision. The object R contains a machine-readable reason, and D contains diagnostics. Conceptual statuses are `answer`, `not identified`, `unsupported`, and `out of scope`; the public application also distinguishes invalid input and unavailable model files. These are software decisions under implemented rules, not proofs that every refused query is

mathematically unidentifiable. Figure 4 shows the input and output objects, and Table 2 lists the causal query classes.

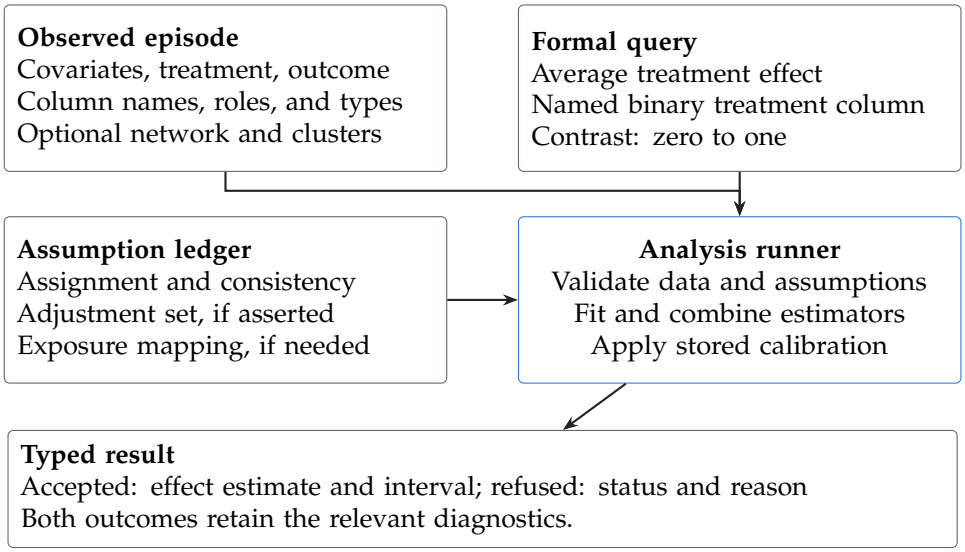

Figure 4. Input and output objects for a static treatment-effect query.

The three inputs have complementary roles: the episode supplies observations, the query fixes the estimand, and the ledger records assumptions. The result brings the decision, estimate, uncertainty, and diagnostics together in one typed object. Section 4.5 explains the conditional meaning of acceptance.

In the treatment example of Section 2.1, each row describes a patient, and named columns record treatment, outcome, and severity. The query requests the average contrast between treatment and no treatment. The ledger states whether assignment was randomized or whether the supplied adjustment set is asserted to be sufficient. The runner validates these inputs, screens support, and checks the required declarations before fitting the eligible estimators. An accepted result reports the combined effect and interval; a failed condition produces a reason instead. In particular, refusal is distinct from an estimated effect of zero.

Table 2. Implemented causal-effect query classes.

Query class	Declared conditions	Support condition	Estimators
Randomized ATE	Randomized treatment assignment	At least ten observations in each arm; empirical overlap screen	Design-based; regression-adjusted
Observational ATE	Declared sufficient pre-treatment adjustment set	Treatment overlap over the target covariate region	Linear, interaction, and spline g-computation
Network spillover	Assignment, consistency, exposure mapping, and observed-network restriction	Support for the requested exposure contrast	Linear and piecewise exposure response

ATE: average treatment effect. All effect queries additionally require asserted consistency and empirical support. The observational and network conditions are supplied by the user; the gates do not verify their truth.

4.3. Typed Encoder and Neural Mechanism Modules

The encoder represents each cell by its value, missingness indicator, variable type, and query role. Alternating attention operations combine information across variables and samples. The query vector also contains normalized treatment, outcome, and exposure column indices. Thus the architecture is presentation-randomized, not position-blind, despite having no separate column-position embedding.

Training and the confirmatory column-permutation probes reorder columns within role groups. The public adapters arrange columns from their query names, but adding covariates can change the normalized indices. Section 6.1 quantifies the resulting stability and remaining sensitivities. Exact invariance to arbitrary query-column relocation is not established; removing those inputs would require a new checkpoint interface and retraining.

Attention combines information from related parts of an input. In CWFM, it operates across variables within a sample and across samples within a variable. These operations capture covariate relationships and distributional patterns. For network data, a message-passing step adds summaries of neighboring units.

Presentation probes check whether changing a dataset's representation changes the answer. Reordering rows should preserve the effect, whereas changing outcome units should rescale it consistently. These are called invariance and equivariance, respectively. Adding a covariate can shift query-role column indices, which are divided by the fixed capacity of 12 columns. Consequently, the query representation can change even when the added column carries no relevant causal information.

The feed-forward layers use a sparse mixture of neural modules, with a learned gate activating a subset for each representation. We call these internal networks *neural modules* and members of the classical library *candidate estimators*. The term `expert` in the code refers to these software components, never to a human participant. A specialist is a classical candidate suited to a particular mechanism. The internal gate shapes representations; the estimator risk router combines fitted effect estimates.

4.4. Structural Heads and Exploratory World Particles

The evaluated regime pathway fits candidate threshold models and computes a Bayesian information criterion (BIC) evidence statistic. A threshold selected on calibration episodes converts this evidence into a split decision. Stationary nonlinear controls help distinguish a smooth mechanism from a structural break (Figure 5).

The architecture also includes a hierarchical neural decoder for the split variable, threshold, and change type. It is trained using matched smooth and piecewise mechanisms. Its outputs are recorded as diagnostics; the reported split decisions use the calibrated classical statistic.

World particles are scored hypotheses combining a graph, mechanism family, and regime hypothesis. They provide an exploratory representation of structural uncertainty. Their scores are effectively uniform at the evaluated checkpoint and do not enter its point estimate (Section 6.7).

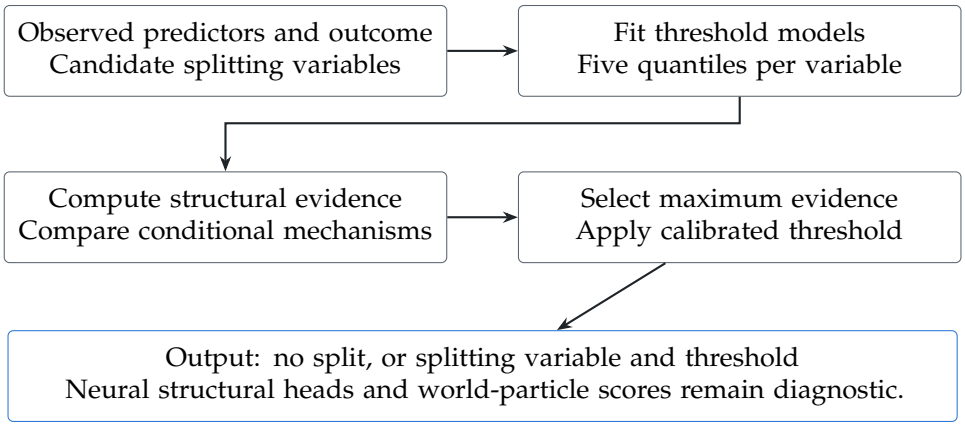

Figure 5. Calibrated regime-determination pathway.

For each eligible splitting variable, candidate models provide evidence at five quantile thresholds. The largest score is compared with the calibrated cutoff. The output records a no-split decision or the selected variable and threshold, making the structural decision traceable to fitted evidence. The splitting threshold is a value of the covariate that divides observations into regions. The calibrated cutoff is instead a value of the evidence statistic that determines whether any proposed split is accepted. Keeping these quantities separate makes the structural decision and the selected boundary easier to interpret.

4.5. Identification and Support Gates

Two deterministic gates connect the requested estimate to its declared assumptions and observed support.

The identification gate enforces a finite library of sufficient conditions in the assumption ledger, including randomized assignment or an asserted sufficient adjustment set. It provides a reproducible decision and a specific reason when a required condition is missing.

This guarantee is conditional on truthful and complete declarations. Incorrect randomization claims, omitted confounders, or a wrong exposure mapping can pass the same rule as correct metadata. Observationally equivalent causal models can imply different intervention effects, so the gate cannot establish causal truth from the observed table. Its guarantee is contract compliance, not correctness of every accepted answer.

The public support gate screens the submitted observations. For static effects, it fits a ridge linear-probability model of treatment on standardized covariates. The fitted score approximates the probability of receiving treatment at a covariate profile. Scores near zero or one flag profiles for which the comparison treatment is poorly represented. The implementation clips scores to $[0, 1]$ and requires at most 10% outside $[0.05, 0.95]$, with the fifth percentile above 0.01 and the 95th below 0.99.

For network effects, support concerns the requested exposure levels as well as own treatment. Both exposure targets must lie inside the fifth-to-95th percentile range overall and within each own-treatment group. Across the dataset, at least ten observations must be near each target. The neighborhood tolerance is the larger of 0.05 and 10% of the observed exposure range. The within-group checks prevent observations from one treatment group alone from supporting a contrast requested across both groups.

Both tasks also require at least ten units in each treatment group. These are fixed heuristics, not calibrated tests of population positivity. The original and confirmatory benchmarks use declared support classes; the separate audit in Section 6.8 evaluates the empirical functions themselves.

A failed check returns a structured refusal and a machine-readable reason. Input validation occurs first; support failure takes precedence when support and identification declarations both fail. Figure 6 shows this order. The explicit reason helps the analyst determine whether the query, available data, or assumption statement needs further examination.

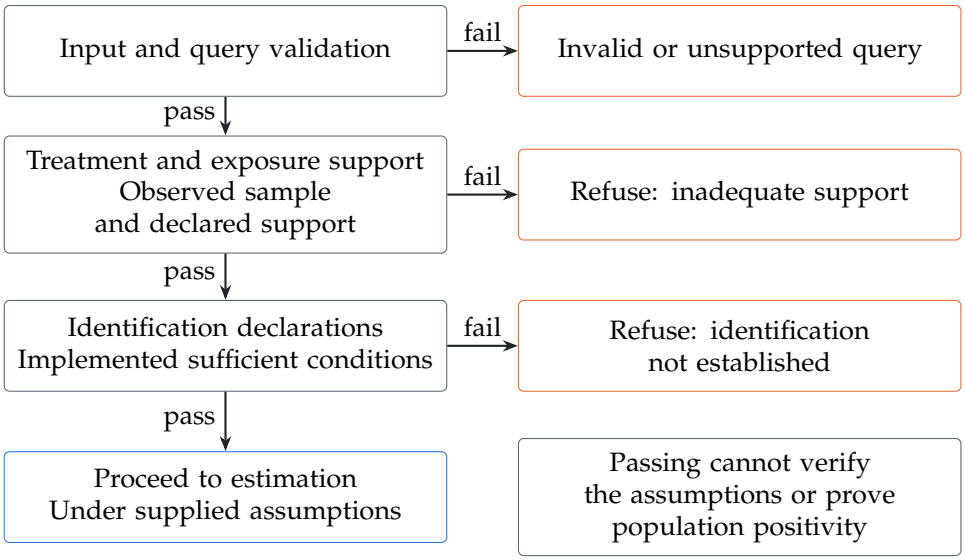

Figure 6. Public validation order and conditional meaning of an accepted request.

Support failure takes precedence when support and identification declarations both fail. Regime determination has a separate structural status rather than a causal-identification claim. Missing or corrupted model files lead to a model-unavailable status before inference.

4.6. A Risk-Routed Library of Classical Estimators

Accepted queries proceed through candidate fitting, aggregation, and interval calibration.

The library contains linear g-computation, a cross-fitted augmented inverse probability weighting (AIPW) estimator with ridge-linear nuisance fits, interaction g-computation, spline g-computation, a piecewise exposure-response candidate, and a design-based or null candidate. Compatibility masks select the applicable candidates. Network cross-fitting uses cluster-level folds when multiple analysis clusters are available. The network candidate occupying the AIPW slot is a cross-fitted linear plug-in, not a derived doubly robust score. Candidates are fitted to each dataset; neural parameters remain fixed.

