Symbolic AI versus LLMs for Cross-Format COSMIC Functional Size Measurement: A Five-Method Benchmark

Nathanael Motta[1], **Cleyton Mário de Oliveira Rodrigues**[1], **Carlo Marcelo Revoredo da Silva**[1]

[1]Universidade de Pernambuco (UPE)
Recife – PE – Brazil

nathanael.carauna@gmail.com, cleyton.rodrigues@upe.br, marcelo.revoredo@upe.br

Abstract. *Automated COSMIC functional size measurement reaches near-expert accuracy with Large Language Models (LLMs), yet each method is validated on a single requirement format, so heterogeneity is untested. Our five-method, four-format heterogeneity matrix pits language models against a zero-cost, auditable symbolic pipeline. Specializing only a per-format normalization layer over an invariant engine, the framework has no significantly negative Standardized Accuracy cell (significantly positive in three of four), whereas every frozen competitor collapses off-format; a small adapter beats GPT-4-turbo on user stories without touching the engine. Heterogeneity is resolved by normalization rather than the engine, and the same layer rescues the LLM.*

1. Introduction

Effective software project management remains a notorious challenge, and at its core lies the need for accurate estimates of effort, cost and duration. A reliable measure of software size is a primary input to any such estimate [Abran 2015]. Physical metrics such as lines of code proved limited by their technology dependence and late availability, which motivated Functional Size Measurement (FSM): function point analysis [Albrecht and Gaffney 1983] pioneered the quantification of delivered functionality independently of implementation, and second-generation methods such as COSMIC (ISO/IEC 19761) [ISO/IEC 2011] refined these concepts for modern, agile development.

Despite its objectivity, FSM faces a practical bottleneck: measurement is predominantly manual, labor-intensive, expert-dependent and subject to interpretation inconsistencies [Abran 2015, Hussain et al. 2013]. In agile settings that demand rapid feedback, the manual process becomes an obstacle, pushing teams toward subjective proxies such as story points. Automating FSM from textual requirements has thus become a central research goal, with COSMIC the most frequent target in recent surveys of LLM-based estimation [Radliński and Swacha 2025]. The literature shows a clear evolution: from symbolic AI based on ontologies and rules [Bagriyanik and Karahoca 2016, Jurkiewicz and Nawrocki 2015], through classical machine learning dependent on feature engineering [Hussain et al. 2013, Zhang et al. 2016], to the current state of the art with deep learning and Large Language Models (LLMs), which reach near-expert performance in extracting functional components [Ochodek et al. 2020, De Vito et al. 2025, Zhao et al. 2024, Ünlü et al. 2024].

The literature also exposes two gaps that limit practical adoption. The first is a *lack of generalization to heterogeneous requirements*: automated methods are typically developed and validated for a single requirement format [De Vito et al. 2025, Ochodek et al. 2020, Hussain et al. 2013, Ünlü et al. 2024], yet different teams and projects adopt different formats (use cases, user stories, isolated sentences, specifications), so a method tied to one format does not carry over to the others, and this cross-format behavior is rarely characterized. The second is a *lack of interpretability*: the most capable models operate as black boxes, producing a number with no auditable justification [De Vito et al. 2025]. This paper targets the first gap, the more fundamental obstacle to deployment, and asks:

> *Can a single approach sustain better-than-trivial measurement across multiple requirement formats, where state-of-the-art methods degrade outside their format of origin?*

We answer with *Hybrid-XAI*, a hybrid and explainable framework whose architectural premise is that heterogeneity is a *normalization* problem, not a measurement problem. A per-format normalization layer (M1) converts any requirement into a common intermediate representation, over which a single, invariant symbolic rule engine (M2) applies COSMIC, and an explainability module (M3) traces each counted CFP to the rule and the text span that produced it. It is hybrid in pairing a data-driven NLP normalization layer (M1) with a symbolic, deterministic rule engine (M2), and explainable by construction through M3. The framework admits two instantiations: a learned pipeline on a local Small Language Model, and a symbolic pipeline that needs no language model. We show that the symbolic instantiation, an interpretable rule engine over classical NLP, is the one that generalizes across formats, so the study also speaks to the broader symbolic-versus-learned question in applied AI, where opaque LLMs collapse off-format but a transparent symbolic method sustains measurement at zero cost. Our contributions are: (i) a five-method, four-format *heterogeneity matrix* with cell-by-cell statistical qualification, an evaluation design without precedent among the reviewed works; (ii) the empirical demonstration that format heterogeneity is resolved in the normalization layer, isolated by a *symmetry probe* that grants our normalization to a competitor and further confirmed by external validation of the measurement engine on independent gold-standard case studies; (iii) a public ground truth of 98 user stories with a formal annotation protocol and three annotators;[1] and (iv) a documented self-correcting protocol, in which two structural methodological errors were detected, quantified and incorporated, with the central claim reformulated from the corrected values.

