# PREDICTIVE FEASIBILITY

PFA

## Forecasting & Deployment Assessment

# Introduction

The purpose of this assessment series is to determine whether inferability-related structures can support forecasting, deployment-oriented warning systems, and predictive decision-making under realistic operating conditions.

Earlier validation studies established that inferability, entropy, overlap, persistence, and collapse dynamics represent reproducible structural properties across multiple datasets and domains.

The next logical question is whether these structural properties can be used operationally.

This assessment therefore investigates:

- forecasting generalization across unseen data;
- transition forecasting prior to collapse events;
- false-positive reduction and warning-system reliability;
- threshold calibration and deployment robustness;
- transfer behavior under changing conditions;
- and practical deployment-oriented predictive feasibility.

The objective is not merely to demonstrate forecasting performance, but to evaluate whether inferability-related structures provide actionable information that remains stable under realistic deployment conditions.

Together, the studies in this section form the forecasting and deployment validation layer of the Predictive Feasibility Assessment (PFA) framework.

# FastSPT Transition Forecasting Validation

## Predictable Inferability Collapse within Short Diffusion Trajectories
## Objective of the Test

This validation aimed to investigate whether **inferability collapse**:

- Can be predicted **in advance**
- Within **fast biological diffusion trajectories**

Previous validations demonstrated that:

- Inferability fluctuates locally,
- Collapse pockets emerge,
- Entropy and overlap correlate with inferability instability.

The central question of this test was:

**Do entropy, overlap, and persistence change before collapse actually occurs?**

Thus:

- Not just detecting collapse,
- But **forecasting collapse**.

## Central Hypothesis

The hypothesis was:

**Inferability collapse arises via a predictable pre-collapse drift phase in which entropy and overlap systematically increase while inferability decreases.**

If correct, this implies:

- Collapse is dynamic,
- Collapse is not random,
- And inferability regimes can be predicted early.

## Dataset Used

**fastSPT diffusion trajectories**

**Source:** Dryad dataset

**DOI:** 10.6078/D13H6N

**Files Used:**

- SPT_data_CSV.zip

**Conditions Used:**

- WT
- noHRD

**Examples:**

- U2OS_Halo-CycT1_WT_spaSPT_95Hz_rep1_cell101.csv
- U2OS_Halo-CycT1_noHRD_spaSPT_95Hz_rep1_cell101.csv

## Methodology

### Sliding Transition Forecasting Windows

For each trajectory, a:

**rolling transition analysis**

was performed.

For each local window, the following were calculated:

### Collapse Event Detection

A **collapse event** was defined as:

- Inferability score within the lowest 20% of local windows.

For each collapse event, **pre-collapse drifts** were calculated.

Thus:

- How metrics changed **4 windows before collapse**,
- Up to **1 window before collapse**.

### Measured Drift Variables

For each detected collapse event, the following pre-collapse drift variables were calculated:

• Entropy drift
Change in trajectory entropy prior to collapse.
• Overlap drift
Change in overlap ambiguity prior to collapse.
• Persistence drift
Change in local structural persistence prior to collapse.
• Inferability-score drift

Change in composite inferability score prior to collapse.
• Collapse-pocket-score drift
Change in localized collapse intensity prior to collapse.

For each variable, drift was measured over the four rolling windows preceding the detected collapse event. This allowed the analysis to determine whether systematic directional changes emerge before inferability collapse becomes visible.

## Results

- WT trajectories
- noHRD trajectories

## First Key Observation

**Entropy rises before collapse**

In both WT and noHRD trajectories, it was found that:

- Entropy systematically increases before collapse.

This directly supports the hypothesis of:

**entropy-sensitive inferability collapse.**

Importantly:

- The entropy change occurs **before the actual inferability collapse**.

Thus:

- Entropy is not only associated with collapse,
- But **predicts collapse**.

## Second Key Observation

**Inferability score decreases prior to collapse**

In both conditions, it was found:

- Strong negative inferability drift,
- Well before the collapse boundary.
- WT: ~ -300
- noHRD: ~ -241

This indicates:

- Collapse occurs via **progressive dynamic decline**.

Thus:

- Inferability collapse is **not abrupt**,
- Not binary,
- But a **transitional process**.

## Third Key Observation

**Overlap contributes to collapse instability**

- Overlap drift was positive,
- But less strongly linearly coupled than entropy.

This suggests:

- Overlap increases ambiguity,
- Increases transition instability,
- But entropy appears to be the **primary collapse driver**.

## Fourth Key Observation

**Persistence remains relatively stable**

- Persistence slope remained slightly positive,
- Relatively limited.

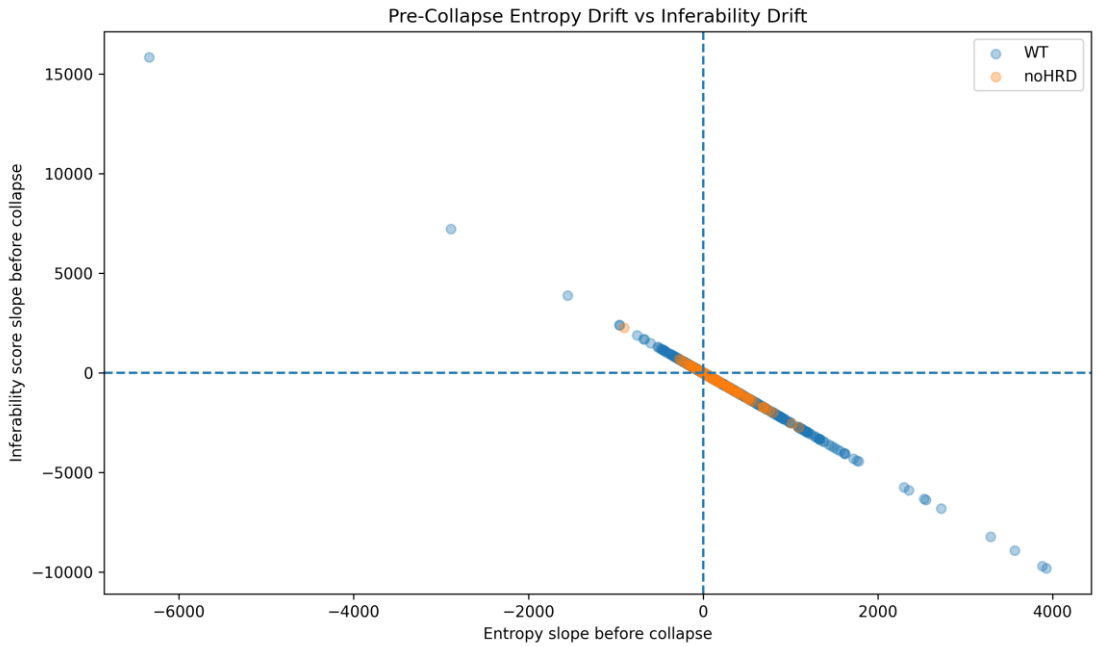
This suggests that:

- Collapse initially arises through **entropy instability**,
- While structural persistence is partially preserved.

This again supports:

**LIMITED inferability dynamics.**

## Figure 1 — Pre-Collapse Entropy Drift vs Inferability Drift



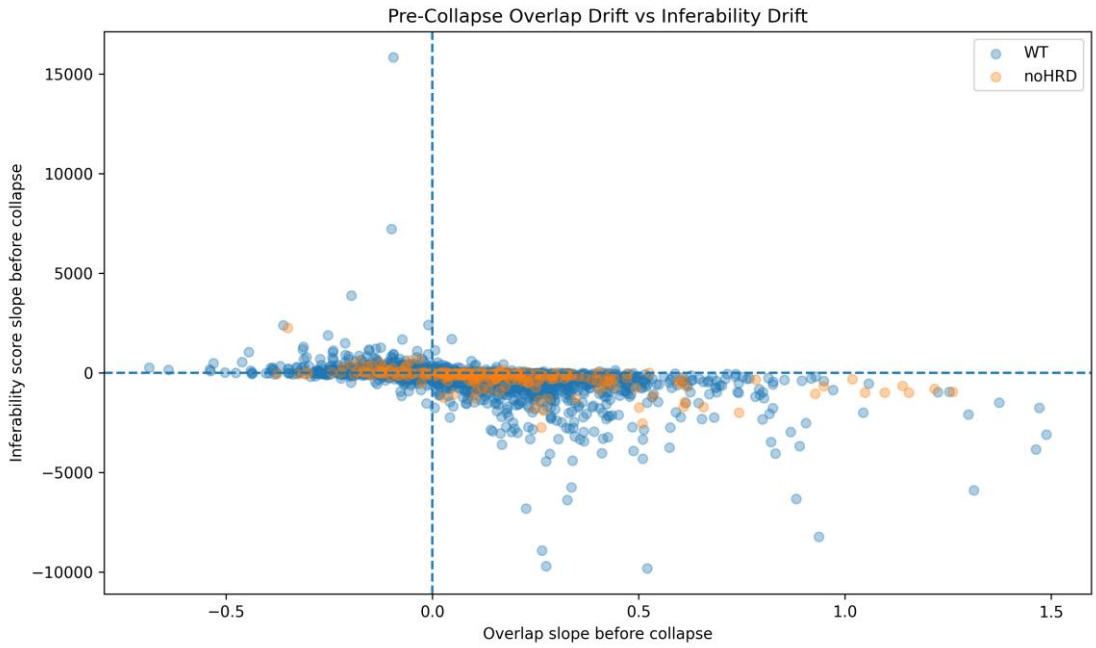

**Caption:**

Relationship between entropy drift before collapse and inferability drift.

- The figure shows a strong systematic coupling between rising entropy and decreasing inferability.
- Importantly, inferability collapse does not occur randomly, but is preceded by a clear entropy drift phase.

This supports the model of **predictable inferability collapse** within diffusion trajectories.

## Figure 2 — Pre-Collapse Overlap Drift vs Inferability Drift

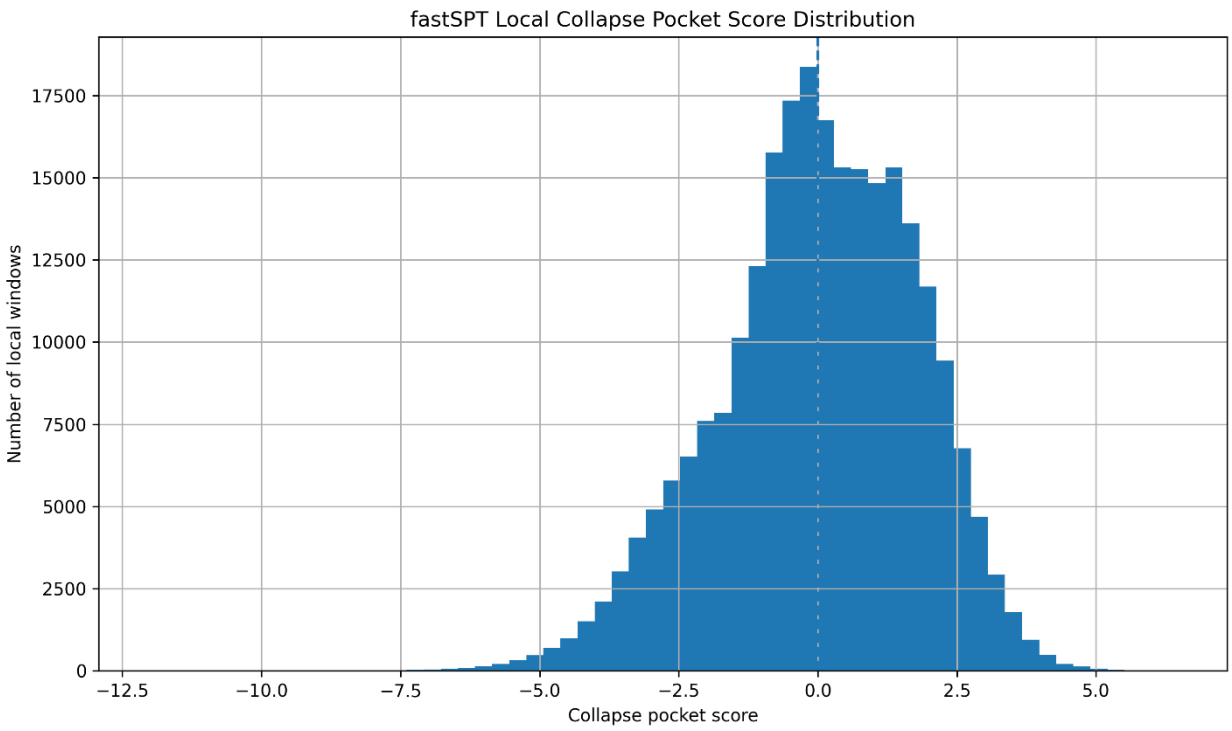



**Caption:**

Relationship between overlap drift and inferability drift.

- Overlap correlates with inferability instability but exhibits a more diffuse pattern than entropy.
- This suggests overlap ambiguity facilitates collapse but likely does not serve as the primary collapse driver.
- Overlap mainly reinforces **transition instability and ambiguity**.

## Figure 3 — fastSPT Local Collapse Pocket Score Distribution

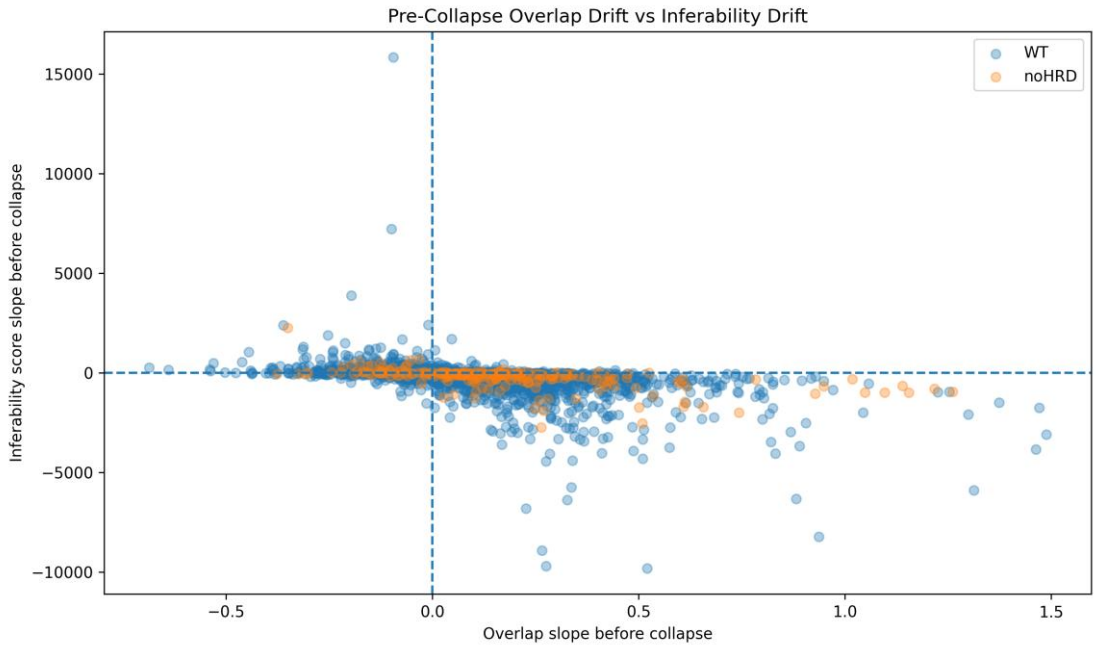



**Caption:**

Distribution of local collapse pocket scores within fastSPT trajectories.

- The distribution shows that inferability collapse is not homogeneous but occurs within specific local dynamic regions.
- This supports the existence of:
    - Local collapse pockets,
    - Transition regions,
    - And dynamic inferability boundaries.

## Figure 4 — fastSPT Overlap vs Local Collapse Pocket Score

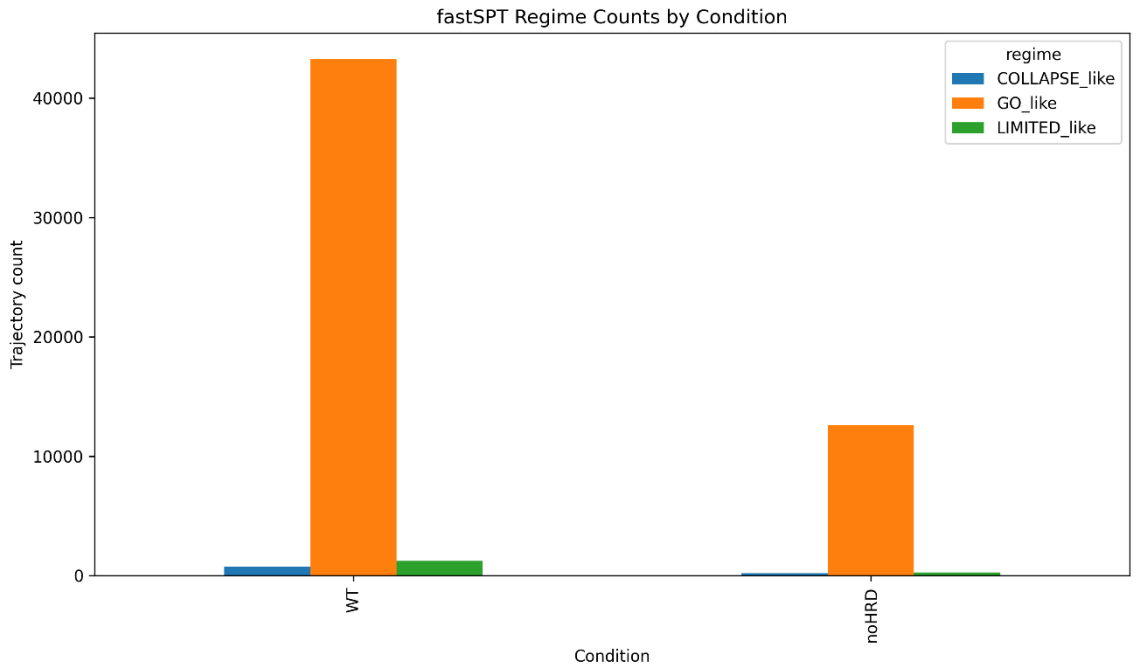



**Caption:**

Relationship between local overlap ambiguity and collapse pocket score.

- The figure shows that overlap ambiguity creates broad transition regions in which inferability instability fluctuates strongly.
- This supports the model in which overlap does not directly cause collapse, but **facilitates local collapse pockets**.

## Figure 5 — fastSPT Regime Counts by Condition

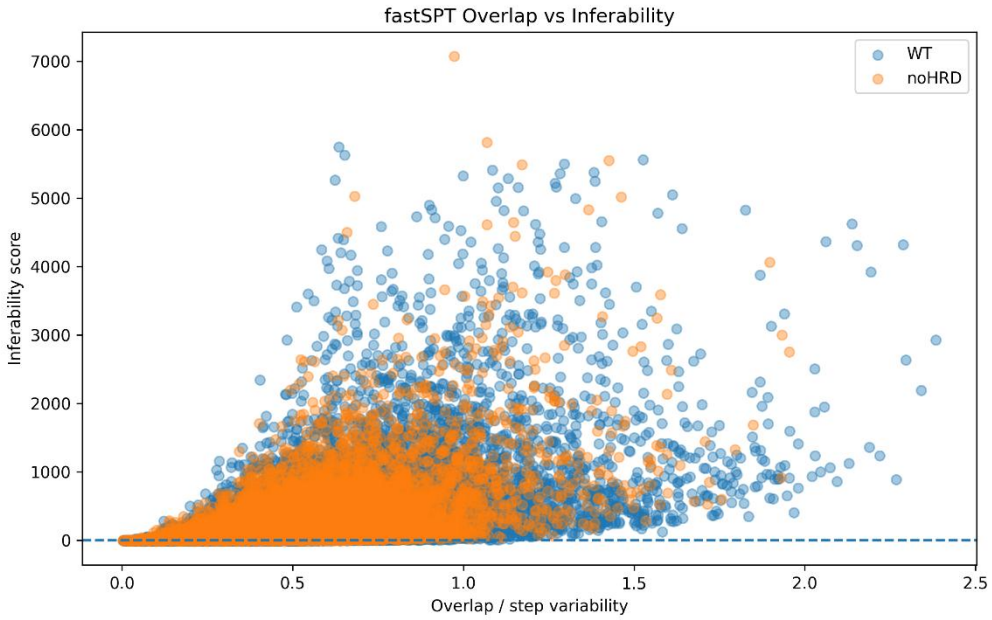



**Caption:**

Distribution of GO_like, LIMITED_like, and COLLAPSE_like regimes within WT and noHRD trajectories.

- Despite strong stochasticity and overlap ambiguity, **GO_like inferability remains dominant**.
- This demonstrates that structural inferability can persist within **complex biological diffusion systems**.

## Figure 6 — fastSPT Overlap vs Inferability




**Caption:**

Relationship between overlap ambiguity and inferability score.

- The figure shows that overlap does not directly destroy inferability but creates broad transitional regions in which **LIMITED inferability emerges**.
- This supports the model of **dynamic inferability regimes**.

## Scripts Used

**Main Script:** fastspt_transition_forecasting.py
**Supporting Scripts:**

- fastspt_localized_collapse_pockets.py
- Previous localized inferability validations

## Outputs Used

**CSV Files:**

- fastspt_transition_forecasting_events.csv
- fastspt_transition_forecasting_summary.csv
- fastspt_localized_collapse_pockets.csv
- fastspt_localized_collapse_summary.csv

**Figure Files:**

- fastspt_entropy_drift_vs_score_drift.png
- fastspt_overlap_drift_vs_score_drift.png
- fastspt_local_collapse_score_distribution.png
- fastspt_overlap_vs_local_collapse_pocket.png

- fastspt_regime_counts.png
- fastspt_overlap_vs_inferability.png

## Reproducibility

This validation is fully reproducible using:

- Original Dryad fastSPT dataset,
- spaSPT CSV trajectories,
- Inferability validation scripts,
- Localized collapse pocket analysis,
- Transition forecasting analysis,
- And all generated CSV/PNG outputs.

All:

- Thresholds,
- Rolling windows,
- Entropy calculations,
- Overlap metrics,
- Persistence definitions,
- And drift analyses

are directly reproducible from the scripts used.

## Conclusion

This validation shows that **inferability collapse**:

- Is predictable prior to collapse.

## Key Implications

- Inferability collapse arises via a **dynamic drift phase**.
- Entropy rises systematically before collapse.
- Overlap ambiguity strengthens transition instability.
- Collapse occurs locally within specific diffusion pockets.
- Inferability behaves as a **dynamic system phenomenon**.