The candidates represent different outcome shapes, including additive, interacting, smooth nonlinear, and piecewise responses. A nuisance model estimates an intermediate quantity, such as the outcome regression or treatment probability, needed to compute an effect. Cross-fitting fits these models on one part of the dataset and evaluates the relevant predictions on another part. Keeping network clusters together in folds avoids splitting a cluster between these fitting and evaluation subsets.

The *estimator risk router* combines the episode representation with candidate estimates and standard errors to score its preferences among candidates. It converts these learned scores into nonnegative weights:

$$\widehat{\tau}_{\text{compiled}} = \sum_{j=1}^{J} w_j(E,Q)\,\widehat{\tau}_j, \qquad \sum_j w_j = 1, \quad w_j \geq 0, \tag{3}$$

Here, $\widehat{\tau}_{\text{compiled}}$ is the combined scalar estimate, J is the number of candidate estimators, j indexes a candidate, $\widehat{\tau}_j$ is its fitted effect estimate, and $w_j(E,Q)$ is its nonnegative weight for episode E and query Q. Incompatible candidates have weight zero. Figure 7 shows the computation; the confirmatory comparison substitutes simpler aggregation policies without changing the candidates or gate logic.

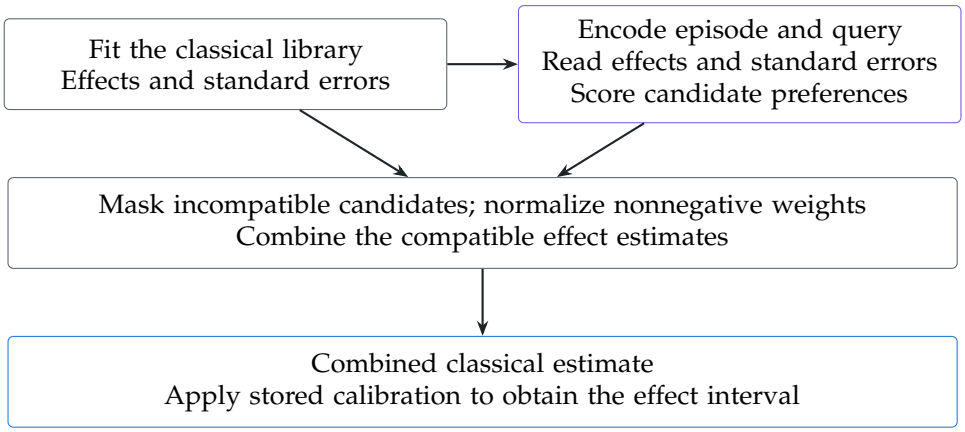

Figure 7. Classical candidate fitting and neural baseline aggregation.

The router learns preferences among candidate estimates. During synthetic training, candidates with smaller errors receive larger target weights. At inference, learned scores are normalized into nonnegative weights that sum to one. The weights describe each candidate's contribution to the combined estimate; they are not probabilities that a causal model is true. Candidate effects and standard errors are available from the submitted data, whereas the true effect used during training is not required at inference. The confirmatory experiment changes this aggregation rule while retaining the same library.

The architecture also permits a gated neural residual, trained with a penalty for degrading the combined estimate. Checkpoint selection applies a development-set non-degradation tolerance. The selected checkpoint precedes residual training, so the reported point estimates are effectively the classical blend. Section 6.7 measures this contribution directly.

4.7. Calibration

Effect intervals use split-conformal calibration on separate episodes. The confirmatory comparison gives every estimator the same scenario-stratified absolute-error score, finite-sample quantile rule, calibration stream, and 90% coverage target.

In that comparison, the aggregation policy is fixed before calibration. It produces an estimate for each calibration episode, and the absolute difference from the known simulated effect becomes a calibration score. Within each scenario, an upper quantile of these scores determines the interval radius, with the finite-sample correction. A test estimate is then extended by the stored radius in both directions. The test episode's true effect is used only to score coverage afterwards. This procedure turns errors on separate, known-effect problems into an uncertainty rule for subsequent episodes.

The original comparison used normalized errors and task strata for CWFM, but absolute errors for its compiled-only comparator and conventional baselines. Equal full and compiled-only point estimates can therefore have different interval coverage in that comparison. The public bundle retains the calibration paired with its checkpoint.

Regime calibration is separate: a threshold maximizes recall subject to a false-positive constraint, with an observed calibration false-positive rate of approximately 0.085. This is an empirical threshold-selection procedure, distinct from the conformal guarantee for effect intervals.

The effect-interval guarantee is *episode-marginal*: for future episodes exchangeable with the calibration stream, coverage is at least the nominal level. It concerns the distribution of complete episodes. Coverage within a particular mechanism or under a different data-generating distribution requires separate evaluation.

The sampling unit for this statement is a complete dataset and its query. Coverage counts how often intervals contain the corresponding true effects across episodes, rather than how many individual rows lie inside an interval. Calibration quantifies estimation uncertainty under its sampling conditions; it does not replace the causal assumptions needed to interpret the effect.

4.8. Reliability Properties and Scope

Table 3 separates the observable behavior of each component from the conditions supporting its interpretation. This distinction makes the framework's guarantees testable: contract enforcement, support screening, and interval coverage each have an explicit evaluation target.

Table 3. Properties of the implemented framework and their conditions.

Component	Observable behavior	Scope and conditions
Identification gate	Withholds an estimate when a required declaration fails	Enforces the ledger; cannot verify its truth or detect undeclared confounding
Support screen	Checks treatment counts and empirical overlap before estimation	A fixed diagnostic; passing does not establish population positivity or model adequacy
Conformal interval	Provides episode-marginal coverage at the nominal level	Requires exchangeability; per-cell and shifted-distribution coverage need separate evaluation
Residual constraint	Selects checkpoints meeting a development non-degradation tolerance	The selected residual is untrained; future non-degradation is not guaranteed
Contract test	Matches declared-invalid cases with structured refusals	Evaluates supplied metadata and rule implementation
Generalization diagnostics	Reports errors on families excluded from gradient training	These families inform selection; automatic prior-shift rejection is not implemented

The three decision layers have distinct evaluation targets: contract compliance, empirical support, and interval coverage. All accepted causal answers remain conditional on the supplied assumptions.

5. Experimental Setup

The experiments separate training, development, calibration, and evaluation data. This section describes the samples, protocols, and Python implementation before the results

in Section 6. A test bank is called held out because it is reserved for evaluation after upstream choices are fixed; its data remain accessible to the evaluator.

The evaluations answer complementary questions. Effect benchmarks compare numerical accuracy and intervals when the declared conditions permit estimation. Stress queries check whether the decision rules enforce those conditions. The empirical support audit examines whether checks computed from observations agree with a known population reference. Reporting these separately distinguishes successful contract enforcement from accurate effect estimation and useful support screening.

5.1. Training and Evaluation Data

The synthetic training distribution specifies which mechanisms the neural baseline encounters and is therefore part of its modeling assumptions.

Synthetic episodes provide known effects and structural labels for controlled evaluation. Each episode samples a graph, mechanism, assignment process, and effect size, then generates observations. Truth labels support training and scoring, while the public inference interface receives observed inputs and declarations. A semi-synthetic experiment adds measured energy-sensor covariates to assess transfer beyond the synthetic covariate distribution.

Episodes come from three task families:

- *Static treatment effects*: datasets have 64 to 128 units, three to eight covariates, a binary treatment, and a continuous outcome. Linear and nonlinear mechanisms cover observed confounding and randomized assignment.
- *Regime determination*: datasets have five to nine covariates. Outcome mechanisms are stationary or change at a covariate threshold, with strong or weak shifts. Stationary smooth nonlinear mechanisms and complete nulls provide controls.
- *Network interference*: a network links units within clusters, and treatment saturation varies across clusters. Exposure is the treated fraction of each unit's neighbors. Spillover mechanisms are linear, threshold-shaped, or exactly null.

Static and network families also include declared hidden confounding and poor support, with gate targets determined by those declarations. Stationary nonlinear controls and complete nulls assess regime decisions. These distinct controls allow contract compliance and structural accuracy to be measured separately.

Training varies row order, column order within role groups, covariate scales, and missingness, masking up to 5% of covariate values. These transformations encourage stable predictions across presentations. Sections 6.1 and 6.7 measure that stability and identify the transformations for which exact invariance remains unresolved.

Disjoint seed ranges separate training, in-distribution development, out-of-distribution development, calibration, and original testing. Figure 8 shows their roles. The optimization run processes 38,400 episodes in 1,600 batches of 24. The selected checkpoint is step 100, after 2,400 episodes. The original test bank is evaluated once training, selection, and calibration choices are fixed.

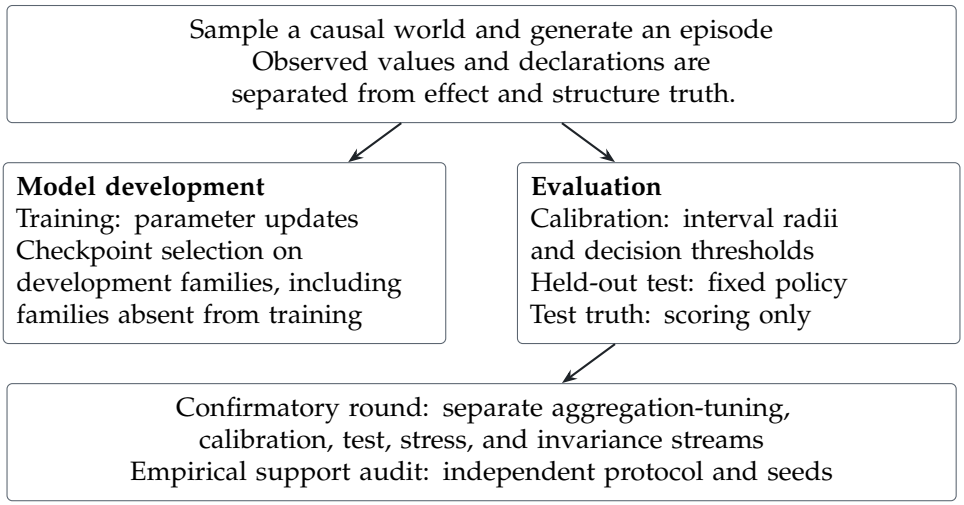

Figure 8. Separation of simulator truth and experimental data streams.

Training updates neural parameters; development selects a checkpoint; calibration sets interval radii and the regime threshold. The held-out test bank then scores the fixed pipeline on 14 scenario cells. Simulator truth is available to these experimental procedures but excluded from public inference inputs.

The out-of-distribution development bank probes hinge, multiplicative, and saturating outcome mechanisms; intercept, variance, and shifted regime changes; alternative network families; and nonmonotone spillovers. These families are excluded from gradient training but inform checkpoint selection. Their results are therefore reported as development diagnostics.

Table 4 summarizes the data sources and sample dimensions. The synthetic evaluation uses 96 units per episode, with six static covariates, seven regime predictors, or four network covariates plus degree. Training instead varies the dimensions within the ranges reported above. Network training uses three to six covariates; all training task families use 64, 80, 96, or 128 units. The external source contains 19,735 time-indexed measurements from one low-energy building, with eight temperature and humidity covariates selected for this study.

Table 4. Data sources and evaluation sample sizes.

Stream or source	Episodes or rows	Units per episode
Synthetic optimization	38,400 episodes; selected checkpoint used 2,400	64 to 128
Original test	420 episodes, 14 cells	96
Aggregation tuning	1,200 episodes, six cells	96
Confirmatory calibration	1,800 episodes, six cells	96
Confirmatory test	1,800 episodes, six cells	96
Confirmatory stress	1,200 episodes, four cells	96
Energy source	19,735 measured rows [51]	Not applicable
Energy calibration and test	200 and 300 semi-synthetic episodes	128
Empirical support audit	40,000 episodes, 80 cells	32 to 512

An episode is one dataset and its query. Synthetic truth is known from the generator. The energy covariates and prognostic response are measured; treatment and effect labels are simulated.