2. Background and Related Work

The COSMIC method. COSMIC [ISO/IEC 2011, COSMIC 2009] measures functional size by counting *data movements* within each functional process. Four types exist: *Entry* (E, data from a functional user into the process), *Exit* (X, from the process to a user), *Read* (R, from persistent storage) and *Write* (W, to persistent storage). Each movement counts one COSMIC Function Point (CFP); the size of a process is the sum of its movements.

[1] Code, the annotation protocol, and the 98-story ground truth are available at `https://anonymous.4open.science/r/cosmichybridxay` (de-anonymized upon acceptance).

Every process has at least one triggering Entry and at least one Exit or Write, and the error and confirmation messages of a process are aggregated into a single Exit.

State-of-the-art methods and their evaluation gap. We re-execute three approaches that represent the field. COSMET [De Vito et al. 2025] (R1) uses GPT-4-turbo with sentence splitting and rule-guided analysis over structured use cases. DEEP-COSMIC-UC [Ochodek et al. 2020] (R2) regresses CFP from use-case *names*, in the comparative configuration of Yokogawa et al. [Yokogawa et al. 2024] (CNN, RF, SVR, BERT). HKO [Hussain et al. 2013] (R3) classifies size from 13 linguistic features and approximates CFP by the class median. Other LLM-based work targets user stories [Ünlü et al. 2024] or function point analysis [Zhao et al. 2024], and commercial tools such as ScopeMaster [Ungan et al. 2018] measure COSMIC from user stories but are proprietary and assume conformant input. The closest work to ours is Laqrichi [Laqrichi 2024], a hybrid framework that combines GPT-4 with a rule system to measure COSMIC across free text, user stories and use cases on the Consortium case studies; it relies on a commercial LLM rather than a zero-cost local symbolic engine, provides no rule-to-excerpt explanation of each movement, and does not subject the methods to a systematic heterogeneity matrix with per-cell statistical qualification. To our knowledge, no prior work pairs cross-format coverage with a locally executable, by-construction-explainable engine, nor evaluates the methods under a statistically qualified matrix, as deployment across diverse requirement formats would require.

Metric for cross-format comparison. Mean absolute error (MAE) is not comparable across datasets, because it carries the scale and dispersion of the ground truth. We therefore center the evaluation on *Standardized Accuracy* (SA) [Shepperd and MacDonell 2012], $SA = 1 - MAE/MAE_{\text{baseline}}$, where the baseline predicts the mean of the ground truth. $SA > 0$ means the method beats an uninformed guess and $SA < 0$ means it is worse than guessing, a quantity that remains comparable across datasets of different scales. SA is therefore the main cross-format comparison metric in this paper, with the caveat that its value depends on the choice of trivial baseline.

3. The Hybrid-XAI Framework

Architectural thesis. The framework comprises a pre-processing stage (M0) and three core modules (Figure 1). M0 detects the source format of each requirement and routes it to the matching normalizer. The *Semantic Normalization module (M1)* converts heterogeneous requirements into a standardized *Intermediate Representation* (IR) that captures the essential functional elements (actor, action, data object, conditions) independently of the source format. The *Measurement Engine (M2)* applies COSMIC rules over the IR, classifying each action into E/X/R/W and aggregating CFP. The *Explainability module (M3)* links each counted CFP to the COSMIC rule that classified it and to the requirement excerpt that triggered it. The fundamental design decision is that heterogeneity is handled exclusively in M1, where only the normalization layer is specialized per format (each module validated on held-out data), while the measurement engine and the explanation remain identical across all formats. As the results show, this decision has direct empirical support.

