Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A small validation budget reliably selects a compact text representation for e-commerce recommendation prediction

Mujahid Shahid

mshahid1@st.knust.edu.gh

Kwame Nkrumah University of Science and Technology

Kwabena Owusu Agyemang

Kwame Nkrumah University of Science and Technology

Oliver Konyo

Kwame Nkrumah University of Science and Technology

Article

Keywords: text representation, resource-aware model selection, e-commerce reviews, TF-IDF, sentence embeddings, LightGBM

Posted Date: September 15th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10963722/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Choosing a text representation for a text-plus-tabular prediction task can cost as much computation as fitting the final model, yet representations are usually compared only after full-budget training. We tested whether a small validation subset could identify the best representation without sacrificing performance, using 22,635 women's clothing reviews to predict purchase recommendation with a fixed LightGBM classifier combining review text with age and product metadata. Across three Clothing-ID-disjoint train–validation–test splits and four candidate representations, a compact combination of TF-IDF and 128-dimensional singular-value decomposition (T128) was the outright validation winner in every repeat at every training budget, including with only 10% of the training data. Locking in this early decision cut candidate row-evaluations by 90% and cached workflow time by 42%, with no measurable loss of held-out test performance (mean macro-F1 0.815) relative to full-budget selection. An uncompressed sparse TF-IDF reference, deliberately withheld from the selectable pool, scored higher still (0.842) but took 67% longer to fit. These results show that small-scale validation trials can make trustworthy, resource-efficient representation choices in text-tabular pipelines, and that restricting the candidate pool – not the validation budget – is the more consequential design decision.

Introduction

Real-world prediction tables increasingly pair numeric and categorical fields with free-text entries such as reviews, notes, or descriptions. Benchmarking work on multimodal tabular learning has shown that this text can carry substantial predictive signal, and that comparatively simple modular pipelines – feature extraction followed by a standard tabular learner – remain competitive with more elaborate end-to-end architectures [1]. Subsequent studies have extended this evaluation to tabular foundation models, string-heavy tables, non-identically distributed splits, and alternative text pre-processing choices [2–4]. Collectively, this body of work reframes representation choice as an empirical systems problem rather than a purely linguistic one: lexical features, dense sentence embeddings, dimensionality compression, the downstream learner, and the data regime jointly determine both accuracy and cost.

Selecting among representations is rarely free. Sparse term-frequency–inverse-document-frequency (TF-IDF) features preserve fine lexical detail [6]; latent semantic compression via singular-value decomposition reduces their dimensionality while retaining most of the signal [7]; frozen sentence encoders such as Sentence-BERT provide reusable dense embeddings without task-specific fine-tuning [8]; and compact checkpoints such as all-MiniLM-L6-v2 offer 384-dimensional normalized embeddings well suited to short texts [9]. Each option carries a distinct feature-extraction, storage, and model-fitting cost, so a representation that wins an exhaustive full-budget comparison may still be the wrong practical choice if that comparison itself is expensive to repeat – across many datasets, many candidate learners, or many deployment cycles.

Resource-aware optimization offers a general answer to this cost problem. Successive Halving allocates increasing resources only to promising configurations [10], and Hyperband formalizes data samples,

training iterations, or feature counts as interchangeable, prunable resource budgets [11]. These methods are well established for hyperparameter search, but their value for representation selection rests on an empirical, not a theoretical, guarantee: a low-fidelity evaluation is useful only if it preserves the decision that a full-fidelity evaluation would have reached. That guarantee must also be checked honestly – reusing the same evidence for both selecting and assessing a model inflates apparent performance, so any credible test of low-budget selection must keep final evaluation strictly isolated from the selection process [12]. For text-representation pipelines specifically, the open, practical question is not whether subsampling can save compute in principle, but whether small-budget evidence is stable enough to trust once failed trials, reusable cached features, and the final full-data refit are all honestly counted.

