Auditing Operating-Condition Normalization in C-MAPSS Remaining Useful Life Evaluation

Aaditya Gupta
University of Illinois Urbana-Champaign, Urbana, IL, USA
ag158@illinois.edu

Abstract

NASA C-MAPSS is a canonical benchmark for turbofan remaining useful life (RUL) prediction, yet conclusions can depend on preprocessing and evaluation protocol in addition to model architecture. We present a protocol and reproducibility audit using compact MLP, temporal CNN, and LSTM baselines as controlled probes. A leakage-safe factorial grid varies the C-MAPSS subset, RUL cap, window length, normalization policy, architecture, and random seed, while enforcing and recording a fixed train/validation/final preprocessing fit policy. All 1080 planned canonical-grid runs completed, with no engine-split overlap or non-finite metrics. On single-regime FD001/FD003, global and per-condition normalization are intentionally equivalent in the harness and are used only as an implementation unit test; on multi-regime FD002/FD004, per-condition normalization lowers RMSE in every matched cell, with mean signed RMSE deltas of -4.39 and -7.40, respectively. A focused 720-run FD002/FD004 ablation further shows that jointly refitting the final model and normalizer on train+validation engines changes reported RMSE by -1.119 to -0.206 on average relative to train-only final fitting, depending on subset and normalization. Across the full grid, subset identity explains 60.8% of cell-mean RMSE variation; within multi-regime subsets, RUL cap and normalization are descriptively large among the evaluated protocol factors, with a seed-balanced sensitivity giving the same qualitative conclusion. These results extend prior PHM studies of operating-condition-specific standardization by quantifying matched protocol effects under a frozen audit contract and by making split unit, fit scope, regime assignment, score convention, and reproducibility records explicit.

1 Introduction

NASA C-MAPSS has become a widely used public benchmark for data-driven prognostics and predictive maintenance. The benchmark is based on a simulated turbofan engine environment and provides run-to-failure trajectories for training and truncated trajectories for testing Frederick et al. (2007); Saxena et al. (2008b); Saxena and Goebel (2008b). Its four common subsets, FD001–FD004, differ in operating-regime multiplicity and fault-mode complexity. This structure makes C-MAPSS useful for PHM research, but it also means that preprocessing choices can affect the reliability and comparability of RUL evaluation. C-MAPSS has supported recurrent neural networks, convolutional networks, hybrid deep architectures, and other deep sequence variants Heimes (2008); Babu et al. (2016); Zheng et al. (2017); Li et al. (2018); Vollert and Theissler (2021).

Despite this popularity, C-MAPSS is not only a model benchmark. It is also a protocol benchmark. Reported results depend on choices outside the model architecture itself: sensor selection, target capping, sliding-window length, normalization policy, operating-regime handling, train/validation splitting, early stopping, final test preprocessing, and metric implementation Ramasso and Saxena (2014); Vollert and Theissler (2021). These choices are not incidental for PHM practice. In RUL

prediction, preprocessing can alter both the input distribution and the effective learning target, while the NASA asymmetric score can amplify a small number of late predictions Saxena et al. (2008a). Thus, two studies can appear to differ in prognostic modeling capability when they actually differ in leakage safety, fit scope, or operating-condition handling.

This paper studies that protocol sensitivity directly. We do not propose a new RUL architecture, and we do not claim state-of-the-art performance. Instead, we ask a narrower evaluation question: under a leakage-safe and fully recorded C-MAPSS protocol, how large is the operating-condition normalization effect relative to architecture, RUL cap, window length, subset identity, and seed variation? This question is especially important because FD001 and FD003 contain a single operating regime, whereas FD002 and FD004 contain multiple operating regimes. Prior PHM work has already shown that operating-condition-specific standardization can improve multi-regime C-MAPSS results and that it reduces to global standardization when there is only one condition Pasa et al. (2019). Our purpose is therefore not to rediscover operating-condition normalization, but to audit this known preprocessing choice under a recorded leakage-safe fit policy, engine-level splitting, matched seeds, target caps, and score reporting. Related multi-condition RUL work has also modeled operational-condition variation directly Huang et al. (2019).

Our contributions are threefold. First, we specify and record a leakage-safe C-MAPSS evaluation protocol in which preprocessing is fit only on the allowed training engines and final test statistics are never used. Second, following Pasa et al.'s single-condition reduction, we use FD001/FD003 as an implementation-equivalence unit test: matched global and per-condition runs must produce identical outputs because only one operating-regime key exists. Third, we quantify signed normalization effects relative to architecture and other protocol factors over a 1080-run factorial grid. FD001 and FD003 are exact intended ties under matched seeds, whereas per-condition normalization lowers RMSE in every matched FD002 and FD004 cell, with mean signed RMSE deltas of -4.39 and -7.40. An independent FD002 seed-0 rerun of six canonical cells reproduces every metric and prediction exactly.

2 Related Work

2.1 NASA C-MAPSS and RUL Evaluation

The C-MAPSS simulator and associated turbofan degradation datasets were introduced through NASA technical documentation and the NASA Prognostics Data Repository Frederick et al. (2007); Saxena and Goebel (2008b); Saxena et al. (2008b). The benchmark is commonly used as a supervised RUL prediction task: training trajectories run to failure, test trajectories are truncated, and a model predicts RUL for each test engine. The PHM08 challenge and associated prognostics metrics further shaped evaluation practice, including asymmetric scoring that penalizes late predictions more heavily than early predictions Saxena and Goebel (2008a); Saxena et al. (2008a).

2.2 Deep RUL Models and Preprocessing

C-MAPSS has supported a long sequence of model-development papers. Early recurrent approaches established the feasibility of sequence modeling for prognostics Heimes (2008). CNN-based approaches framed RUL estimation as regression over sliding sensor windows and helped popularize deep temporal feature extraction for this benchmark Babu et al. (2016); Li et al. (2018). LSTM-based work further established recurrent neural networks as strong baselines for RUL sequence modeling Zheng et al. (2017).