Intermediate Representation. The IR is a hierarchical schema that decomposes a requirement into functional processes, steps and atomic actions. Each atomic action

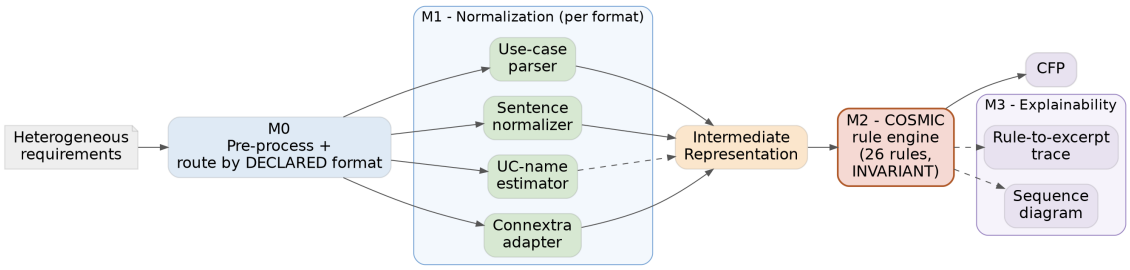


Figure 1. The Hybrid-XAI framework. Only normalization (M1) is specialized per format; the measurement engine (M2) and explainability (M3) are invariant.

records the acting party (functional user or system), the lemmatized verb, the data group, the object of interest and auxiliary flags (negation, condition, interface element). It enforces method-aligned constraints: every functional process must hold at least one triggering Entry, and identical movements (same type and same normalized data group) within a process are counted once.

Symbolic measurement engine. Our main instantiation is a symbolic pipeline with no language model. It chains a dependency parse [Honnibal and Montani 2017], a decomposer that splits compound actions into verb–theme atoms, and a semantic-role extractor applied with narrow scope (only to detect negation, suppressing movements of the negated action). Its core is an *ontology of 26 identified classification rules* implementing the measurement manual [COSMIC 2009]: each rule maps a pattern (actor, verb class, presence of a data group, mention of storage or interface) to a data movement or to the deliberate absence of one. For instance, selecting a UI element yields no movement, whereas selecting a domain datum yields an Entry, the difference between counting data movements and counting clicks. After classification, deduplication collapses identical movements over the same normalized data group, per ISO 19761. Every classification records the rule identifier and the text span that fired it, making the engine auditable by construction. Here "ontology" denotes an operational rule set rather than a formal one, and a few rules are gated by the detected format (for example, an implicit-read rule active only on isolated sentences), which counts as per-format specialization rather than an engine rewrite. An earlier version included a trained named-entity recognizer; an ablation showed it was decorative, since removing it does not change CFP in any format, so it was dropped from the final architecture.

The Connextra adapter. The cross-format evaluation revealed that the symbolic pipeline, designed over use cases, collapses on Connextra user stories (*"As a ⟨role⟩, I want ⟨action⟩, so that ⟨benefit⟩"*): the parser extracted the verb of the frame ("want"), never the real action, and every story fell back to a single Entry. The fix is a normalization component, not an engine change (Figure 2): an adapter that converts a user story into one or more canonical pseudo use cases, with all rules derived from the annotation protocol, the measurement standard fixed before evaluation, and not from the ground truth. Each process receives a triggering Entry and a result Exit; query verbs yield Read and Exit, update and delete verbs yield a prior Read and a Write; coordinated distinct writes ("*create, update and delete X*") are decomposed into separate processes and summed, whereas read-only coordination ("*filter, sort and search*") stays one process. To prevent overfitting, the adapter was calibrated on a 20-story development subset and evaluated once on the

remaining 78.

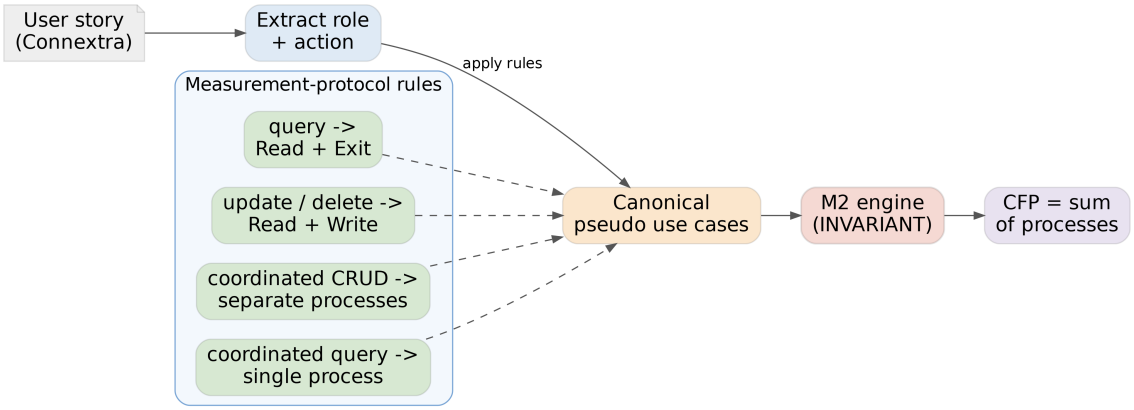


Figure 2. The Connextra adapter: a user story is converted into one or more pseudo use cases by rules anchored in the measurement protocol, which then feed the invariant measurement engine.

