

Supplementary Information for “Structural support constraints prevent invalid surrogate predictions in branched agent-based simulations”

Alaa Aldin Hajjar^{1,*} and Armen Beklaryan²

¹Lab of Machine Learning and Knowledge Representation, Research Center of the Artificial Intelligence Institute, Innopolis University, Innopolis, Russia

²Faculty of Computer Science, HSE University, Moscow, Russia

*Correspondence: a.hajjar@innopolis.university

ABSTRACT

This file provides supporting details for the Scientific Reports submission. It contains extended background, simulator and representation notes, model and validation details, supplementary diagnostics, and the runtime-estimate derivation. It does not introduce new experiments beyond the submitted manuscript.

Supplementary Note 1: Extended related work

The main text summarizes the relationship between this study and surrogate modeling for expensive simulation, agent-based model emulation, and constraint-aware prediction. The broader literature also includes output-constrained regression, zero-inflated and hurdle models, regime-specific surrogates, mixture-of-experts methods, and shape-constrained learning. These traditions share the concern that predictions should respect known support, zero-generating mechanisms, or regime structure. The present workflow differs in one key respect: the zero-support rule is not latent or statistically inferred. It is deterministic and observed through the simulator branch indicator.

Physics-informed and rule-informed machine learning provide a related motivation: predictions should respect scientific or mechanistic structure when that structure is known. Here the constraint is not a differential-equation residual or a learned symbolic rule. It is a post-hoc simulator-derived support rule applied to selected outputs under an observed branch. The scope is therefore narrower and more auditable than a general neurosymbolic or physics-informed learning claim.

Supplementary Note 2: Simulator and branch details

Each simulator invocation is defined by lattice size, simulator seed, affordability margin, and vacancy fraction. Each invocation expands into four condition rows: `baseline_noecon`, `baseline_econ`, `with_influencer_noecon`, and `with_influencer_econ`. The branch indicator `use_economic_constraint` determines whether affordability blocking can occur. If the economic constraint is inactive, the two primary blockage outcomes have structural-zero support.

The expanded dataset contains 400 unique invocations and 1600 condition-specific rows. Branch-integrity checks confirmed that each invocation expands to exactly four condition rows, all four conditions are balanced at 400 rows each, and the two primary blockage outcomes are zero for all 800 no-economy rows.

Supplementary Note 3: Feature encodings and preprocessing

The reduced representation contains lattice size, seed, affordability margin, and vacancy fraction. The minimal branch-aware representation adds simulator variant and the economic-constraint indicator. The full branch-aware representation also includes the named condition. Sensitivity checks showed that seed and condition were not required as primary predictive features after variant and `use_economic_constraint` were included. The selected model uses `size_L`, affordability margin, vacancy fraction, variant, and `use_economic_constraint`.

Categorical simulator fields are encoded in the released scripts. The final model configuration is stored in the release package at `results/model_selection/selected_model_config.json`.

Supplementary Note 4: Model hyperparameters and training settings

The selected model is `posthoc_masked_mlp_8`. It uses two hidden layers of width 8, ReLU activations, Adam optimization, learning rate 0.01, weight decay 0.0001, 300 training epochs, and mean-squared error on the standardized target. The model has 137 trainable parameters under the final encoded input representation. The selected random seed is 20260514, with fold-specific training seeds 20260515–20260519.

The evaluated comparison families include mean prediction, ridge regression, Random Forest, ExtraTrees, Gradient Boosting, MLP-8, MLP-16, post-hoc masked variants, and two-stage active-regime baselines. Full hyperparameters are specified in the release scripts.