5.2. Pretraining and Constrained Model Selection

The training schedule first optimizes estimator routing with the residual frozen, then trains the gated residual under a non-degradation penalty, and finally applies joint fine-tuning at a low learning rate. Classical candidate estimators are fitted within each episode rather than pretrained as neural parameters.

Checkpoint selection jointly considers effect estimation and structural decisions. Eligibility requires a residual non-degradation tolerance, a false-split cap on stationary nonlinear controls, and continued use of multiple candidates. Eligible checkpoints are ranked by worst-group error on development families. Step 100 satisfies these criteria. Later saved checkpoints have lower unconstrained scores but exceed the 0.30 false-split cap. Refusal remains governed by the deterministic runtime gate.

A checkpoint is a saved snapshot of the neural parameters. Continuing optimization after that snapshot does not mean the later parameters are used for evaluation. Here, the selection rule makes the tradeoff between effect error and false structural alarms explicit. The selected snapshot therefore provides a routing baseline that meets the declared development criteria, even though the full training schedule includes later stages.

The selected checkpoint provides a clear evaluation of pretrained routing over a classical library. It precedes residual training at steps 241 to 1040 and subsequent joint fine-tuning. The residual and world-effect mixture objective consequently remain untrained at this checkpoint; their implemented pathways are treated as exploratory.

The evaluated implementation covers static tabular effects, observed-regime determination, and network interference. Temporal mechanism changes, latent-variable structure, and identification with partially specified graphs remain outside this scope.

5.3. Implementation and Tool Support

The Python implementation exposes the same analysis pipeline to example scripts and a Streamlit application. Named-query adapters map observed columns to model roles. Validation and decision-policy modules enforce input, support, and declaration rules. NumPy and SciPy fit the classical candidates; PyTorch executes the frozen encoder and router; calibration and result modules construct typed outputs. Scikit-learn supplies the conventional random-forest estimator and the shallow confirmatory selector. Figure 9 maps these components.

The analyst selects a query and declares assumptions; fitting, aggregation, and decision reporting are then automated. Network analysis includes an explicit declaration about unmeasured confounding. Support checks preserve earlier treatment-group failures, including missing groups. Accepted results state their dependence on the supplied assumptions.

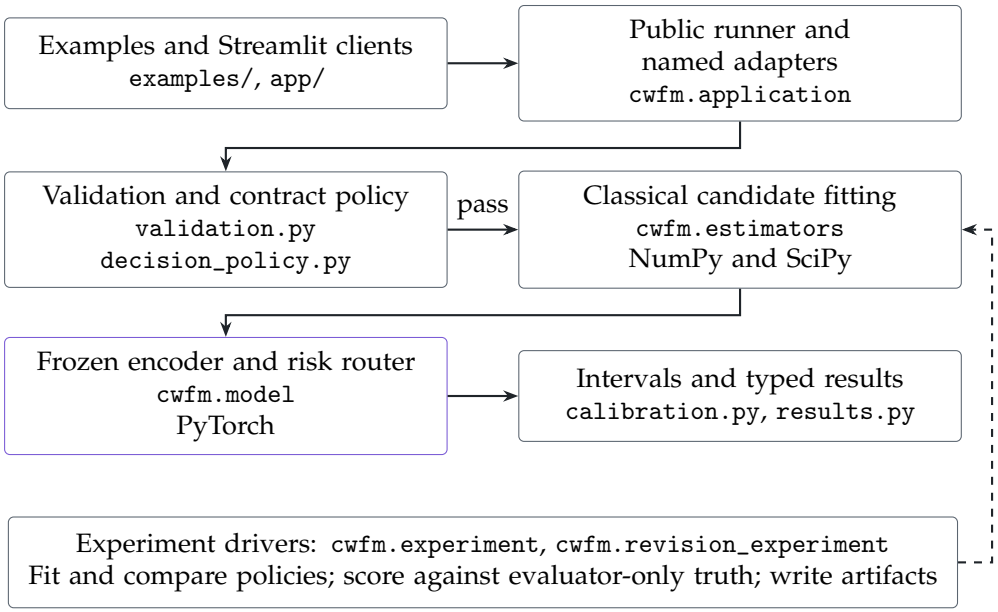

Figure 9. Python components implementing the method and its evaluation.

The runner coordinates input adaptation, empirical validation, inference, and result serialization. Confirmatory drivers use the same candidate-fitting code and declared-contract logic while evaluating alternative aggregators, including a scikit-learn shallow selector. The separate support driver evaluates the public empirical checks without neural inference.

The model bundle records a Secure Hash Algorithm checksum with a 256-bit digest (SHA-256) for its weights and pairs the checkpoint with calibration artifacts. A missing or mismatched model produces `MODEL_UNAVAILABLE`. Example scripts cover static, regime, network, batch, and diagnostic workflows, with reproducible seeds and machine-readable output. A dry-run option validates a contract before inference.

The refusal demonstration, `examples/04_safety_and_abstention.py`, reports accepted queries and declared-condition failures with machine-readable reasons.

Running `streamlit run app/streamlit_app.py` opens views for analysis, case comparison, stored benchmarks, and model diagnostics. All clients call the public runner. The original experiment evaluates generator episodes through the declared-design gate. The confirmatory driver compares aggregation policies using cached candidates and the same declared-contract wrapper. The support audit separately evaluates the public empirical functions and treatment-group count condition.

5.4. Experimental Design

The original held-out test bank contains 14 scenario cells with 30 independent seeds each, shared by all methods (420 episodes). Six cells contain answerable effects: static linear, nonlinear, and randomized treatment effects, and linear, threshold, and null network spillovers (180 queries). Four cells contain regime tasks: strong split, weak split, stationary nonlinear control, and complete null (120 episodes). Four stress cells contain declared hidden confounding or poor support in static and network settings (120 queries). The confirmatory and support studies use separate streams specified below; the development diagnostics are not held-out confirmatory tests.

The original effect comparison includes linear, interaction, spline, piecewise exposure-response, and random-forest g-computation. These are candidate estimators with differing mechanism assumptions. Random-forest g-computation is an outcome estimator, distinct from the shallow random-forest risk selector that chooses among library estimates in the

confirmatory study. The regime comparator is a compact BIC-penalized procedure with at most one linear split, inspired by model-based recursive partitioning [13]. Conclusions about it do not apply to all partitioning methods. CWFM with its neural residual removed is also evaluated to identify the active estimation pathway.

The six RQ4 families were excluded from gradient training but used for checkpoint ranking (Section 5.1). Each family contributes four development episodes; Section 6.11 reports their individual errors. The neural baseline has hidden dimension 96, three attention blocks, and four neural mechanism modules; the classical library has six members.

The full neural model contains 917,040 parameters, including the exploratory heads. Its float32 parameters occupy 3.67 MB, and the saved checkpoint occupies 3.74 MB (decimal units). The original run took 2,000.5 seconds, or 33.3 minutes, on one graphics processing unit (GPU); its model was not recorded. It performed 1,600 updates with 24 episodes per batch, processing 38,400 training episodes. The selected step-100 checkpoint follows 2,400 episodes. These counts distinguish the cost of the full optimization schedule from the selected snapshot. Classical candidates are fitted separately for each input dataset and are not included in the neural parameter count.

Acceptance is reported separately from estimation error. All 180 answerable original effect queries and all 1,800 confirmatory effect queries receive estimates. The 120 original and 1,200 confirmatory stress queries belong to separate cells with failed declarations. Within the answerable banks, the gate does not remove episodes according to their difficulty or prediction error; all compared policies are scored on the same queries. These counts describe the declared-contract benchmarks. The public empirical screen has separate finite-sample acceptance behavior, evaluated in Section 6.8.

Effect accuracy is measured by MAE against the known generator effect. Interval evaluation records empirical coverage and mean width at the 90% target. Contract evaluation counts estimates returned when the declared conditions fail. Regime structure accuracy requires the correct split variable or a correct no-split decision; threshold localization is not scored. Method differences use paired bootstrap confidence intervals over shared seeds. Manifests record seeds, configurations, package versions, and checkpoint hashes to support reproduction, subject to numerical differences across environments.

MAE averages the magnitude of effect-estimation errors, so positive and negative errors do not cancel. Coverage and width describe the uncertainty interval attached to each effect estimate. A paired bootstrap interval instead summarizes uncertainty in the average performance difference between methods. Pairing keeps each method's results on the same episodes together, allowing comparisons to account for shared variation in episode difficulty.

5.5. Confirmatory Protocol and Comparators

The confirmatory protocol fixes seed streams, methods, hyperparameter policies, coverage targets, and bootstrap comparisons in `experiments/revision_protocol.json`. Its hash and the released checkpoint's SHA-256 are recorded in the manifest. The estimator-layer repair described in Section 6.7 was completed before drawing these streams. Neural parameters remain fixed.

Separate streams contain 1,200 aggregation-tuning episodes, 1,800 calibration episodes, 1,800 confirmatory effect queries, and 1,200 stress queries. The effect test has 300 episodes in each of six scenarios. Presentation probes and an external semi-synthetic benchmark provide additional evaluations.

Comparators.

Applicable methods share the same episodes and calibration protocol. The comparison includes individual compatible library candidates and four conventional estimators: linear and random-forest g-computation, cross-fitted AIPW with ridge-linear nuisance models, and a cross-fitted doubly robust learner with spline nuisance models.

Nine policies combine the same fitted library. Neural variants use the frozen soft blend, its highest-weight candidate, or temperature-scaled weights. Simple rules use the uniform mean, median, or best fixed candidate from the tuning stream. Learned alternatives are a metadata-only ridge selector, a shallow random-forest risk selector, and a nonnegative convex stack fitted by task. The shallow selector reads candidate estimates and standard errors as well as metadata. Per-cell and per-episode oracles select candidates using unavailable truth and serve as reference bounds.

Every method uses scenario-stratified absolute-error split-conformal intervals, the finite-sample quantile rule, the same calibration stream, and a 90% target. This shared construction supports direct comparison of the aggregation policies.

The shallow selector and convex stack use the library differently. The selector predicts each eligible candidate's absolute error from the episode summaries and returns the candidate with the smallest predicted error. Its choice can vary between datasets. The convex stack instead learns nonnegative weights for each task on the tuning stream and reuses those weights to combine new estimates. Both policies are fitted before calibration and testing. Their distinction is therefore adaptive candidate choice versus a learned combination shared within a task.

5.6. External Semi-Synthetic Design

The external source is the University of California, Irvine (UCI) Appliances Energy Prediction dataset [51]. The analysis selects T1, RH_1, T2, RH_2, T3, RH_3, T_out, and RH_out without using treatment effects. These temperature and humidity covariates are standardized, and standardized log-transformed appliance energy use supplies the prognostic component. The fixed simulator adds a covariate-dependent logistic treatment assignment, a heterogeneous treatment effect, and outcome noise. Thus measured outcomes do not provide observed intervention ground truth; the known effect comes from the added simulation.

Each episode has 128 units. Even source rows supply 200 calibration episodes; odd rows supply 300 test episodes. The treatment and effect equations were fixed before evaluation, and the aggregation policies learned from synthetic tuning episodes remain fixed. Source pools are disjoint, but adjacent sensor measurements are temporally related and episodes can reuse rows. The design is a semi-synthetic transfer check, not independent validation of a real intervention or a new building.

The design introduces covariate patterns from measured sensors while preserving a known effect for scoring. The prognostic component represents baseline outcome variation before the simulated treatment contribution is added. This allows the study to examine how policies learned on synthetic tasks perform with measured covariates, without treating an observed energy association as an established intervention effect.

5.7. Finite-Sample Empirical Support Audit

The direct support audit applies the public screening functions and minimum treatment-group count condition to 40,000 datasets. Its fixed protocol, `experiments/support_protocol.json`, uses sample sizes 32, 64, 128, 256, and 512, with 500 independent replicates per cell. Seed 9102026 is combined with family, size, severity, and replicate

indices. This inexpensive experiment isolates support decisions, with thresholds fixed and no neural inference or effect fitting.