A neural instantiation. For reference we also built a neural pipeline whose M1 is a local Small Language Model (Qwen3-8B [Qwen Team 2025]) with parametric correction and a BERT–Ridge ensemble [Devlin et al. 2019]; it collapses outside its home format and so does not carry the heterogeneity thesis. The M3 artifacts are a rule-to-excerpt trace and a COSMIC-isomorphic sequence diagram, evaluated functionally below.

4. Experimental Design

The study follows a design-science methodology. We use four public datasets (Table 1) chosen to cover four distinct requirement formats. The De Vito ground truth was produced by two COSMIC-certified professionals ($\kappa = 0.976$) [De Vito et al. 2025]; the Zenodo and Ochodek datasets ship with per-item ground truth [De Vito et al. 2025, Ochodek et al. 2020].

Table 1. Datasets used in the evaluation.

Dataset	Format	Source	N	CFP range
De Vito	structured use cases	[De Vito et al. 2025]	119	2–27
Zenodo	isolated sentences	[De Vito et al. 2025]	2,040	0–4
Ochodek	use-case names	[Ochodek et al. 2020]	437	2–43
User Stories	user stories (Connextra)	[Dalpiaz 2018]	98	3–11

A purpose-built ground truth for user stories. The only user-story dataset with COSMIC ground truth [Ünlü et al. 2024] is unavailable, so we built our own. We sampled 100 stories from the Dalpiaz collection [Dalpiaz 2018] (22 projects, stratified by project, fixed seed) and wrote a formal annotation protocol operationalizing the manual for the format. Annotation used three annotators over two rounds: an LLM applied the protocol and produced a pre-annotation (A1); two professionals annotated independently, without access to the LLM pre-annotation so as to avoid anchoring (A2, A3); and a human adjudicator, blind to any evaluated model, resolved divergences. Inter-annotator reliability reached human-to-human κ of 0.659 (exact) and 0.705 (weighted) [Cohen 1960], and

a Fleiss $\kappa = 0.672$ for the three annotators [Fleiss 1971], which is *substantial* agreement. The AI pre-annotation agreed with each human about as much as the humans agreed with one another, supporting the assisted flow. The final ground truth has 98 items, with two excluded by a documented criterion.

An external validation set. The four datasets above are development-and-evaluation data: the pipeline was iterated while observing three of them. To assess generalization beyond these data, we add an external validation set that was genuinely unused in development, namely the official COSMIC Consortium case studies, whose measurements are produced and validated by the Measurement Practices Committee under ISO/IEC 19761 and therefore constitute a gold standard, already used as a benchmark in recent COSMIC-automation work [Laqrichi 2024]. We use the three business-domain studies in use-case form, ACME Car Hire (8 functional processes), RestoSys (26) and Course Registration (C-REG, 19) [COSMIC Measurement Practices Committee 2019], for a total of 53 processes; requirements were transcribed from each document's natural-language descriptions rather than its measurement tables, with CFP sizes checked against the official summaries. On this set we evaluate the two methods that natively measure from use-case text, the proposed symbolic pipeline and R1-COSMET, each measuring once with no tuning, under the same statistical apparatus (bootstrap CIs and a paired Wilcoxon test) and stratified by process complexity. Because these are didactic, deliberately clear documents, they should be read as a performance ceiling under clean conditions rather than a measure of field difficulty.

The heterogeneity matrix and the symmetry probe. The central instrument crosses all methods with all formats (20 cells). Competitors are frozen in their home configuration and applied to the other formats; the proposed method specializes only the M1 layer per format, with each module calibrated on training data and evaluated on held-out data, keeping M2 and M3 invariant. R3 is an exception, marked per cell: it trains *within* each dataset (cross-validation), so it is a model rebuilt per format, not a single method applied to four. This asymmetry, normalization granted to our method while competitors are frozen, is itself tested by a *symmetry probe*: we grant our normalization layer (the Connextra adapter) to competitor R1 in the format where it collapses (user stories) and re-measure with the same GPT-4-turbo engine, isolating the effect of normalization from that of the engine.