C-MAPSS preprocessing conventions vary across studies. Common choices include dropping

near-constant sensors, applying piecewise-linear RUL caps, converting trajectories to fixed-length sliding windows, scaling sensors globally or by operating condition, and using different validation-split practices and reporting conventions Ramasso and Saxena (2014); Vollert and Theissler (2021). Operating-condition treatment has been part of C-MAPSS prognostics since early PHM challenge work: Peel used neural-network ensembles on the PHM08 turbofan data, while Wang et al. developed a similarity-based RUL method and explicitly connected RUL estimation to operating-regime handling Peel (2008); Wang et al. (2008); Wang (2010). Lim and Mba used switching Kalman filters for condition monitoring and RUL prediction, and Li et al. proposed operating-condition-classified prognostics, reinforcing that regime-aware prognostics predate recent deep-learning pipelines Lim and Mba (2014); Li et al. (2014). Pasa et al. later evaluated feedforward, convolutional, and LSTM networks over a large C-MAPSS configuration set and directly studied operating-condition-specific standardization; their study reported that this preprocessing is equivalent to global standardization on single-condition subsets and can materially improve results on multi-condition subsets Pasa et al. (2019). Ragab et al. similarly apply working-condition-specific Min-Max normalization because the same sensors can exhibit different readings under different operating conditions Ragab et al. (2021). Against this lineage, the present study is positioned as a reproducibility and protocol audit: it does not claim novelty for the idea of operating-condition normalization, but instead freezes the normalization fit scope, split unit, target cap, regime assignment, matched-seed comparison, and variance decomposition so that the contribution is a transparent account of how large the protocol factor is under controlled leakage-safe conditions.

Table 1 makes this positioning explicit. The closest prior work is Pasa et al., which already studies operating-condition-specific standardization with neural models on C-MAPSS Pasa et al. (2019). The difference is that the present paper treats normalization as one audited evaluation factor in a matched-cell protocol rather than as one design choice inside a broad hyperparameter search or model-comparison study. This distinction matters for reproducibility: the audit asks which preprocessing statistics are fit on which engines, whether validation engines can influence fitted statistics, whether official test records are ever used for preprocessing, whether paired deltas use identical seeds and protocol cells, and which fields should be reported so that another PHM study can reproduce the comparison. Recent C-MAPSS model papers continue to report strong architectures and protocol-dependent benchmark values, including semi-supervised deep architectures and interaction models Ellefsen et al. (2019); Zhevnenko et al. (2023). Those works are not direct normalization audits, but they illustrate why protocol records are needed when results are compared across model families.

3 Methods

3.1 Dataset and Prediction Task

We evaluate NASA C-MAPSS FD001–FD004. Each subset contains multivariate sensor trajectories for multiple engines. Training engines are observed until failure; official test engines are truncated and associated with held-out RUL labels. The task is sequence-to-one regression: for each official test engine, predict RUL using the final observed window. The study uses the standard 14-sensor set $\{2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, 21\}$, corresponding to columns `s2`, `s3`, `s4`, `s7`, `s8`, `s9`, `s11`, `s12`, `s13`, `s14`, `s15`, `s17`, `s20`, and `s21`. Operating settings are retained for regime assignment but are not included as input features in the main grid.

3.2 Protocol Grid

Table 2 summarizes the controlled factors. The full V3 design contains 1080 runs. Global-normalization cells use 10 seeds, while per-condition cells use 5 seeds. Paired normalization deltas use matched seeds 0–4 for both normalization policies.

3.3 Protocol Audit Design

The study is designed as a protocol audit rather than as a leaderboard comparison. The compact MLP, temporal CNN, and LSTM are used as controlled diagnostic probes because they represent common feedforward, convolutional, and recurrent modeling families while keeping training cost, capacity, and stochastic variation small enough for a complete matched factorial grid. A stronger architecture could improve absolute RMSE, but it would also introduce additional confounders such as attention mechanisms, larger hyperparameter spaces, heavier regularization choices, and longer training schedules. For the present question, the relevant evidence is not whether a particular model is best, but whether the same architecture, cap, window length, subset, and seed produce different conclusions when the normalization protocol changes. Holding these other factors fixed makes the signed paired deltas interpretable as protocol effects within the audited grid.

The grid also separates model-selection decisions from final-test reporting. Model selection is performed on engine-level training and validation partitions. Final reported metrics are then produced by retraining with the selected epoch count on the official training data allowed by the declared final fit scope. In each run, the seed affects both the engine-level train/validation split and stochastic model initialization, so paired seed comparisons should be read as matched protocol comparisons rather than independent replicates over fixed data splits. This structure mirrors common benchmark practice while making the two places where preprocessing statistics are fit explicit and auditable.

After the canonical V3 grid, we ran a focused fit-scope ablation on FD002/FD004 to estimate the sensitivity of final reported metrics to the declared final refit policy. This ablation holds subset, architecture, RUL cap, window length, normalization, and seed fixed while comparing `train_only` against `train_plus_validation` final refitting. It contains 720 completed runs and 360 paired cells. Deltas are defined as `train_plus_validation` minus `train_only`; negative values indicate that the train+validation final fit lowers the metric. Because this policy changes both the normalizer fit scope and the labeled engines used for the final model fit, it should be interpreted as a joint final-refit-policy ablation rather than a normalizer-only attribution. The ablation is separate from the frozen 1080-run V3 grid and is used only to quantify fit-scope sensitivity.

3.4 Labels, Windowing, and Test Protocol

Training labels use piecewise-linear capped RUL targets with cap values 100, 125, and 150. Input trajectories are framed into fixed-length sliding windows of length 30 or 50. During training, each window is paired with the capped RUL target at its final cycle. For official test evaluation, each engine contributes exactly one prediction from the final available window, and the official test RUL labels are used as provided rather than capped. Model predictions are clipped only at the lower bound of zero before computing RMSE and NASA score. If a trajectory is shorter than the required window length, the harness uses deterministic left padding by repeating the first observed cycle.