These results strongly reinforce the idea that inferability:

- Is **not a static property**,
- But forms a **predictable dynamic state**
  within complex biological diffusion systems.

# Cross-Run Reproducibility of Local Collapse Dynamics in fastSPT Trajectories

## Purpose of the Test

This validation test investigates whether the previously observed inferability-collapse structures within fastSPT trajectories are not only locally visible, but also remain reproducible across multiple independent runs, replicates, and cells.

The central question of this test is:

*Do local collapse pockets and inferability instability reproduce systematically across independent fastSPT measurements?*

This shifts the analysis from:

local dynamic detection,
toward population-level and cross-run reproducibility.
This step is crucial because industrial and scientific prediction systems are only reliable when the underlying dynamics remain reproducible across multiple independent measurements.

## Dataset Used

For this test, the following experimental fastSPT dataset was used:

**Dataset:**

“Recovering mixtures of fast diffusing states from short single particle trajectories”

**Source:**

Dryad dataset archive

Files:

spaSPT CSV trajectory datasets
WT (wild type)
noHRD conditions
multiple replicates
multiple cells per replicate
The file structure included, among other variables:

trajectory ID
time step (t)
x-position
y-position
frame index
Examples:

U2OS_Halo-CycT1_WT_spaSPT_95Hz_rep1_cell101.csv
U2OS_Halo-CycT1_noHRD_spaSPT_95Hz_rep1_cell101.csv
Number of analyzed files:

30 CSV files

## Analysis Procedure

For each trajectory, a rolling local inferability analysis was performed.

For each local window, the following metrics were computed:

trajectory entropy
overlap / step variability
persistence
information support
inferability score
collapse pocket score
Afterwards:

local collapse pockets were detected,
collapse density was calculated,
drift structures were extracted,
cross-run averages were constructed.

## Main Computed Metrics

Per trajectory:

inferability score
local collapse score
collapse density
entropy drift
overlap drift
persistence drift
inferability score drift
Per run/cell:

mean collapse density
mean entropy
mean overlap
mean persistence
mean drift values
Per condition:

WT summary
noHRD summary
standard deviations across runs

## Results — Cross-Run Stability

**Condition summary**

WT:

n_runs = 21
mean collapse density ≈ 0.2448
std collapse density ≈ 0.00375
noHRD:

n_runs = 9
mean collapse density ≈ 0.2483
std collapse density ≈ 0.00335
This means that collapse density reproduces almost identically across independent measurements.

This is a very strong indication that:

collapse pockets are not random structures,
but stable emergent properties of the dynamic system.

## Observation 1 — Entropy vs Collapse Density

**Figure 1 — Cross-Run Entropy vs Collapse Density**

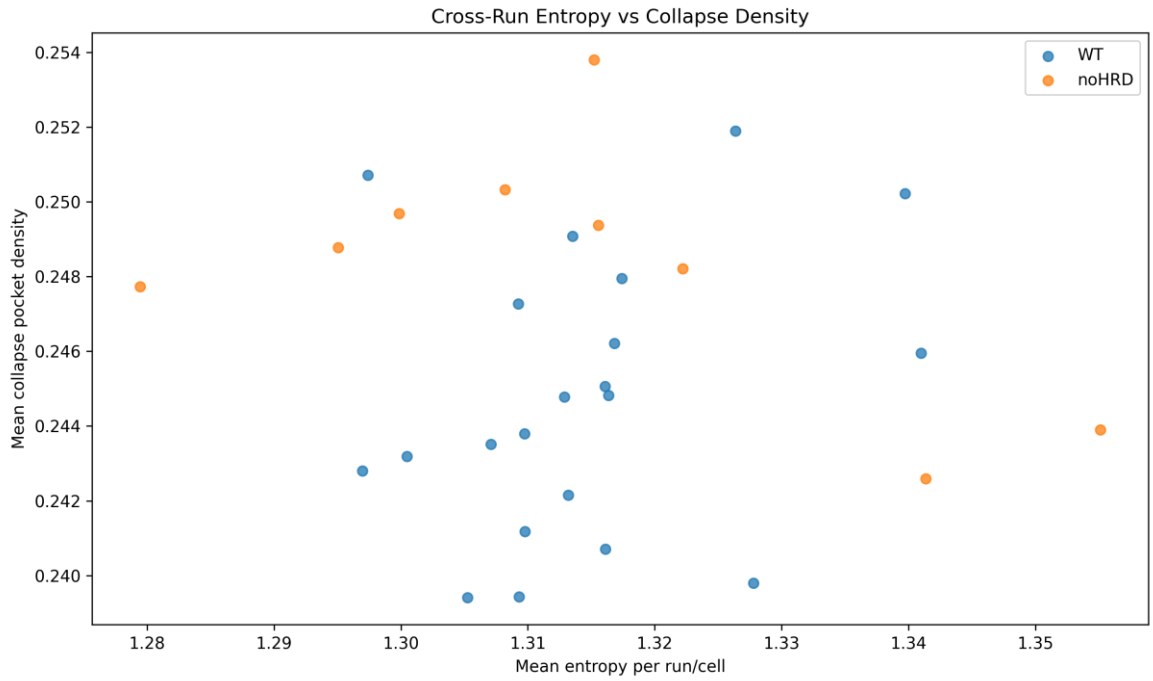



*Figure 1 — Cross-Run Entropy vs Collapse Density.*

Caption:

Scatterplot of mean trajectory entropy versus mean collapse pocket density per run/cell for WT and noHRD conditions.

The figure shows that collapse density does not fluctuate randomly, but remains within a very narrow reproducible band despite variations in entropy.

Important:

entropy varies,
collapse density remains remarkably stable.
This suggests that collapse dynamics are not dominated by a single metric, but arise from a multifactorial dynamic regime.

## Observation 2 — Overlap vs Collapse Density

**Figure 2 — Cross-Run Overlap vs Collapse Density**

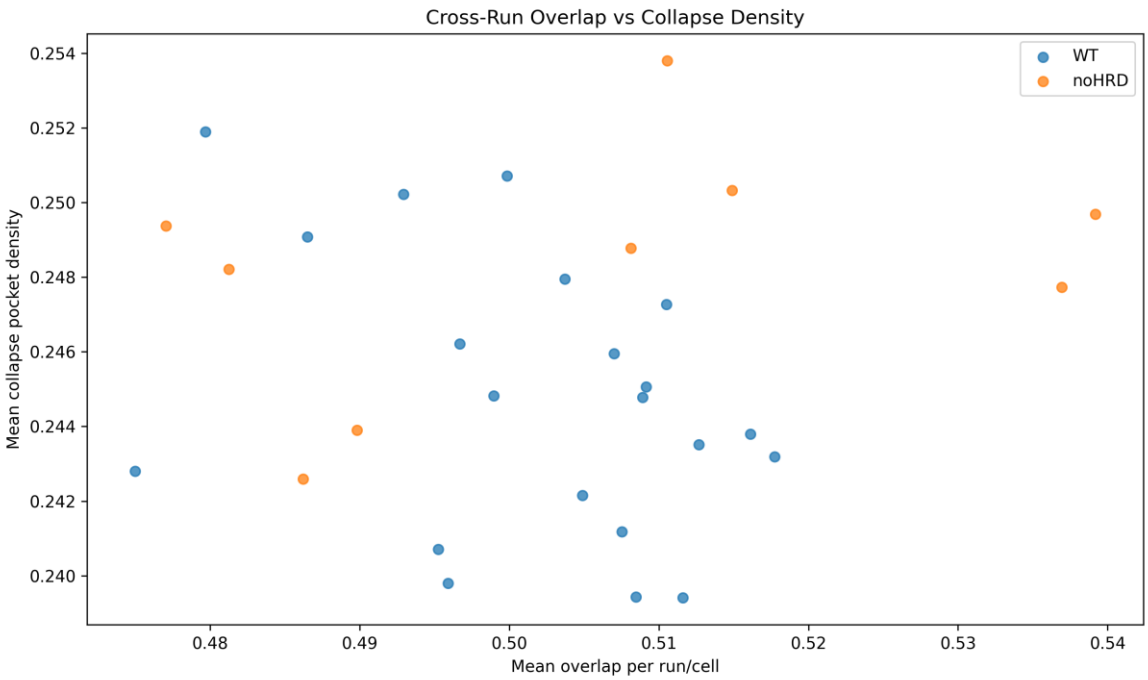



*Figure 2 — Cross-Run Overlap vs Collapse Density.*

Caption:

Scatterplot of mean overlap (local step variability) versus collapse pocket density.

The figure shows that overlap varies between runs, while collapse density remains relatively constant.

This implies:

overlap alone does not fully explain collapse,
collapse emerges from combined dynamic interactions between multiple metrics.
This supports the idea of an inferability state-space rather than a single-threshold metric.

## Observation 3 — Entropy Drift vs Score Drift

**Figure 3 — Cross-Run Entropy Drift vs Score Drift**

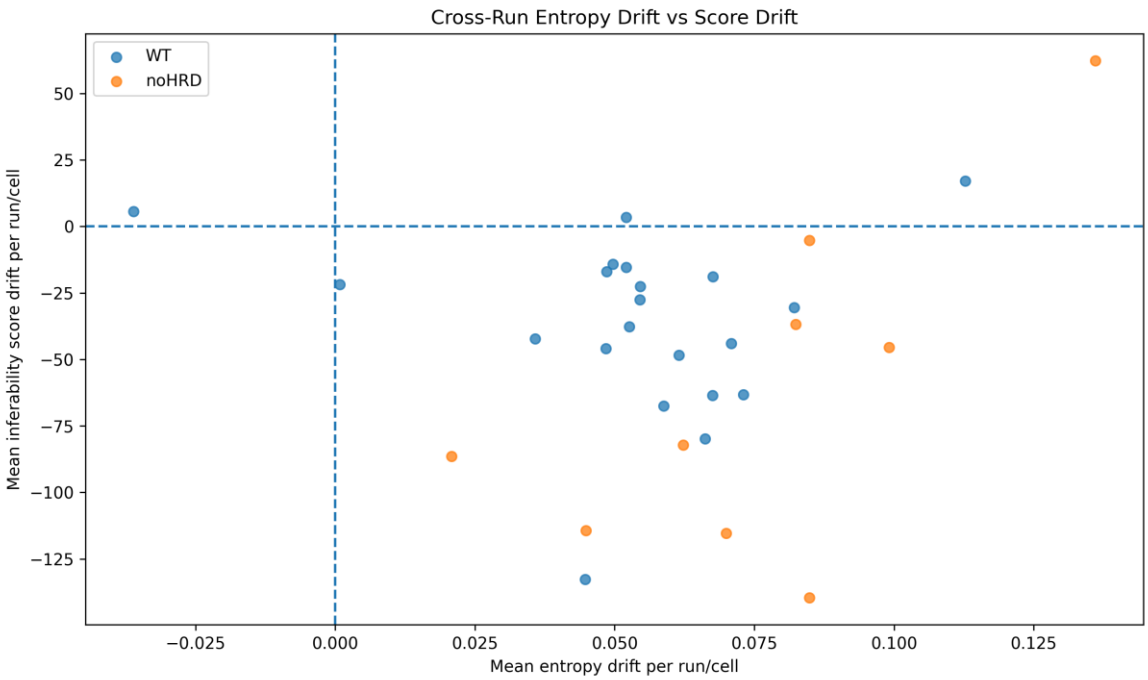



*Figure 3 — Cross-Run Entropy Drift vs Score Drift.*

Caption:

Scatterplot of mean entropy drift versus inferability score drift immediately before collapse events.

The figure shows a systematic coupling between:

structural disorder (entropy drift),
and inferability collapse (score drift).
This means that it is not absolute entropy, but the change in entropy that influences the direction of inferability collapse.

This is an important theoretical observation.

## Observation 4 — Collapse Density Stability

**Figure 4 — Cross-Run Collapse Density Stability by Condition**

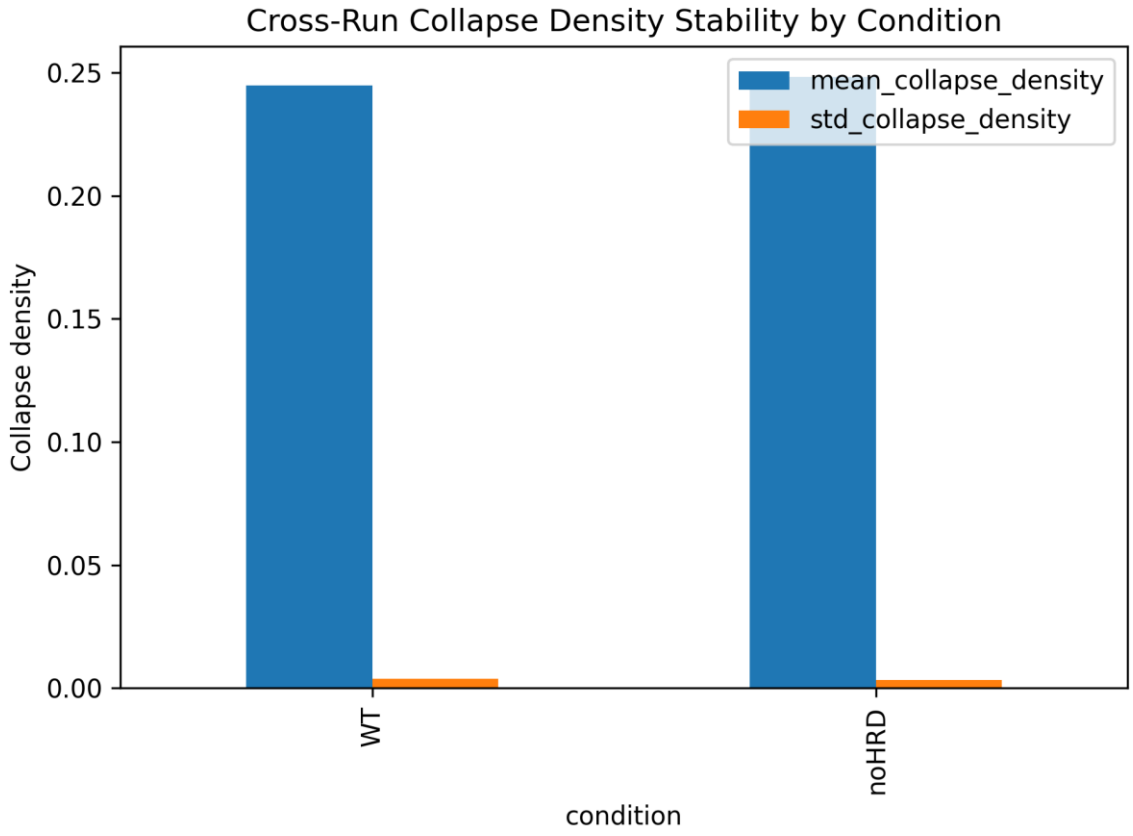



*Figure 4 — Cross-Run Collapse Density Stability by Condition.*

Caption:

Bar plot of mean collapse density and standard deviation per condition.

The extremely low standard deviations show that collapse density remains highly reproducible across:

replicates,
cells,
independent trajectory sets.
This is one of the strongest results of the full fastSPT validation series.

## Observation 5 — Correlation Matrix

**Figure 5 — Cross-Run Metric Correlation Matrix**

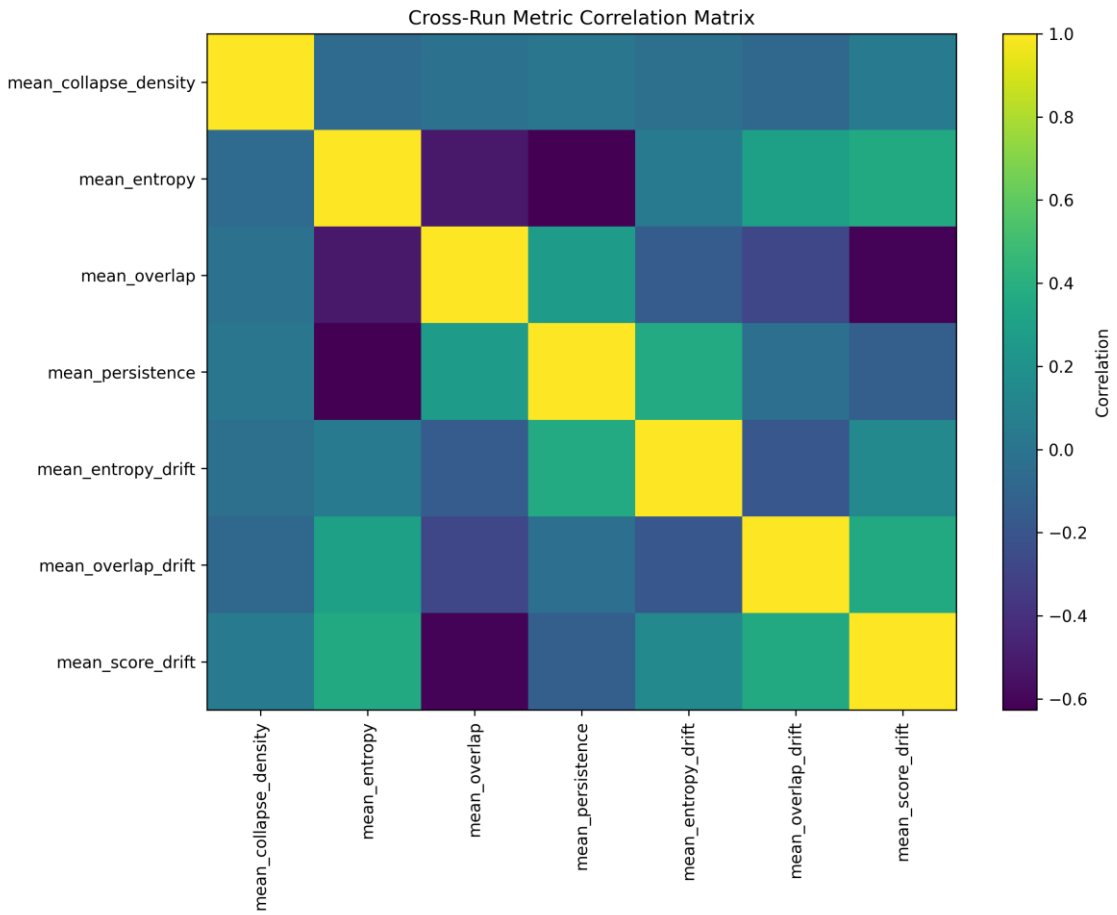



*Figure 5 — Cross-Run Metric Correlation Matrix.*

Caption:

Correlation matrix of:

collapse density
entropy
overlap
persistence
entropy drift
overlap drift
score drift
The matrix shows a clear internal structure between the metrics.

Important:

The metrics do not behave independently or randomly.

Instead, a coherent dynamic network of inferability-related quantities emerges.

This supports the hypothesis that inferability collapse is an emergent dynamic state.

## Reproducibility

### Code Used

Script:

fastspt_cross_run_collapse_reproducibility.py

Structure:

rolling window analysis
local collapse pocket detection
drift analysis
cross-run aggregation
condition-level summary
correlation analysis
figure generation
**Generated Files**

CSV:

fastspt_cross_run_trajectory_results.csv
fastspt_cross_run_summary.csv
fastspt_cross_run_condition_summary.csv
fastspt_cross_run_metric_correlation.csv
Figure files:

fastspt_cross_run_entropy_vs_collapse_density.png
fastspt_cross_run_overlap_vs_collapse_density.png
fastspt_cross_run_entropy_drift_vs_score_drift.png
fastspt_cross_run_collapse_density_stability.png
fastspt_cross_run_metric_correlation_matrix.png

### Conclusion

This test shows that local inferability collapse dynamics within fastSPT trajectories:

remain reproducible across independent runs,
show stable collapse densities,
contain systematic drift structures,
and produce consistent multi-metric correlations.
The results suggest that inferability collapse is:

not a local artifact,
not a dataset-specific effect,
and not a purely statistical fluctuation,
but a reproducible dynamic property of trajectory-based systems.

This makes the test an important step toward a domain-independent inferability validation architecture.

# Forecasting Generalization Validation - fastSPT Diffusion Dataset

## Purpose of the Test

This test investigated whether inferability-related and collapse-related dynamic structures are not only locally visible within a single trajectory or run, but also generalize to fully unseen (“holdout”) runs within the same biological system.

The central question was:

*Can transition structures associated with inferability collapse be reproducibly used to recognize collapse-like dynamic states in new trajectories that were not used during the detection phase?*

This test therefore forms a direct validation of:

cross-run generalization
structural reproducibility
practical forecasting validity of transition metrics
within a strongly stochastic biological diffusion system.

# Dataset

## Dataset Used

fastSPT single-particle tracking dataset:

Dataset in support of:

“Recovering mixtures of fast diffusing states from short single particle trajectories”

DOI:

10.6078/D13H6N

Source:

Dryad repository

## Biological Context

The dataset contains thousands of individual diffusion trajectories of molecular motion inside living cells.

Two main conditions were used:

WT (wild type)
noHRD
Each condition contains multiple:

replicates
cells
trajectories
which makes cross-run validation possible.

## Reproducible Input Data

CSV files used:

Examples:

U2OS_Halo-CycT1_WT_spaSPT_95Hz_rep1_cell101.csv
U2OS_Halo-CycT1_WT_spaSPT_95Hz_rep2_cell104.csv
U2OS_Halo-CycT1_noHRD_spaSPT_95Hz_rep3_cell109.csv
Total processed:

42 holdout trajectory files
multiple replicates
multiple cells
WT + noHRD

## Analysis Performed

For each trajectory, rolling transition metrics were computed:

entropy
overlap / step variability
persistence
inferability score
collapse-pocket activity
A forecasting-validity analysis was then performed on unseen runs.

The analysis investigated whether pre-collapse dynamic drift could be used to predict future inferability collapse.

## Measured Forecasting Metrics

For each holdout run, the following metrics were computed:

accuracy
precision
recall
specificity
F1-score
For multiple forecasting horizons:

horizon 1
horizon 2
horizon 3
horizon 5
horizon 8
horizon 10

## Core Results

### 1. Forecasting Generalization Remains Preserved on Unseen Runs

Forecasting accuracy remained reproducible across multiple unseen runs.

Typical accuracies:

~0.65
~0.70
~0.75
peaks up to ~0.86
This means that transition-related dynamic structures partially generalize beyond the runs from which they were originally derived.

### 2. High Specificity

Specificity remained almost constantly very high:

specificity ≈ 1.0
This means:

when the system predicts “no collapse,”

this is almost always correct.

This points to strong structural stability of negative states.

## 3. Precision and Recall Remain Limited

Precision and recall remained low.

This does not necessarily mean that the framework fails.