Metrics and statistics. We report MAE, MMRE, PRED(25) and SA [Shepperd and MacDonell 2012]. Uncertainty is quantified by 95% percentile bootstrap confidence intervals (10,000 resamples, fixed seed) for each cell; paired comparisons use the Wilcoxon signed-rank test over per-item absolute errors, with the Cliff's δ effect size. Family-wise type-I error across paired comparisons is controlled by Holm-Bonferroni, and all reported comparisons survive the adjustment.

Two self-detected corrections. The protocol incorporates two structural corrections detected during the research and documented in a versioned audit log. The first is *few-shot leakage* in the neural M1: six examples drawn from the evaluation system were replaced by curated synthetic ones, and the honest error proved 56.3% higher than the inflated value ($0.597 \rightarrow 0.933$ under cross-validation), so the contaminated number is reported only as a demonstration of the leakage effect. The second is an *inconsistency in the SA metric*: the symbolic evaluator computed $1 - MAE/\bar{y}$ instead of the Shepperd and

MacDonell definition [Shepperd and MacDonell 2012], which inflates values on narrow distributions; the metric was fixed at the source, all results regenerated, and the central claim reformulated from the corrected values. To guard against overfitting, a stratified hold-out of 30 use cases was separated from De Vito, with hyperparameters calibrated on the remaining 89 and a single final evaluation on the hold-out.

5. Results

Re-executions in the home format. The R2 CNN reproduces the published benchmark within 0.1% (MAE 3.475 vs. 3.479), a faithful re-implementation. R1 does not reproduce the near-perfect original (MAE 0.039 reported vs. 0.377 re-executed on the same 114 processes); the order-of-magnitude gap, attributed to the contemporary GPT-4-turbo snapshot, illustrates the reproducibility fragility of methods coupled to commercial LLMs, and the re-executed value (MAE 0.370 on 119 processes) is adopted as a realistic baseline. In the home format, the hierarchy R1 $\gg$ R3 $>$ R2 holds (Wilcoxon, $p < 0.001$ for all pairs).

The heterogeneity matrix. Table 2 reports SA for every method in every format. The proposed method holds a positive point estimate in all four formats, and bootstrap intervals place De Vito $[+0.412, +0.668]$, Zenodo $[+0.094, +0.221]$ and use-case names $[+0.019, +0.108]$ (leave-one-project-out) significantly above zero, while only User Stories (test, $n = 78$) $[-0.024, +0.492]$ is not distinguishable from zero, so there the method degrades gracefully rather than winning. In contrast, *every* off-home cell of R1 and the neural pipeline is significantly negative (for example, R1 on Zenodo $[-1.292, -0.449]$). This yields the central claim:

> Specializing only the per-format normalization layer (each module validated on held-out data) and keeping the measurement engine and explanation invariant, the framework has no significantly negative SA cell across the four formats, with significantly positive SA in three. Every competitor has all of its off-home cells significantly negative. The framework shows good performance in three formats and graceful degradation in the fourth, against the systematic collapse of the competitors.

Table 2. Heterogeneity matrix: SA per method $\times$ format. (H) = home format; (w) = trained within-dataset; N/A = format-incompatible; the second row is an ablation of the proposed method without the format adapter. The proposed method specializes normalization per format (validated on held-out data); competitors operate frozen.

Method	De Vito	Zenodo	Ochodek	User Stories
Hybrid-XAI (symbolic)	$+0.555$ (H)	$+0.161$	$+0.067$	$+0.261$
without format adapter	$+0.555$ (H)	$+0.161$	$+0.067$	-2.606
Hybrid-XAI (neural)	$+0.285$ (H)	-1.756	-0.475	-1.065
R1-COSMET (GPT-4-turbo)	$+0.824$ (H)	-0.802	-0.336	-0.230
R2-DEEP-COSMIC	-0.475	-13.126	$+0.240$ (H)	N/A
R3-HKO	$+0.190$ (w)	$+0.297$ (w)	$+0.096$ (w)	$+0.166$ (w)

Three cells need careful reading. R3 is positive in all four, but it trains *within* each dataset, so it is a model rebuilt per format from its own ground truth, not a single

method applied to four, and its ceiling (SA 0.10–0.30) is structural, a consequence of the class-median approximation. R1 on user stories shows why the matrix reports SA and not MAE. Its MAE (0.888) looks competitive, but the negative SA reveals prediction near the dataset mean, with no ordering ability beyond a trivial predictor. R2's extreme value on Zenodo (SA -13.126) is real, not a typo. A regressor trained on use-case names (mean CFP ≈ 9) predicts roughly 5–6 for every input, so on Zenodo's narrow distribution (CFP mostly ≤ 1), where the trivial baseline's error is already tiny, the error ratio explodes into a textbook domain-shift collapse.