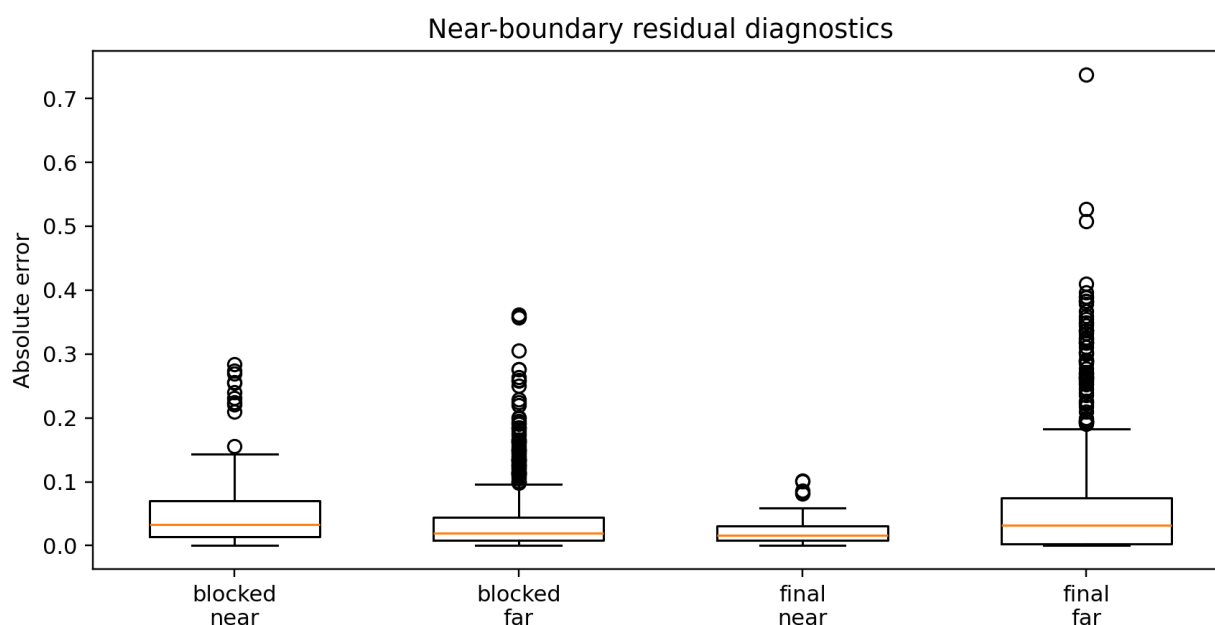
Supplementary Note 5: Full model-family results

The main manuscript reports a compact subset of the model-family comparison. The full generated CSV and LaTeX artifacts are available in the release package:

- `results/model_family_comparison/model_family_comparison_expanded.csv`
- `results/model_family_comparison/model_family_comparison_expanded_by_fold.csv`
- `tables/table_model_family_comparison_expanded.csv`

Supplementary Note 6: Residual diagnostics

Near-boundary diagnostics separate rows whose observed outcome lies within 0.05 of the relevant high-regime threshold from the remaining rows. This diagnostic is used to identify where the surrogate is weakest after structural masking. The final affordability-blocked share shows the largest residual concentration in difficult regimes, especially around transition and high-gradient regions of the response surface.