We addressed this question in a concrete case study: recommendation prediction from the Women's E-Commerce Clothing Reviews dataset, in which free-text reviews are naturally paired with reviewer age and product metadata [5]. We asked whether a small validation-based trial could select a compact representation for this mixed text-tabular task without sacrificing performance relative to a full-budget search. Four compressed lexical and frozen neural candidates were compared at 10%, 25%, and 100% of the training partition, across three Clothing-ID-disjoint splits, under a locked, test-blind selection protocol. Metadata-only, uncompressed sparse TF-IDF, and native MiniLM representations were retained as fixed references but deliberately withheld from the selectable set, so that the study could separately quantify how well low-budget selection performs within a restricted candidate pool and how much overall performance that restriction itself forfeits. The central contribution is a transparent, fully auditable case-study estimate of selection stability, held-out regret, and end-to-end cost under this locked protocol. We do not propose a new encoder or search algorithm, and we intentionally limit our conclusions to the evaluated dataset, learner, candidate set, and hardware.

Methods

Study design and reproducibility protocol

The study was implemented as four linked Kaggle notebooks: data audit and splits; text representations; models and selection experiments; and final analysis. Each notebook consumed versioned upstream artifacts and emitted configuration files, row-level manifests, predictions, timing logs, checksums, and summary tables. Python 3.12.13 was used. The final experiment environment included pandas 2.3.3, NumPy 2.0.2, scikit-learn 1.6.1, LightGBM 4.6.0, sentence-transformers 5.4.1, and PyTorch 2.10.0 + cu128. Random seeds were fixed in advance.

Dataset and cleaning

We used the publicly available Women's E-Commerce Clothing Reviews dataset [13]. The source contained 23,486 rows and 11 columns. The exported index column was discarded; a traceable source-row identifier was retained. Records without both title and review text were ineligible. Exact duplicate

records were removed after excluding the exported index. Review bodies were normalized for duplicate detection; within each repeated normalized-body group, the first record was retained and later occurrences removed. This procedure removed 851 rows in total and yielded 22,635 records.

The binary target was Recommended IND (0 = no, 1 = yes). Title and Review Text were concatenated with a blank-line separator after missing values were replaced with empty strings. Age was the numeric feature. Division Name, Department Name, and Class Name were categorical features. Rating was excluded because it is a closely related evaluative outcome, and Positive Feedback Count was excluded because it is observed after the review. Clothing ID was used only for split grouping. Missing categorical values were imputed with a constant indicator during model preprocessing.

Grouped repeated splits and training budgets

Three repeated group-disjoint partitions were generated with base seeds 42, 142, and 242. Each partition assigned 705 Clothing IDs to training and 236 each to validation and test; exact row proportions varied because group sizes were unequal. The test partition was inaccessible to representation and policy selection. Within each training partition, nested group-preserving subsets targeted 10% and 25% of the training rows; the 10% groups were contained in the 25% groups, which were contained in the 100% partition. Candidate transformers and metadata preprocessing were fitted only on rows permitted by the current budget. Validation remained fixed within each repeat.

Text and metadata representations

Seven representations were registered. M0 contained metadata only. T0 added sparse TF-IDF and served as an uncompressed fixed reference. T32 and T128 applied truncated singular-value decomposition to TF-IDF with 32 and 128 components. E32 and E128 applied principal-component analysis to frozen MiniLM embeddings with 32 and 128 components. E384 used native 384-dimensional MiniLM embeddings and served as a fixed reference. The selectable candidate set was fixed as T32, T128, E32, and E128 before validation experiments began.

TF-IDF used lowercase word unigrams and bigrams, Unicode accent stripping, minimum document frequency 2, maximum document frequency 0.98, a maximum of 20,000 features, sublinear term frequency, L2 normalization, and float32 storage [6]. Truncated SVD was fitted within each training budget [7]. Metadata processing used median imputation for age, constant imputation for categorical missingness, and one-hot encoding with unknown categories ignored; this treatment was shared across all arms.