Static episodes contain five independent standard-normal covariates and a binary treatment. The two treatment-assignment models are

$$e_s(x) = \sigma(sx_1) \quad \text{or} \quad e_s(x) = \sigma\big(s(x_1^2 - 1)\big), \qquad s \in \{0, 0.5, 1, 2, 4, 8\}. \tag{4}$$

Here, $e_s(x)$ is the probability of treatment conditional on covariate vector x, x_1 is its first coordinate, s controls assignment severity, and $\sigma(z) = 1/(1 + \exp(-z))$ is the logistic function for a real-valued argument z, with exp denoting the exponential function. Treatment is sampled independently conditional on these probabilities. The linear case probes sampling behavior; the quadratic case probes model misspecification. An outcome column is supplied but never used by either screen.

The population reference applies the static gate's tail policy to the known propensity distribution: at most 10% of true probabilities outside $[0.05, 0.95]$, a fifth percentile above 0.01, and a 95th percentile below 0.99. These quantities are computed analytically from the normal or chi-squared distribution. Slopes zero, 0.5, and one meet the policy; slopes two, four, and eight fail it in both families. All finite logistic slopes retain strict positivity, so this reference concerns practical overlap under a specified policy, not mathematical non-identification.

Network episodes use a ring whose units have neighbors at offsets minus two, minus one, one, and two. Treatments are independent Bernoulli draws with probabilities 0.5, 0.3, 0.1, or 0.02. Exposure is the treated fraction of the four neighbors and is marginally binomial with four trials, divided by four; own treatment is independent of own exposure. The requested exposure contrast is 0.25 to 0.75. The population reference requires these targets inside the fifth-to-95th percentile range of that known exposure distribution. Probabilities 0.5 and 0.3 pass; 0.1 and 0.02 fail. Minimum nearby counts and treatment-group sizes are finite-sample restrictions, not population properties. Exposure dependence between neighboring units is preserved.

A false refusal is rejection in a cell meeting its population reference; a false acceptance is acceptance in a cell failing that reference. Thus a "positive" support-failure decision has false-positive rate equal to the false-refusal rate. We report both directions to avoid ambiguity. Refusal rates use all 500 replicates in each cell. Their pointwise 95% Wilson intervals treat episodes, not units, as independent observations. The complete design comprises 30,000 static and 10,000 network episodes. This audit characterizes support decisions only; it does not estimate causal error conditional on acceptance.

5.8. Pretraining Ablation and Router Diagnostics

We isolate the contribution of neural pretraining by comparing the selected checkpoint with five untrained copies of the same architecture, initialized with seeds 1729 to 1733. Each copy receives the same observed inputs, fitted classical estimates, compatibility masks, and declared gate. No copy receives test-time optimization. Residual effects are below 2.1×10^{-7} in magnitude in all six models, so the comparison isolates their estimator weights to numerical precision. Simulator targets are excluded from model inputs and used only for scoring.

The analysis reuses the 1,800 confirmatory effect episodes, with 300 per scenario. It is a post-review ablation on an existing bank, rather than a newly independent confirmatory study. Uniform weights provide a control without learned aggregation. A second control fixes pretrained weights to their mean on the separate 1,200-episode tuning stream, grouping by task and compatibility mask. This comparison tests whether episode-specific routing improves on the learned average preferences. We report point

errors, all initialization results, weight summaries, and paired bootstrap intervals from 4,000 resamples stratified by scenario. The intervals condition on the selected checkpoint and fixed initializations; they do not measure training-run stability. Protocol, code, timings, and per-episode records are supplied in `experiments/pretraining_protocol.json`, `cwfm/pretraining_experiment.py`, and `artifacts/cwfm/pretraining_revision/`.

6. Results

6.1. Confirmatory Comparison of Estimator-Selection Policies

The shallow risk selector and convex stack achieve pooled MAEs of 0.141 and 0.147 on 1,800 fresh effect queries, compared with 0.161 for the frozen deep router. Figure 10 shows the nine aggregation policies, and Table 5 provides coverage, worst-cell error, and paired comparisons. All policies use the same fitted library and declared-contract wrapper.

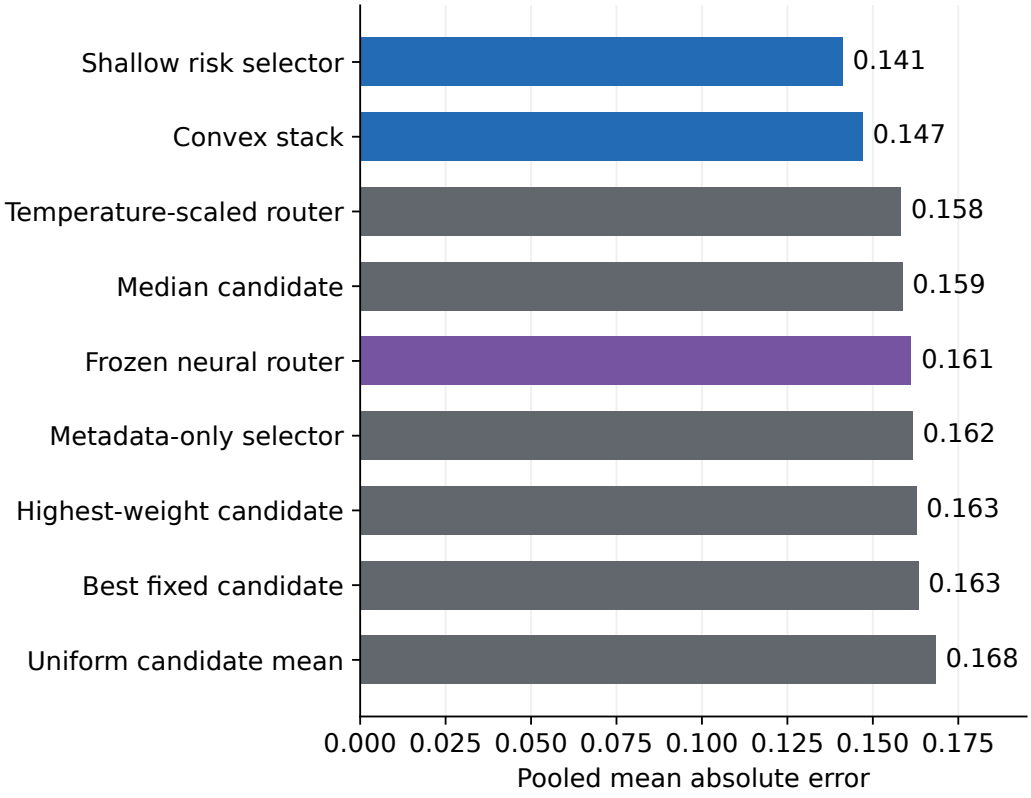

Figure 10. Confirmatory effect error for the nine aggregation policies.

Each bar uses the same 1,800 queries. The shallow selector and convex stack combine the existing candidate library and have the lowest pooled errors among these policies. The deep router provides the neural reference. Paired confidence intervals appear in Table 5.

Table 5. Confirmatory comparison on 1,800 fresh effect queries.

Method	MAE	Worst cell	Coverage	Router − method (95% CI)
Shallow risk selector	0.141	0.191	0.888	+**0.0200** [0.0152, 0.0249]
Convex stack	0.147	0.179	0.894	+**0.0142** [0.0111, 0.0171]
Temperature-scaled router	0.158	0.295	0.888	+**0.0030** [0.0014, 0.0046]
Median candidate	0.159	0.303	0.888	+**0.0026** [0.0004, 0.0047]
Frozen neural router	0.161	0.286	0.886	(reference)
Metadata-only selector	0.162	0.307	0.882	−0.0004 [−0.0034, 0.0026]
Highest-weight candidate	0.163	0.308	0.886	−0.0015 [−0.0048, 0.0016]
Best fixed candidate	0.163	0.307	0.879	−0.0021 [−0.0053, 0.0010]
Uniform candidate mean	0.168	0.279	0.888	−**0.0071** [−0.0087, −0.0056]
Oracle: per-cell candidate	0.117	0.166	Not applicable	Not applicable
Oracle: per-episode candidate	0.067	0.092	Not applicable	Not applicable

MAE: mean absolute error. Worst cell: largest per-scenario MAE. Coverage: empirical coverage at the 90% target. CI: scenario-stratified paired bootstrap confidence interval. The last column is router error minus method error, so positive differences favor the listed method. Bold differences have intervals excluding zero. Oracles use unavailable truth and cannot be deployed. Not applicable denotes quantities not evaluated for oracle policies.

Accuracy.

Relative to the frozen router, the shallow selector reduces pooled MAE by 0.0200, with a 95% paired interval of $[0.0152, 0.0249]$. The convex stack reduces it by 0.0142, with an interval of $[0.0111, 0.0171]$. Temperature scaling and the median also have lower errors with paired intervals excluding zero. Threshold-spillover MAEs are 0.191 for the shallow selector, 0.179 for the stack, and 0.286 for the router.

Paired intervals include zero for the router's comparisons with the best fixed candidate, highest-weight candidate, and metadata-only selector. The additional AIPW and spline-nuisance doubly robust comparators apply to the 900 static queries, with MAEs 0.373 and 0.211. Those subset results have a different evaluation population from the six-cell pooled comparison.

Calibration.

Pooled coverage is 0.888 for the shallow selector, 0.894 for the stack, and 0.886 for the frozen router at the 0.9 target. The router's 95% Wilson interval is $[0.871, 0.900]$. Across all evaluated methods, coverage ranges from 0.879 to 0.940. Several methods have coverage between 0.82 and 0.87 in the static nonlinear cell. These realized rates are reported alongside the pooled results because scenario-stratified calibration does not ensure nominal coverage in every finite test sample.

Contract compliance.

The shared gate correctly refuses all 1,200 stress episodes with declared poor support or hidden confounding. Each of the 1,200 distinct decisions is recorded under 13 method

labels, giving 15,600 statuses. The ungated indicator is true by definition; numerical effects are not recomputed for every label. The result establishes reproducible enforcement of the supplied contract independently of the aggregation policy.

Presentation stability.

The confirmatory probes contain 300 episodes per transformation, except treatment-label recoding, which has 150 eligible episodes. After predictions are mapped back to the original scale, maximum absolute differences are 2.9×10^{-6} for row permutation, 2.4×10^{-7} for within-role covariate permutation, 2.4×10^{-7} for outcome translation, and 8.4×10^{-5} for positive rescaling. These four transformations pass tolerance 10^{-5} in 97.25% of cases.

Mean changes are 0.019 for irrelevant-covariate insertion (maximum 0.278), 0.023 for negative outcome scaling, and 0.010 for treatment recoding. The latter two are unresolved equivariance errors. The column-permutation probes keep query-role indices fixed, so they do not test arbitrary relocation of treatment and outcome columns.

External semi-synthetic results.

Simple policies also achieve lower errors than the frozen router on the 300 energy-based test episodes. MAEs are 0.157 for the linear candidate, 0.160 for the convex stack, 0.162 for the temperature-scaled router, 0.163 for the median, 0.165 for the shallow selector, and 0.171 for the frozen router. Paired intervals favor the temperature-scaled, median, and stacked policies over the router. Coverage ranges from 0.883 to 0.930, with 0.913 for the router. The design uses measured covariates and simulated treatment effects from one building.

The complete confirmatory tables, per-seed results, and protocol are recorded in `artifacts/cwfm/revision/` and `experiments/revision_protocol.json`.

6.2. RQ1: Accuracy on Answerable Queries

On the original 180 answerable queries, CWFM has pooled MAE 0.166, compared with 0.167 for interaction g-computation, 0.169 for linear g-computation, and 0.329 for random-forest g-computation. The paired differences are -0.0028 with 95% bootstrap interval $[-0.0132, 0.0080]$ against linear g-computation, -0.0003 with $[-0.0111, 0.0109]$ against interaction g-computation, and -0.163 with $[-0.203, -0.122]$ against random-forest g-computation.