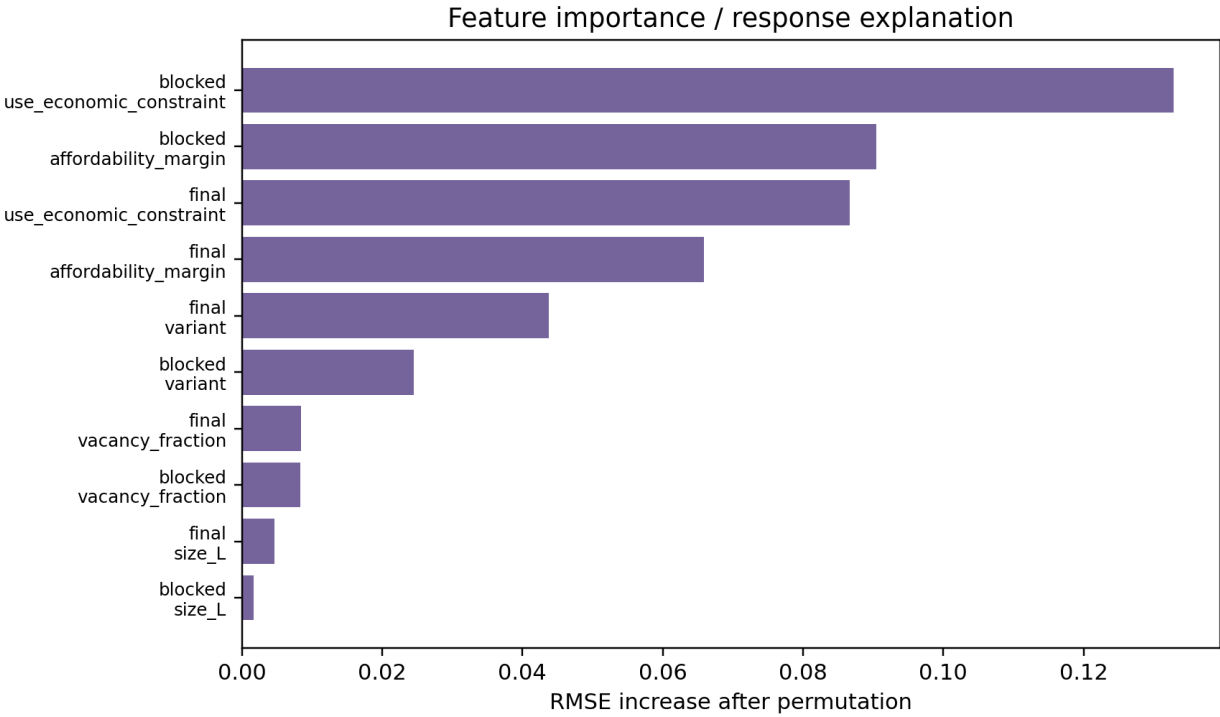


Supplementary Fig. S1. Supplementary residual diagnostics for the selected masked MLP-8 model. Absolute errors are shown separately for near-boundary and non-near-boundary rows for the two primary blockage outcomes.

Supplementary Note 7: Surrogate response explanation

Permutation importance was used as a diagnostic explanation of the surrogate response surface. The dominant variables are the economic-constraint indicator, affordability margin, simulator variant, and vacancy fraction. These importances should be

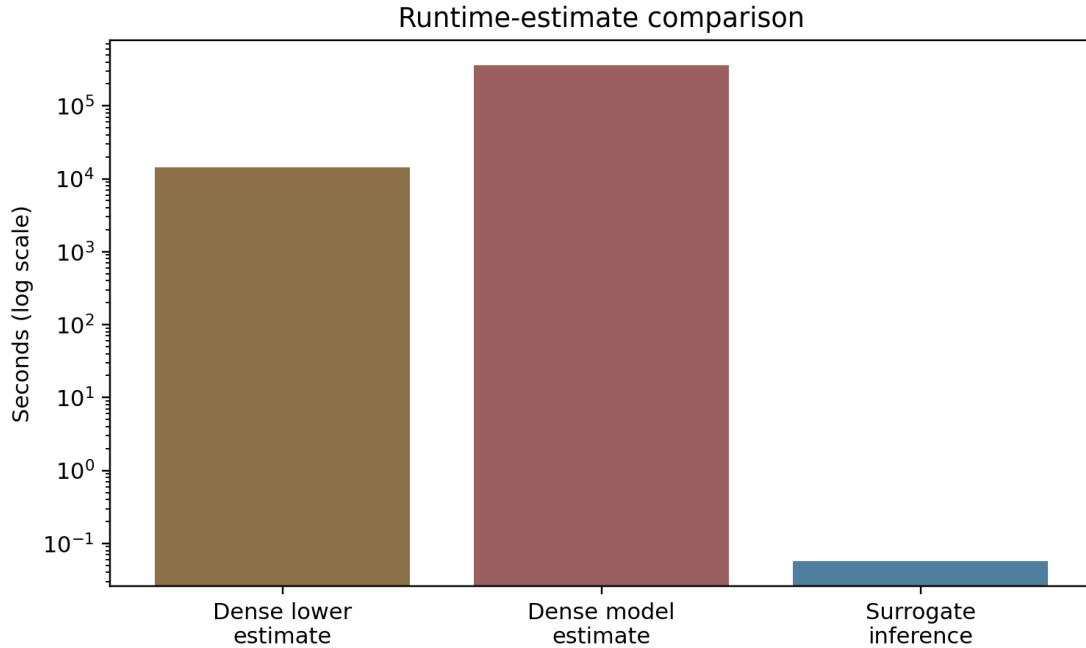
interpreted as surrogate explanations of the implemented simulator response under the evaluated design, not as empirical causal effects.