3.5 Normalization and Leakage Policy

Normalization is the primary protocol factor. The global policy fits one mean and standard deviation per feature. The per_condition policy assigns each record to a deterministic operating regime and fits feature statistics within each regime. Operating regimes are assigned using

$$(\text{round}(op_1),\ \text{round}(100op_2),\ \text{round}(op_3))\,. \tag{1}$$

This rule recovers one regime for FD001/FD003 and six regimes for FD002/FD004 in the canonical V3 logs. For per_condition normalization, regime assignment is record-level: each cycle is scaled using the statistics associated with that cycle's deterministic operating-regime key before windows are assembled, so a window spanning a regime transition may contain records transformed under different regime statistics. For FD001/FD003, the per_condition path explicitly uses the same transformation as global normalization because only one regime key exists; the resulting equality is therefore an intended-equivalence check, not an independent discovery about single-regime normalization.

For model selection, normalization statistics are fit only on the training-engine partition after an engine-level 80/20 train/validation split using the run seed. Validation engines are transformed with those training-engine statistics and are used only to choose the epoch count through validation RMSE. They are not used to fit model-selection preprocessing or model parameters during selection. For final reported test metrics, the selected epoch count is treated as a locked hyperparameter. The model is reinitialized and retrained from scratch for that epoch count, and the normalizer is refit on all official training engines using the explicit train_plus_validation scope. Official test records are never used to fit preprocessing statistics, select epochs, tune hyperparameters, or estimate model parameters.

3.6 Models and Training

The architecture factor includes three compact baselines rather than leaderboard-optimized systems. The MLP flattens the input window and uses hidden widths 96 and 48. The temporal CNN uses 48-channel convolutions with kernel sizes 5 and 3, followed by adaptive average pooling and a 48-unit head. The LSTM has one 48-unit recurrent layer and a 48-unit prediction head. All models use ReLU activations where applicable and dropout 0.15. We minimize mean squared error with AdamW (learning rate 10^{-3}, weight decay 10^{-4}) and batch size 512. Model selection uses at most 15 epochs with patience 4 on validation RMSE. The selected epoch count is the only selection outcome transferred into final testing; final model weights and final preprocessing statistics are recomputed under the locked final-test policy. Python, NumPy, and PyTorch random states are seeded; deterministic PyTorch algorithms are requested, and cuDNN benchmarking is disabled.

3.7 Metrics and Variance Decomposition

We report RMSE and the NASA asymmetric score. Let $d_i = \hat{y}_i - y_i$, where positive error means RUL is overpredicted. The per-engine NASA score is

$$s_i = \begin{cases} \exp(-d_i/13) - 1, & d_i < 0, \\ \exp(d_i/10) - 1, & d_i \geq 0, \end{cases} \tag{2}$$

with total score $S = \sum_i s_i$.

To analyze protocol effects, we compute cell-mean functional ANOVA over the complete factorial grid. For a metric cell mean $y(x)$ indexed by factors x, the grand mean is $\bar{y}$, a main-effect function is $f_A(a) = E[y(x) \mid x_A = a] - \bar{y}$, and each higher-order effect is the corresponding conditional mean

minus $\bar{y}$ and all lower-order effects contained in that interaction. The variance share for term T is $\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} f_T(x_T)^2$ divided by $\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}(y(x)-\bar{y})^2$. The design is balanced at the cell-mean level, so these components sum to the total cell-mean variance. The decomposition partitions variance across evaluated protocol-cell means, not across individual test engines and not across all possible C-MAPSS protocols; we therefore use descriptive terms such as "share" rather than inferential ANOVA significance.

Reported "±" values are sample standard deviations across random seeds. Global-normalization cells contain 10 seeds and per-condition cells contain 5. Normalization comparisons therefore use only matched seeds 0–4. Paired deltas are defined as per_condition minus global, so negative signed deltas mean that per-condition normalization lowers the metric. To reduce false precision from shared test engines and repeated factor settings, we summarize paired deltas at three levels: the 90 matched cells, the 18 architecture/cap/window configuration blocks per subset, and the five seed blocks per subset. For the mean absolute paired differences, we also report 95% percentile bootstrap intervals obtained by resampling the 90 matched cells within each subset; these intervals describe the audited grid and are not population-level confidence intervals.

3.8 C-MAPSS Reporting Checklist

Table 3 gives the reporting fields suggested by this audit for leakage-safe C-MAPSS RUL evaluation.

4 Protocol Verification

Protocol verification is central to the study. Table 4 summarizes the completion and leakage checks.

The single-regime equivalence check is the most important guard against misreading FD001/FD003. Since these subsets each contain one operating regime, the V3 harness intentionally uses the same transformation for global and per-condition normalization under matched architecture, cap, window, and seed. The check therefore verifies that the recorded outputs follow the intended equivalence: max $|\Delta\text{RMSE}| = 0.0$ and max $|\Delta\text{NASA score}| = 0.0$. It should not be interpreted as independent evidence about numerical reduction order or as a new scientific result.

The multi-regime sanity check verifies that the same implementation still preserves real normalization differences when multiple operating regimes exist. FD002 has mean signed $\Delta\text{RMSE} = -4.39$ and max $|\Delta\text{RMSE}| = 8.74$. FD004 has mean signed $\Delta\text{RMSE} = -7.40$ and max $|\Delta\text{RMSE}| = 14.29$. In both subsets, per-condition normalization lowers RMSE in all 90 matched cells.

5 Results

5.1 Best RMSE Configurations

Table 5 reports the best RMSE configuration for each subset. These rows are descriptive summaries of the evaluated grid, not claims of state-of-the-art performance.

The best FD002 and FD004 configurations use per_condition normalization. The single-regime rows are reported as tied because global and per_condition normalization are exact intended ties for matched FD001/FD003 seeds.

5.2 Paired Normalization Deltas

Table 6 compares per_condition against global normalization for matched subset, architecture, RUL cap, window length, and seed. Figure 1 visualizes the signed RMSE deltas by protocol stratum.