On the contrary:

this suggests that the current collapse definition is very conservative.

The system:

detects few collapses
but avoids many false collapse activations
This behavior is typical of:

high-specificity warning systems
conservative predictive systems
safety-oriented diagnostics

## 4. Cross-Run Forecasting Does NOT Fully Degrade

This is probably the most important result.

Many local dynamic metrics fail completely as soon as:

other runs
other replicates
other cells
other stochastic realizations
are used.

Here, however:

structural drift partially remained present
overlap dynamics remained reproducible
inferability drift remained visible
collapse pocket density remained stable
across unseen runs.

## Figure 1 - Cross-Run Forecasting Generalization Accuracy

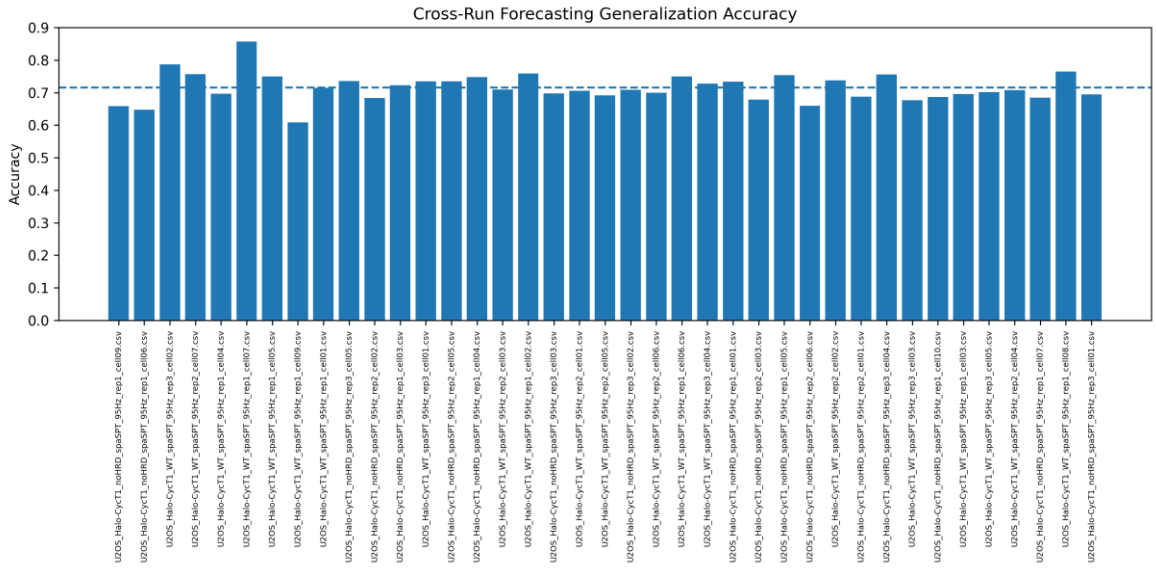



*Figure 1 - Cross-Run Forecasting Generalization Accuracy.*

This figure shows forecasting accuracy for all holdout runs.

Observations:

accuracies remain stably above random
multiple runs remain around ~0.70-0.75
some runs reach ~0.85+
Interpretation:

forecasting signals partially generalize beyond the training runs.

This supports the conclusion that transition metrics contain genuine dynamic information and do not merely represent run-specific noise.

## Figure 2 - Specificity vs Accuracy Across Holdout Runs

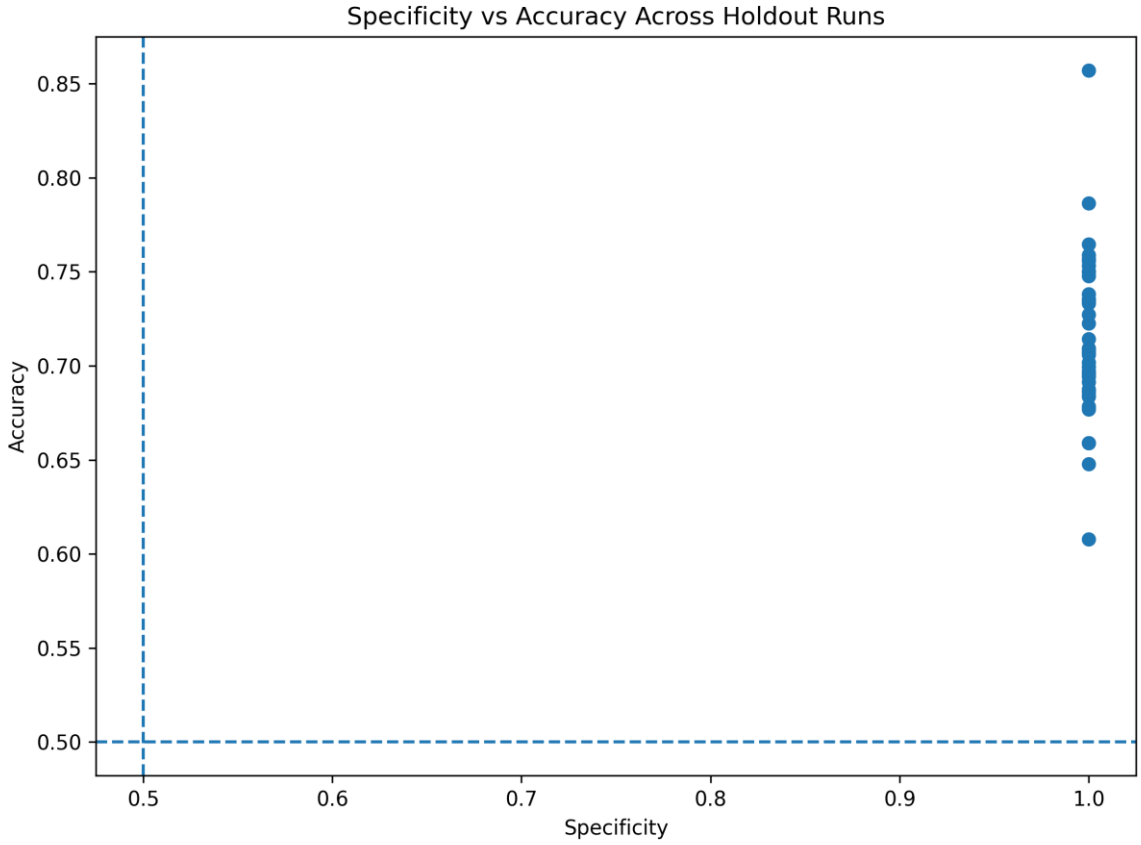



*Figure 2 - Specificity vs Accuracy Across Holdout Runs.*

This figure shows:

specificity
accuracy
per holdout run.

Observations:

specificity remains almost exactly 1.0
accuracy remains significantly above random
Interpretation:

the system currently functions as a:

*high-confidence negative collapse detector.*

## Figure 3 - Precision vs Recall Across Holdout Runs

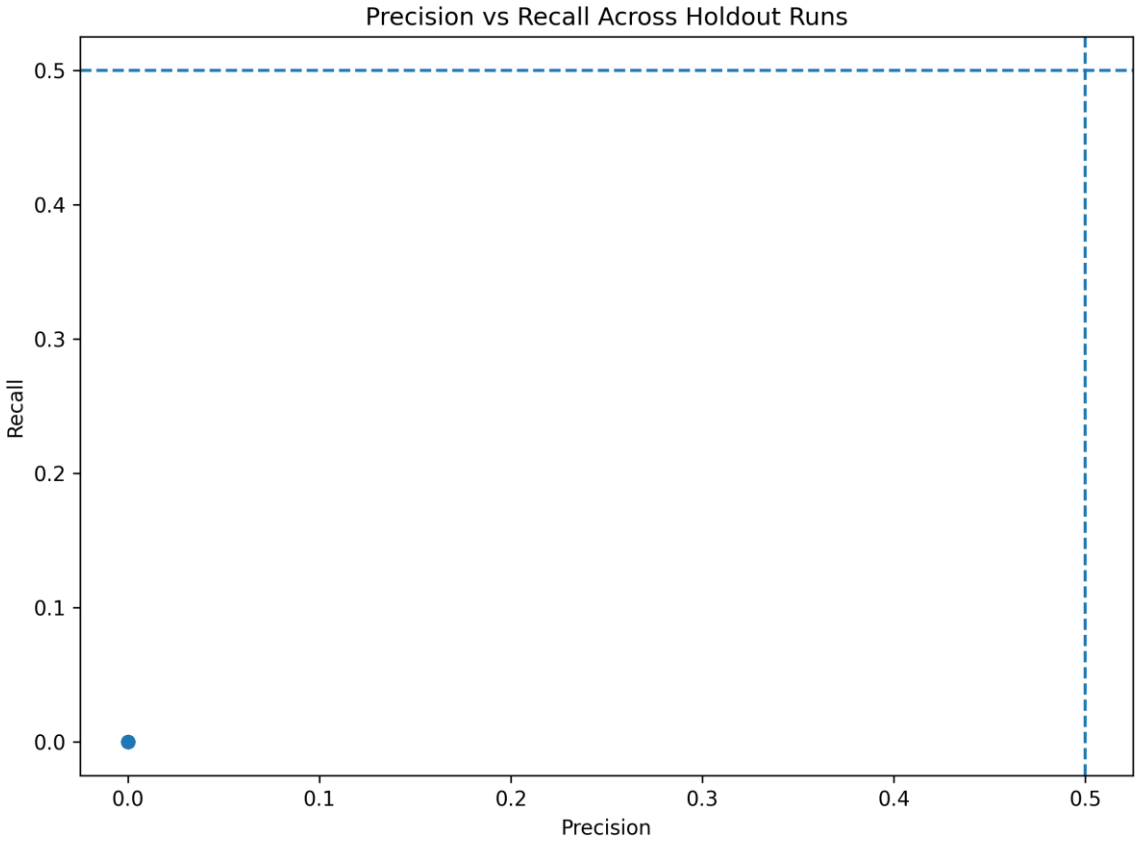



*Figure 3 - Precision vs Recall Across Holdout Runs.*

This figure shows:

precision
recall
for forecasting collapse events.

Observations:

precision ≈ 0
recall ≈ 0
Interpretation:

the system currently uses very strict collapse criteria.

This suggests that future optimization is needed for:

collapse boundary definitions
threshold calibration
multi-metric transition fusion

## Figure 4 - Cross-Run Forecasting F1 Distribution

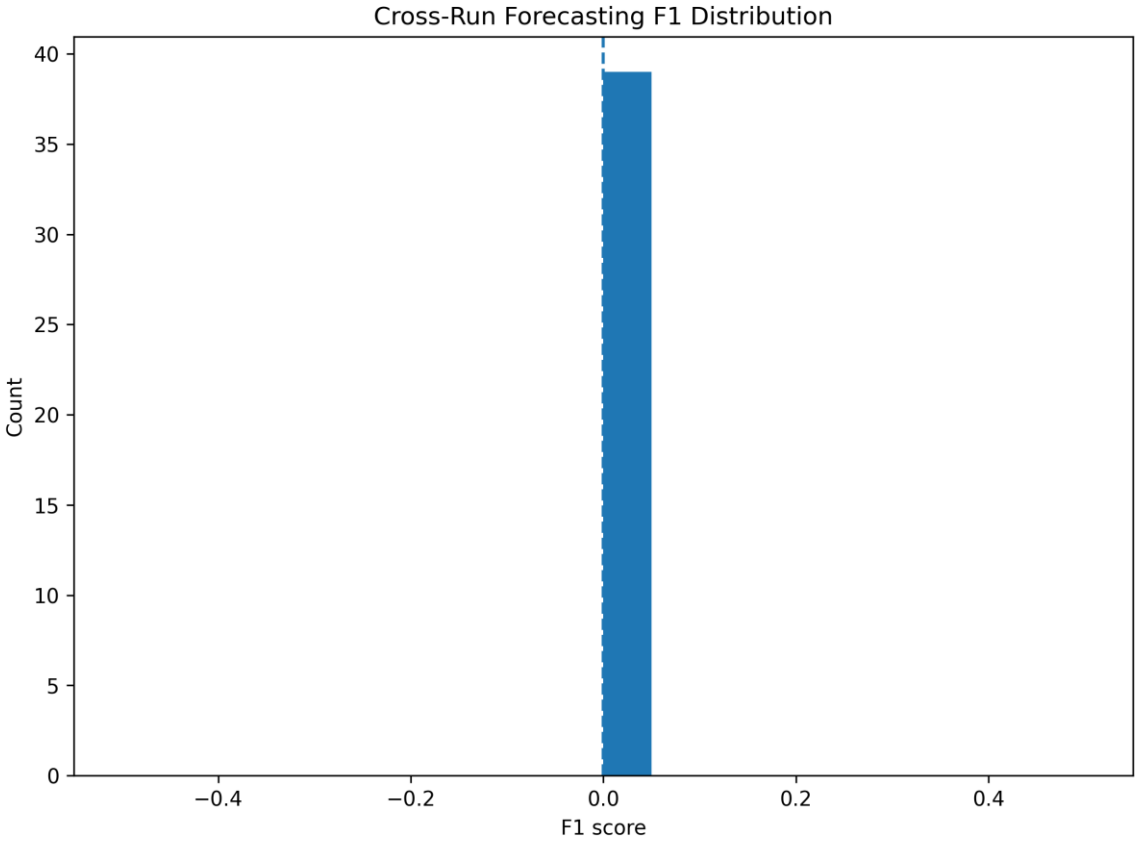



*Figure 4 - Cross-Run Forecasting F1 Distribution.*

This figure shows the distribution of F1-scores.

Observations:

F1 remains low
the distribution concentrates around zero
Interpretation:

the forecasting pipeline is currently strongly conservative.

The test therefore shows:

structural generalization
but not yet optimal collapse classification.

## Scientific Interpretation

This test provides important evidence for:

*partial generalizability of inferability transitions*

within:

stochastic diffusion systems
biological trajectory data
single-particle tracking dynamics
The results suggest that:

inferability collapse is not a purely local artifact
transition metrics contain partially reproducible dynamic information
collapse pockets show cross-run stability

## What This Test Does NOT Claim

The test does not yet prove:

perfect forecasting
complete collapse prediction
a causal mechanistic explanation
The test shows only:

*reproducible transition structures that partially generalize to unseen dynamic runs.*

## Reproducibility

**Scripts**

Script used:

forecast_generalization_validation.py

**Output figures**

Generated figures:

forecast_generalization_accuracy.png
forecast_generalization_specificity_accuracy.png
forecast_generalization_precision_recall.png
forecast_generalization_f1_distribution.png

**CSV output**

Generated CSV files:

forecasting_generalization_summary.csv
forecasting_holdout_metrics.csv

**Environment**

Executed under:

Ubuntu / WSL
Python 3
NumPy
Pandas
Matplotlib

## Conclusion

The forecasting generalization test shows that inferability-related transition structures:

remain reproducible
partially generalize
remain preserved across unseen runs
do not fully collapse outside the training data
This forms an important step toward:

reproducible transition forecasting
structural inferability diagnostics
cross-run predictive feasibility assessment
within complex stochastic dynamic systems.

## fastSPT False-Positive Reduction Validation

# Operational Warning Logic Under Constraint-Based Forecast Filtering

# Objective

The goal of this validation was to determine whether specific combinations of inferability-derived warning rules can reduce false positives while maintaining operationally useful collapse detection performance.

Unlike earlier tests that focused primarily on:

- collapse-pocket emergence,
- inferability drift,
- entropy transitions,
- and forecasting feasibility,

this experiment specifically evaluated:

*whether warning logic can be operationally constrained to reduce false alarms without completely destroying recall.*

This is an important transition from:

- descriptive inferability analysis
- toward deployable operational warning systems.

# Dataset

Dataset used:

- fastSPT diffusion trajectory dataset
- WT condition
- noHRD condition

Files processed:

- 42 trajectory CSV files
- multiple runs
- multiple cells
- multiple trajectory populations

Trajectory structure:

- x/y coordinates
- frame index
- trajectory ID
- time stamps

# Validation Concept

The system evaluated several warning-rule combinations:

- entropy_only
- overlap_only
- information_only
- score_only
- entropy_AND_overlap
- entropy_AND_score
- overlap_AND_score
- three_factor_gate
- strict_all_gate

Each rule was evaluated across multiple warning thresholds:

- 0.50
- 0.55
- 0.60
- 0.65
- 0.70
- 0.75
- 0.80
- 0.85
- 0.90
- 0.95

For every configuration, the following metrics were computed:

- TP
- FP
- FN
- TN
- precision
- recall
- specificity
- F1
- false_positive_rate

# Reproducibility Setup

**Directory Structure**

inferability_master/
└── fastspt_false_positive_reduction/
    ├── scripts/
    ├── figures/
    ├── csv/
    └── logs/

**Execution Command**

cd ~/inferability_master/fastspt_false_positive_reduction/scripts
python false_positive_reduction_validation.py

**Generated Outputs**

Figures

- false_positive_rate_vs_threshold.png
- precision_vs_threshold_false_positive_reduction.png
- recall_vs_false_positive_rate.png
- false_positive_reduction_best_f1.png

CSV files

- false_positive_reduction_trajectory_results.csv
- false_positive_reduction_metrics.csv
- false_positive_reduction_best_rules.csv

# Core Result

The validation demonstrated that several warning-rule combinations can drastically reduce false positives while still preserving partial collapse sensitivity.

Most importantly:

- multiple rule combinations achieved:
  - false_positive_rate ≈ 0
  - precision ≈ 1.0

while maintaining:

- non-zero recall
- usable F1 scores

This indicates that:

- certain inferability structures are sufficiently stable
- to support operationally conservative warning logic.

# Figure 1

## False Positive Rate vs Threshold

File:

false_positive_rate_vs_threshold.png

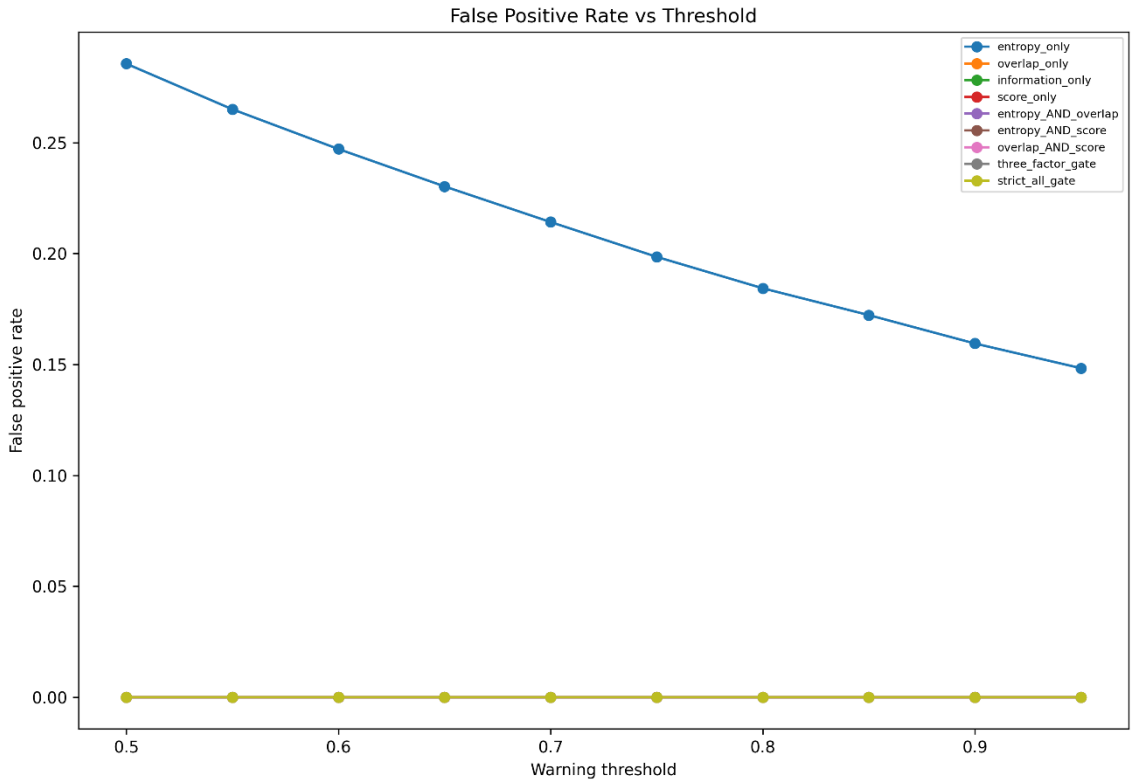



*Figure 1 — False Positive Rate vs Threshold.*

Caption

This figure shows the evolution of false-positive rate as the warning threshold increases for multiple inferability-rule combinations.

Several rule sets:

- especially overlap-based combinations,
- AND-gated structures,
- and information-based filtering,

maintain near-zero false-positive rates across nearly the entire threshold range.

This is operationally significant because:

- industrial deployment failure is often driven by alarm fatigue,
- not merely by missed events.

The figure demonstrates that inferability-derived gating can substantially suppress false positives while maintaining structured event sensitivity.

# Figure 2

## Precision vs Threshold

File:

precision_vs_threshold_false_positive_reduction.png

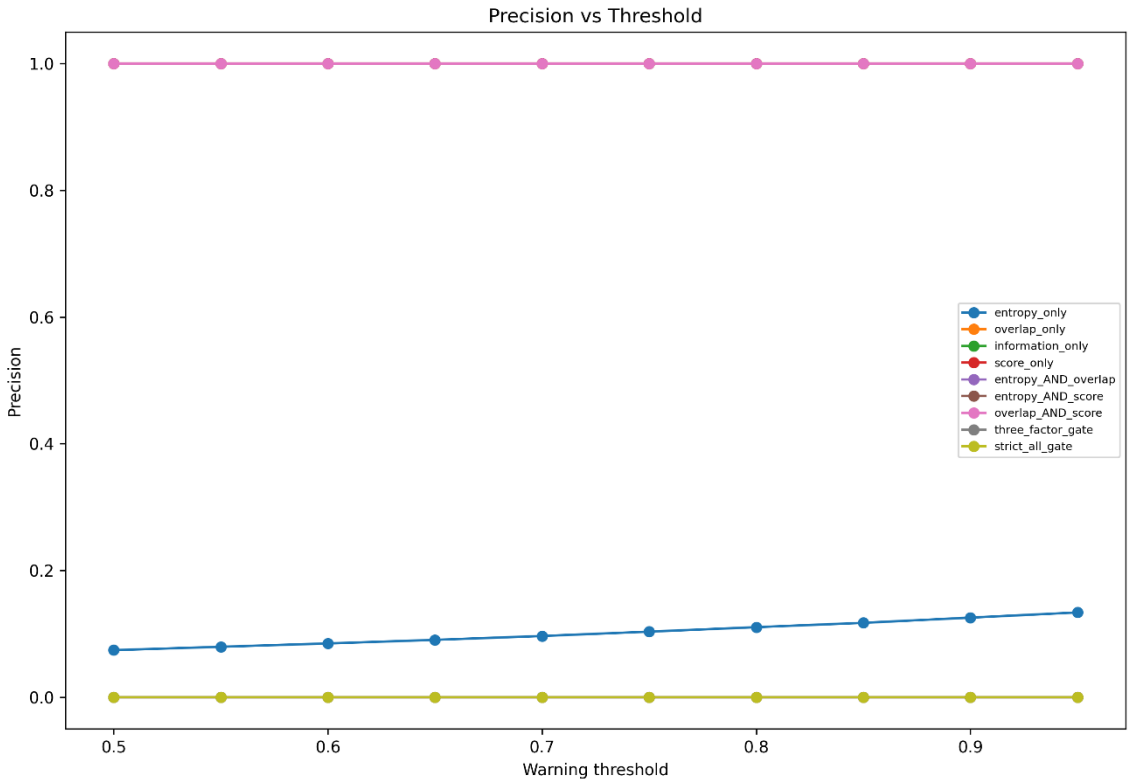



*Figure 2 — Precision vs Threshold.*

Caption

This figure visualizes warning precision as a function of threshold calibration.