Supplementary Fig. S2. Supplementary permutation-importance summary for the constrained blockage models. The plot describes surrogate response sensitivity within the simulator design.

Supplementary Note 8: Runtime-estimate derivation

The dense diagnostic grid contains 2542 unique simulator invocations and 10,168 condition rows. The observed mean runtime from the expanded campaign gives a lower-bound dense direct runtime estimate of 14,209.0174 s. The conservative runtime model gives 361,228.3647 s. The measured surrogate inference time is 0.056635 s. The resulting estimate-based speedup range is 2.51×10^5 to 6.38×10^6 . The dense direct sweep was not executed.



Supplementary Fig. S3. Supplementary runtime-estimate comparison. Direct dense-sweep runtimes are estimated, whereas surrogate inference time is measured.

Supplementary Table S1: Representation ablation

Supplementary Table S1. Representation and hard-mask ablation on the expanded dataset.

Mode	Blocked RMSE / econ. RMSE	Afford. RMSE / econ. RMSE	Invalid (b,a)	Takeaway
Reduced representation + unmasked MLP-8	0.1344 / 0.1485	0.1348 / 0.1650	800, 800	Omitting branch-defining variables collapses economy-active and no-economy regimes.
Minimal branch-aware representation + unmasked MLP-8	0.0496 / 0.0699	0.0735 / 0.1038	800, 800	Branch-aware features recover predictive signal but do not guarantee no-economy support for all model families.
Minimal branch-aware representation + hard post-hoc mask	0.0494 / 0.0698	0.0746 / 0.1055	0, 0	Hard simulator-derived masks eliminate impossible no-economy predictions while preserving useful economy-active accuracy.
Full branch-aware representation + unmasked MLP-8	0.0499 / 0.0705	0.0753 / 0.1065	800, 800	Adding the named condition did not materially improve performance after variant and branch indicator were retained.

The support-mask result is guaranteed after masking; the empirical evidence is that unmasked surrogates can violate support and that masked predictors retain active-branch accuracy.

Supplementary Table S2: Runtime-estimate quantities

Supplementary Table S2. Estimated runtime range for dense diagnostic mapping.

Quantity	Value	Interpretation
Expanded experimental dataset	400 invocations / 1600 rows	Measured simulator campaign used for the final evidence base.
Observed expanded runtime	2235.880 s	Measured campaign wall time.
Observed mean runtime per invocation	5.590 s	Input to the lower-bound direct-runtime estimate.
Dense-grid unique invocations	2542	Grid over affordability margin, vacancy fraction, two lattice sizes, and one representative seed.
Dense-grid condition rows	10,168	Each dense invocation expands to four conditions.
Lower-bound dense direct runtime	14,209.017 s	Dense invocations times observed expanded mean runtime.
Conservative dense direct runtime	361,228.365 s	Runtime-model estimate; the dense sweep was not executed.
Measured surrogate inference time	0.056635 s	Wall-clock time for the selected model inference pass.
Estimated speedup range	2.51×10^5 to 6.38×10^6	Direct-runtime estimates divided by measured surrogate inference time.

The dense direct sweep was estimated rather than executed; the table reports a runtime-estimate comparison, not a measured dense-sweep benchmark.

Supplementary Table S3: Primary model-selection summary

Supplementary Table S3. Primary model-selection summary under grouped validation.

Role / model	Params	Blocked RMSE	Blocked econ. RMSE	Afford. RMSE	Afford. econ. RMSE
Primary: post-hoc masked MLP-8	137	0.0494 ± 0.0037	0.0698	0.0746 ± 0.0042	0.1055
Sensitivity: post-hoc masked MLP-16	401	0.0495 ± 0.0036	0.0700	0.0748 ± 0.0041	0.1058
Comparison: masked Random Forest	19,500	0.0496 ± 0.0035	0.0701	0.0748 ± 0.0040	0.1058
Comparison: two-stage MLP-8	137	0.0496 ± 0.0036	0.0701	0.0747 ± 0.0042	0.1057

The primary model is chosen by zero structural-support violations, economy-active predictive accuracy, grouped-CV stability, simplicity, and reproducibility. The selection does not claim that the MLP architecture is uniquely superior.