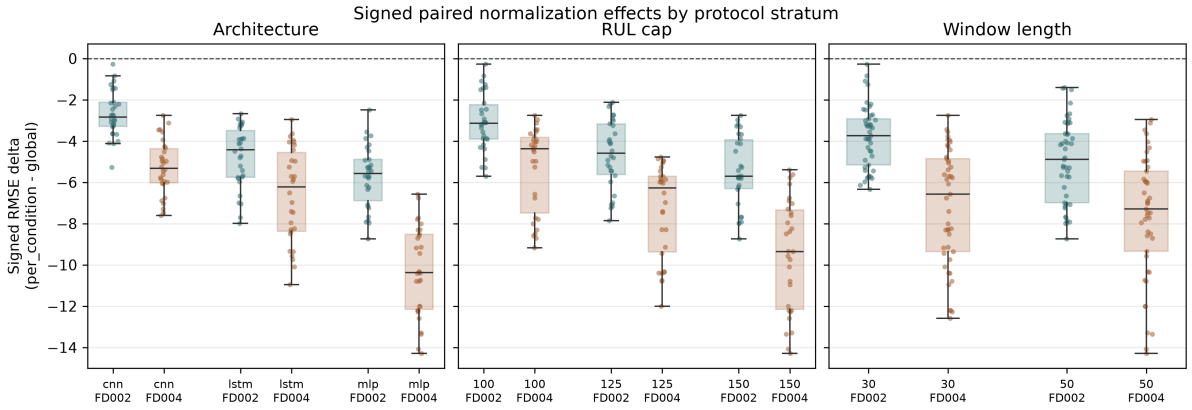

Figure 1: Signed RMSE deltas for multi-regime subsets, stratified by architecture, RUL cap, and window length. Each point is a matched seed/protocol-cell comparison with Δ = per_condition minus global. All plotted FD002 and FD004 deltas are negative, but the magnitude varies by architecture and RUL cap.

The paired pattern remains after blocking by configuration and by seed. On FD002, the 18 architecture/cap/window block means range from -7.71 to -1.18 RMSE, and the five seed-block means range from -4.57 to -4.01. On FD004, the corresponding configuration-block range is -13.40 to -3.59, and the seed-block range is -7.69 to -7.14. A supplementary matched-seed bootstrap summary gives 95% percentile intervals for the mean absolute paired deltas. The intervals remain exactly zero on FD001/FD003 and clearly positive on FD002/FD004; these intervals are descriptive uncertainty summaries over the audited matched cells, not claims about all possible C-MAPSS protocols.

The NASA-score deltas are directionally consistent with RMSE but more concentrated, as expected from the exponential late-prediction penalty. Across the 90 matched FD002 comparisons, the mean paired NASA-score delta in Table 6 sums to a net improvement of approximately 1.44×10^6 score units, and the largest 5% of engine-pair observations account for 89.5% of the positive per-engine improvement. For FD004, the corresponding summed improvement is also approximately 1.44×10^6 score units, with the largest 5% of engine-pair observations accounting for 73.5% of positive per-engine improvement. These diagnostics support reporting NASA score alongside RMSE rather than treating it as a uniformly distributed improvement over engines.

5.3 Variance Decomposition

Table 7 summarizes the top variance-decomposition terms, and the supplementary artifact lists every term with share at least 1%. These are descriptive shares over evaluated cell means. Overall RMSE variation has its largest share in subset identity: subset explains 60.8%. Overall NASA score also has its largest share in subset identity, which explains 48.7%. Within subsets, the largest descriptive factor changes. Architecture explains 75.0% of FD001 RMSE cell-mean variation and 74.8% of FD003 RMSE variation. In contrast, FD002 and FD004 place their largest descriptive shares on protocol factors: for FD002 RMSE, RUL cap explains 55.9% and normalization explains 27.3%; for FD004 RMSE, normalization explains 46.7% and RUL cap explains 37.8%. The largest omitted RMSE interactions are architecture+normalization on FD002 (2.1%) and architecture+normalization plus RUL cap+normalization on FD004 (3.3% and 3.0%), so the interaction structure is present but does not change the main ordering inside this audited grid.

The top-3 stability analysis is retained as supplementary material rather than a main-paper

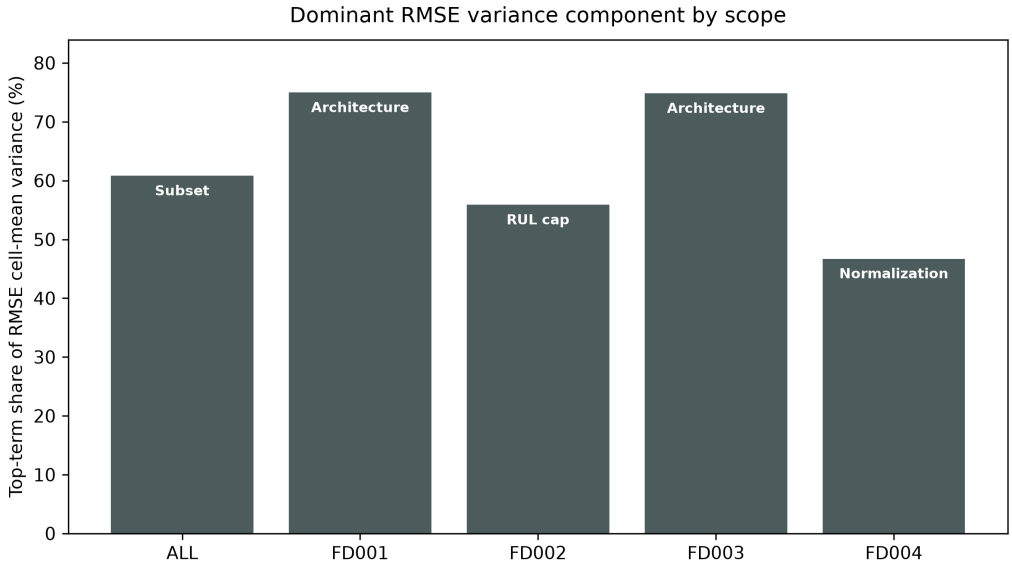

Figure 2: Largest descriptive RMSE variance components by scope. The largest share shifts from subset identity overall to architecture in single-regime subsets and normalization/RUL cap in multi-regime subsets.

result. This preserves the main PHM message: the evidence is stronger for protocol sensitivity than for fine-grained model ranking.

Because the canonical grid has 10 global seeds but 5 per-condition seeds, we also recomputed the variance decomposition using only matched seeds 0–4. The qualitative interpretation is unchanged: subset remains the largest overall share (60.7% of RMSE cell-mean variance), and FD002 keeps its largest shares in RUL cap (54.9%) and normalization (28.7%). FD004 is the only notable sensitivity: RUL cap (41.7%) and normalization (40.2%) swap order but remain the two largest terms. We therefore interpret the decomposition as descriptive evidence that multi-regime results are sensitive to both RUL cap and normalization, not as a precise universal ordering of those two factors.