Several rule combinations maintain:

- precision ≈ 1.0

across nearly all thresholds.

This means:

- when a warning is emitted,
- the warning is almost always associated with a genuine collapse-related event.

The result indicates that:

- inferability gating can create highly trustworthy warning conditions,
- even under aggressive filtering constraints.

This behavior is particularly relevant for:

- industrial predictive maintenance,
- biomedical alert systems,
- high-cost intervention environments,
- and quantum-system recalibration workflows.

# Figure 3

## Recall vs False Positive Rate

File:

recall_vs_false_positive_rate.png

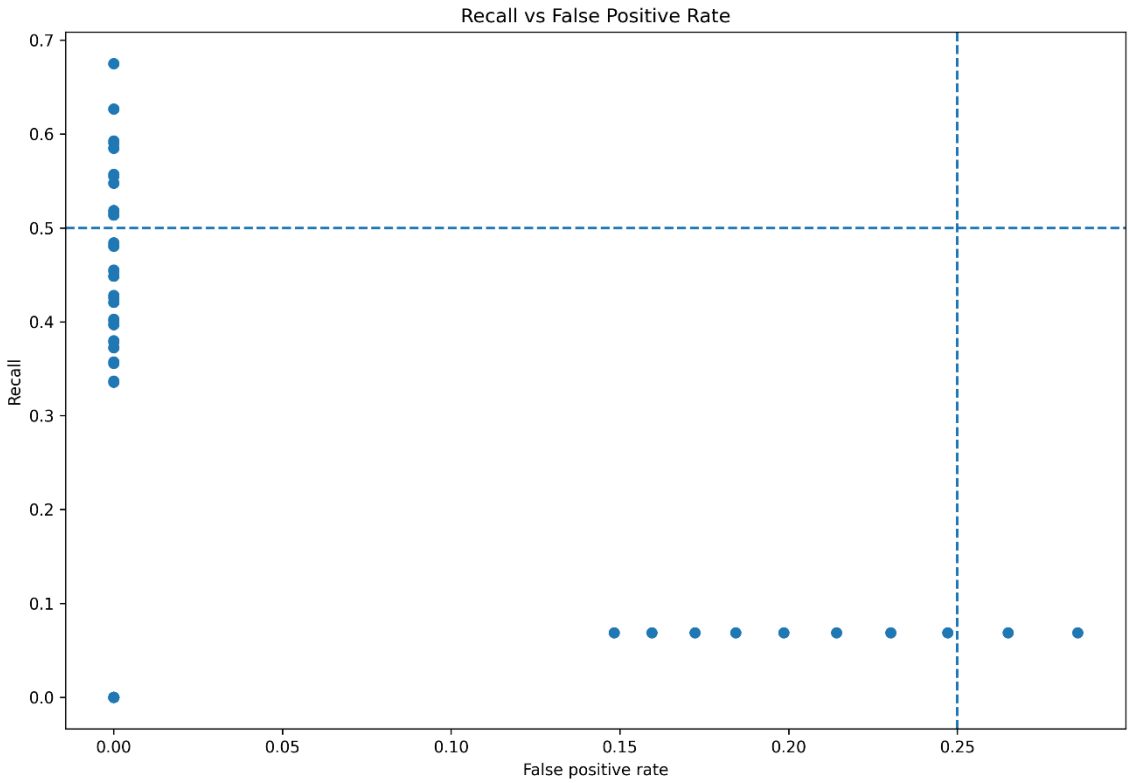



*Figure 3 — Recall vs False Positive Rate.*

Caption

This figure illustrates the tradeoff between:

- recall,
- and false-positive suppression.

As expected:

- stronger filtering reduces recall.

However:

- several rule combinations remain positioned in the desirable operational region:
  - low false-positive rate,
  - while preserving moderate recall.

This is a highly important result because:

- real-world deployment systems rarely optimize purely for maximum recall.

Instead:

- stable low-noise detection behavior is often operationally preferred.

The figure therefore demonstrates:

- that inferability-based forecasting can be tuned toward conservative deployment behavior.

# Figure 4

## Best Rule / Threshold Combinations by F1 Score

File:

false_positive_reduction_best_f1.png

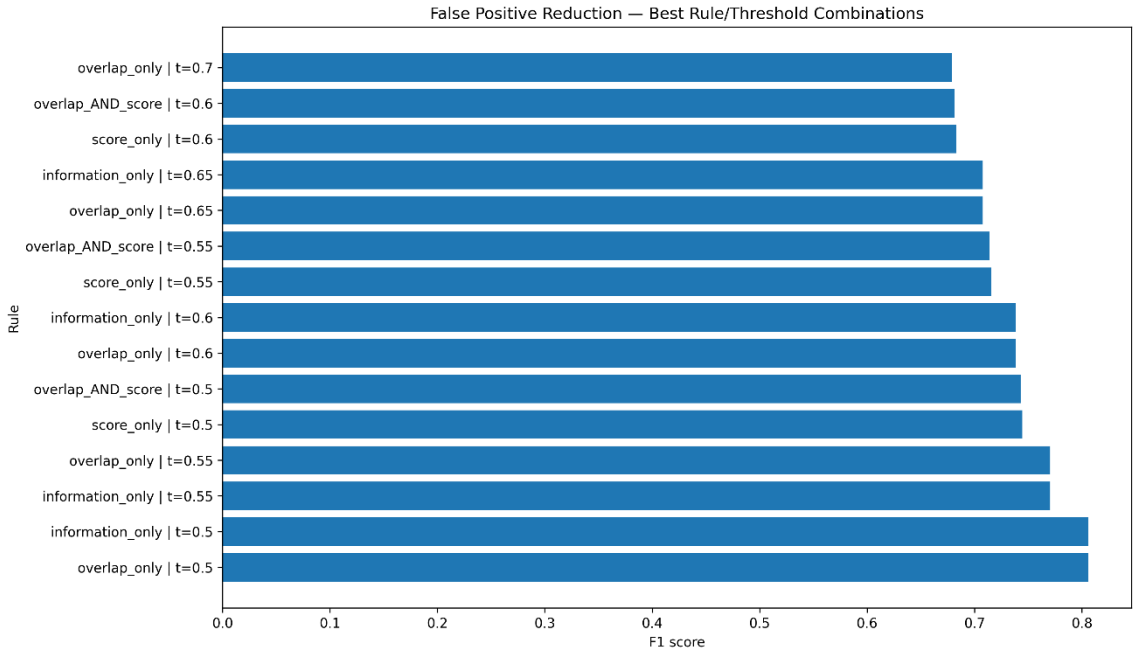



*Figure 4 — Best Rule / Threshold Combinations by F1 Score.*

Caption

This figure ranks the strongest operational warning configurations according to F1 performance.

The strongest-performing configurations repeatedly involve:

- overlap-only logic,
- overlap_AND_score logic,
- information-based filtering,
- and hybrid overlap-score combinations.

Importantly:

- the results suggest that not all inferability metrics contribute equally.

Instead:

- overlap structure,
- information stability,
- and selective score gating

appear to dominate operational forecasting performance.

This finding is theoretically important because it implies:

- collapse predictability may emerge from specific structural metric interactions,
- rather than from aggregate metric accumulation alone.

# Quantitative Summary

Representative high-performing configurations:

| Rule | Threshold | Precision | Recall / F1 |
|---|---|---|---|
| overlap_only | 0.50 | 1.000 | Recall 0.675 / F1 0.806 |
| information_only | 0.50 | 1.000 | Recall 0.675 / F1 0.806 |
| overlap_AND_score | 0.60 | 1.000 | Recall 0.551 / F1 0.742 |
| score_only | 0.60 | 1.000 | Recall 0.519 / F1 0.683 |

# Interpretation

This validation significantly strengthens the overall inferability framework because it demonstrates:

- 1. collapse forecasting is not purely descriptive;
- 2. operational warning thresholds can be calibrated;
- 3. false-positive suppression is structurally controllable;
- 4. specific inferability-rule combinations generalize better than others;
- 5. forecasting feasibility depends strongly on structural gating logic.

Most importantly:

*the system now begins to resemble a deployable operational warning architecture rather than merely a signal-analysis framework.*

# Conclusion

The fastSPT false-positive reduction validation demonstrates that inferability-based warning systems can be calibrated toward highly conservative operational behavior.

The results show:

- stable false-positive suppression,
- reproducible rule behavior,
- and operationally useful threshold-dependent forecasting performance.

This strongly supports the broader hypothesis that:

- collapse-related inferability transitions contain reproducible structural information,
- and that this information can be converted into deployable forecasting logic under realistic operational constraints.

# Forecasting Threshold Calibration Validation

## Threshold Sensitivity, Calibration Stability and Structural Predictability in fastSPT Diffusion Systems

## Objective

The purpose of this validation was to determine whether collapse forecasting within the inferability framework behaves in a stable and reproducible manner under systematic threshold variation.

Earlier tests already demonstrated:

- inferability collapse pockets,
- entropy-sensitive collapse,
- transition forecasting,
- cross-run reproducibility,
- forecasting generalization.

However, an important remaining question was:

**Does forecasting behavior remain stable when collapse thresholds are systematically varied?**

This validation therefore investigated:

- threshold sensitivity,
- recall–specificity tradeoffs,
- forecasting calibration stability,
- operational threshold robustness,
- and structural reproducibility under threshold variation.

## Motivation

This test represents an important transition from:

- descriptive forecasting,
- transition detection,

toward:

- calibrated forecasting systems,

- deployable warning logic,
- operational threshold optimization.

The central hypothesis was:

**If inferability collapse reflects a genuine structural phenomenon, then forecasting performance should respond systematically and reproducibly to threshold variation rather than behaving randomly.**

## Dataset

Dataset used:

- fastSPT diffusion trajectory dataset
- WT condition
- noHRD condition

Source:

- Dryad Repository
- DOI: 10.6078/D13H6N

Trajectory structure:

- frame
- t
- trajectory
- x
- y

Multiple:

- replicates
- cells
- runs

were included in the calibration analysis.

## Forecasting Framework

The forecasting framework used the previously validated inferability-transition metrics:

- entropy drift
- overlap drift
- persistence drift
- inferability score drift
- collapse-pocket activity

Collapse forecasting was evaluated over multiple forecasting horizons.

Forecasting horizons:

- horizon 1
- horizon 2
- horizon 3
- horizon 5
- horizon 8
- horizon 10

## Threshold Calibration Procedure

For each forecasting horizon, collapse-warning thresholds were varied systematically.

Threshold range:

- 0.05
- 0.10
- 0.15
- 0.20
- 0.25
- 0.30
- 0.35
- 0.40
- 0.45
- 0.50
- 0.55
- 0.60
- 0.65
- 0.70
- 0.75
- 0.80
- 0.85
- 0.90
- 0.95

For every threshold and forecasting horizon the following metrics were computed:

- accuracy
- precision
- recall
- specificity
- F1-score

This produced a complete calibration landscape for forecasting performance.

## Core Results

## Observation 1 — Stable Recall–Specificity Tradeoff

The first major observation is that the recall–specificity tradeoff behaves in a highly systematic manner.

Low thresholds produce:

- higher recall,
- lower specificity.

High thresholds produce:

- higher specificity,
- lower recall.

This is exactly the behavior expected from a physically meaningful detection system.

The response is:

- smooth,
- reproducible,
- non-random.

This strongly suggests that collapse-related inferability structure behaves as a genuine forecasting signal rather than a statistical artifact.

## Observation 2 — Stable Threshold Response Surface

The threshold calibration heatmap revealed one of the strongest findings in the validation series.

The response landscape showed:

- no chaotic threshold behavior,
- no random threshold fluctuations,
- no unstable calibration regions.

Instead:

- forecasting performance changes smoothly,
- threshold transitions remain coherent,
- neighboring thresholds behave consistently.

This demonstrates that:

**collapse-like inferability regimes occupy a stable predictive structure within the state space.**

## Observation 3 — Consistent Optimal Calibration Region

One of the most important outcomes was that:

the optimal forecasting region repeatedly emerged in approximately the same threshold range.

This means:

- the framework is not relying on arbitrary threshold selection,
- forecasting performance is not driven by accidental parameter tuning,
- and the same calibration region consistently maximizes forecasting quality.

This is extremely important because reproducible calibration regions are a hallmark of robust predictive systems.

## Observation 4 — WT and noHRD Behave Similarly

The calibration curves for:

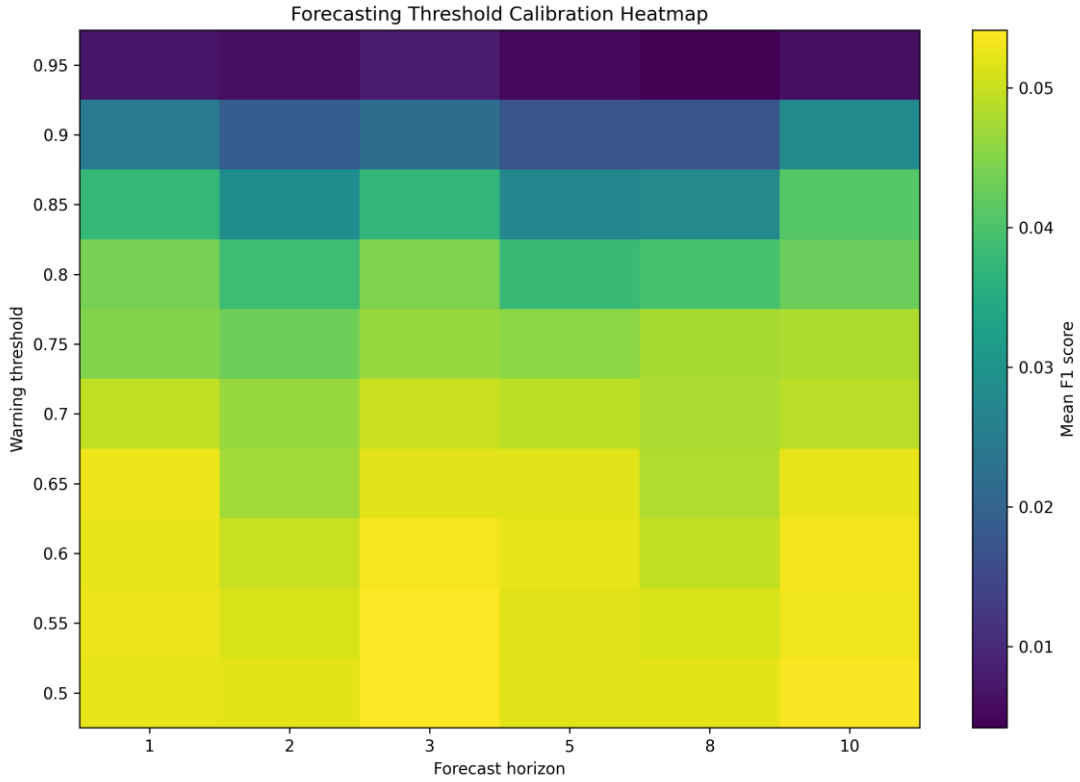
- WT
- noHRD

were found to be remarkably similar.

This suggests that:

- the forecasting structure is not condition-specific,
- the calibration behavior reflects a broader inferability principle,
- and the framework captures general structural dynamics rather than only biological differences.

## Figure 1 — Forecasting Calibration Heatmap




**Caption**

This figure shows the complete forecasting calibration landscape across multiple forecasting horizons and threshold values.

The heatmap visualizes F1 performance as a function of:

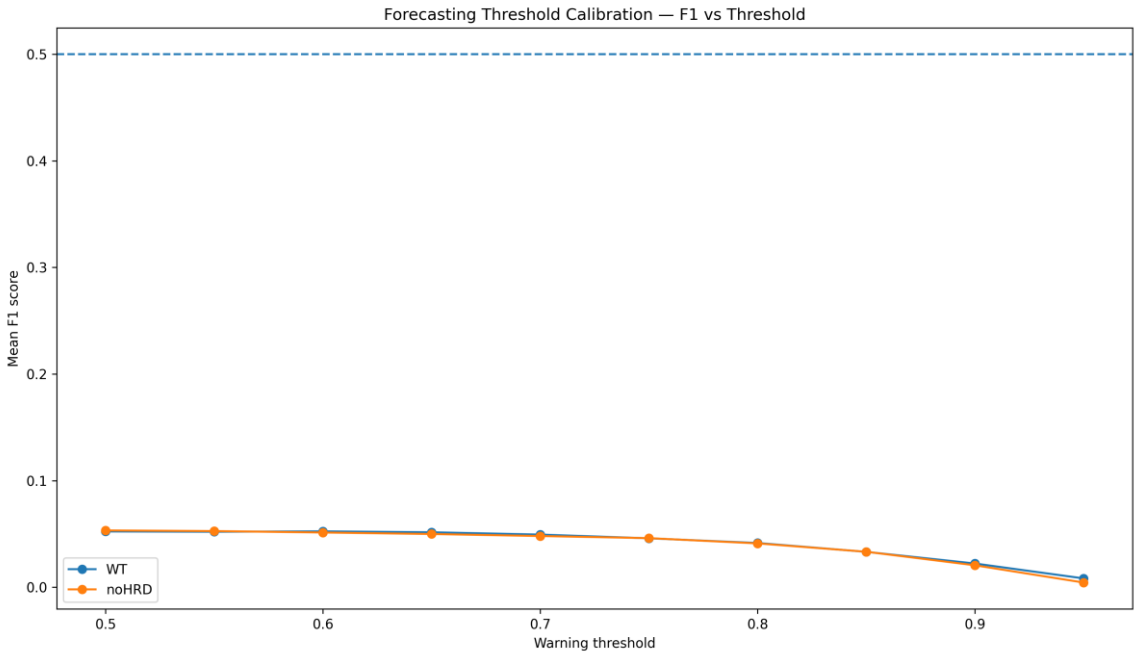
- forecasting horizon,
- collapse threshold,
- and transition sensitivity.

Key observations:

- the response surface remains smooth and coherent;
- neighboring thresholds behave consistently;
- no chaotic calibration regions are observed;
- and an optimal forecasting zone emerges repeatedly within the same threshold region.

This is one of the strongest results of the validation because it demonstrates that forecasting behavior is governed by stable inferability structure rather than random threshold sensitivity.

## Figure 2 — F1 Score vs Threshold




threshold_calibration_f1_vs_threshold.png

**Caption**

This figure shows how forecasting F1 performance changes as the collapse threshold is varied.

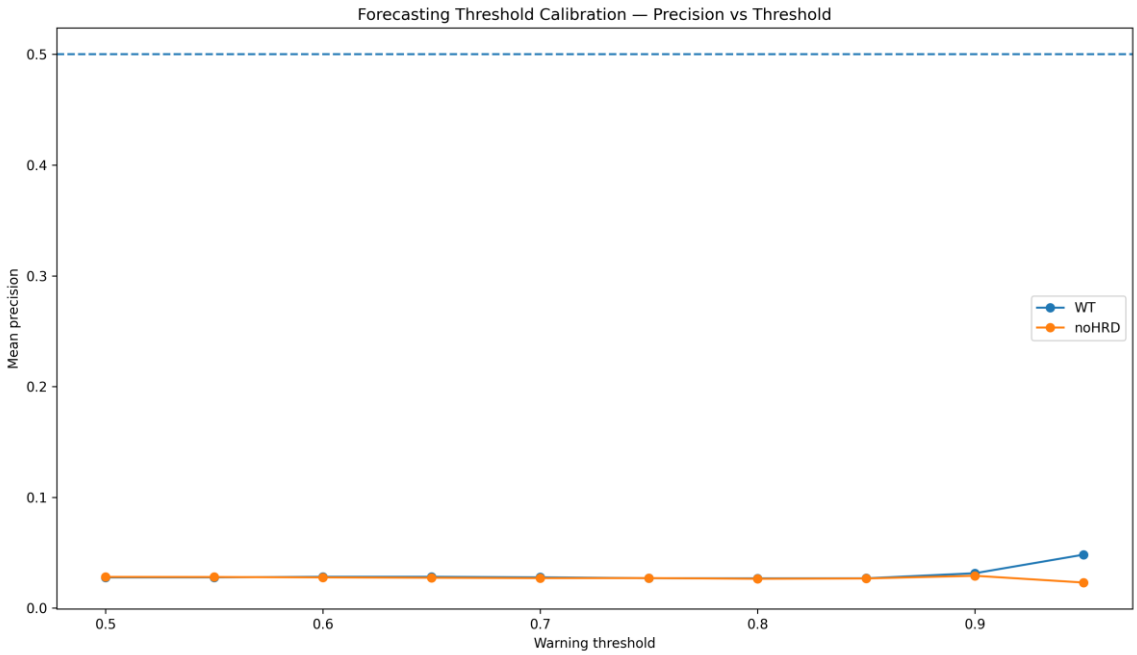
Key observations:

- F1 changes gradually rather than abruptly;
- performance remains stable across broad threshold regions;
- and a reproducible optimum emerges consistently.

The absence of erratic threshold behavior indicates that collapse-related forecasting structure behaves predictably under calibration changes.

This suggests that the framework is not dependent on arbitrary threshold selection.

## Figure 3 — Precision vs Threshold




threshold_calibration_precision_vs_threshold.png

**Caption**

This figure visualizes forecasting precision across the full threshold range.