Pooled coverage is 0.906 at the 90% target, with mean width 0.793. Other fixed-baseline widths range from 0.772 to 0.817, and the random-forest width is 1.337. Per-cell coverage ranges from 0.900 to 0.967 except for threshold spillovers, with 0.700 (21 of 30; 95% Wilson interval 0.52 to 0.83). The confirmatory comparison uses the subsequently adopted scenario-stratified calibration protocol.

Table 6 shows how candidate accuracy varies with the mechanism. The best method differs across cells; for threshold spillovers, the piecewise candidate has MAE 0.173 compared with 0.283 for CWFM. The per-cell oracle over the listed methods reaches approximately 0.132, while the library oracle reaches approximately 0.122 because its null candidate has zero error in the network-null cell. The router's pooled MAE is 0.166, and it is not the lowest-error method within any original cell. The confirmatory experiment evaluates practical aggregation policies on fresh data.

Table 6. Effect error by scenario in the original test bank (30 seeds per cell).

Scenario	CWFM	Linear	Interaction	Spline or piecewise	Random forest
Static linear	0.168	**0.160**	0.189	0.181	0.412
Static nonlinear	0.190	0.218	**0.163**	0.169	0.502
Static randomized	0.150	**0.141**	0.154	0.145	0.449
Network linear	0.107	**0.097**	0.104	0.201	0.207
Network threshold	0.283	0.311	0.312	**0.173**	0.351
Network null	0.100	0.090	0.079	0.206	**0.055**

Entries are mean absolute errors. Bold marks the lowest error per row. The combined column uses spline g-computation for static rows and piecewise exposure response for network rows.

6.3. Contextual Comparison with Published Causal Models

Table 7 places the results in the context of published causal foundation models and amortized discovery methods. Precision in estimation of heterogeneous effects (PEHE) measures individual-effect error, while the present study evaluates population-effect MAE. Since the cited systems use different estimands and benchmark distributions, the table compares evaluated targets rather than numerical accuracy.

Table 7. Principal causal targets in the cited model evaluations.

Model	ATE	CATE/ITE	Do-query	Graph	Principal published evaluation
CWFM	✓				Population effects
CausalPFN [8]	✓	✓			Treatment effects
Do-PFN [7]	✓	∼	✓		Interventional queries
CausalFM [6]	∼	✓	∼		Conditional effects
AVICI [31]				✓	Graph recovery
ACD [29]				✓	Temporal graph recovery

✓: principal evaluated target. ∼: capability evaluated in some experiments. Blank: not a principal target in the cited evaluation. ATE: average treatment effect; CATE: conditional average treatment effect; ITE: individual treatment effect; PEHE: precision in estimation of heterogeneous effects. AVICI: amortized variational inference for causal discovery; ACD: amortized causal discovery. A do-query requests an interventional distribution.

A shared dataset, estimand, split protocol, and calibration target would enable a numerical comparison across these systems. The interface and target tables identify the common tasks on which such a comparison could be organized.

6.4. RQ2: Enforcement of the Declared Contract

CWFM returns a structured refusal for each of the 120 original stress queries whose declared identification or support conditions fail. The always-answer comparators return estimates. The confirmatory test extends the same contract check to 1,200 queries (Section 6.1).

The status and reason consistently match the supplied contract in both evaluations. This tests enforcement of declared conditions; the separate audit in Section 6.8 evaluates empirical support decisions from observations.

6.5. RQ3: Regime Detection and False Splits

Table 8 reports pooled structure accuracy of 0.617 for CWFM and 0.458 for the one-split linear baseline. On stationary nonlinear controls, CWFM selects the correct no-split

structure in 83.3% of episodes; the baseline splits every episode. Both select no split in every complete-null case.

On strong splits, accuracy is 0.633 for CWFM and 0.800 for the baseline. Weak-split accuracy is 0.000 and 0.033, respectively. These results use the calibrated classical evidence rule in Section 4.4; the neural split head is diagnostic. Structure accuracy scores the splitting variable or no-split decision, without threshold localization.

Table 8. Regime determination in the original test bank (30 seeds per cell).

	CWFM		One-split baseline	
Scenario	**Structure**	**Split rate**	**Structure**	**Split rate**
Strong split	0.633	0.633	**0.800**	0.833
Weak split	0.000	0.000	**0.033**	0.033
Stationary nonlinear (hard negative)	**0.833**	0.167	0.000	1.000
Complete null	**1.000**	0.000	**1.000**	0.000
Pooled structure accuracy	**0.617**		0.458	

Structure denotes the proportion selecting the correct splitting variable or correctly selecting no split. Threshold localization is not scored. Split rate is the proportion declaring a split. Bold marks the higher structure accuracy per row; pooled accuracy weights the four cells equally.

6.6. RQ4: Mechanisms Excluded from Gradient Training

Figure 11 reports effect MAE for six families excluded from gradient training but used in checkpoint selection, with four episodes per family. Errors are 0.157 for hinge outcomes, 0.172 for multiplicative outcomes, 0.163 for Erdős-Rényi random graphs, 0.307 for saturating outcomes, 0.361 for small-world graphs, and 1.357 for nonmonotone interference.

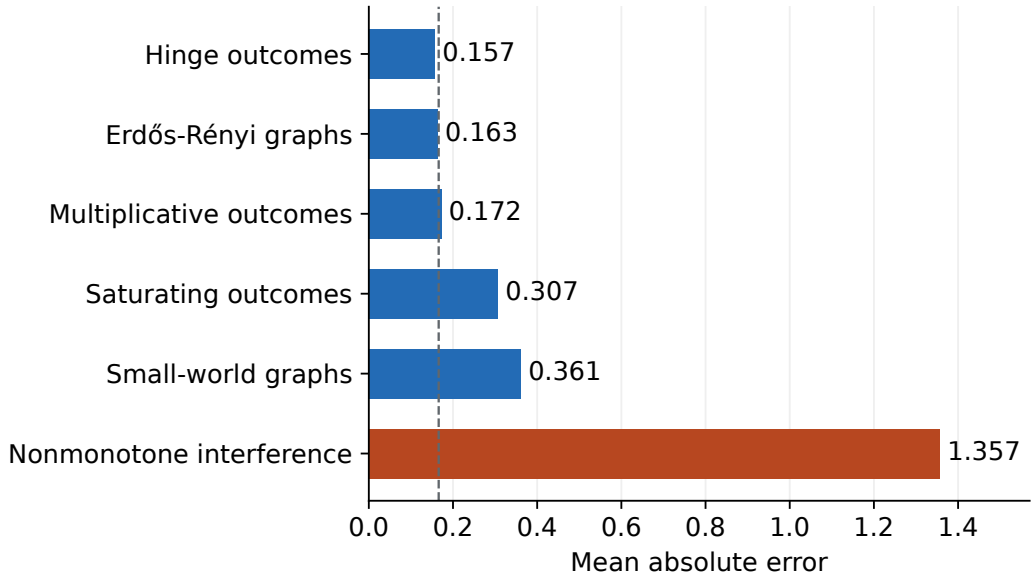

Figure 11. Effect error on families excluded from gradient training.

The dashed line marks the original pooled MAE of 0.166. Hinge, multiplicative, and random-graph cases have errors near this reference, while the largest error occurs for nonmonotone interference. Each bar uses four development episodes from families that also informed checkpoint selection.

All six families satisfy their declared contracts and receive answers. Their varying errors characterize sensitivity to mechanisms excluded from gradient training. Because

these families also informed selection, this evaluation is a development diagnostic; an independent distribution-shift test remains a separate requirement.

6.7. RQ5: Contribution of Computational Pathways

The reported effects can be traced to the classical blend: full and compiled-only CWFM estimates agree to numerical precision. Mean residual-gate activation is 0.0067, and mean absolute correction is below 3×10^{-8}. The checkpoint precedes residual training, and world-particle scores are effectively uniform. These measurements identify the active computation at the evaluated checkpoint.

The recorded weights show contributions from all six candidate estimators. In threshold-spillover episodes, the piecewise candidate receives about 15% of the weight and has MAE 0.173, compared with 0.283 for the blend. The null candidate receives about 15% in the network-null cell, with zero error compared with 0.100 for the blend.

Table 9 records the original single-episode checks that motivated the estimator-layer repair. Row shuffling and positive rescaling already produce differences near numerical precision; irrelevant-covariate insertion changes the estimate by 0.039 and covariate reordering by 0.170. Cross-fitting previously used lexicographic covariate ordering, and the hinge basis used only the first three covariates. Both dependencies were removed before the confirmatory experiment, which recalibrates intervals on fresh data. The resulting stability is reported in Section 6.1; the neural query-index inputs remain as described in Section 4.3.

Table 9. Original presentation probes on one static linear episode.

Probe	Absolute change in estimate
Row shuffling	1.2×10^{-7}
Rescaling the outcome by a positive factor	4.9×10^{-8}
Insertion of an irrelevant covariate	0.039
Covariate reordering within roles	0.170

Entries are absolute differences after transforming the estimate back to the original outcome scale when needed. Adding a covariate changes the input dimension. These single-episode values precede the estimator-layer repair and are not averages over the confirmatory invariance bank.

6.8. Empirical Support: Refusals and Screening Errors

Figure 12 and Table 10 report the direct empirical audit. In the linear-assignment family with slope one, false refusals fall from 0.528 at 32 units to 0.106 at 128 and zero at 512. With slope two, false acceptance falls from 0.112 to 0.066 and 0.016 at the same sizes. Slopes four and eight are refused in every replicate.

In the quadratic-assignment family, all 500 episodes at slope two are accepted at both 128 and 512 units despite failing the population overlap reference. At slope eight, false acceptance is 0.988 at 128 units and 1.000 at 512.

For the network reference with treatment probability 0.5, false refusal falls from 0.976 at 32 units to 0.006 at 128 and zero at 512. With probability 0.3 it remains 0.246 at 512 units. Probabilities 0.1 and 0.02 produce refusal in every replicate at every size. Zero observed events in 500 replicates still have an upper 95% Wilson bound of approximately 0.008. These empirical error rates differ from the zero contract-violation rate in the declared-class benchmarks.