5.4 Joint Final-Refit Policy Sensitivity

The focused joint final-refit-policy ablation completed all 720 planned FD002/FD004 runs with 0 failures, 0 non-finite metrics, 0 engine-split overlaps, and 360 paired cells. Figure 3 summarizes paired deltas for final refitting on train+validation engines relative to final refitting on train-only engines. Train+validation final fitting lowers RMSE on average in every subset/normalization stratum: FD002/global has mean delta -0.594 with bootstrap interval [-0.787, -0.408], FD002/per-condition has -0.398 [-0.528, -0.267], FD004/global has -1.119 [-1.333, -0.918], and FD004/per-condition has -0.206 [-0.333, -0.075]. NASA-score deltas are also negative on average but noisier, particularly in FD002/global. These results show that the joint final-refit policy is a measurable protocol choice even when official test records remain excluded from preprocessing and model fitting; they do not identify how much of the change comes from refitting preprocessing statistics rather than from adding validation engines to final model training.

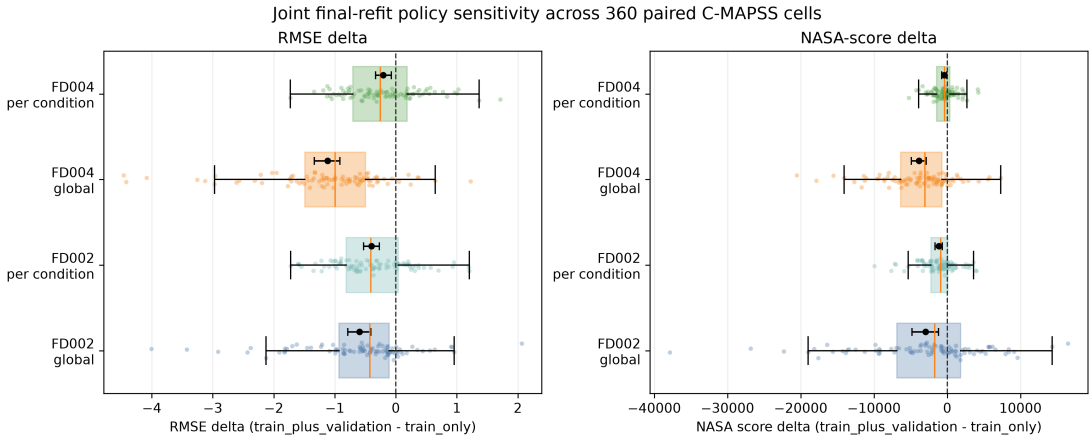

Figure 3: Paired joint final-refit-policy deltas across the 360 FD002/FD004 ablation cells. Each point is one matched configuration/seed pair; boxplots summarize the distribution and black markers show bootstrap mean 95% intervals. The dashed vertical line marks no change. Negative values indicate that jointly refitting the final model and normalizer on train+validation improves the metric relative to refitting on train only.

6 Discussion

The main result is that operating-condition normalization is large in the multi-regime C-MAPSS subsets under this audited grid and descriptively large among the evaluated protocol factors, but not a separate factor in the single-regime subsets. This finding is consistent with prior PHM work on operating-condition-specific standardization Pasa et al. (2019). The present contribution is the controlled audit around that known effect: preprocessing fit scope is explicit, test statistics are excluded, split units are engine-level, paired comparisons use matched seeds, and each result file records the protocol fields needed to reproduce the cell. On FD001 and FD003, where the operating-regime set contains a single element, the V3 harness intentionally maps per-condition normalization to the same transformation as global normalization. The exact equality is therefore best read as an implementation and reporting check for intended equivalence, not as an independent proof about floating-point reduction order or as a new scientific discovery.

On FD002 and FD004, multiple operating regimes create sensor-distribution shifts that can interact with degradation trends. Per-condition normalization changes the input representation by normalizing within deterministic operating-regime keys. The signed paired deltas show the direction of this effect: per-condition normalization lowers RMSE and NASA score in all 90 matched FD002 cells and all 90 matched FD004 cells. Because these effects are several RMSE points, they are large enough to affect benchmark interpretation when compared methods differ in preprocessing policy; we avoid making a stronger ranking-swap claim without a dedicated cross-method ranking analysis. We exclude operating settings from the model inputs to isolate normalization policy as the audited protocol lever; models that ingest operating settings directly may learn part of the same condition structure, and testing that interaction is an important extension.

These findings motivate a small set of suggested reporting fields for C-MAPSS. A paper can make comparisons easier to interpret by stating whether normalization statistics were fit globally or by operating condition, whether they were fit before or after validation splitting, whether test statistics were used, how operating regimes were identified, which RUL cap was used, which sensors were retained, how windows were created, and whether validation was engine-level.

The fit-scope ablation adds a second, smaller sensitivity result. Refitting the final model and normalizer on train+validation engines tends to improve held-out official test metrics relative to refitting on train-only engines, but the magnitude depends on subset and normalization policy. This is not evidence of test leakage: both policies exclude official test records. Instead, it quantifies the consequence of a common benchmark choice about whether validation engines are returned to the final training pool after epoch selection. Because the ablation changes both preprocessing-statistic scope and final labeled training data, we treat it as a joint final-refit-policy sensitivity rather than as a normalizer-only result.

6.1 Practical Implications for PHM Benchmarking

The practical implication is that C-MAPSS comparisons should not be interpreted from architecture names alone. A reported LSTM, CNN, Transformer, or domain-adaptation model is also tied to a preprocessing and evaluation contract. If that contract is not stated, differences in RMSE or NASA score can mix modeling improvements with differences in normalization scope, target capping, windowing, and validation practice. This is especially consequential for FD002 and FD004, where the present audit shows that normalization alone can shift RMSE by several points under matched seeds and identical model capacity.