The symmetry probe. It is fair to ask whether the observed robustness comes from the design, in which normalization is granted to our method while competitors are frozen, rather than from the architecture. Two pieces of evidence answer this. First, one competitor already receives per-format adaptation: R3 trains within-dataset and still ceilings at SA 0.10–0.30. Second, the symmetry probe grants our normalization to a competitor. Given the Connextra adapter and re-measured with the same GPT-4-turbo engine, R1 moves from collapse (SA -0.230, CI $[-0.551, -0.005]$) to SA $+0.222$ (CI $[+0.046, +0.378]$, $n = 98$), significantly positive. Normalization, not the engine, is the portable lever, since it rescues even an opaque commercial engine. The proposed method (SA $+0.364$ on the same set) still leads R1-with-adapter while preserving auditability and zero cost. The cross-format advantage is therefore attributable to the normalization layer, the contribution of this work, and not to a calibration privilege granted only to our method.

Case study: the Connextra collapse and the adapter. The user-story cell gives the most direct evidence of the architectural thesis. Without the adapter the pipeline collapses (SA $= -2.606$; 100% of stories fall back to a single Entry; ordering correlation is null). With the adapter (calibrated on 20 development stories, rules taken from the protocol), the single test evaluation ($n = 78$, frozen) yields MAE 0.500 and SA $+0.261$, with 45 of 78 exact matches; the gap to the development subset (MAE 0.300) is the expected generalization gap. Across the complete set of 98 stories, the method (MAE 0.459, SA $+0.364$) outperforms both R3-HKO trained on the format itself (MAE 0.602, SA $+0.166$) and R1-COSMET on GPT-4-turbo (MAE 0.888, SA -0.230), the latter with statistical significance (Wilcoxon $p = 8.4 \times 10^{-4}$; Cliff's $\delta = -0.232$, small). The comparison against the adapter-free pipeline quantifies the value of normalization ($p = 6.5 \times 10^{-20}$, $\delta = -0.940$, large). A roughly 200-line adapter, with rules anchored in the measurement protocol, took a collapsing pipeline to outperform the frontier LLM on user stories, with no change to the measurement engine.

External validation on canonical case studies. Table 3 reports SA on the 53 independent processes. In aggregate the symbolic pipeline scores SA $+0.386$ with a bootstrap CI of $[+0.118, +0.587]$, entirely above zero, so it performs significantly better than chance on data that took no part in development. This is direct evidence that the home-format result (De Vito, $+0.555$) is not an artifact of fitting our own datasets. R1-COSMET attains a higher point SA ($+0.621$), as on De Vito, but a paired Wilcoxon test over per-item error does not separate the two methods on this set ($p = 0.095$; Cliff's $\delta = 0.12$, negligible). This is absence of evidence of a difference rather than evidence of equivalence, which would require a formal equivalence test, and the power is limited at $n = 53$; the LLM retains the higher point estimate (we return to the per-study spread in the threats to validity). By process complexity, both methods reach low error on simple processes (CFP ≤ 4:

MAE 0.333 symbolic, 0.091 R1), the symbolic one at zero cost, and both undercount on complex processes (CFP ≥ 5: MAE 1.800 and 1.300). This is the same large-process undercounting seen on De Vito, now confirmed on independent gold-standard data as a structural limit of text-based measurement common to all methods.

Table 3. External validation on the COSMIC Consortium canonical case studies (use-case form, gold-standard ground truth). "Symbolic" is the proposed symbolic pipeline (zero cost); R1 is COSMET/GPT-4-turbo. SA with a mean baseline; 95% bootstrap CI (10,000 resamples).

Case study	n	SA symbolic [CI95]	SA R1 [CI95]
ACME Car Hire	8	$+0.784$ $[+0.500, +1.000]$	$+1.000$ $[+1.000, +1.000]$
RestoSys	26	$+0.471$ $[-0.158, +0.751]$	$+0.565$ $[+0.190, +0.772]$
C-REG	19	-0.023 $[-0.767, +0.386]$	$+0.452$ $[-0.056, +0.731]$
Aggregate	**53**	$\mathbf{+0.386}$ $[+0.118, +0.587]$	$+0.621$ $[+0.430, +0.758]$

Accuracy versus operational cost. The symbolic pipeline reaches the second-best home accuracy (SA $+0.555$, behind only R1's $+0.824$) at zero cost, on CPU, two to three orders of magnitude faster than the LLM (0.004–0.076 vs. 5–16 s per item). R1 is the only method with a monetary cost (US\$$0.062$–$0.077$ per item on a proprietary API), so only a zero-cost local method makes continuous per-change re-measurement feasible.