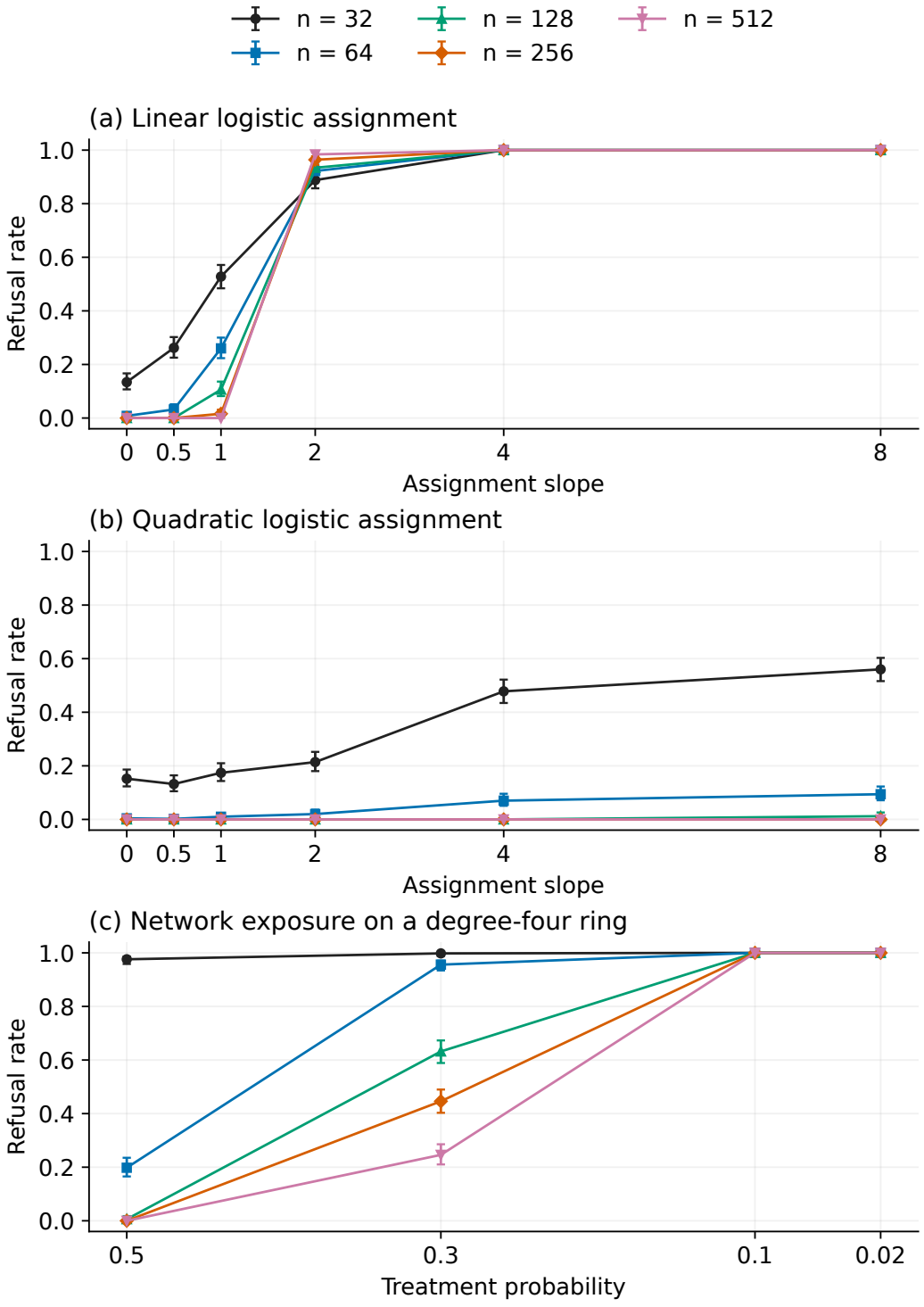

Figure 12. Empirical support refusal rates by sample size and assignment severity.

Each point summarizes 500 independent episodes; error bars are pointwise 95% Wilson intervals. Lines distinguish sample sizes, with n denoting the number of units. The first two panels use increasing logistic slopes, while the network panel moves toward rarer treatment from left to right. Refusal in population-adequate cells is false refusal; acceptance in population-inadequate cells is false acceptance under the reference policies of Section 5.7.

Table 10. Selected empirical support error rates (500 episodes per cell).

Family, severity	**Error type**	$n = 32$	$n = 128$	$n = 512$
Linear, 1	False refusal	0.528	0.106	0.000
Linear, 2	False acceptance	0.112	0.066	0.016
Quadratic, 2	False acceptance	0.786	1.000	1.000
Quadratic, 8	False acceptance	0.440	0.988	1.000
Network, 0.5	False refusal	0.976	0.006	0.000
Network, 0.3	False refusal	0.998	0.632	0.246
Network, 0.1	False acceptance	0.000	0.000	0.000

Severity is the logistic slope for static families and treatment probability for the network. n is the number of observed units. False refusal is rejection in a population-adequate cell; false acceptance is acceptance in a population-inadequate cell. All sizes, severities, rates, and interval endpoints are available in the support-audit artifacts.

6.9. Qualitative Case Studies

The worked examples show how fitted models recover mechanisms on individual datasets. They use companion classical pipelines in the repository. Representative success cases are selected by the seed closest to the median primary metric, with generator truth used for evaluation; one failure case is included explicitly. Temporal examples illustrate a potential extension beyond CWFM's implemented tasks.

6.9.1. Observed-Regime Recovery

The representative mechanism splits at $X_5 \leq 0.000$, and the fitted tree selects the correct variable with threshold 0.027. Here, X_5 denotes the fifth predictor. Across twelve predictor-outcome relationships, estimated coefficient changes preserve every sign except the smallest, essentially null entry. The largest absolute error is 0.104. This companion-pipeline example illustrates recovery of both the splitting variable and local coefficient changes.

6.9.2. Local-Model Misspecification

Figure 13 compares two analyses of one dataset generated by a smooth quadratic mechanism without a regime change. A correctly specified quadratic local model yields no split (permutation $p = 0.6$). A linear local model yields four regions (permutation $p = 0.05$). Both fitted curves approximate the same smooth mechanism. The example corresponds to the stationary nonlinear control used in the regime evaluation.

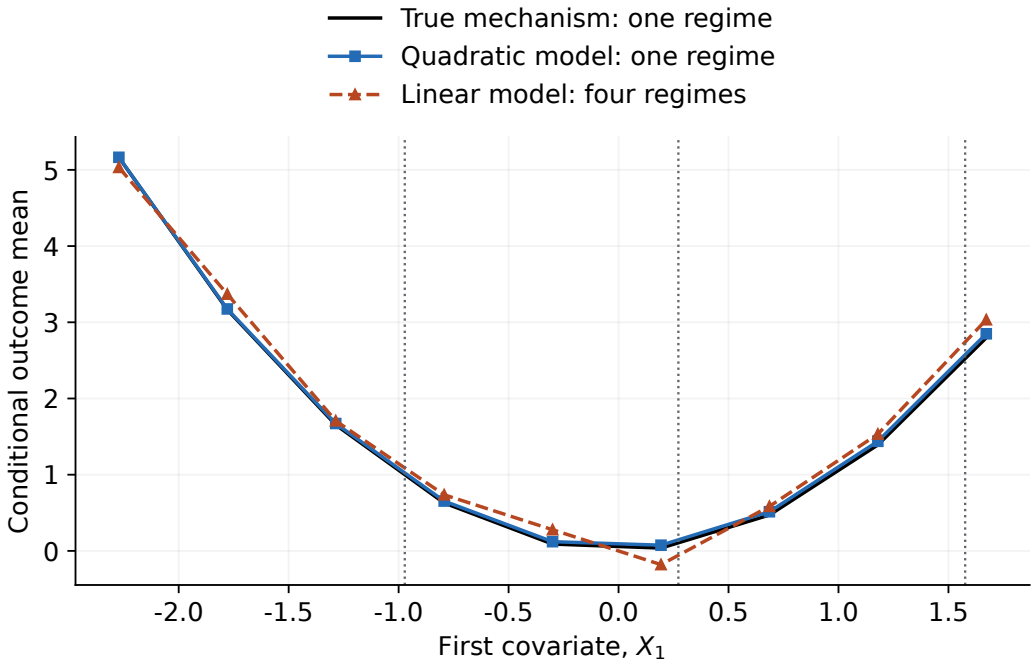

Figure 13. Quadratic mechanism and two fitted approximations for one dataset.

The horizontal axis is the first covariate, X_1, and the vertical axis is the conditional outcome mean. The quadratic model retains one regime. The linear model follows the curve using four regions separated by the vertical dotted lines. Similar fitted responses can therefore accompany different structural decisions.

6.9.3. Distinguishing Change Types over Time

The temporal examples examine whether a detected change is assigned to the correct mechanism. A true slope change is retained, while a matched intercept-only change is not classified as a slope change ($p = 0.705$). Vector autoregression with exogenous inputs (VARX) gives a similar distinction: a lagged-coefficient change is retained, and a contemporaneous-only change is rejected ($p = 0.951$). In the included failure case, an intercept-only change is classified as a lagged-coefficient change ($p = 0.032$), despite small estimated coefficient differences. These are companion-pipeline illustrations rather than CWFM temporal evaluations.

6.10. Contribution of Pretraining and Learned Routing

Table 11 compares pretrained and untrained routing on the same episodes. Pretraining reduces pooled MAE from 0.1673, averaged across five untrained initializations, to 0.1612. The reduction is 0.0061, with paired 95% interval [0.0047, 0.0076], or 3.7% relative to the untrained mean. The five untrained pooled MAEs range from 0.1654 to 0.1711. Uniform aggregation yields 0.1683. Pretraining lowers error in five of the six scenario means; threshold interference favors untrained and uniform aggregation.

Table 11. Effect errors with and without neural pretraining.

Scenario	Pretrained	Untrained	Uniform	Fixed mean
Static linear	0.1548	0.1654	0.1651	0.1553
Static nonlinear	0.1830	0.2072	0.2093	0.1833
Randomized	0.1537	0.1565	0.1558	0.1539
Network linear	0.1028	0.1089	0.1099	0.1024
Network threshold	0.2859	0.2733	0.2790	0.2905
Network null	0.0870	0.0926	0.0907	0.0872
Pooled	0.1612	0.1673	0.1683	0.1621

Entries are mean absolute errors over 300 episodes per scenario; all episodes are answered. Untrained averages errors across five random initializations, without averaging their predictions. Uniform uses equal compatible weights. Fixed mean uses pretrained weights averaged on separate tuning episodes with the same task and compatibility mask.

The fixed-mean control attains pooled MAE 0.1621. Episode-specific routing reduces it by 0.0009, with paired 95% interval [0.0003, 0.0015]. Learned weights remain distributed across candidates: mean routing entropy divided by the logarithm of the eligible candidate count ranges from 0.977 to 0.987 across scenarios, where one denotes uniform weights. In static linear and nonlinear cells, mean interaction weights are 0.238 and 0.236, whereas augmented inverse probability weighting receives 0.110 and 0.111. In network cells, interaction weights average 0.230 to 0.234 and spline weights 0.114 to 0.126. The released records provide every weight, candidate estimate, initialization result, and paired comparison.

The complete audit took 126.7 seconds on an NVIDIA L40S with batch size 32 and four CPU threads. It includes 1,200 tuning and 1,800 test episodes, with candidate fitting shared across models. Pretrained forward passes over both streams took 31.2 seconds; generating episodes and fitting candidates took 3.2 seconds. These are batched audit timings, not interactive latency measurements. The pretrained estimates reproduce the earlier confirmatory records within 4.4×10^{-7}, and all neural states remain unchanged.

6.11. Individual Development-Episode Results

Table 12 reports every episode behind the six RQ4 effect means in Figure 11. The values are exported directly from the saved step-100 diagnostics, preserving the historical estimator implementation. All 24 episodes receive estimates. The released CSV also contains targets, predictions, and reconstructed generation seeds, together with the 12 development regime records. These families informed checkpoint selection, and four episodes per family provide descriptive checks rather than a precise estimate of independent generalization performance.

Table 12. Individual effect errors in the development families.

Family	Episode 1	Episode 2	Episode 3	Episode 4	Mean
Hinge	0.108	0.049	0.117	0.355	0.157
Multiplicative	0.016	0.245	0.296	0.130	0.172
Saturation	0.635	0.343	0.188	0.062	0.307
Small-world graph	0.120	0.559	0.343	0.423	0.361
Random graph	0.103	0.114	0.390	0.044	0.163
Nonmonotone interference	1.863	1.653	1.265	0.646	1.357

Episode columns give absolute effect errors in the saved generation order. Means use unrounded errors. Random graph denotes the Erdős-Rényi family. These are development observations used in checkpoint selection, not an independent test bank.

7. Discussion

7.1. Interpretation of the Results

The experiments support a practical approach to causal automation: combine an inspectable statistical library with explicit decision rules, and select the aggregation policy by evidence. The shallow selector reduces pooled MAE by 12.4% relative to the deep router, and the convex stack by 8.8%. Their paired intervals exclude zero. These gains show that the framework can benefit from simple aggregation while retaining the same candidates, query contract, and calibration protocol.

The pretraining ablation gives the neural baseline a concrete role: it learns useful candidate preferences beyond random initialization or equal weights. The fixed-mean control accounts for most of this gain, while episode-specific weighting adds a smaller improvement. Thus the evaluated router mainly learns broad aggregation preferences, with limited adaptation to individual datasets. Its weights describe contributions to an estimate, not probabilities that a causal model is true. The stronger shallow policies in the main comparison remain the most effective evaluated use of the shared library.