For authors, the suggested fields in Table 3 are intended as a low-cost addition to C-MAPSS papers. They do not require extra experiments; they require recording and disclosing choices that already exist in the pipeline. For reviewers and readers, the same fields can help separate architecture claims from preprocessing and evaluation choices. If a paper reports a new RUL architecture but omits normalization fit scope, test-label policy, or split unit, then the comparison may still be useful, but it is incomplete as a benchmark claim. This does not imply that every paper must reproduce the full factorial grid here; it means that reported metrics should be accompanied by enough protocol detail for another group to reconstruct the evaluation.

For future benchmark studies, the audit suggests separating three claims that are often conflated. A modeling claim says that one architecture better captures degradation dynamics. A preprocessing claim says that a transformation better handles operating-regime structure. A protocol claim says that a reported metric is leakage-safe and comparable under a stated evaluation contract. The present paper contributes primarily to the third category while quantifying one important preprocessing factor. Keeping these claims separate should make C-MAPSS results more reusable for PHM researchers and less sensitive to hidden implementation choices.

6.2 Reproducibility and Data Availability

The audit is designed so that each numerical result can be traced back to an explicit protocol cell. The public NASA C-MAPSS data are used as the source dataset, and the code and result artifacts are maintained at `https://github.com/Aaditya-Gupta24/cmapss-protocol-harness`. The repository includes the grid driver, dependency specification, table-generation scripts, plotting scripts, result summaries, and revision history for the manuscript. Each result record stores the dataset subset, architecture, RUL cap, window length, normalization policy, seed, split audit, fitted-preprocessing scope, predictions, targets, RMSE, NASA score, validation history, and runtime. This file-level provenance is part of the contribution: it makes preprocessing and evaluation choices inspectable rather than implicit.

6.3 Claims and Boundaries

The strongest claim supported by this study is that, under an audited leakage-safe C-MAPSS protocol with operating settings excluded from the model inputs, operating-condition normalization is large and descriptively large among protocol factors in the multi-regime subsets, while remaining an intended tie on single-regime subsets. The focused follow-up ablation supports the narrower claim that joint final refit policy is also measurable on FD002/FD004, with train+validation final refitting improving RMSE on average relative to train-only final refitting. The study does not claim that per-condition normalization is universally better, does not claim novelty for operating-condition-specific standardization itself, does not claim that the evaluated compact baselines are state-of-the-art, and does not claim that the same factor ordering will hold for every possible model class, sensor set, or real-world prognostics dataset. Because operating settings are not crossed as model inputs in the frozen grid, the reported normalization magnitudes should be read as conditional on that design and non-transportable to pipelines that provide the operating settings directly without a separate crossed audit. The most direct way to expand the audit beyond the known Pasa effect would be a new ablation of regime assignment, controlled leakage, or operating settings as model inputs; those experiments are not part of the frozen V3 evidence.

7 Threats to Validity

The results are specific to NASA C-MAPSS FD001–FD004. C-MAPSS is simulated and may not capture all sources of noise, maintenance intervention, sensor drift, or fleet heterogeneity present in operational predictive-maintenance settings. Newer variants such as N-CMAPSS incorporate real flight conditions and richer degradation structure Arias Chao et al. (2021). The suggested reporting fields proposed here should transfer naturally to such datasets, but the numerical variance decomposition and paired normalization magnitudes should not be assumed to transfer without a separate audit. The model set includes compact MLP, temporal CNN, and LSTM baselines. It does not include every modern architecture, such as Transformers, attention hybrids, ensembles, or physics-informed models. This is by design: the objective is protocol sensitivity, not leaderboard optimization.

The canonical grid varies subset, architecture, RUL cap, window length, normalization, and seed; the separate fit-scope ablation is limited to FD002/FD004 and should not be read as evidence for FD001/FD003. Other choices may matter, including optimizer schedule, hidden size, dropout, sensor subset, loss function, early-stopping patience, final preprocessing fit scope, and inclusion of operating settings as input features. Because operating settings are excluded from model inputs here, the estimated normalization effect may be larger than in pipelines that provide the model with operating-condition features directly. The functional ANOVA partitions variance over evaluated cell means. It should not be interpreted as a universal decomposition over all possible C-MAPSS protocols or over individual test-engine errors. We evaluate one deterministic regime definition, chosen for transparency and exact reproducibility because it recovers one regime on FD001/FD003 and six regimes on FD002/FD004. Clustered alternatives such as K-means or GMM assignments may be useful, but would introduce additional stochastic or modeling choices and are left for follow-up sensitivity studies; claims about operating-condition normalization should therefore be read as claims under this deterministic key. Official test RUL labels are not capped, so the RUL-cap factor changes the training target rather than the evaluation labels. The large FD002 RUL-cap variance share should be interpreted partly as sensitivity to train-target cap choice under uncapped official test labels, not as a property of degradation dynamics alone. Finally, the NASA score is asymmetric and can be dominated by a small number of late predictions; we therefore report RMSE,

NASA score, and NASA-score influence diagnostics together.

Broader impacts. More reproducible RUL evaluation can reduce misleading comparisons and support safer maintenance research. However, C-MAPSS is simulated, and treating benchmark performance as evidence of deployment readiness could create safety and economic risks in real maintenance decisions. Results should therefore be validated on representative operational data, under distribution shift, and with domain-specific uncertainty and safety procedures before deployment. The study uses no personal data or human participants and does not release a high-risk model.

8 Conclusion

This paper reframes NASA C-MAPSS RUL prediction as a protocol-sensitive evaluation problem. Under a frozen, leakage-safe V3 grid, global and per-condition normalization are exact intended ties on single-regime FD001 and FD003, while per-condition normalization lowers RMSE and NASA score in every matched FD002 and FD004 cell. The full V3 grid completed 1080 / 1080 runs with 0 failures, 0 engine split overlaps, and 0 non-finite metrics, and an FD002 seed-0 canonical cap=125/window=30 six-cell reproducibility rerun produced max metric/prediction diff = 0.0. A separate 720-run FD002/FD004 ablation shows that joint final train+validation refitting also shifts reported RMSE relative to train-only final fitting, with mean deltas from -1.119 to -0.206 across subset/normalization strata. Variance decomposition shows descriptively that subset has the largest overall share, architecture has the largest single-regime RMSE shares, and RUL cap plus normalization have the largest multi-regime RMSE shares. These findings support a protocol-first view of C-MAPSS benchmarking: preprocessing choices and final fit scope should be audited and reported with the same care as model architecture.