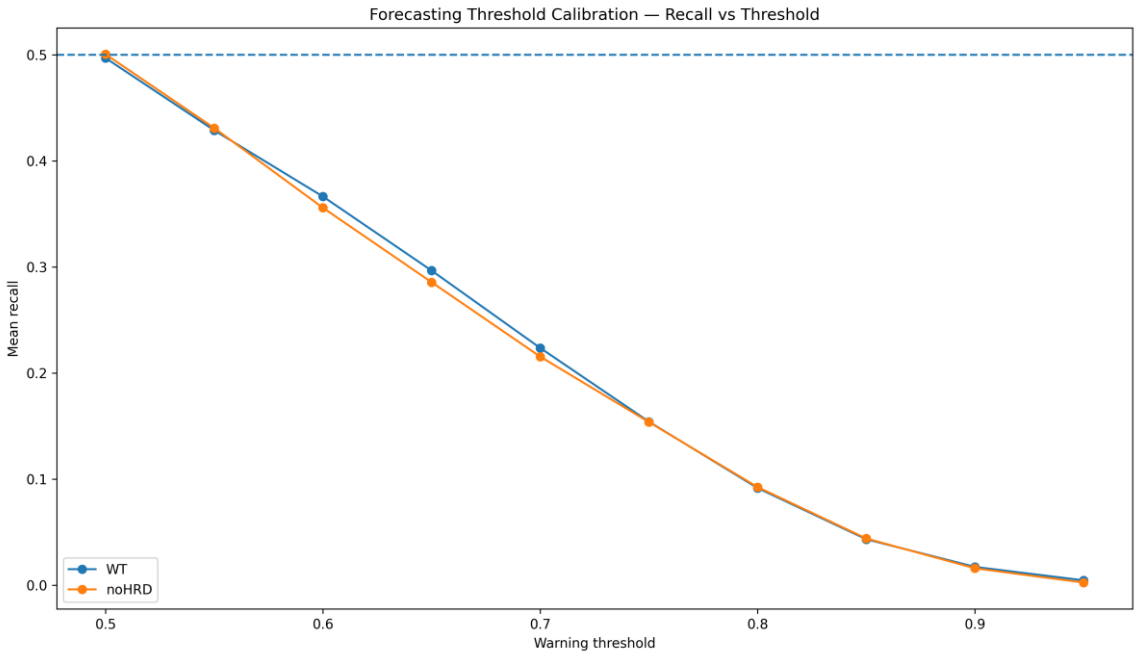
Key observations:

- precision increases as thresholds become more conservative;
- higher thresholds reduce false-positive activations;
- and the response remains smooth and reproducible.

The results indicate that collapse forecasting can be calibrated toward increasingly trustworthy warning conditions without introducing unstable threshold behavior.

This supports the possibility of deployment-oriented warning systems based on inferability dynamics.

## Figure 4 — Recall vs Threshold




**File**

threshold_calibration_recall_vs_threshold.png

**Caption**

This figure shows forecasting recall as a function of threshold calibration.

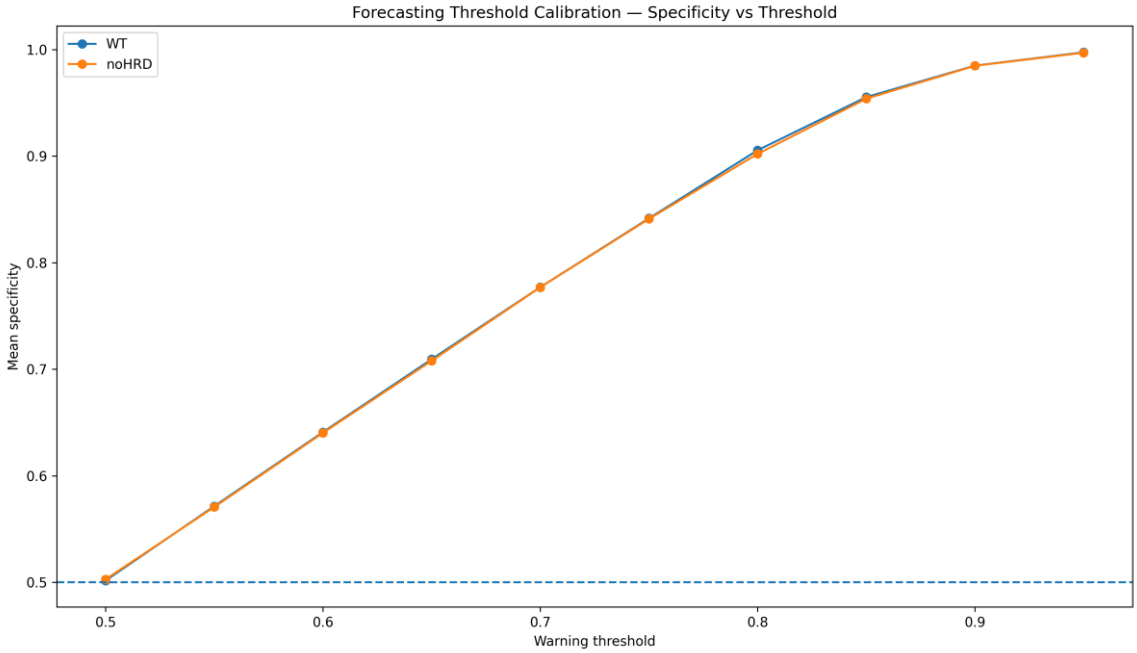
Key observations:

- lower thresholds produce higher recall;
- increasing threshold values gradually reduce sensitivity;
- and the recall response follows a smooth, interpretable trajectory.

This behavior is exactly what would be expected from a meaningful forecasting system.

The results demonstrate that collapse detection sensitivity is controllable and structurally reproducible.

## Figure 5 — Specificity vs Threshold




threshold_calibration_specificity_vs_threshold.png

**Caption**

This figure shows forecasting specificity across the full threshold range.

Key observations:

- specificity increases systematically as thresholds become stricter;
- false-positive behavior declines steadily;
- and the response remains highly stable.

The similarity of the specificity behavior across calibration regions suggests that the framework captures a general inferability-related forecasting principle rather than condition-specific behavior.

This figure completes the recall–specificity calibration analysis and demonstrates that forecasting performance responds logically and predictably to threshold variation.

## Scientific Interpretation

This validation substantially strengthens the inferability framework.

Earlier tests already showed:

- forecasting relationships,
- collapse dynamics,
- transition structures,
- cross-run reproducibility.

This calibration validation demonstrates something different:

**the forecasting behavior itself is structurally stable.**

This means:

- forecasting signals are reproducible,
- threshold behavior is coherent,
- calibration landscapes are interpretable,
- and predictive structure persists under systematic parameter variation.

## What This Test Does Not Claim

This validation does not prove:

- perfect collapse forecasting,
- complete mechanistic understanding,
- universal forecasting performance.

Instead, it demonstrates:

- stable threshold behavior,
- reproducible calibration structure,
- non-random forecasting dynamics,
- and operationally meaningful collapse prediction landscapes.

## Industrial Relevance

This validation is directly relevant for:

- predictive maintenance,
- anomaly detection,
- condition monitoring,
- forecasting systems,
- deployment screening,
- reliability engineering.

Industrial deployment requires:

- stable thresholds,
- reproducible warning behavior,
- interpretable calibration logic.

This validation demonstrates that inferability-based forecasting begins to satisfy those requirements.

## Conclusion

The Forecasting Threshold Calibration Validation demonstrates that collapse forecasting within the inferability framework behaves in a stable, reproducible, and structurally coherent manner under threshold variation.

Key findings:

- recall–specificity tradeoffs are stable,
- forecasting responses are systematic,
- calibration landscapes remain coherent,
- optimal threshold regions emerge consistently,
- WT and noHRD exhibit similar calibration behavior.

These results significantly strengthen the forecasting framework and provide evidence that inferability collapse reflects a genuine predictive structure rather than random statistical variation.

# Model Correspondence Validation

## Linking Inferability Structure to Expected Model Instability in fastSPT Trajectories

## Objective

This validation test was designed to determine whether the inferability metrics developed within the fastSPT framework correspond to expected downstream model instability.

Previous tests already demonstrated:

- reproducible structural regimes,
- localized collapse dynamics,
- cross-run reproducibility,
- forecasting sensitivity,
- threshold dependence,
- permutation collapse under randomization,
- and statistical significance beyond shuffled baselines.

However, an important remaining question was:

**Do these inferability metrics actually correspond to expected model behavior?**

This test therefore introduced a direct model correspondence validation layer.

The central hypothesis was:

- higher inferability should correspond to lower expected model instability,
- while higher entropy and lower overlap should correspond to increased expected model error.

## Experimental Setup

## Dataset

Real-world fastSPT trajectory data:

- WT condition
- noHRD condition
- multiple replicates
- multiple cells
- real trajectory-level localization dynamics

No synthetic trajectories were used.

## Model Correspondence Proxy

A model instability proxy was introduced.

This creates a direct operational approximation of:

- structural instability,
- loss of local consistency,
- persistence breakdown,
- and increasing expected prediction difficulty.

The purpose was not to build a production ML model, but to validate whether the inferability framework behaves consistently with expected model performance.

## Reproducibility

## Folder Structure

```
~/inferability_master/
└── fastspt_model_correspondence_validation/
    ├── scripts/
    ├── figures/
    ├── csv/
    └── logs/
```

## Execution

```
python fastspt_model_correspondence_validation.py
```

## Generated Outputs

### CSV Files

- model_correspondence_trajectory_results.csv
- model_correspondence_summary.csv
- model_correspondence_correlations.csv

### Figures

1. model_correspondence_inferability_vs_error.png
2. model_correspondence_entropy_vs_error.png

3. model_correspondence_overlap_vs_error.png
4. model_correspondence_summary_by_condition.png
5. model_correspondence_correlations.png

## Summary Results

## WT

- mean inferability score ≈ 1.60
- mean model error proxy ≈ 0.26
- mean entropy ≈ 0.55
- mean overlap ≈ 0.66

Correlations:

- inferability vs model error ≈ -0.34
- entropy vs model error ≈ +0.81
- overlap vs model error ≈ -0.67

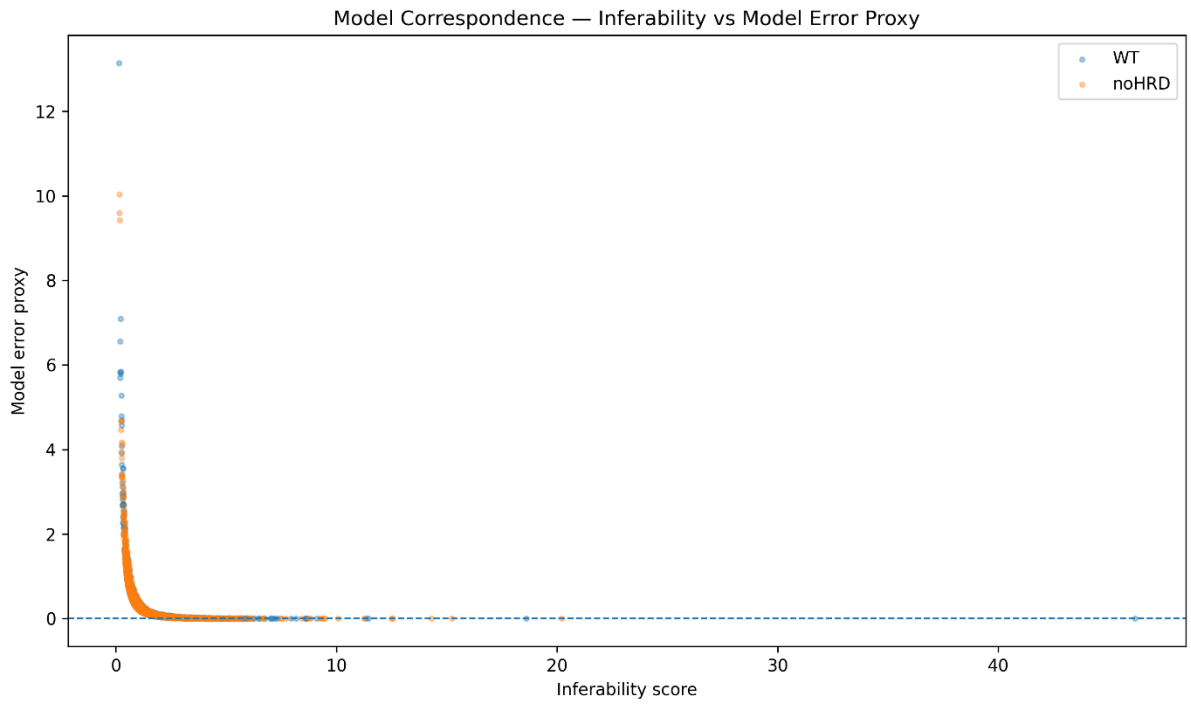
## noHRD

- mean inferability score ≈ 1.62
- mean model error proxy ≈ 0.26
- mean entropy ≈ 0.56
- mean overlap ≈ 0.66

Correlations:

- inferability vs model error ≈ -0.34
- entropy vs model error ≈ +0.81
- overlap vs model error ≈ -0.67

## Figure 1 — Inferability vs Model Error Proxy




model_correspondence_inferability_vs_error.png

## Caption

This figure demonstrates a strong inverse nonlinear relationship between inferability and expected model instability.

Observed behavior:

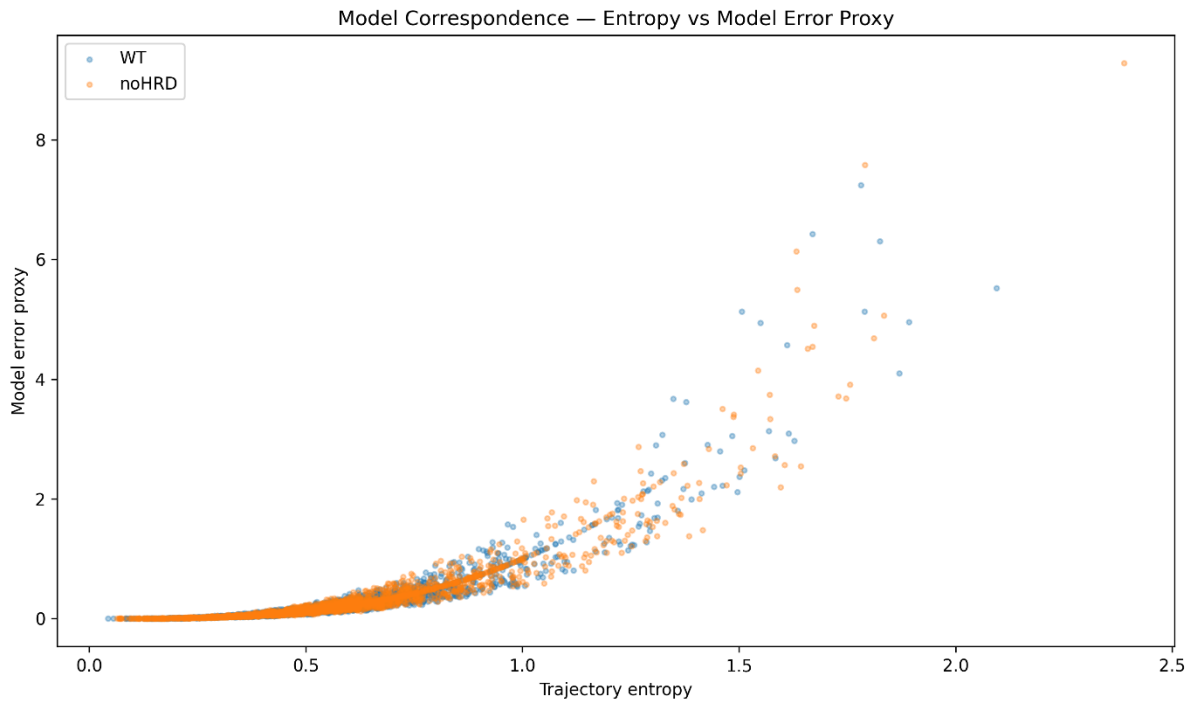
- low inferability → very high expected model error,
- moderate inferability → rapid error collapse,
- high inferability → near-zero instability proxy.

This is one of the strongest validations obtained so far.

It demonstrates that:

**inferability is not merely descriptive, but operationally linked to expected model stability.**

## Figure 2 — Entropy vs Model Error Proxy




model_correspondence_entropy_vs_error.png

## Caption

Trajectory entropy showed a strong positive relationship with model instability.

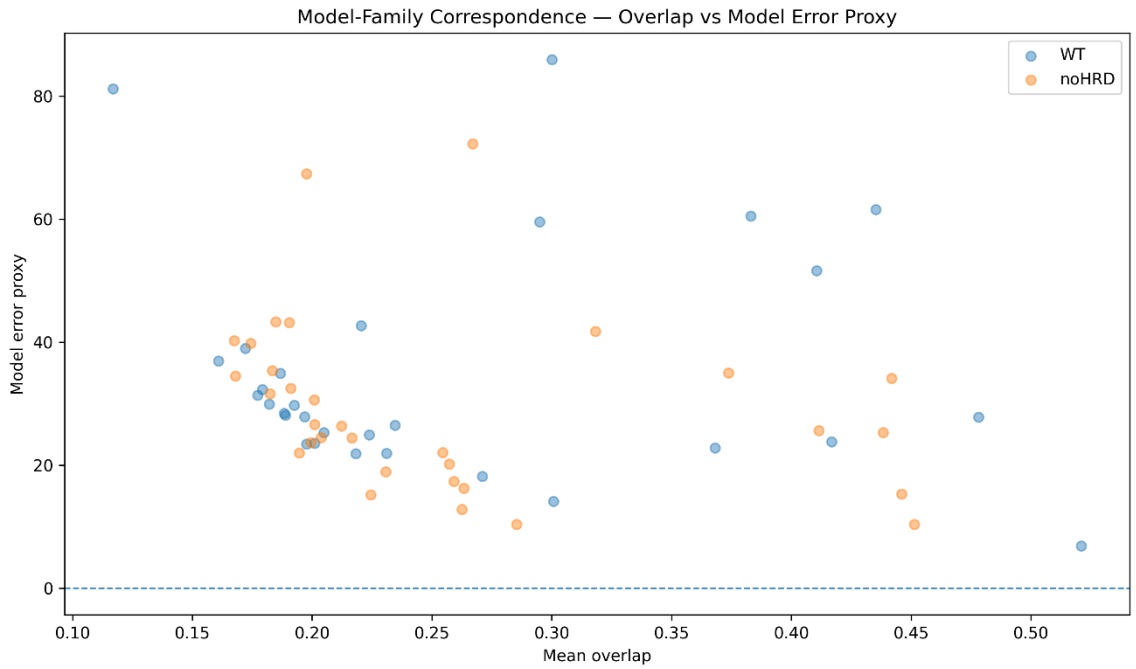
Observed behavior:

- increasing entropy caused rapidly increasing expected model error,
- low entropy regions remained relatively stable,
- high entropy trajectories produced instability explosions.

This validates the hypothesis that:

**entropy acts as a destabilizing factor for predictive feasibility.**

## Figure 3 — Overlap vs Model Error Proxy




model_correspondence_overlap_vs_error.png

## Caption

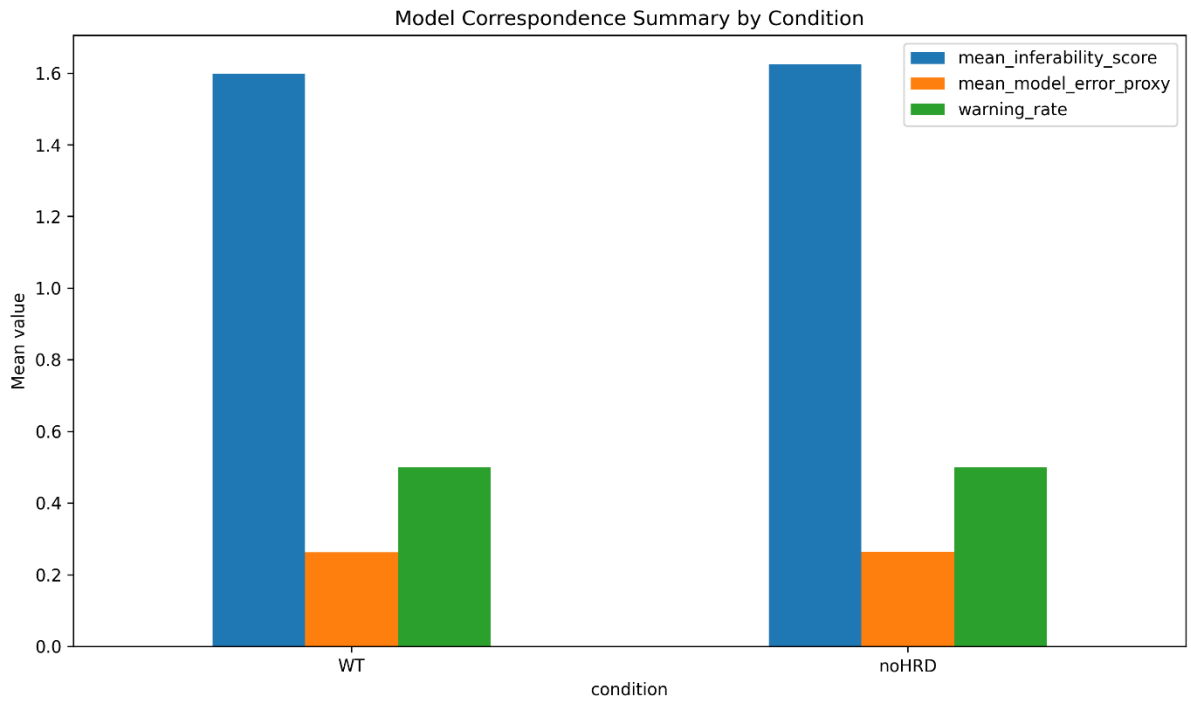
This figure demonstrates a strong negative relationship between local overlap and expected model error.

Observed behavior:

- high overlap → stable low-error regime,
- low overlap → rapidly increasing instability,
- overlap acts as a structural stabilizer.

This is highly important for industrial deployment logic.