Frozen neural features were produced with sentence-transformers/all-MiniLM-L6-v2 at revision 1110a243fdf4706b3f48f1d95db1a4f5529b4d41 [8, 9]. The checkpoint maps text to normalized 384-dimensional vectors. The observed maximum input length was 168 MiniLM tokens, below the configured 256-token limit, so no record was truncated. Full-corpus encoding used batch size 256 on a Tesla T4 and

produced a 22,635 × 384 float32 matrix. PCA was fitted only on the permitted training rows for E32 and E128.

Classifier and metrics

All representations used the same LightGBM gradient-boosted decision-tree classifier [14]. The fixed configuration was binary objective, 300 estimators, learning rate 0.05, 31 leaves, unrestricted maximum depth, minimum child samples 20, unit subsampling and feature-subsampling rates, no L1 or L2 regularization, balanced class weights, four CPU threads, deterministic mode, and column-wise construction. No early stopping or hyperparameter tuning was performed. The model seed matched the repeat seed.

The decision threshold was fixed at 0.5. Macro-F1 was the primary metric because it gives equal weight to both classes under imbalance [15]. Secondary metrics were balanced accuracy, accuracy, receiver-operating-characteristic area under the curve, log loss, class-specific precision and recall, and Brier score. With three repeated splits, summaries use means, observed ranges, and paired descriptive differences; no inferential p-values are reported.

Selection policies and pre-test lock

For each repeat and budget, all four candidates were trained on the permitted training subset and evaluated on the repeat's validation partition. The raw winner had the highest validation macro-F1. Candidates within 0.01 absolute macro-F1 of that score were considered eligible; among eligible candidates, the representation with the lowest observed runtime was selected, with a fixed code order as a final deterministic tie-break. The 0.01 tolerance was a pragmatic protocol choice, not a clinically or universally established margin.

We compared fixed-10%, fixed-25%, and full-100% policies. A Successive-Halving-style policy evaluated all four candidates at 10%, retained the top two, and evaluated those at 25% before selecting one for full-data fitting [10, 11]. The schedule was deliberately small and non-geometric because the candidate space contained only four pipelines. After all validation trials, selections were written to CSV and JSON, the validation-results file was hashed, and the lock status was set to LOCKED_BEFORE_TEST_EVALUATION. Only then were locked test predictions generated. Test selection regret was defined as the difference between the test macro-F1 of the full-budget selected candidate and that of the candidate selected by the lower-budget policy within the same repeat.

Cost accounting

Observed wall-clock time was recorded for representation fitting and transformation, model fitting, and policy-level candidate evaluation. Cached workflow cost summed all trials required by a policy plus the final full-training fit. Candidate row-evaluations counted the number of training rows processed across

selection trials. MiniLM load time (4.93 s) and full-corpus encoding time (32.03 s) were reported separately, summing to 36.96 s. Shared-cache cold cost added this overhead once across all three repeats, whereas conservative independent-run cost added it to each repeat whenever a policy evaluated a MiniLM candidate. These accounting scenarios expose, rather than conceal, the effect of reusable embeddings.

Confidence and subgroup analyses

For the selected representation, confusion components and class-specific metrics were calculated separately by repeat. Maximum predicted class probability was grouped into pre-specified bins from 0.50–0.59 through 0.90–1.00 and pooled descriptively across repeated test predictions. Subgroup metrics were summarized for observed age bands and product categories only when support checks passed. Because the same source records can appear across different repeats, pooled prediction rows and subgroup estimates are not independent observations; these analyses were descriptive and were not used for selection.

Ethics and use of generative artificial intelligence

Ethical approval was not required because this study used an existing, publicly available, anonymized dataset and involved no interaction with human participants.

OpenAI ChatGPT was used to assist with methodological structuring, code drafting and debugging, interpretation checks, and manuscript language preparation. All generated code, analyses, numerical results, citations, interpretations, and manuscript

Results

Cohort and leakage-controlled partitions

The source table contained 23,486 reviews. We excluded 844 records with neither a title nor a review body, one exact duplicate record, and six later occurrences of repeated, normalized review bodies, for 851 removed rows in total. The final modelling table contained 22,635 reviews from 1,177 Clothing IDs. The recommendation target was imbalanced: 18,535 reviews (81.89%) were positive and 4,100 (18.11%) were negative. Review title and body were concatenated as text; age, division, department, and class were retained as metadata. Rating and positive-feedback count were excluded because they were, respectively, a closely related outcome and a response observed only after the review.