The semi-synthetic energy results extend this finding to measured covariates: the convex stack and other simple policies improve on the frozen router, and the linear candidate has the lowest error. The original evaluation provides complementary evidence. CWFM's pooled error is close to the strongest fixed estimators, and its calibrated regime rule reduces spurious splitting on stationary nonlinear controls. The regime results also make the sensitivity tradeoff explicit: the one-split baseline detects more strong changes.

The decision layer contributes a separate, testable property. Every declared-invalid stress query is refused, with the decision determined by the contract rather than the estimator. This separation lets an analyst inspect why an estimate was returned or withheld and allows aggregation policies to be compared under the same rules. As detailed in Section 4.5, truthful and complete metadata remain a fundamental condition; the gate cannot verify the underlying causal assumptions.

The support audit provides concrete guidance about the empirical screen. For several linear and network settings, false refusals decline as sample size grows, while severe support failures are consistently detected. Quadratic assignment exposes a different issue: a misspecified linear-probability model can accept poor overlap even with more data. The audit therefore identifies where the current inexpensive screen is useful and where a more flexible, independently validated diagnostic is needed. Its results are reported with fixed thresholds rather than used to tune the same benchmark.

The quadratic assignment helps explain this distinction. Treatment probability rises at both large positive and large negative values of the first covariate. A linear model cannot represent that symmetric pattern and can produce nearly flat fitted scores. More observations reduce sampling noise without supplying the missing model shape. In the network audit, sparse upper-exposure observations and empirical quantile variation help explain continued refusal at treatment probability 0.3. These mechanisms distinguish restrictions that more data can alleviate from restrictions requiring a different diagnostic.

7.2. Limitations

The following limitations define the scope of the reported evidence.

1. *Assumptions and scope.* The gates enforce user declarations; false or incomplete metadata can still support an invalid causal answer. The implemented tasks are static effects, observed regimes, and network interference. General graphical identification, latent variables, and temporal effect estimation require further development.
2. *Neural evaluation.* One optimization run supplies the evaluated checkpoint. Its saved trajectory and reproducible inference do not establish training stability across initial-

izations. The pretraining ablation compares five untrained initializations with this same checkpoint; it is not a study of repeated training. Multiple optimization runs would be needed to assess the variability of training and checkpoint selection. The shallow policies improve on this router, without establishing a general ranking of deep and shallow methods. Residual and world-particle pathways remain exploratory.

3. *Presentation sensitivity.* The estimator-layer repairs improve the tested row and covariate permutations. Query indices remain inputs, and negative outcome scaling and treatment recoding retain equivariance errors (Section 6.1).
4. *Statistical resolution and generalization.* Original regime results use 30 episodes per cell; the confirmatory effect comparison uses 300. RQ4 uses four development episodes per family and informs checkpoint selection. Their individual errors are disclosed in Table 12; a larger independent study is needed to estimate performance on those families reliably.
5. *Support and calibration.* The support audit covers five covariates, two logistic assignment families, and a fixed-degree network. Its false refusals and false acceptances do not characterize arbitrary data settings. Conformal coverage is marginal over exchangeable episodes, with finite-sample per-cell variation and no guarantee under a changed episode distribution.
6. *External validity.* Effect truth is synthetic or semi-synthetic. The energy experiment uses one building, with adjacent time measurements and reused rows across episodes. It assesses transfer to measured covariates, not a real intervention or an independent building or time period.

7.3. Practical Use and Interpretation

The framework gives the analyst a clear division of responsibilities. Domain knowledge supports the query and causal declarations; the software records those declarations, checks its implemented conditions, fits compatible estimators, and reports uncertainty and decision reasons. The released bundle fixes the neural baseline and its calibration for reproducibility. Deploying one of the stronger confirmatory policies would require packaging its fitted parameters with matching calibration. This makes the next implementation step concrete while preserving an auditable reference result.

For an accepted effect query, the useful record consists of the requested contrast, its assumptions, the contributing estimators, and the calibrated interval. Candidate weights explain how the reported number was assembled; they do not assign causal importance to individual variables. For a refusal, the reason identifies the condition that prevented the requested analysis. This form of reporting supports review of the analysis choices while preserving the distinction between an automated calculation and its causal interpretation.

7.4. Future Work

The next priorities are to package the stronger simple aggregation policies with matching calibration and to validate more flexible support diagnostics. Independent studies can then vary assignment models, covariate dimension, network structure, and data source. Additional training runs would clarify when deep routing adds value. Temporal mechanisms, partial graphs, and sensitivity bounds offer further extensions. Sensor networks, process control, and energy systems provide application settings for evaluation with real interventions or defensible quasi-experimental designs.

8. Conclusion

CWFM provides a reproducible framework for causal estimation with explicit assumptions, classical estimators, calibrated intervals, and documented refusals. Its central

result is that simple aggregation makes effective use of a shared estimator library. On 1,800 confirmatory queries, the shallow selector and convex stack achieve MAEs of 0.141 and 0.147, compared with 0.161 for the evaluated deep router. The regime evaluation and semi-synthetic energy benchmark provide complementary evidence for the integrated approach.

All 1,200 declared-invalid stress queries are refused by the shared gate. This consistency is conditional on the supplied contract, whose truth and completeness remain the analyst's responsibility. The empirical support audit identifies both improved screening with larger samples in several settings and persistent errors under nonlinear assignment. The framework thus offers an inspectable basis for further causal applications, with clear evaluation targets for estimator selection, support diagnostics, and external validation.

Author Contributions: Conceptualization, A.B.; methodology, A.B.; software, A.B.; validation, A.B.; formal analysis, A.B.; investigation, A.B.; data curation, A.B.; writing: original draft preparation, A.B.; writing: review and editing, A.B.; visualization, A.B. The author has read and agreed to the published version of the manuscript.

Funding: Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and from the German Federal Ministry of Research, Technology and Space (BMFTR), the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW), and the Hessian Ministry of Science and Research, Arts and Culture (HMWK) under grant agreement No 101250682.

Institutional Review Board Statement: Not applicable. This study used synthetic data and a public energy-sensor dataset; it did not involve human or animal participants.

Informed Consent Statement: Not applicable.

Data Availability Statement: The baseline code, synthetic generators, released model bundle, and original and confirmatory artifacts are available at https://github.com/fit-alessandro-berti/foundation-causality0. The revision files include the empirical support protocol, experiment driver, complete per-seed results, and manifests under `experiments/` and `artifacts/cwfm/support_revision/`. The external source is the UCI Appliances Energy Prediction dataset [51], licensed under Creative Commons Attribution 4.0. Checkpoint and calibration checksums are recorded in the release manifest.

Acknowledgments: Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and from the German Federal Ministry of Research, Technology and Space (BMFTR), the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW), and the Hessian Ministry of Science and Research, Arts and Culture (HMWK) under grant agreement No 101250682.

Use of Artificial Intelligence: During the preparation of this work, the author used generative artificial intelligence tools to improve the wording and presentation of the text and to assist in the implementation of the approach. The author has reviewed and edited all output and takes full responsibility for the content of this publication.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AIPW	Augmented Inverse Probability Weighting
BIC	Bayesian Information Criterion
CI	Confidence Interval
CWFM	Causal World Foundation Model
SCM	Structural Causal Model
ATE	Average Treatment Effect
CATE	Conditional Average Treatment Effect
ACD	Amortized Causal Discovery
AVICI	Amortized Variational Inference for Causal Discovery
DECI	Deep End-to-End Causal Inference
ITE	Individual Treatment Effect
PEHE	Precision in Estimation of Heterogeneous Effects
PFN	Prior-Data Fitted Network
MAE	Mean Absolute Error
RQ	Research Question
GPU	Graphics Processing Unit
SHA-256	Secure Hash Algorithm, 256-bit
UCI	University of California, Irvine
VARX	Vector Autoregression with Exogenous Inputs

References

1. Pearl, J. *Causality: Models, Reasoning, and Inference*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2009. https://doi.org/10.1017/cbo9780511803161.
2. Hernán, M.A.; Robins, J.M. *Causal Inference: What If*; Chapman & Hall/CRC: Boca Raton, FL, USA, 2020. Available online: https://miguelhernan.org/whatifbook (accessed on 8 September 2026).
3. Blöbaum, P.; Götz, P.; Budhathoki, K.; Mastakouri, A.A.; Janzing, D. DoWhy-GCM: An Extension of DoWhy for Causal Inference in Graphical Causal Models. *Journal of Machine Learning Research* **2024**, *25*, 1 to 7. Available online: https://jmlr.org/papers/v25/22-1258.html (accessed on 8 September 2026).
4. Alaa, A.; van der Schaar, M. Validating Causal Inference Models via Influence Functions. In Proceedings of the 36th International Conference on Machine Learning. PMLR, 2019, Vol. 97, *Proceedings of Machine Learning Research*, pp. 191 to 201. Available online: https://proceedings.mlr.press/v97/alaa19a.html (accessed on 8 September 2026).
5. Lan, H.; Syrgkanis, V. Causal Q-Aggregation for CATE Model Selection. In Proceedings of the 27th International Conference on Artificial Intelligence and Statistics. PMLR, 2024, Vol. 238, *Proceedings of Machine Learning Research*, pp. 4366 to 4374. Available online: https://proceedings.mlr.press/v238/lan24a.html (accessed on 8 September 2026).
6. Ma, Y.; Frauen, D.; Javurek, E.; Feuerriegel, S. Foundation Models for Causal Inference via Prior-Data Fitted Networks. In Proceedings of the International Conference on Learning Representations, 2026, Vol. 2026, pp. 79065 to 79098. Available online: https://proceedings.iclr.cc/paper_files/paper/2026/hash/7f8cabaf2de70e1a9d3eb187f02bd58c-Abstract-Conference.html (accessed on 8 September 2026).
7. Robertson, J.; Reuter, A.; Guo, S.; Hollmann, N.; Hutter, F.; Schölkopf, B. Do-PFN: In-Context Learning for Causal Effect Estimation. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2025, Vol. 38, pp. 174811 to 174848. https://doi.org/10.52202/085713-5814.
8. Balazadeh Meresht, V.; Kamkari, H.; Thomas, V.; Ma, J.; Li, B.; Cresswell, J.; Krishnan, R. CausalPFN: Amortized Causal Effect Estimation via In-Context Learning. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2025, Vol. 38, pp. 154945 to 154984. https://doi.org/10.52202/085713-5184.
9. Hudgens, M.G.; Halloran, M.E. Toward Causal Inference with Interference. *Journal of the American Statistical Association* **2008**, *103*, 832 to 842. https://doi.org/10.1198/016214508000000292.
10. Aronow, P.M.; Samii, C. Estimating Average Causal Effects under General Interference, with Application to a Social Network Experiment. *The Annals of Applied Statistics* **2017**, *11*, 1912 to 1947. https://doi.org/10.1214/16-aoas1005.
11. Sävje, F. Causal Inference with Misspecified Exposure Mappings: Separating Definitions and Assumptions. *Biometrika* **2024**, *111*, 1 to 15. https://doi.org/10.1093/biomet/asad019.
12. D'Amour, A.; Ding, P.; Feller, A.; Lei, L.; Sekhon, J. Overlap in Observational Studies with High-Dimensional Covariates. *Journal of Econometrics* **2021**, *221*, 644 to 654. https://doi.org/10.1016/j.jeconom.2019.10.014.