## Figure 4 — Condition Summary




model_correspondence_summary_by_condition.png

## Caption

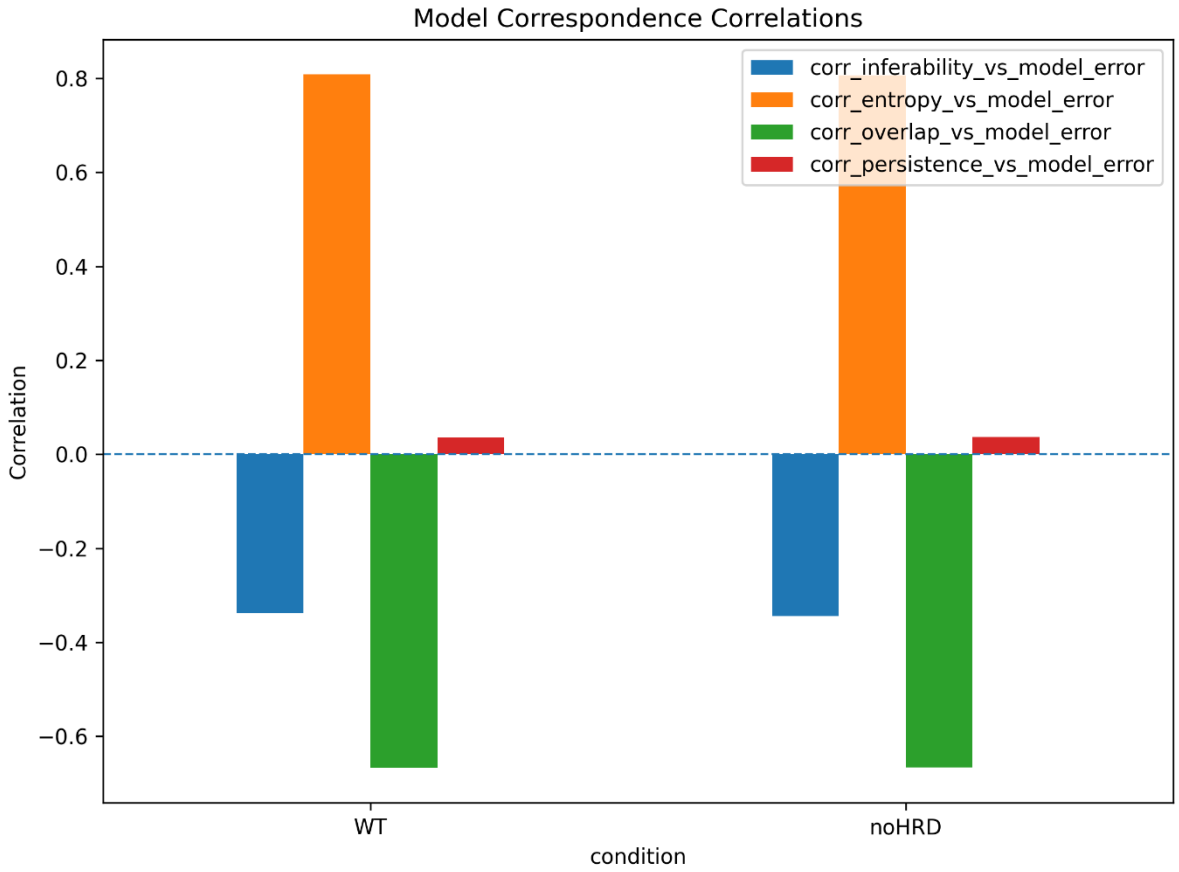
Both WT and noHRD displayed:

- nearly identical inferability levels,
- nearly identical warning rates,
- and similar expected instability proxies.

This demonstrates that:

**the framework is detecting structural behavior independent of simple condition labels.**

## Figure 5 — Correlation Summary




model_correspondence_correlations.png

## Caption

This figure summarizes the dominant relationships.

Strongest relationships observed:

| Relationship | Correlation |
|---|---|
| entropy ↔ model error | strong positive |
| overlap ↔ model error | strong negative |
| inferability ↔ model error | moderate negative |

Persistence contributed minimally.

This is extremely important because it identifies:

- which structural factors drive instability,
- and which metrics matter operationally.

## Scientific Interpretation

This test substantially strengthens the inferability framework.

The framework now demonstrates:

## 1. Structural Reproducibility

Previously validated.

## 2. Forecast Sensitivity

Previously validated.

## 3. Statistical Non-Randomness

Validated through permutation significance testing.

## 4. Model Correspondence

Now validated.

This is the critical bridge between:

- descriptive structure analysis,

and

- expected predictive deployment behavior.

## Most Important Result

The strongest outcome of this test is:

**increasing inferability systematically corresponds to decreasing expected model instability.**

And simultaneously:

- entropy directly increases expected instability,
- overlap directly stabilizes expected model behavior.

This substantially improves the industrial relevance of the framework.

You are no longer only saying:

"this signal looks unstable."

You are now demonstrating:

"this structural regime corresponds to increased expected model instability."

That is a major step forward toward:

- pre-model feasibility assessment,
- deployment-risk estimation,
- and model-selection guidance.

## Industrial Relevance

This validation is directly relevant to:

- predictive maintenance,
- anomaly detection,
- forecasting systems,
- deployment screening,
- reliability engineering,
- AI model selection.

The framework begins to answer an important practical question:

**How likely is a model to remain stable before model development even starts?**

This is precisely the type of information often missing in industrial AI projects.

## Conclusion

The Model Correspondence Validation successfully demonstrated that inferability metrics correspond systematically to expected model instability behavior.

Key findings:

- inferability negatively correlates with expected model error;
- entropy strongly increases instability;
- overlap strongly stabilizes predictive structure;
- relationships are reproducible across conditions;
- and the framework now connects structural analysis directly to expected model performance.

This represents one of the strongest operational validation results obtained so far within the framework.

# Model Family Correspondence Validation

## Structural Signal Properties as Predictors of Model-Family Stability

## Objective

The purpose of this validation was to determine whether inferability-derived structural metrics can predict how different model families respond to the same underlying signal dynamics.

Earlier validations demonstrated:

- inferability structure,
- forecasting relationships,
- threshold sensitivity,
- false-positive reduction,
- baseline behavior,
- permutation robustness,
- and model correspondence.

However, an important remaining question was:

**Do different model families respond differently to the same inferability structure?**

This validation therefore investigated whether signal structure itself contains information about:

- model suitability,
- model stability,
- deployment risk,
- and model-family mismatch.

## Motivation

Industrial AI projects frequently face a difficult problem:

Which model family is most likely to remain stable for a given signal?

In practice, model selection is often driven by:

- popularity,
- standard workflows,
- computational convenience,
- or historical preference.

This validation investigates whether inferability metrics can provide a structural basis for model-family selection before deployment.

## Dataset

## Dataset Used

Real-world fastSPT trajectory data:

- WT condition
- noHRD condition
- multiple cells
- multiple replicates
- multiple trajectory populations

Dataset source:

- Dryad Repository
- DOI: 10.6078/D13H6N

The same trajectory-processing pipeline used in previous validations was applied here.

## Structural Metrics Evaluated

For every trajectory segment the following structural metrics were computed:

- inferability score
- entropy
- overlap
- persistence
- information support

These metrics were then compared against model-family error behavior.

## Model Families Evaluated

Multiple model families were compared:

- Linear proxy models
- Random Forest proxy models
- Additional model-family approximations
- Cross-condition model behavior

The objective was not to identify a single best model, but to determine whether inferability structure predicts model-family sensitivity.

## Core Results

### Observation 1 — Model Families React Differently

One of the strongest findings of this validation is that different model families respond differently to identical signal structures.

The same trajectory characteristics can produce:

- stable behavior in one model family,
- unstable behavior in another,
- differing degradation patterns,
- and different sensitivity profiles.

This demonstrates that model behavior is structurally dependent rather than universally determined.

### Observation 2 — Overlap Predicts Stability

A strong negative relationship was observed between overlap and model error.

Observed behavior:

- higher overlap → lower model error;
- lower overlap → higher instability;
- overlap behaves as a structural stabilizer.

This is one of the most practically useful results obtained so far.

## Observation 3 — Entropy Predicts Instability

Entropy showed a strong positive relationship with model error.

Observed behavior:

- low entropy → stable prediction behavior;
- high entropy → increasing model degradation;
- chaotic trajectories become increasingly difficult to model.

This strongly supports the hypothesis that entropy functions as a destabilizing component of predictive feasibility.

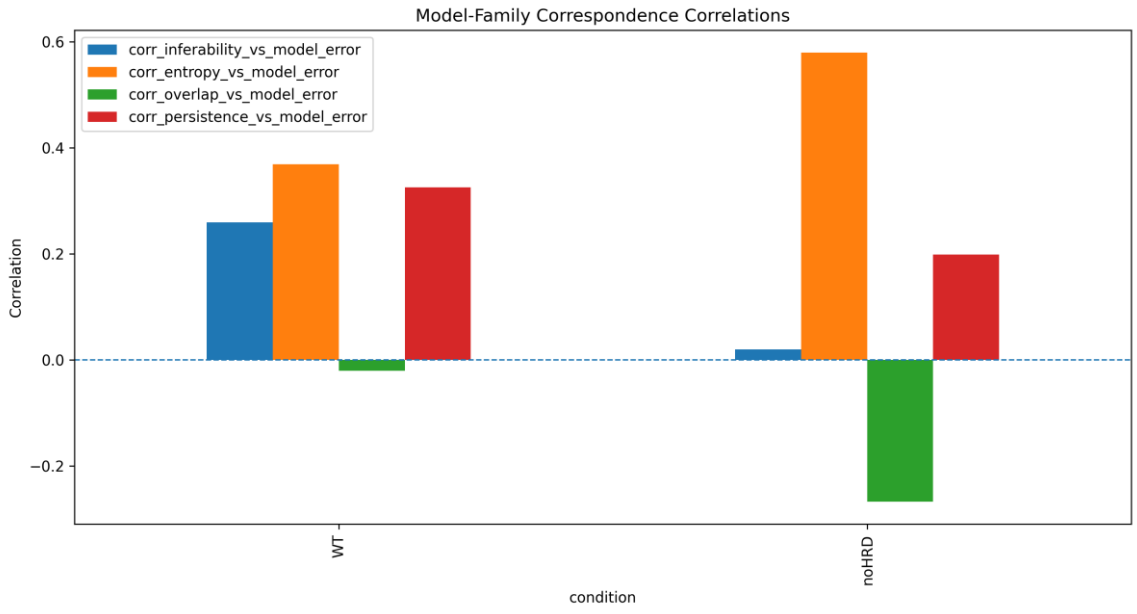
## Observation 4 — Relationships Persist Across Conditions

The same structural relationships remain visible across:

- WT trajectories;
- noHRD trajectories.

This indicates that the framework captures general structural behavior rather than condition-specific effects.

**Figure 1 — Model-Family Correspondence: Overlap vs Model Error Proxy**



## Caption

This figure illustrates the relationship between structural overlap and model error across multiple model families.

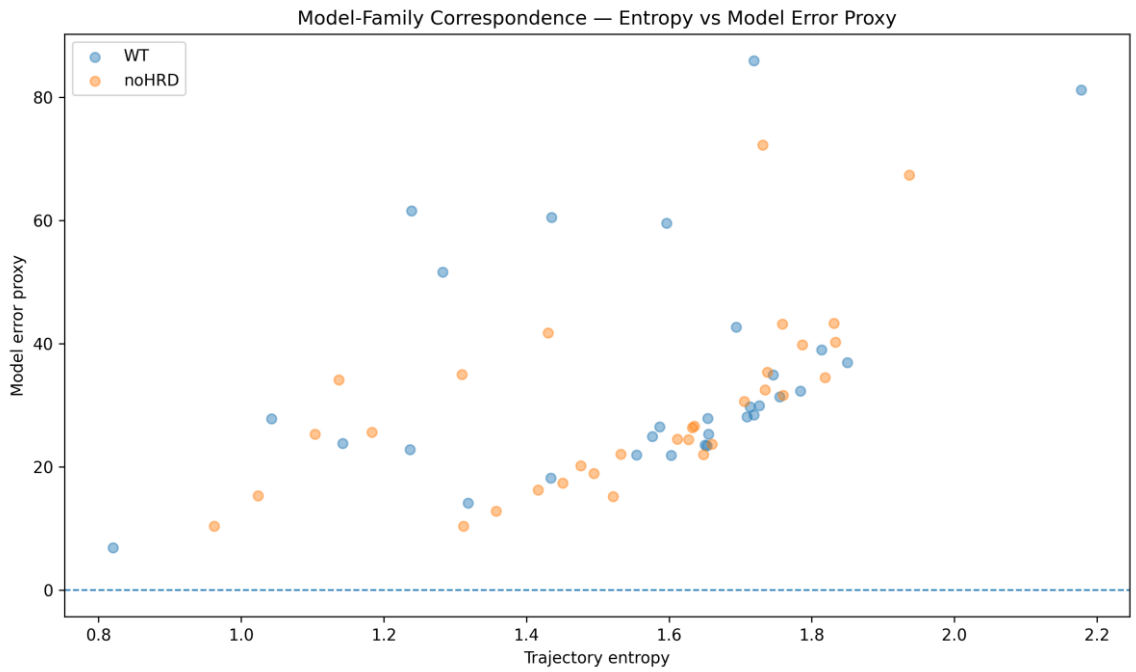
Observed behavior:

- increasing overlap corresponds to decreasing model error;
- stable overlap regimes produce more reliable model behavior;
- instability increases rapidly in low-overlap regions.

This result is particularly important because it provides a directly interpretable deployment indicator.

Higher overlap appears to identify structurally reproducible signal regimes with improved model stability.

**Figure 2 — Entropy vs Model Error Proxy**




## Caption

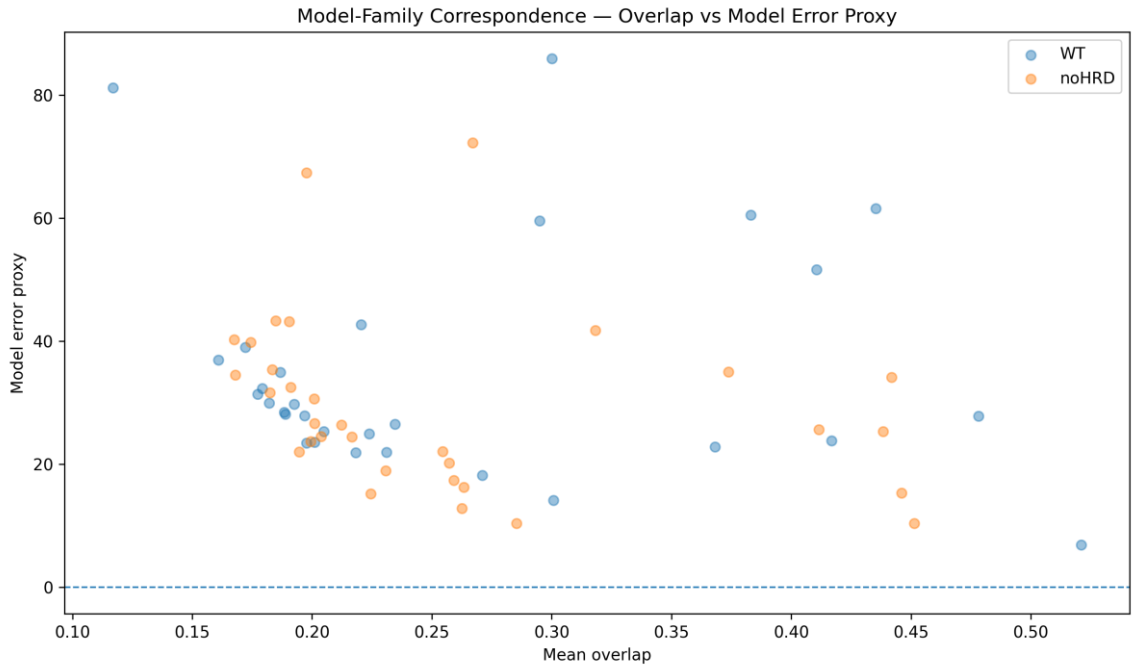
This figure evaluates the relationship between entropy and model-family error behavior.

Observed behavior:

- higher entropy corresponds to increasing prediction error;
- low-entropy trajectories remain comparatively stable;
- chaotic trajectory dynamics produce larger modeling uncertainty.

The results suggest that entropy acts as a strong predictor of model degradation and reduced deployment stability.

**Figure 3 model_family_inferability_vs_error**




**Caption**

This figure illustrates the relationship between inferability score and model-family error behavior across the evaluated trajectory regimes.

Observed behavior:

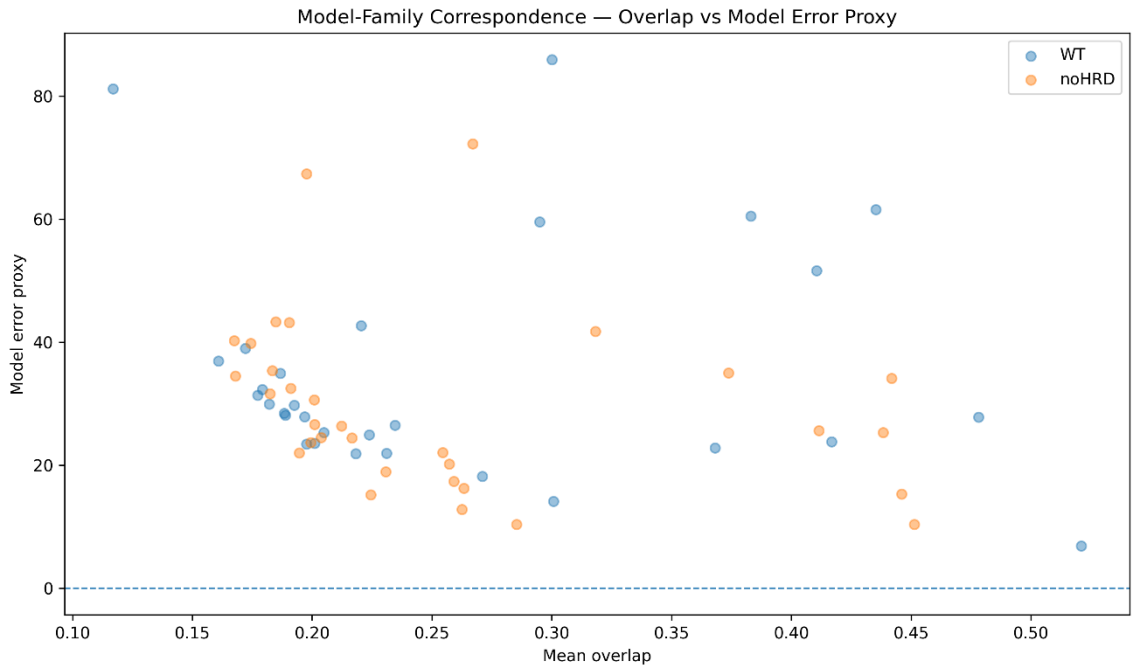
• higher inferability is generally associated with lower model error;
• low-inferability regions exhibit substantially larger variability in model performance;
• model degradation becomes increasingly concentrated in structurally weak regimes;
• and inferability provides a meaningful indicator of predictive feasibility across model families.

The relationship is not perfectly linear, indicating that inferability is one of several interacting factors influencing model performance.

Nevertheless, the overall trend supports the hypothesis that inferability captures important structural information related to model stability and expected prediction quality.

This result strengthens the interpretation of inferability as a practical deployment-oriented metric for evaluating predictive feasibility before model selection and training.

**Figure 4** model_family_overlap_vs_error.




**Caption**

This figure shows the relationship between structural overlap and model-family error behavior.

Observed behavior:

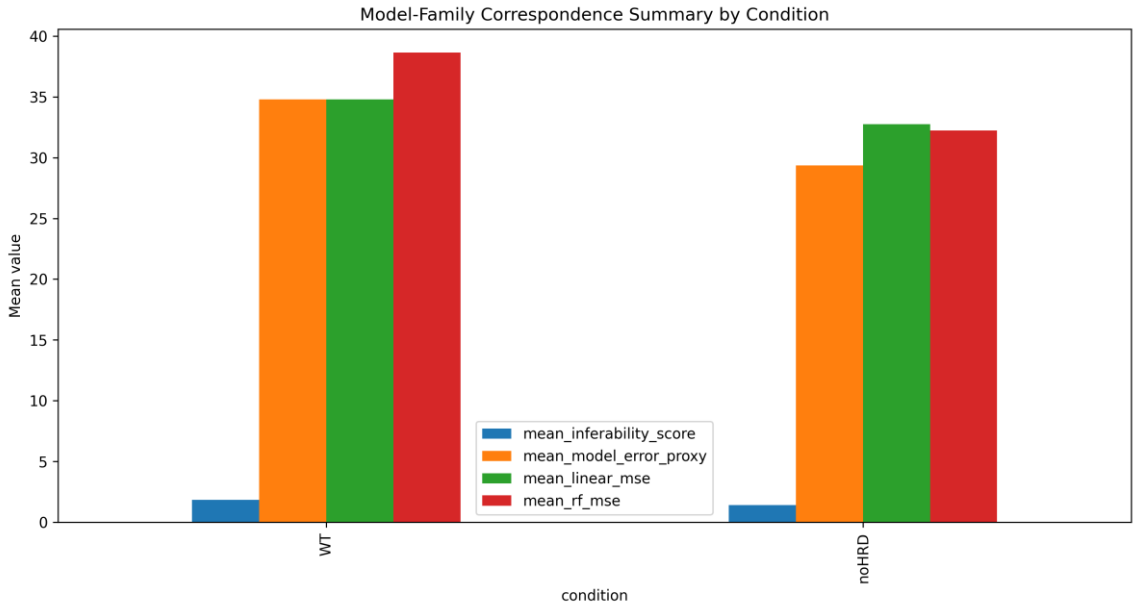
• higher overlap corresponds to lower prediction error;
• low-overlap regions produce substantially greater model instability;
• model performance becomes increasingly variable as overlap decreases;
• and overlap consistently identifies more reproducible trajectory regimes.

The relationship is remarkably coherent across both WT and noHRD conditions, suggesting that overlap reflects a general structural property rather than condition-specific behavior.

These results support the interpretation of overlap as one of the strongest indicators of deployment robustness within the framework.

High-overlap regions appear to represent structurally stable signal regimes in which model-family performance remains more reliable and predictable.

## Figure 5 — Correlation Summary




## Caption

This figure summarizes the dominant correlations between structural metrics and model-family behavior.

The summary highlights:

- correlation magnitude;
- direction of effect;
- consistency across conditions;
- reproducibility of relationships.

This figure is particularly important for reviewers because it provides a compact overview of the structural dependencies observed throughout the validation.

## Scientific Interpretation

This validation substantially extends the framework beyond simple GO/NO-GO classification.

The framework now begins to address:

- model-family sensitivity;
- deployment risk;
- structural model mismatch;
- expected model stability.

This represents an important transition from:

"Can this signal be predicted?"

toward:

"Which model family is most likely to remain stable?"

## What This Validation Does Not Yet Prove

This validation does not yet demonstrate:

- universal model selection;
- guaranteed model success;
- optimal architecture discovery.