Explainability: faithful by construction. We compare the symbolic trace with a post-hoc attribution (LIME [Ribeiro et al. 2016] on the BERT regressor) by functional metrics, with no human subjects (eight LIME re-runs per requirement, 400 perturbations; De Vito hold-out). Because the trace makes each CFP cite the rule and the span that produced it, its *fidelity* equals the movement-level accuracy already reported (per-type F1 0.77–0.91), a check no post-hoc method allows. Table 4 gives the other two properties. The trace is deterministic, identical on re-run, whereas LIME keeps only about half its top terms across runs (Jaccard 0.505); and all of the trace's mass falls on cited excerpts, whereas LIME reaches 0.570 and *drops* to 0.449 once stopwords are removed, so its weight sits on surface tokens rather than data movements. This is the point rather than an unfair white-box advantage: a symbolic method explains by exposing its own computation, while post-hoc attribution on a black box is unstable and weakly anchored. Perceived usefulness by practitioners is left to future work.

Table 4. Functional explanation quality on the De Vito hold-out (30 processes). LIME: eight re-runs, 400 perturbations; 95% bootstrap CI.

Property	Symbolic trace	LIME (post-hoc)
Stability (Jaccard top-8)	1.000 (determ.)	0.505 [0.463, 0.549]
Stability (Spearman of weights)	1.000 (determ.)	0.684 [0.654, 0.713]
Semantic anchoring (all terms)	1.000 (by constr.)	0.570 [0.525, 0.612]
Semantic anchoring (no stopwords)	1.000 (by constr.)	0.449 [0.403, 0.491]

6. Discussion

The research question receives an affirmative but calibrated answer. The proposed method shows no significantly negative cell, whereas every frozen competitor collapses outside

its home format, and the adapter case study and symmetry probe isolate the mechanism, showing that heterogeneity is resolved by per-format normalization rather than by a single parser or a larger model. The matrix does not license a reading of strong performance everywhere. On use-case names and, to a lesser degree, isolated sentences, the positive SA is of modest magnitude and sensitive to the choice of trivial baseline, so the method's merit there is graceful degradation rather than a strong result. A per-item analysis confirms two limits common to all methods, the undercounting of large processes (GT ≥ 10 CFP) and the information ceiling of short use-case names.

On raw accuracy the frontier LLM remains superior in its home format (MAE 0.370 vs. 0.867 for our best variant), and beating it on use cases was never the goal. What the framework adds is good performance in three of four formats with a single auditable engine, a statistically significant win over the same GPT-4 on user stories (Wilcoxon $p = 8.4 \times 10^{-4}$), rule-level traceable decomposition, zero per-item cost, and local execution; its explanation is faithful, deterministic and anchored by construction, where post-hoc attribution is not (Table 4). The methodological rigor is itself a result: two structural corrections, the few-shot leakage (ΔMAE $+56.3\%$) and the SA inconsistency, were self-detected, quantified and incorporated, with the central claim reformulated from the corrected values.

6.1. Threats to Validity

Construct. R1's accuracy could reflect contamination, since the public datasets predate its training cut-off; but under memorization it should do best on the oldest sets, and instead it collapses on Ochodek and Dalpiaz while excelling on the newer De Vito, and on low-content formats part of the error reflects ground-truth granularity rather than method quality. *Internal.* The pipeline was iterated observing three of the four datasets (user stories the only unseen format, partly mitigated by the external validation), and the adapter was calibrated on 20 stories with visible ground truth (mitigated by a single frozen test on the other 78). *External.* The 53 gold-standard processes support generalization, but the positive aggregate (SA $+0.386$) is driven mainly by RestoSys, the largest study and the one we re-extracted, while on C-REG the method only matches the trivial baseline (SA -0.023), so the evidence is suggestive, not conclusive; the user stories are a single English collection and the hold-out is small ($n = 30$). *Conclusion.* Cells carry bootstrap CIs and paired tests are Holm-Bonferroni-adjusted, so the central claim rests only on the two-sided-significant cells and the paired user-story comparison ($p = 8.4 \times 10^{-4}$); the user-story SA cell is not two-sided significant ($[-0.024, +0.492]$) and is reported as graceful degradation.