References

Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, 6(1):5, 2021. doi: 10.3390/data6010005.

Giduthuri Sateesh Babu, Peilin Zhao, and Xiao-Li Li. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *Database Systems for Advanced Applications*, pages 214–228. Springer, 2016. doi: 10.1007/978-3-319-32025-0_14.

André Listou Ellefsen, Emil Bjørlykhaug, Vilmar Æsøy, Sergey Ushakov, and Houxiang Zhang. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, 183:240–251, 2019. doi: 10.1016/j.ress.2018.11.027.

Dean K. Frederick, Jonathan A. DeCastro, and Jonathan S. Litt. User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS). Technical Report NASA/TM-2007-215026, NASA Glenn Research Center, 2007. URL `https://ntrs.nasa.gov/citations/20070034949`.

Felix O. Heimes. Recurrent neural networks for remaining useful life estimation. In *2008 International Conference on Prognostics and Health Management*, pages 1–6, 2008. doi: 10.1109/PHM.2008.4711422.

Cheng-Geng Huang, Hong-Zhong Huang, and Yan-Feng Li. A bidirectional lstm prognostics method under multiple operational conditions. *IEEE Transactions on Industrial Electronics*, 66(11): 8792–8802, 2019. doi: 10.1109/TIE.2019.2891463.

Qi Li, Zhan Bao Gao, and Li Qun Shao. An operating condition classified prognostics approach for remaining useful life estimation. In *2014 International Conference on Prognostics and Health Management*, 2014. doi: 10.1109/ICPHM.2014.7036396.

Xiang Li, Qian Ding, and Jian-Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018. doi: 10.1016/j.ress.2017.11.021.

Reuben Lim and David Mba. Condition monitoring and remaining useful life prediction using switching kalman filters. *International Journal of Strategic Engineering Asset Management*, 2(1): 54–71, 2014. doi: 10.1504/IJSEAM.2014.063881.

Gabriel Duarte Pasa, Ivo Paixão de Medeiros, and Takashi Yoneyama. Operating condition-invariant neural network-based prognostics methods applied on turbofan aircraft engines. In *Annual Conference of the PHM Society*, volume 11, 2019. doi: 10.36001/phmconf.2019.v11i1.786.

Leto Peel. Data driven prognostics using a kalman filter ensemble of neural network models. In *2008 International Conference on Prognostics and Health Management*, pages 1–6, 2008. doi: 10.1109/PHM.2008.4711423.

Mohamed Ragab, Zhenghua Chen, Min Wu, Chuan Sheng Foo, Chee Keong Kwoh, Ruqiang Yan, and Xiaoli Li. Contrastive adversarial domain adaptation for machine remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, 17(8):5239–5249, 2021. doi: 10.1109/TII.2020.3032690.

Emmanuel Ramasso and Abhinav Saxena. Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. *International Journal of Prognostics and Health Management*, 5 (2), 2014. doi: 10.36001/ijphm.2014.v5i2.2236.

Abhinav Saxena and Kai Goebel. PHM08 challenge data set. NASA Prognostics Data Repository, NASA Ames Research Center, 2008a. URL `https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/`. Accessed July 8, 2026; NASA repository page lists the PHM08 Challenge Data Set but states direct download and evaluation are currently unavailable.

Abhinav Saxena and Kai Goebel. Turbofan engine degradation simulation data set. NASA Prognostics Data Repository, NASA Ames Research Center, 2008b. URL `https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/`. Accessed July 8, 2026; NASA repository page lists the Turbofan Engine Degradation Simulation data set and download link.

Abhinav Saxena, Jose Celaya, Edward Balaban, Kai Goebel, Bhaskar Saha, Sankalita Saha, and Mark Schwabacher. Metrics for evaluating performance of prognostic techniques. In *2008 International Conference on Prognostics and Health Management*, pages 1–17, 2008a. doi: 10.1109/PHM.2008.4711436.

Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pages 1–9, 2008b. doi: 10.1109/PHM.2008.4711414.

Simon Vollert and Andreas Theissler. Challenges of machine learning-based RUL prognosis: A review on NASA's C-MAPSS data set. In *2021 IEEE 26th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021. doi: 10.1109/ETFA45728.2021.9613682.

Tianyi Wang. *Trajectory Similarity Based Prediction for Remaining Useful Life Estimation*. PhD thesis, University of Cincinnati, 2010. Doctoral dissertation.

Tianyi Wang, Jianbo Yu, David Siegel, and Jay Lee. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *2008 International Conference on Prognostics and Health Management*, pages 1–6, 2008. doi: 10.1109/PHM.2008.4711421.

Shuai Zheng, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta. Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management*, pages 88–95, 2017. doi: 10.1109/ICPHM.2017.7998311.

Dmitry Zhevnenko, Mikhail Kazantsev, and Ilya Makarov. Interaction models for remaining useful life estimation, 2023.

A Reproducibility details

The repository at `https://github.com/Aaditya-Gupta24/cmapss-protocol-harness` contains the evaluation harness, exact grid driver, dependency specification (`pyproject.toml`), current environment lockfile (`requirements-lock.txt`), analysis scripts, and instructions for obtaining the public NASA data. The seed list is 0–4 for matched paired comparisons and 0–9 for global-normalization cells. The canonical command is

```
python run_full_grid.py --epochs 15 \
  --patience 4 --batch-size 512 \
  --hidden-size 48 \
  --output-dir runs_full_grid_v3
```

Each result file records the complete protocol and training configuration, engine-level split audit, fitted-preprocessing scope, validation history, runtime, predictions, and targets. The final-test policy is explicitly recorded: model selection fits preprocessing statistics on the seed-specific training engines only, whereas final reported test metrics refit the normalizer on all official training engines and never use official test records for preprocessing. The training configuration records `device=auto`; the original result records did not preserve the resolved accelerator model, driver version, or aggregate energy use, so that hardware provenance limitation is reported explicitly rather than reconstructed after the fact.