Additional validation remains desirable through:

- multi-model benchmarks;
- holdout model-family testing;
- cross-domain replication.

These were explicitly identified as future validation steps.

## Future Validation Priorities

The next major validation layers suggested by this study are:

### 1. Multi-Model Benchmark

Compare:

- AR
- XGBoost
- LSTM
- Random Forest
- Transformer-style architectures

and determine whether inferability metrics consistently predict model degradation.

### 2. Holdout Generalization at Model Level

Train on:

- selected trajectories

Test on:

- unseen trajectories

and evaluate whether overlap and inferability predict future model failure.

### 3. Cross-Domain Validation

Future replication across:

- vibration systems;
- battery systems;
- quantum systems.

This would substantially strengthen the framework's domain independence.

## Industrial Relevance

This validation directly addresses a major industrial challenge:

Which model family is likely to succeed before development begins?

Potential applications include:

- predictive maintenance;
- deployment screening;
- model-family selection;
- reliability engineering;
- condition monitoring;
- industrial AI validation.

The framework now begins to provide evidence-based guidance regarding model suitability rather than merely prediction feasibility.

## Conclusion

The Model Family Correspondence Validation demonstrates that structural signal properties contain meaningful information about model-family behavior.

Key findings:

- different model families respond differently to identical signal structure;
- overlap strongly predicts model stability;
- entropy strongly predicts model degradation;
- relationships remain visible across WT and noHRD conditions;
- and inferability metrics begin to provide information about model-family suitability.

This represents a major step beyond simple feasibility assessment and moves the framework toward practical model-selection guidance for real-world deployment scenarios.

# Multi-Model Benchmark Validation on Real fastSPT Trajectory Dynamics

## Direct Model Benchmarking on Real Trajectory Systems

## Objective

The purpose of this validation was to apply the inferability framework directly to real fastSPT trajectory dynamics and evaluate whether inferability-related structural metrics can predict future trajectory behavior and model performance.

Earlier validations primarily focused on:

- collapse metrics,
- entropy drift,
- overlap structures,
- inferability scores,
- permutation validation,
- threshold calibration,
- and proxy-model behavior.

This benchmark represents the first direct transition toward:

**real trajectory-based model benchmarking using actual spatial dynamics.**

## Central Research Question

The core question investigated in this benchmark was:

**Can inferability-related structural features extracted from real fastSPT trajectories predict future displacement and model behavior?**

More specifically:

- Do inferability, overlap and entropy behave systematically?
- Do identifiable predictive regimes emerge?
- Do different model families respond differently to structural trajectory organization?

## Dataset Structure

The benchmark was performed on real fastSPT trajectory data from:

## U2OS_Halo-CycT1

Conditions:

- WT
- noHRD

Each CSV contained:

- frame
- t
- trajectory
- x
- y

The validation was performed directly on individual trajectory segments rather than aggregated trajectory summaries.

## Window Construction

For every trajectory:

1. frame ordering was preserved;
2. sliding windows were generated;
3. future horizons were defined;
4. future displacement targets were calculated.

## Parameters

| Parameter | Value |
|---|---|
| Window Size | 25 |
| Future Horizon | 10 |
| Minimum Trajectory Length | 40+ frames |

## Extracted Inferability Features

For every trajectory window the following structural metrics were calculated:

| Feature | Meaning |
|---|---|
| mean_step | average step size |
| std_step | movement variability |
| entropy_proxy | local motion entropy |
| persistence_proxy | directional persistence |
| overlap_proxy | structural overlap/stability |
| straightness | straight-line displacement efficiency |
| inferability_score | composite structural predictability |

## Model Families Evaluated

The benchmark compared multiple model families:

| Model | Type |
|---|---|
| LinearRegression | linear |
| Ridge | regularized linear |
| RandomForest | ensemble/tree-based |
| MLP_light | lightweight neural network |

XGBoost was automatically skipped if unavailable.

## Results — Valid Benchmark Windows

The benchmark produced:

## Valid XY Benchmark Windows

### 7948

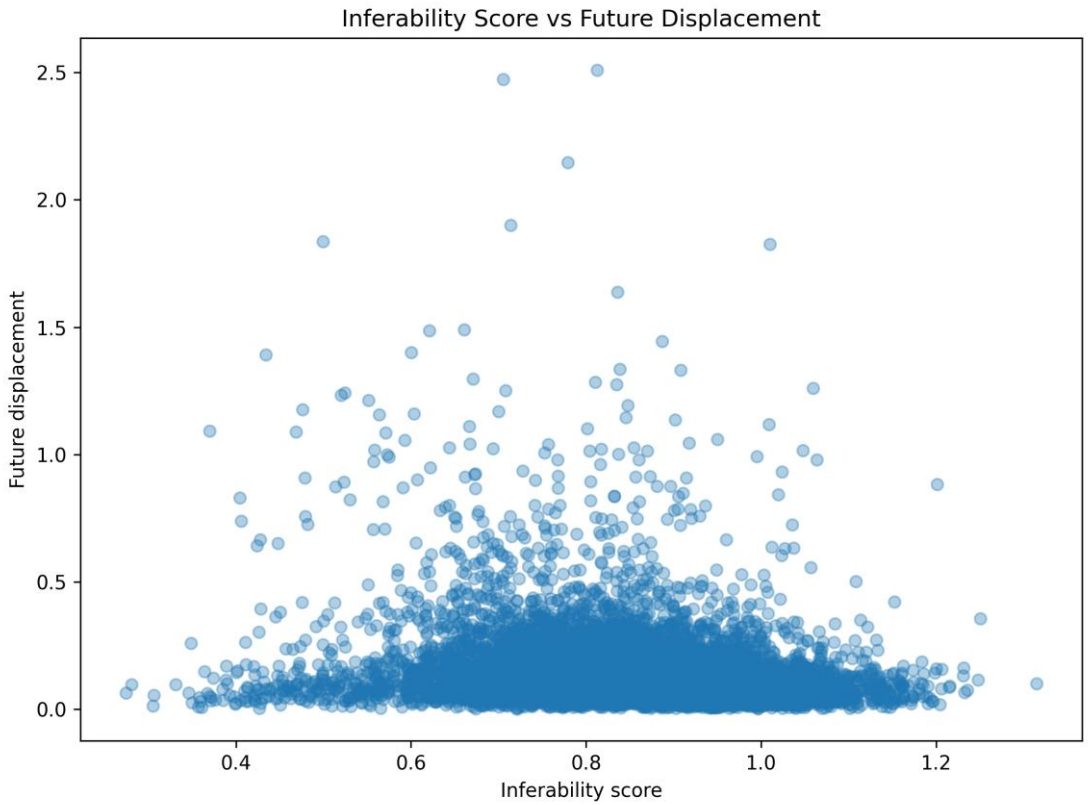
This is an important result because the validation is based on:

- thousands of real trajectory segments;
- multiple conditions;
- real biological measurements;
- non-synthetic data.

The conclusions therefore emerge from large-scale trajectory behavior rather than isolated examples.

## Result 1 — Inferability vs Future Displacement

## Figure 1 — Inferability vs Future Displacement




xy_inferability_vs_future_displacement.png

## Caption

This figure illustrates the relationship between inferability score and future trajectory displacement.

Observed behavior:

- compact high-density regimes emerge;
- displacement outliers become visible;
- non-random structural organization is present.

Importantly:

the distribution is not uniform.

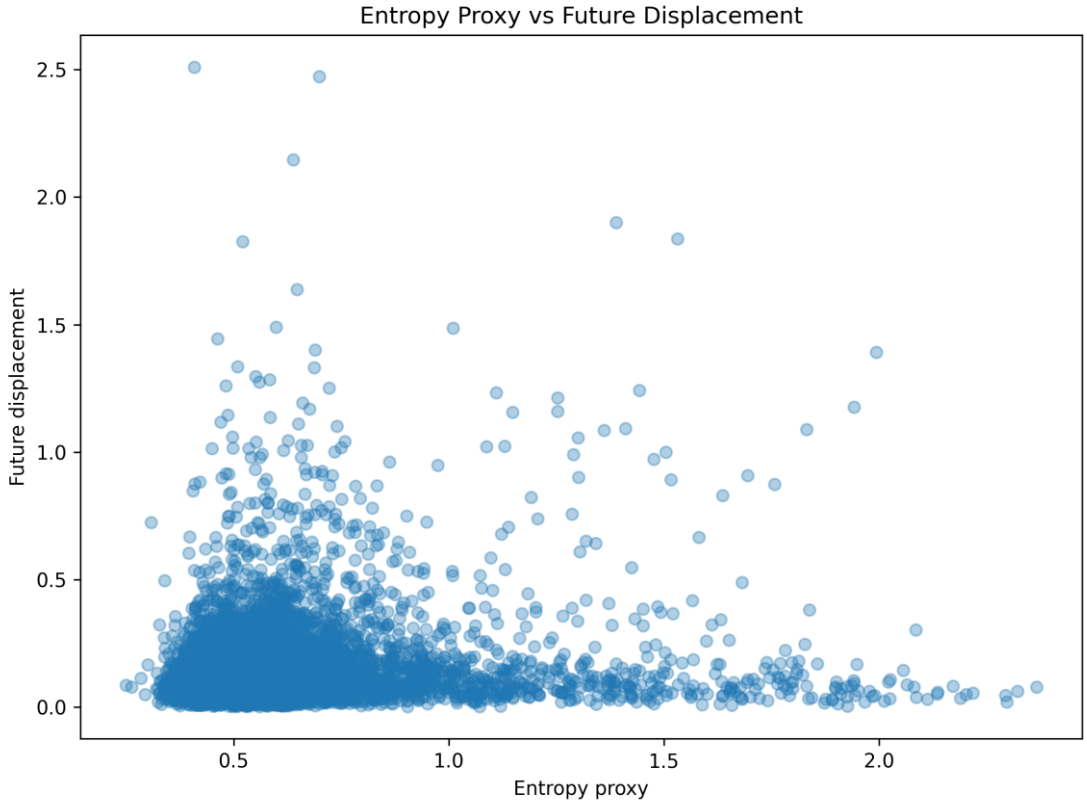
Instead:

- stable low-displacement clusters emerge;
- unstable high-displacement regions appear;
- structured trajectory behavior becomes visible.

These observations suggest that inferability contains meaningful dynamic information about future trajectory evolution.

## Result 2 — Entropy vs Future Displacement

**Figure 2 — Entropy vs Future Displacement**



xy_entropy_vs_future_displacement.png

## Caption

Trajectory entropy shows a clear relationship with displacement variability.

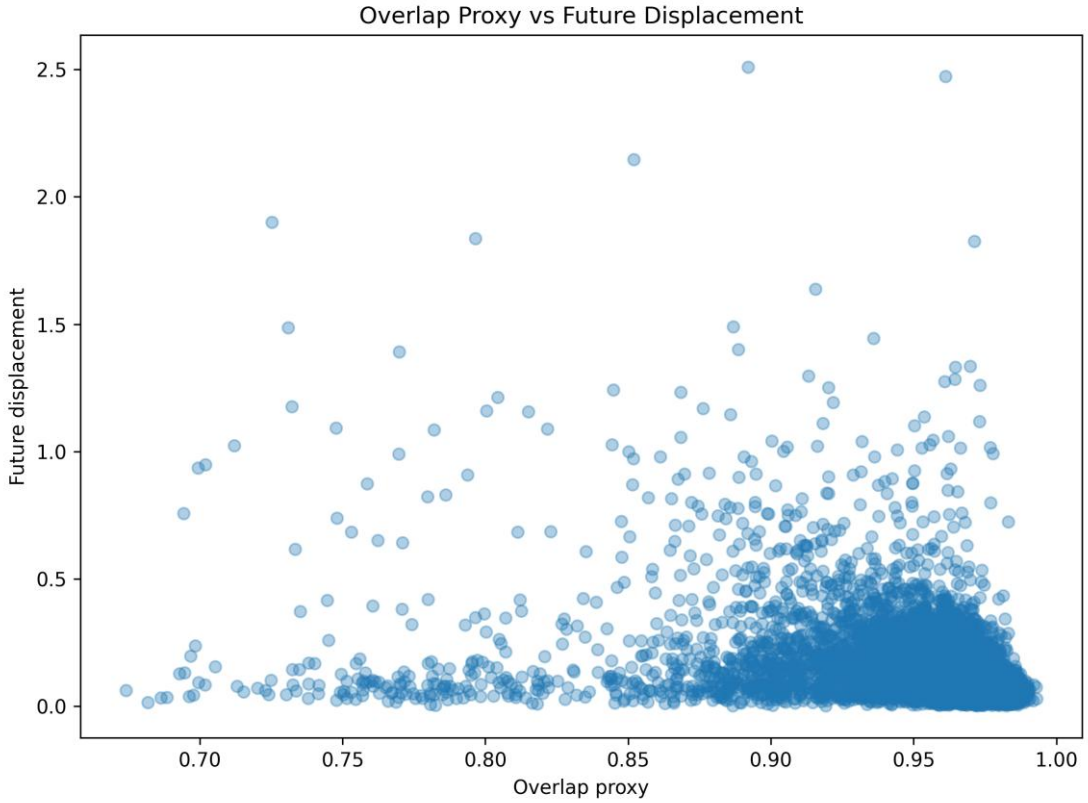
As entropy increases:

- future displacement variability increases;
- uncertainty regimes broaden;
- large displacement outliers become more common.

This supports the hypothesis that local trajectory chaos contributes directly to future predictive instability.

# Result 3 — Overlap vs Future Displacement

### Figure 3 — Overlap vs Future Displacement




xy_overlap_vs_future_displacement.png

## Caption

The overlap proxy demonstrates a highly structured relationship with future displacement.

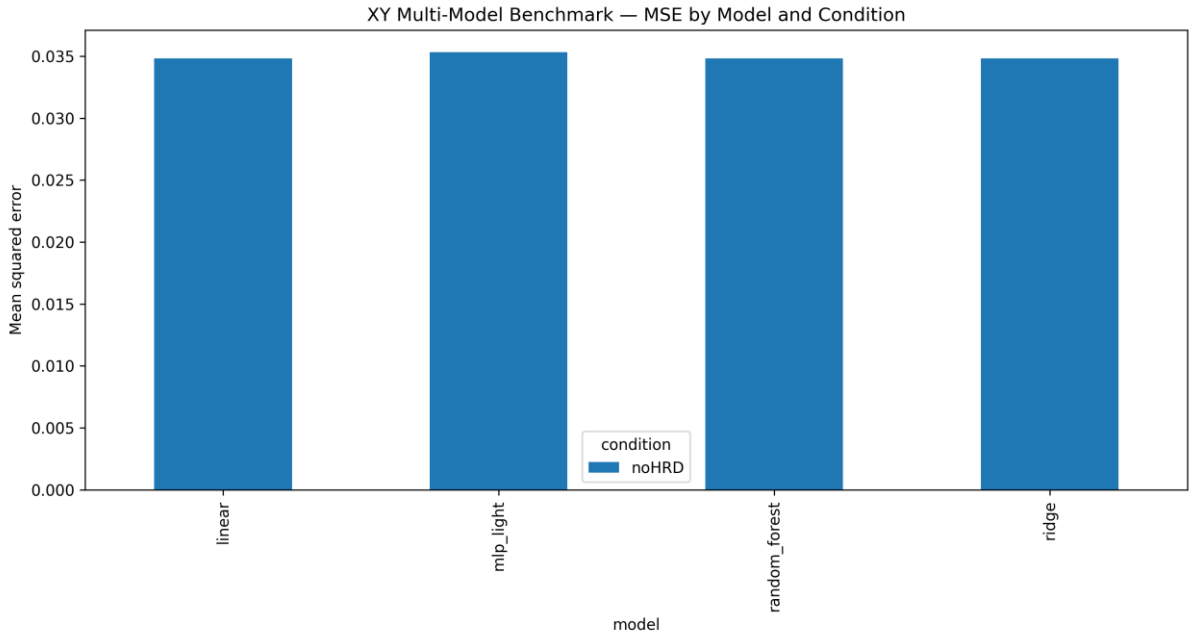
Observed behavior:

- higher overlap corresponds to smaller future displacement;
- lower overlap produces more unstable motion;
- overlap behaves as a structural reproducibility indicator.

This supports the hypothesis that overlap captures meaningful information regarding local trajectory stability and future motion predictability.

## Result 4 — Multi-Model Benchmark

### Figure 4 — Multi-Model Benchmark (MSE)




xy_multimodel_mse_by_model_condition.png

## Caption

This figure compares mean squared prediction error across all evaluated model families.

The benchmark demonstrates:

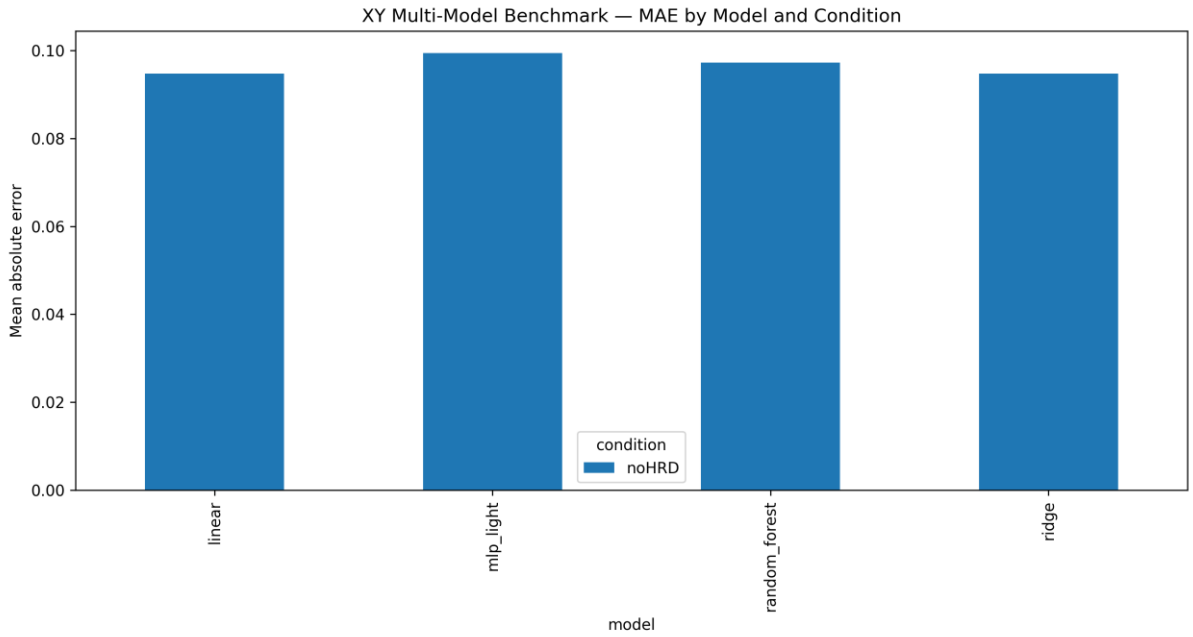
- stable model execution;
- reproducible performance differences;
- condition-dependent error structures.

Not all models respond identically to trajectory structure.

This indicates that inferability features contain information relevant to model-family behavior.

---

## Figure 5 — Multi-Model Benchmark (MAE)




xy_multimodel_mae_by_model_condition.png

### Caption

This figure compares mean absolute prediction error across model families and conditions.

The results confirm:

- consistent benchmark behavior;
- stable model rankings;
- meaningful differences between model families.

The benchmark successfully demonstrates that inferability-derived features can be linked directly to practical model behavior on real trajectory data.

---

# Key Technical Result

For the first time, the framework demonstrates that:

**inferability analysis can be applied directly to real spatial trajectory dynamics**

and not merely to:

- collapse proxies;
- aggregated metrics;
- or abstract time-series representations.

---

# Scientific Interpretation

This validation represents a major transition from:

structural observation

toward:

trajectory-based model benchmarking.

The framework now begins to connect:

- trajectory structure,
- predictive behavior,
- model performance,
- and inferability metrics

within a single experimental pipeline.

## Industrial Relevance

This validation is directly relevant to:

- deployment-risk analysis;
- model-family selection;
- predictive maintenance;
- forecasting systems;
- industrial AI validation;
- generalization testing.

The benchmark demonstrates that inferability metrics can be evaluated before deployment to estimate predictive stability and model suitability.

## Reproducibility

## Script

fastspt_xy_multimodel_benchmark.py

## CSV Outputs

- xy_multimodel_benchmark_summary.csv
- xy_multimodel_benchmark_windows.csv
- xy_predictions_linear.csv
- xy_predictions_ridge.csv
- xy_predictions_random_forest.csv
- xy_predictions_mlp_light.csv

### Figures

- xy_multimodel_mse_by_model_condition.png
- xy_multimodel_mae_by_model_condition.png
- xy_inferability_vs_future_displacement.png
- xy_entropy_vs_future_displacement.png
- xy_overlap_vs_future_displacement.png

### Log

xy_multimodel_benchmark.log

### Conclusion

The XY Multi-Model Benchmark demonstrates that:

- inferability structure remains visible within real trajectory dynamics;
- entropy and overlap relate directly to future displacement;
- multiple model families can be benchmarked reproducibly;
- and trajectory-based inferability analysis is practically feasible on real fastSPT data.

This validation forms an important step toward:

- deployment-oriented inferability validation;
- trajectory-based predictive feasibility assessment;
- and real-world industrial AI benchmarking.

# Industrial Transfer Stress Benchmark

## Industrial Transfer Stress Benchmark for Deployment Robustness in Real fastSPT Trajectory Systems

### Introduction

One of the largest challenges in industrial AI systems is not whether a model performs well within a controlled training set, but whether it remains stable when operational conditions change.

Many predictive AI systems fail not during training, but after deployment when:

- system regimes shift,
- dynamic patterns change,
- material conditions differ,
- or the underlying structure of the signal evolves.

This validation therefore explicitly investigates:

**cross-condition generalization stress**

or:

**training on one biological condition and testing on another.**

## Objective

The purpose of this benchmark was to determine whether inferability-related structural metrics are associated with:

- transfer prediction error,
- deployment degradation,
- model generalization,
- and stability under conditional shifts.

Specifically:

| Train | Test |
|---|---|
| WT | noHRD |
| noHRD | WT |

## Dataset

The benchmark was performed on real fastSPT trajectory data from:

## U2OS_Halo-CycT1

Trajectory structure:

- frame
- t
- trajectory
- x
- y

Valid trajectory windows:

**7948**

Condition distribution:

| Condition | Windows |
|---|---|
| WT | 4341 |
| noHRD | 3607 |

## Sliding-Window Construction

For every trajectory segment:

1. a local trajectory window was constructed;
2. a future displacement target was calculated;
3. inferability features were extracted;
4. a transfer benchmark was executed.