13. Zeileis, A.; Hothorn, T.; Hornik, K. Model-Based Recursive Partitioning. *Journal of Computational and Graphical Statistics* **2008**, *17*, 492 to 514. https://doi.org/10.1198/106186008x319331.
14. Athey, S.; Imbens, G. Recursive Partitioning for Heterogeneous Causal Effects. *Proceedings of the National Academy of Sciences* **2016**, *113*, 7353 to 7360. https://doi.org/10.1073/pnas.1510489113.
15. Peters, J.; Bühlmann, P.; Meinshausen, N. Causal Inference by Using Invariant Prediction: Identification and Confidence Intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **2016**, *78*, 947 to 1012. https://doi.org/10.1111/rssb.12167.
16. Huang, B.; Zhang, K.; Zhang, J.; Ramsey, J.; Sanchez-Romero, R.; Glymour, C.; Schölkopf, B. Causal Discovery from Heterogeneous/Nonstationary Data. *Journal of Machine Learning Research* **2020**, *21*, 1 to 53. Available online: https://jmlr.org/papers/v21/19-232.html (accessed on 8 September 2026).
17. Müller, S.; Hollmann, N.; Pineda Arango, S.; Grabocka, J.; Hutter, F. Transformers Can Do Bayesian Inference. In Proceedings of the International Conference on Learning Representations, 2022. Available online: https://openreview.net/forum?id=KSugKcbNf9 (accessed on 8 September 2026).
18. Vovk, V.; Gammerman, A.; Shafer, G. *Algorithmic Learning in a Random World*; Springer: New York, NY, USA, 2005. https://doi.org/10.1007/b106715.
19. Lei, J.; G'Sell, M.; Rinaldo, A.; Tibshirani, R.J.; Wasserman, L. Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association* **2018**, *113*, 1094 to 1111. https://doi.org/10.1080/01621459.2017.1307116.
20. Rosenbaum, P.R.; Rubin, D.B. Assessing Sensitivity to an Unobserved Binary Covariate in an Observational Study with Binary Outcome. *Journal of the Royal Statistical Society: Series B (Methodological)* **1983**, *45*, 212 to 218. https://doi.org/10.1111/j.2517-6161.1983.tb01242.x.
21. Manski, C.F. Nonparametric Bounds on Treatment Effects. *American Economic Review* **1990**, *80*, 319 to 323. Available online: https://www.jstor.org/stable/2006592 (accessed on 8 September 2026).
22. Shpitser, I.; Pearl, J. Identification of Joint Interventional Distributions in Recursive Semi-Markovian Causal Models. In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), 2006, pp. 1219 to 1226. Available online: https://aaai.org/papers/01219-aaai06-191-identification-of-joint-interventional-distributions-in-recursive-semi-markovian-causal-models/ (accessed on 8 September 2026).
23. Geffner, T.; Antorán, J.; Foster, A.; Gong, W.; Ma, C.; Kiciman, E.; Sharma, A.; Lamb, A.; Kukla, M.; Pawlowski, N.; et al. Deep End-to-end Causal Inference. *Transactions on Machine Learning Research* **2024**. Available online: https://openreview.net/forum?id=e6sqttxEGX (accessed on 16 September 2026).
24. Spirtes, P.; Glymour, C.; Scheines, R. *Causation, Prediction, and Search*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2001. https://doi.org/10.7551/mitpress/1754.001.0001.
25. Zheng, X.; Aragam, B.; Ravikumar, P.K.; Xing, E. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2018, Vol. 31. Available online: https://proceedings.neurips.cc/paper/2018/hash/e347c51419ffb23ca3fd5050202f9c3d-Abstract.html (accessed on 8 September 2026).
26. Runge, J.; Nowack, P.; Kretschmer, M.; Flaxman, S.; Sejdinovic, D. Detecting and Quantifying Causal Associations in Large Nonlinear Time Series Datasets. *Science Advances* **2019**, *5*, eaau4996. https://doi.org/10.1126/sciadv.aau4996.
27. Hollmann, N.; Müller, S.; Purucker, L.; Krishnakumar, A.; Körfer, M.; Hoo, S.B.; Schirrmeister, R.T.; Hutter, F. Accurate Predictions on Small Data with a Tabular Foundation Model. *Nature* **2025**, *637*, 319 to 326. https://doi.org/10.1038/s41586-024-08328-6.
28. Reuter, A.; Dhir, A.; Diaconu, C.; Robertson, J.; Ossen, O.; Hutter, F.; Weller, A.; van der Wilk, M.; Schölkopf, B. Use What You Know: Causal Foundation Models with Partial Graphs. In Proceedings of the 43rd International Conference on Machine Learning, 2026. Available online: https://icml.cc/virtual/2026/poster/61982 (accessed on 16 September 2026).
29. Löwe, S.; Madras, D.; Zemel, R.; Welling, M. Amortized Causal Discovery: Learning to Infer Causal Graphs from Time-Series Data. In Proceedings of the First Conference on Causal Learning and Reasoning. PMLR, 2022, Vol. 177, *Proceedings of Machine Learning Research*, pp. 509 to 525. Available online: https://proceedings.mlr.press/v177/lowe22a.html (accessed on 16 September 2026).
30. Melnychuk, V.; Frauen, D.; Feuerriegel, S. Causal Transformer for Estimating Counterfactual Outcomes. In Proceedings of the 39th International Conference on Machine Learning. PMLR, 2022, Vol. 162, *Proceedings of Machine Learning Research*, pp. 15293 to 15329. Available online: https://proceedings.mlr.press/v162/melnychuk22a.html (accessed on 16 September 2026).
31. Lorch, L.; Sussex, S.; Rothfuss, J.; Krause, A.; Schölkopf, B. Amortized Inference for Causal Structure Learning. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2022, Vol. 35, pp. 13104 to 13118. https://doi.org/10.52202/068431-0952.
32. Nie, X.; Wager, S. Quasi-Oracle Estimation of Heterogeneous Treatment Effects. *Biometrika* **2021**, *108*, 299 to 319. https://doi.org/10.1093/biomet/asaa076.

33. Saito, Y.; Yasui, S. Counterfactual Cross-Validation: Stable Model Selection Procedure for Causal Inference Models. In Proceedings of the 37th International Conference on Machine Learning. PMLR, 2020, Vol. 119, *Proceedings of Machine Learning Research*, pp. 8398 to 8407. Available online: https://proceedings.mlr.press/v119/saito20a.html (accessed on 8 September 2026).
34. Cui, Y.; Tchetgen Tchetgen, E.J. Selective Machine Learning of Doubly Robust Functionals. *Biometrika* **2024**, *111*, 517 to 535. https://doi.org/10.1093/biomet/asad055.
35. Curth, A.; van der Schaar, M. In Search of Insights, Not Magic Bullets: Towards Demystification of the Model Selection Dilemma in Heterogeneous Treatment Effect Estimation. In Proceedings of the 40th International Conference on Machine Learning. PMLR, 2023, Vol. 202, *Proceedings of Machine Learning Research*, pp. 6623 to 6642. Available online: https://proceedings.mlr.press/v202/curth23b.html (accessed on 8 September 2026).
36. Mahajan, D.; Mitliagkas, I.; Neal, B.; Syrgkanis, V. Empirical Analysis of Model Selection for Heterogeneous Causal Effect Estimation. In Proceedings of the International Conference on Learning Representations, 2024, Vol. 2024, pp. 26673 to 26702. Available online: https://proceedings.iclr.cc/paper_files/paper/2024/hash/71484f17ae8ddd0500c8571bed59926d-Abstract-Conference.html (accessed on 16 September 2026).
37. Vanderschueren, T.; Verdonck, T.; van der Schaar, M.; Verbeke, W. AutoCATE: End-to-End, Automated Treatment Effect Estimation. In Proceedings of the 42nd International Conference on Machine Learning. PMLR, 2025, Vol. 267, *Proceedings of Machine Learning Research*, pp. 60880 to 60904. Available online: https://proceedings.mlr.press/v267/vanderschueren25a.html (accessed on 8 September 2026).
38. Crump, R.K.; Hotz, V.J.; Imbens, G.W.; Mitnik, O.A. Dealing with Limited Overlap in Estimation of Average Treatment Effects. *Biometrika* **2009**, *96*, 187 to 199. https://doi.org/10.1093/biomet/asn055.
39. Li, F.; Morgan, K.L.; Zaslavsky, A.M. Balancing Covariates via Propensity Score Weighting. *Journal of the American Statistical Association* **2018**, *113*, 390 to 400. https://doi.org/10.1080/01621459.2016.1260466.
40. Dorn, J.; Guo, K. Sharp Sensitivity Analysis for Inverse Propensity Weighting via Quantile Balancing. *Journal of the American Statistical Association* **2023**, *118*, 2645 to 2657. https://doi.org/10.1080/01621459.2022.2069572.
41. Melnychuk, V.; Balazadeh, V.; Feuerriegel, S.; Krishnan, R.G. Frequentist Consistency of Prior-Data Fitted Networks for Causal Inference. In Proceedings of the 43rd International Conference on Machine Learning, 2026. Available online: https://icml.cc/virtual/2026/poster/61548 (accessed on 16 September 2026).
42. Lei, L.; Candès, E.J. Conformal Inference of Counterfactuals and Individual Treatment Effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **2021**, *83*, 911 to 938. https://doi.org/10.1111/rssb.12445.
43. van der Laan, L.; Ulloa-Pérez, E.; Carone, M.; Luedtke, A. Causal Isotonic Calibration for Heterogeneous Treatment Effects. In Proceedings of the 40th International Conference on Machine Learning. PMLR, 2023, Vol. 202, *Proceedings of Machine Learning Research*, pp. 34831 to 34854. Available online: https://proceedings.mlr.press/v202/van-der-laan23a.html (accessed on 8 September 2026).
44. Geifman, Y.; El-Yaniv, R. Selective Classification for Deep Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2017, Vol. 30. Available online: https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html (accessed on 8 September 2026).
45. Jesson, A.; Mindermann, S.; Shalit, U.; Gal, Y. Identifying Causal-Effect Inference Failure with Uncertainty-Aware Models. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2020, Vol. 33, pp. 11637 to 11649. Available online: https://proceedings.neurips.cc/paper_files/paper/2020/file/860b37e28ec7ba614f00f9246949561d-Paper.pdf (accessed on 16 September 2026).
46. Faller, P.M.; Vankadara, L.C.; Mastakouri, A.A.; Locatello, F.; Janzing, D. Self-Compatibility: Evaluating Causal Discovery without Ground Truth. In Proceedings of the 27th International Conference on Artificial Intelligence and Statistics. PMLR, 2024, Vol. 238, *Proceedings of Machine Learning Research*, pp. 4132 to 4140. Available online: https://proceedings.mlr.press/v238/faller24a.html (accessed on 8 September 2026).
47. Jin, Z.; Chen, Y.; Leeb, F.; Gresele, L.; Kamal, O.; Lyu, Z.; Blin, K.; Gonzalez Adauto, F.; Kleiman-Weiner, M.; Sachan, M.; et al. CLadder: Assessing Causal Reasoning in Language Models. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2023, Vol. 36, pp. 31038 to 31065. https://doi.org/10.52202/075280-1353.
48. Geng, L.; Ouyang, A.; Wu, T.; Barretto, D.; Hayes, M.J.; Cooper, R.; Zeng, Y.; Vijay, S.; Ancone, G.; Rai, A.; et al. CausalT5k: Diagnosing Refusal and Failure Modes in Trustworthy Causal Reasoning Across Causal Rungs. In Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2. ACM, 2026, pp. 8918 to 8929. https://doi.org/10.1145/3770855.3817567.
49. Ma, Y.; Tresp, V. Causal Inference under Networked Interference and Intervention Policy Enhancement. In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. PMLR, 2021, Vol. 130, *Proceedings of Machine Learning Research*, pp. 3700 to 3708. Available online: https://proceedings.mlr.press/v130/ma21c.html (accessed on 8 September 2026).

50. Zhang, H.; Yang, X.; Luo, Y.; Chen, F.; Lu, N.; Liu, Y.; Deng, Y. A Novel Hybrid Algorithm for Damage Detection in Bridge Foundations under Complex Underwater Environments Using ROV Capture Pictures. *Engineering Structures* **2026**, *352*, 122131. https://doi.org/10.1016/j.engstruct.2026.122131.
51. Candanedo, L. Appliances Energy Prediction. UCI Machine Learning Repository, 2017. CC BY 4.0, https://doi.org/10.24432/C5VC8G.