Three pre-specified, group-disjoint splits used seeds 42, 142, and 242. Each repeat assigned 705 Clothing IDs to training and 236 each to validation and test, yielding training sizes of 12,824, 12,790, and 13,065 reviews; validation sizes of 5,371, 5,023, and 4,729; and test sizes of 4,440, 4,822, and 4,841 (Table 1). No Clothing ID and no identical normalized review text crossed a partition boundary, and the

nested 10% and 25% training budgets preserved both group membership and class support within each repeat.

Table 1
Study cohort, split design, candidate representations, and fixed evaluation protocol across three Clothing-ID-disjoint repeats.

Component	Definition	Repeat 0	Repeat 1	Repeat 2
Training rows	*Group-disjoint training partition*	12,824	12,790	13,065
Validation rows	*Fixed selection partition*	5,371	5,023	4,729
Test rows	*Held out until lock*	4,440	4,822	4,841
Training Clothing IDs	*Grouping unit*	705	705	705
Validation Clothing IDs	*Grouping unit*	236	236	236
Test Clothing IDs	*Grouping unit*	236	236	236
10% budget rows	*Nested training subset*	1,282	1,279	1,306
25% budget rows	*Nested training subset*	3,206	3,198	3,266
Candidate set	*T32, T128, E32, E128*	Fixed	Fixed	Fixed
Primary metric	*Macro-F1 at threshold 0.5*	Fixed	Fixed	Fixed

Representation learning curves and selection stability

Across the four selectable representations, T128 (TF-IDF compressed to 128 dimensions by truncated singular-value decomposition) had the highest mean validation macro-F1 at every budget: 0.752 at 10%, 0.782 at 25%, and 0.819 at 100% of the training partition. The corresponding means were 0.733, 0.766, and 0.793 for T32; 0.717, 0.747, and 0.769 for E32; and 0.712, 0.747, and 0.791 for E128. Performance improved with training budget for every candidate, but the ranking among candidates was already stable at the smallest budget tested (Fig. 1).

T128 was the outright winner in all nine repeat-by-budget comparisons and was the only candidate within the pre-specified 0.01 macro-F1 selection tolerance of the winner at every budget. At 10% of the training data, its margin over the best alternative was 0.0293, 0.0133, and 0.0154 macro-F1 across the three repeats – a minimum margin of 0.0033 above the tolerance. Consequently, the fixed-10%, fixed-25%, full-100%, and Successive-Halving policies all selected T128 in every repeat. Each decision was written to a machine-readable lock file before any test-set evaluation occurred. Relative to full-budget selection, every lower-cost policy therefore achieved 100% agreement and zero observed validation-based selection regret.

Held-out representation performance

The locked T128 model achieved a mean test macro-F1 of 0.8151 across the three repeats (observed range 0.8056–0.8225). Validation-to-test change was small: the mean signed gap was − 0.0033, the mean absolute gap was 0.0078, and the maximum absolute gap was 0.0085. T128 outperformed every other compressed or dense candidate in all three paired test comparisons, with a mean advantage of 0.0213 over T32, 0.0232 over E128, 0.0459 over E32, and 0.0196 over the native E384 reference. Because only three repeated splits were available, we report these as descriptive paired differences rather than inferential statistics.

The strongest representation overall was the fixed reference T0, uncompressed sparse TF-IDF, with mean macro-F1 0.8419 (range 0.8296–0.8508) – 0.0268 higher than T128 on average (range 0.0240–0.0283). Metadata alone was weak (mean 0.4585), confirming that review text carried most of the predictive signal. Validation and test rankings agreed closely: repeat-level Spearman rank correlations across all seven representations were 0.893, 0.964, and 1.000, although the complete seven-way rank order matched exactly in only one of the three repeats (Fig. 4). Table 2 separates the deployable selection result (T128) from the fixed-reference comparison (T0, E384, M0).