B Supplementary analyses

The matched-seed bootstrap resamples the 90 paired cells per subset (three architectures, three RUL caps, two window lengths, and five seeds). The 95% percentile intervals for mean absolute RMSE difference are $[0, 0]$ for FD001, $[4.01, 4.78]$ for FD002, $[0, 0]$ for FD003, and $[6.81, 7.99]$ for FD004. Corresponding NASA-score intervals are $[0, 0]$, $[14270.2, 17860.9]$, $[0, 0]$, and $[14382.4, 17598.8]$. These are descriptive uncertainty summaries over the evaluated cells. Additional review-fallout artifacts include `paired_delta_config_block_summary.csv`, `paired_delta_seed_block_summary.csv`, `paired_delta_factor_summary.csv`, and `major_variance_terms_ge1pct.csv`.

Table 1: Positioning relative to closely related C-MAPSS and operating-condition work.

Study	What it establishes	Remaining gap for this paper	How this audit extends it
Peel; Wang et al.; Lim and Mba; Li et al. Peel (2008); Wang et al. (2008); Wang (2010); Lim and Mba (2014); Li et al. (2014)	PHM08 and early post-challenge turbofan prognostics already tied RUL estimation to operating-regime-aware modeling and evaluation.	Does not isolate modern neural-normalization choices under matched C-MAPSS protocol cells.	Places operating-condition normalization in the PHM08 lineage rather than treating it as a new ML preprocessing idea.
Ramasso and Saxena Ramasso and Saxena (2014)	C-MAPSS is a benchmark whose metrics and subset structure affect method comparison.	Leaves preprocessing fit scope and matched normalization deltas outside the main benchmark comparison.	Adds leakage-safe fit-scope recording, paired normalization deltas, and suggested reporting fields.
Pasa et al. Pasa et al. (2019)	Operating-condition-specific standardization can improve multi-regime C-MAPSS neural prognostics and is equivalent to global standardization in single-regime subsets.	Does not focus on train/validation/final normalization scopes, matched-seed deltas, or variance attribution across protocol factors.	Treats operating-condition normalization as a controlled audit variable rather than a new method contribution.
Ragab et al. Ragab et al. (2021)	Working-condition-specific normalization can be useful inside a domain-adaptation RUL pipeline.	Normalization is part of a modeling pipeline, not the object of a protocol audit.	Freezes architecture/cap/window/seed and reports the isolated normalization effect.
Ellefsen et al.; Zhevnenko et al. Ellefsen et al. (2019); Zhevnenko et al. (2023)	Contemporary C-MAPSS model papers can report strong results under architecture-specific pipelines.	Protocol details are not the primary object of study.	Motivates reporting normalization, fit scope, operating settings, and score conventions alongside model claims.
This paper	Protocol choices can be varied or recorded as first-class PHM benchmark variables.	–	Quantifies signed matched-cell effects, records a frozen fit scope, uses FD001/FD003 as a unit test, and recommends reporting fields.

Table 2: Protocol grid for the canonical V3 evaluation.

Factor	Levels
Dataset subset	FD001, FD002, FD003, FD004
Architecture	MLP, temporal CNN, LSTM
RUL cap	100, 125, 150
Window length	30, 50
Normalization	global, per_condition
Sensor set	standard 14
Seeds	5 per cell; 10 for global

Table 3: Suggested C-MAPSS reporting fields for leakage-safe RUL evaluation.

Item	What to report
RUL cap	Uncapped/capped and cap value
Window length	Sliding-window length and padding
Sensor set	Retained sensors and operating settings as inputs
Normalization policy	Global, per-condition, residualized, or other rule
Normalization fit scope	Engines used to fit preprocessing statistics
Regime assignment	Rule or model used to define regimes
Split unit	Engine-level, window-level, or other
Test prediction rule	Final-window prediction or other rule
NASA score convention	Error direction and asymmetric denominators

Table 4: Verification checks for the canonical V3 result set.

Check	Result
Canonical V3 grid completed	1080 / 1080
Failures	0
Engine split overlaps	0
Non-finite metrics	0
FD002 seed-0 reproducibility diff	0.0

Table 5: Best seed-balanced RMSE configuration for each C-MAPSS subset using matched seeds 0–4. For FD001/FD003, global and per_condition are exact intended ties under the listed configuration.

Subset	Architecture/cap/window	Normalization	RMSE	NASA score
FD001	LSTM, cap 125, window 30	tied	15.97 ± 0.45	595.2 ± 96.3
FD002	LSTM, cap 150, window 50	per_condition	22.73 ± 0.28	4086.0 ± 311.8
FD003	MLP, cap 125, window 50	tied	16.76 ± 0.22	533.6 ± 57.6
FD004	LSTM, cap 150, window 50	per_condition	21.78 ± 0.11	2925.0 ± 165.6

Table 6: Paired normalization deltas. Signed deltas are per_condition minus global; negative values favor per-condition normalization because lower RMSE and NASA score are better. "Better" counts strict per-condition improvements; "Ties" counts exact matched-cell ties.

Subset	Metric	Mean Δ	Mean |Δ|	Better	Ties
FD001	RMSE	0.00	0.00	0/90	90/90
FD001	NASA score	0.00	0.00	0/90	90/90
FD002	RMSE	-4.39	4.39	90/90	0/90
FD002	NASA score	-15978.94	15978.94	90/90	0/90
FD003	RMSE	0.00	0.00	0/90	90/90
FD003	NASA score	0.00	0.00	0/90	90/90
FD004	RMSE	-7.40	7.40	90/90	0/90
FD004	NASA score	-15970.74	15970.74	90/90	0/90

Table 7: Top variance-decomposition terms. Percentages partition cell-mean variance across evaluated factors and interactions.

Scope	Metric	Dominant terms
Overall	RMSE	subset: 60.8%; architecture: 8.2%; RUL cap: 7.3%
Overall	NASA score	subset: 48.7%; subset+normalization: 12.4%; normalization: 11.7%
FD001	RMSE	architecture: 75.0%; window length: 9.5%; RUL cap: 6.9%
FD002	RMSE	RUL cap: 55.9%; normalization: 27.3%; architecture: 11.3%
FD003	RMSE	architecture: 74.8%; architecture+RUL cap: 12.1%; RUL cap: 11.4%
FD004	RMSE	normalization: 46.7%; RUL cap: 37.8%; architecture: 7.5%