7. Conclusion

We asked whether a single approach can sustain better-than-trivial COSMIC measurement across heterogeneous requirement formats. Through a five-method, four-format heterogeneity matrix with cell-by-cell statistical qualification, the answer is that it can, provided heterogeneity is treated where it belongs, in a per-format normalization layer over an invariant measurement engine. The clearest evidence is a roughly 200-line format adapter that took the symbolic pipeline from total collapse to outperforming GPT-4-turbo on user stories, with no change to the engine, together with a symmetry probe showing that the

same normalization rescues even the commercial engine. Within the limits of this evaluation, format heterogeneity is resolved by normalization, not by the measurement engine. Future work follows from the observed failure modes: decomposing large processes before measurement, extending the per-format adapters to further formats such as structured specifications and epics, as localized extensions that never touch the engine, and adapting the symbolic pipeline to languages other than English for use in production.

References

Abran, A. (2015). *Software Project Estimation: The Fundamentals for Providing High-Quality Information to Decision Makers*. John Wiley & Sons, Hoboken, NJ, USA.

Albrecht, A. J. and Gaffney, J. E. (1983). Software function, source lines of code, and development effort prediction: A software science validation. *IEEE Transactions on Software Engineering*, SE-9(6):639–648.

Bagriyanik, S. and Karahoca, A. (2016). Automated COSMIC function point measurement using a requirements engineering ontology. *Information and Software Technology*, 72:189–203.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

COSMIC (2009). *The COSMIC Functional Size Measurement Method, Version 3.0.1: Measurement Manual*. The Common Software Measurement International Consortium.

COSMIC Measurement Practices Committee (2019). COSMIC case studies: ACME car hire, RestoSys, and course registration (C-REG). cosmic-sizing.org.

Dalpiaz, F. (2018). Requirements data sets (user stories). V1. DOI: 10.17632/7zbk8zsd8y.1.

De Vito, G., Di Martino, S., Ferrucci, F., Gravino, C., and Palomba, F. (2025). LLM-based automation of COSMIC functional size measurement from use cases. *IEEE Transactions on Software Engineering*, 51(5):1500–1523.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. Software: https://spacy.io.

Hussain, I., Kosseim, L., and Ormandjieva, O. (2013). Approximation of COSMIC functional size to support early effort estimation in Agile. *Data & Knowledge Engineering*, 85:2–14.

ISO/IEC (2011). ISO/IEC 19761:2011 — software engineering — COSMIC: a functional size measurement method.

Jurkiewicz, J. and Nawrocki, J. (2015). Automated events identification in use cases. *Information and Software Technology*, 58:110–122.

Laqrichi, S. (2024). A hybrid framework for COSMIC measurement: Combining large language models with a rule-based system. In *Joint Proceedings of IWSM-Mensura 2024*, volume 3852 of *CEUR Workshop Proceedings*, Montréal, Canada.

Ochodek, M., Kopczyńska, S., and Staron, M. (2020). Deep learning model for end-to-end approximation of COSMIC functional size based on use-case names. *Information and Software Technology*, 123:106310.

Qwen Team (2025). Qwen3 technical report. arXiv:2505.09388.

Radliński, Ł. and Swacha, J. (2025). Large language models for early-stage software project estimation: A systematic mapping study. *Applied Sciences*, 15(24):13099.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Shepperd, M. and MacDonell, S. (2012). Evaluating prediction systems in software project estimation. *Information and Software Technology*, 54(8):820–827.

Ungan, E., Hammond, C. A., and Abran, A. (2018). Automated COSMIC measurement and requirement quality improvement through ScopeMaster tool. In *Proceedings of IWSM-Mensura 2018*, volume 2207 of *CEUR Workshop Proceedings*.

Ünlü, H., Tenekeci, S., Çiftçi, C., et al. (2024). Predicting software functional size using natural language processing: An exploratory case study. In *2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 188–193. IEEE.

Yokogawa, K., Hiroishi, M., Amasaki, S., Aman, H., and Yokogawa, T. (2024). A comparative study on COSMIC FP approximation with deep learning and conventional machine learning. In *2024 IEEE/ACIS 22nd International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 346–351. IEEE.

Zhang, C., Tong, S., Mo, W., Zhou, Y., Xia, Y., and Shen, B. (2016). ESSE: An early software size estimation method based on auto-extracted requirements features. In *Proceedings of the 8th Asia-Pacific Symposium on Internetware*, pages 112–115. ACM.

Zhao, Z., Jiang, H., Zhao, R., and He, B. (2024). Emergence of a novel domain expert: A generative AI-based framework for software function point analysis. In *2024 39th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 2245–2250. IEEE.