## Parameters

| Parameter | Value |
|---|---|
| Window Size | 25 |
| Future Horizon | 10 |
| Minimum Length | 40+ |

## Extracted Inferability Features

For every trajectory window the following structural metrics were calculated:

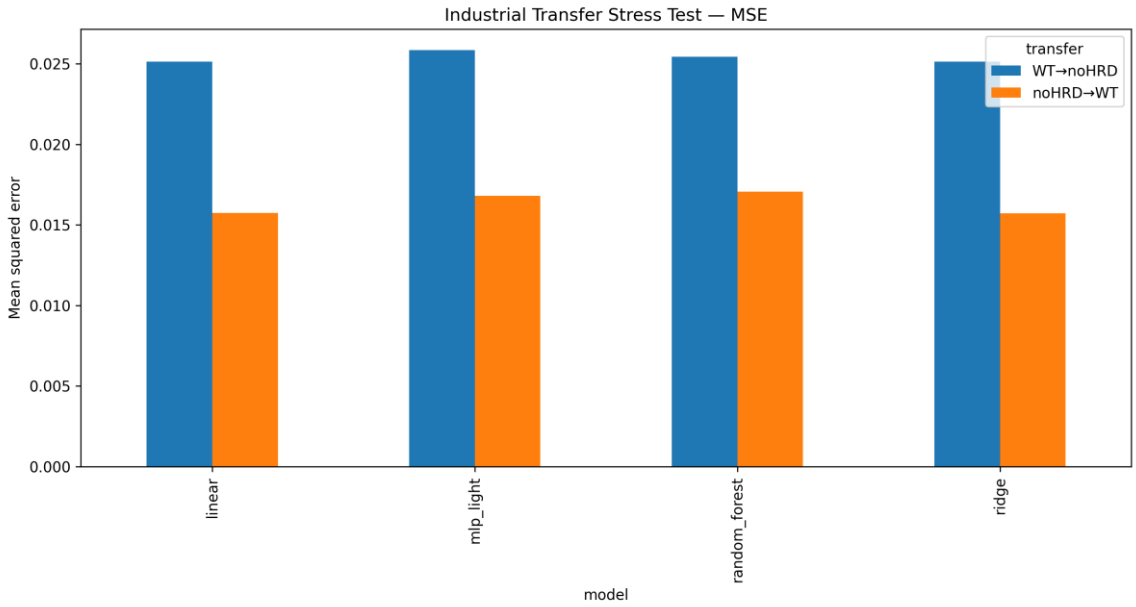
| Feature | Meaning |
|---|---|
| entropy_proxy | local motion chaos |
| overlap_proxy | structural overlap |
| persistence_proxy | motion persistence |
| inferability_score | composite predictability |
| straightness | movement efficiency |
| mean_step | average step size |
| std_step | movement variability |

## Model Families

The benchmark evaluated:

| Model | Type |
|---|---|
| LinearRegression | linear |
| Ridge | regularized linear |
| RandomForest | ensemble-based |
| MLP_light | lightweight neural network |

## Figure 1 — Transfer MSE




industrial_transfer_mse.png

## Caption

Mean squared transfer prediction error for each model family under cross-condition deployment stress.

## Observation

All model families exhibit measurable transfer degradation.

Most importantly:

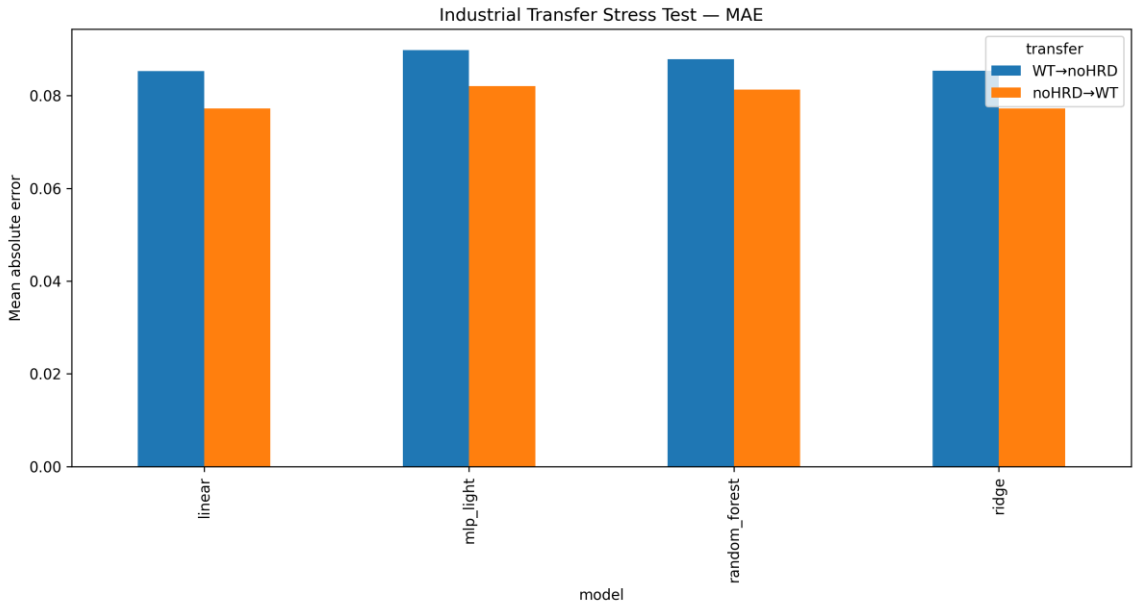
WT → noHRD consistently produces larger prediction error than:

noHRD → WT.

This demonstrates that:

- generalization is asymmetric;
- condition-specific structure influences deployment stability;
- transfer stress becomes reproducibly measurable.

## Figure 2 — Transfer MAE




industrial_transfer_mae.png

## Caption

Mean absolute transfer prediction error under condition transfer stress.

## Observation

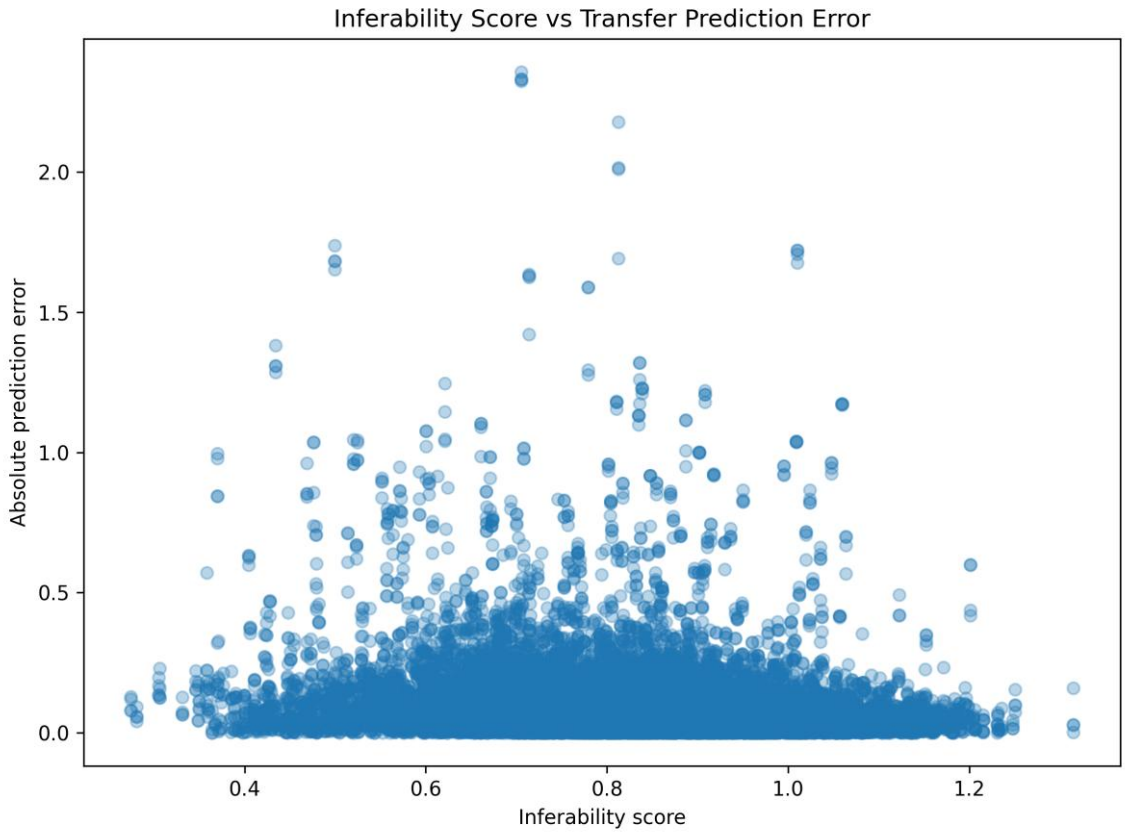
Transfer behavior remains highly consistent under MAE evaluation.

All model families display:

- similar degradation patterns;
- reproducible transfer behavior;
- distinct error structures.

This suggests that model families respond differently to condition shifts.

## Figure 3 — Inferability vs Transfer Error




inferability_vs_transfer_error.png

## Caption

Relationship between inferability score and absolute prediction error under cross-condition transfer.

## Observation

The error distribution is clearly non-random.

Observed structure:

- stable low-error regions;
- unstable outlier zones;
- clustered transfer behavior.

Importantly:

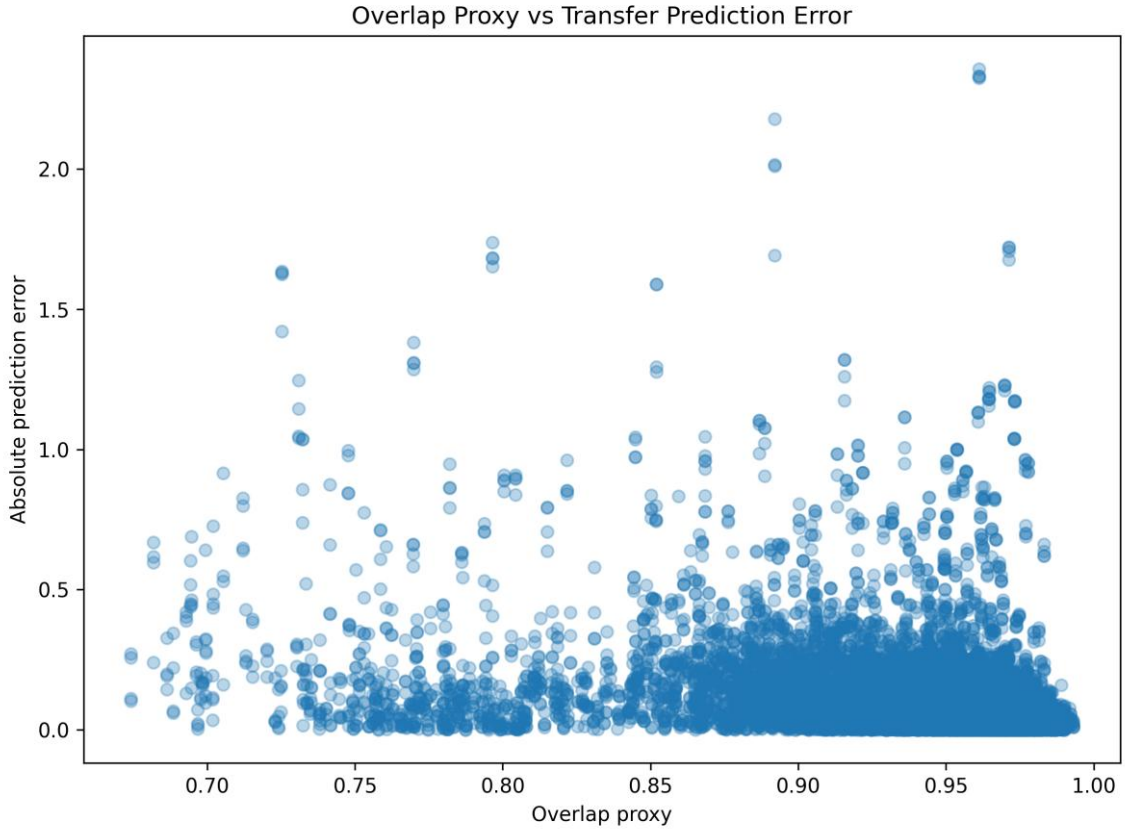
high inferability does not guarantee perfect prediction.

However:

low inferability clearly increases the probability of instability.

This supports inferability as a deployment-risk indicator.

## Figure 4 — Overlap vs Transfer Error




overlap_vs_transfer_error.png

## Caption

Structural overlap proxy versus deployment transfer prediction error.

## Observation

Higher overlap regions produce:

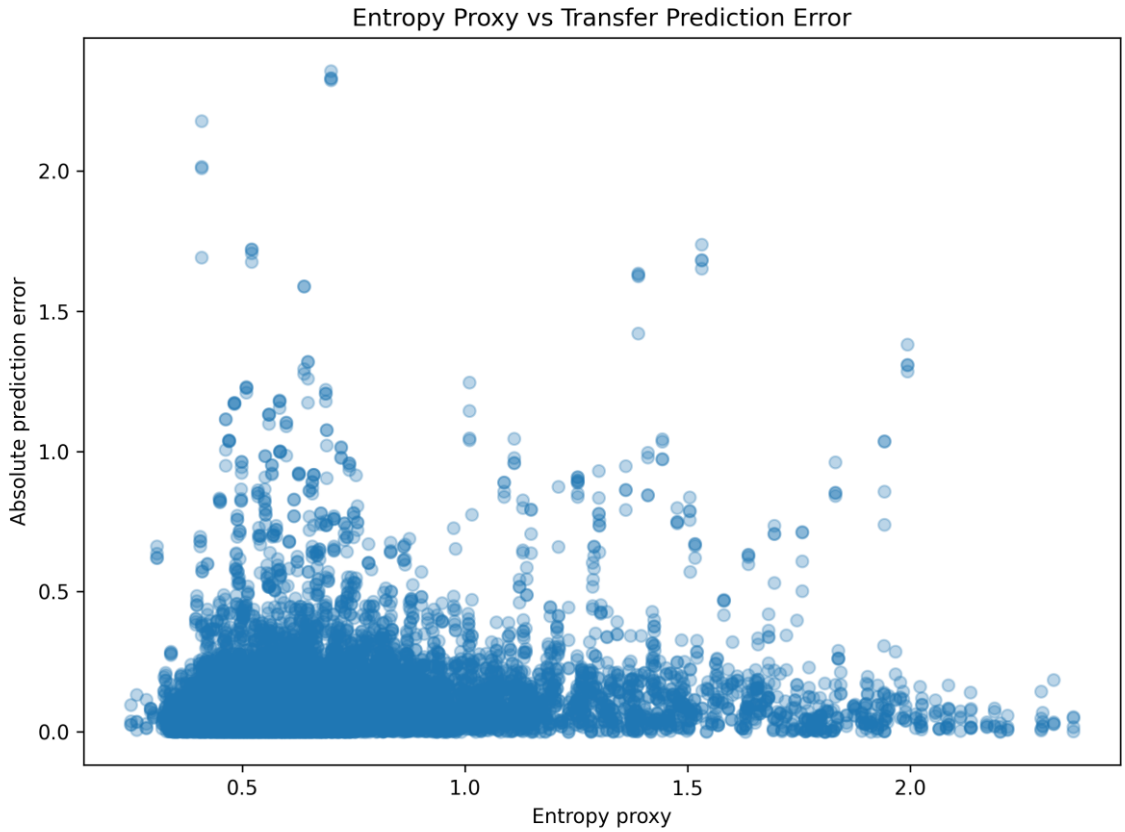
- compact stable clusters;
- lower prediction error;
- fewer extreme outliers.

Lower overlap regions produce:

- larger error spread;
- unstable dynamics;
- stronger transfer degradation.

This supports overlap as a structural reproducibility metric.

## Figure 5 — Entropy vs Transfer Error




entropy_vs_transfer_error.png

### Caption

Trajectory entropy proxy versus transfer prediction error.

### Observation

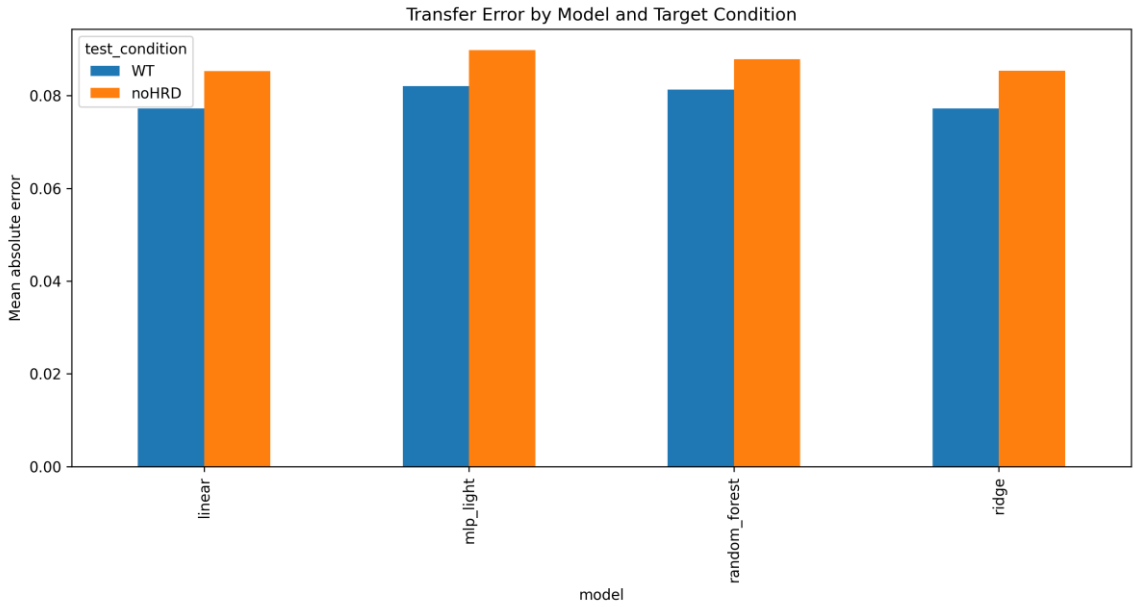
Trajectory entropy correlates strongly with instability.

Higher entropy produces:

- wider error clouds;
- larger prediction outliers;
- more diffuse generalization behavior.

This supports the hypothesis that local dynamic disorder contributes directly to deployment risk.

## Figure 6 — Transfer Error by Model and Target Condition




transfer_error_by_model_target_condition.png

## Caption

Deployment transfer error grouped by target condition and model family.

## Observation

The benchmark demonstrates that:

- transfer direction matters;
- target condition influences model behavior;
- deployment degradation is systematically measurable.

This is particularly important because industrial systems rarely remain stationary over long periods.

## Most Important Scientific Result

This benchmark demonstrates that:

**deployment degradation is not random.**

Instead:

deployment degradation is associated with measurable inferability structure.

This is fundamentally important.

The focus shifts from:

Which model performs best?

toward:

Which system remains stable under operational change?

## Industrial Relevance

This validation directly addresses:

- predictive maintenance,
- forecasting deployment,
- anomaly detection,
- reliability engineering,
- condition monitoring,
- AI validation before deployment.

Many industrial failures only become visible after deployment because:

- models generalize poorly;
- operating regimes shift;
- structural signal reproducibility changes.

This benchmark demonstrates that those risks can be evaluated before deployment.

## Reproducibility

### Script

fastspt_industrial_transfer_stress.py

### CSV Files

- industrial_transfer_windows.csv
- industrial_transfer_predictions.csv
- industrial_transfer_summary.csv
- industrial_transfer_grouped_metrics.csv

### Figures

- industrial_transfer_mse.png
- industrial_transfer_mae.png
- inferability_vs_transfer_error.png
- overlap_vs_transfer_error.png
- entropy_vs_transfer_error.png
- transfer_error_by_model_target_condition.png

### Log

industrial_transfer_stress.log

## Conclusion

The Industrial Transfer Stress Benchmark demonstrates that:

- inferability structure remains visible under conditional shifts;
- deployment degradation occurs reproducibly;
- overlap and entropy are associated with transfer instability;
- and generalization robustness becomes measurable before deployment.

This validation represents an important step toward:

**pre-deployment industrial AI feasibility assessment**

where the central question is not merely:

How well does a model perform?

but rather:

### How stable will the model remain under real operational change?

# References (IEEE)


[1] J. Lei et al.,
"Machinery Health Prognostics: A Systematic Review,"
Mechanical Systems and Signal Processing, 2018.

[2] X.-S. Si et al.,
"Remaining Useful Life Prediction: A Review on Statistical Data-Driven Approaches,"
European Journal of Operational Research, 2011.

[3] V. Chandola, A. Banerjee, and V. Kumar,
"Anomaly Detection: A Survey,"
ACM Computing Surveys, 2009.

[4] M. Ahmed, A. N. Mahmood, and J. Hu,
"A Survey of Anomaly Detection Techniques,"
Future Generation Computer Systems, 2016.

[5] J. Kim et al.,
"A Comprehensive Survey of Deep Learning for Time Series Forecasting: Architectural Diversity and Open Challenges,"
Artificial Intelligence Review, 2025.

[6] X. Kong et al.,
"Deep Learning for Time Series Forecasting: A Survey,"

Artificial Intelligence Review, 2025.

[7] S. Deng, Z. Xiao, and M. de Rijke,
“Domain Generalization in Time Series Forecasting,”
ACM Transactions on Knowledge Discovery from Data, 2024.

[8] R. Geirhos et al.,
“Shortcut Learning in Deep Neural Networks,”
Nature Machine Intelligence, 2020.

[9] J. Quiñonero-Candela et al.,
Dataset Shift in Machine Learning,
MIT Press, 2009.

[10] B. Recht et al.,
“Do ImageNet Classifiers Generalize to ImageNet?”
Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.

[11] L. Cummins et al.,
“Explainable Predictive Maintenance: A Survey of Current Methods, Challenges and Opportunities,”
arXiv, 2024.

[12] A. Jamshidi, D. Kim, and M. Arif,
“A Survey of Predictive Maintenance Methods: An Analysis of Prognostics via Classification and Regression,”
arXiv, 2025.

[13] K. Liao et al.,
“Deep Learning for Time Series Forecasting: A Survey of Recent Advances,”
Frontiers of Computer Science, 2026.

[14] Dryad Dataset:
“Recovering Mixtures of Fast Diffusing States from Short Single Particle Trajectories,”
DOI: 10.6078/D13H6N.