Table 2
Held-out test macro-F1 and observed final-fit cost across all seven representations. Values are means over three group-disjoint repeats; parentheses show observed ranges (n = 3). T0, E384, and M0 were fixed references and were not eligible for selection.

Code	Representation	Role	Test macro-F1 (range)	Final fit (s)	Δ vs T128
T0	Sparse TF-IDF	*Reference*	0.8419 (0.8296–0.8508)	21.91	−0.0268
T128	TF-IDF + SVD-128	*Selected candidate*	0.8151 (0.8056–0.8225)	7.16	0 (reference)
E384	Native MiniLM-384	*Reference*	0.7955 (0.7848–0.8045)	10.24	+ 0.0196
T32	TF-IDF + SVD-32	*Candidate*	0.7939 (0.7908–0.7987)	3.48	+ 0.0213
E128	MiniLM + PCA-128	*Candidate*	0.7920 (0.7821–0.7971)	4.71	+ 0.0232
E32	MiniLM + PCA-32	*Candidate*	0.7693 (0.7596–0.7804)	1.96	+ 0.0459
M0	Metadata only	*Reference*	0.4585 (0.4443–0.4687)	0.33	+ 0.3566

Selection cost and practical trade-offs

Under cached-feature accounting, mean workflow time (candidate selection plus the locked final fit) was 14.55 s for fixed 10%, 17.54 s for fixed 25%, 20.53 s for Successive Halving, and 24.91 s for full-budget selection. Fixed 10% therefore cut cached workflow time by 41.6% and candidate row-evaluations by

90.0% relative to full selection; fixed 25% cut time by 29.6% and row-evaluations by 75.0%; and Successive Halving cut time by 17.6% and row-evaluations by 77.5% (Fig. 3).

Accounting for the one-time cost of loading and encoding with MiniLM (36.96 s on a Tesla T4) changed the magnitude of the savings but not their ordering. When this overhead was amortized once across the three repeats, fixed 10% required 80.62 s in total versus 111.69 s for full selection – a 27.8% reduction; under a conservative accounting that charged the overhead independently to every repeat, the mean reduction for fixed 10% was 16.7%. Finally, T128's mean final-fit time was 7.16 s versus 21.91 s for T0, a 67.3% reduction, but T0 retained its 0.0268 mean macro-F1 advantage. The overall quality-cost frontier therefore depends on whether the goal is to select efficiently among compact candidates or to maximize predictive quality irrespective of feature sparsity (Fig. 2).

Error profile of the selected representation

At the fixed 0.5 decision threshold, T128 had a mean negative-class recall of 0.7757 (range 0.7520–0.7880) and a mean positive-class recall of 0.9082 (range 0.9049–0.9115), with mean balanced accuracy 0.8420 and mean accuracy 0.8851. The predicted-positive proportion was 78.94%, somewhat lower than the observed 81.89% positive prevalence, consistent with the balanced class weighting used during training. Confidence stratification was descriptive rather than causal: pooled accuracy rose from 0.543 among predictions with a maximum class probability of 0.50–0.59 to 0.977 among predictions at 0.90–1.00, and the highest-confidence bin alone comprised 66.34% of repeated test predictions. This pattern suggests that an uncertainty-aware review workflow could concentrate human attention on a comparatively small, genuinely ambiguous subset of reviews, although we did not test this prospectively.

Discussion

This case study shows that a small validation budget was sufficient to identify the same compact text representation as an exhaustive, full-budget search. T128 won at every budget and every repeat, remained the unique candidate within the selection tolerance, generalized from validation to test with only small drift, and produced zero observed test-set selection regret among the four eligible candidates. The efficiency gain was largest for the most aggressive policy: fixed 10% selection cut candidate row-evaluations by 90.0%, cached workflow time by 41.6%, and total time by 27.8% once one-time neural feature extraction was amortized across repeats. This is not evidence for a new search algorithm; it is direct evidence that, in a mixed text-tabular task with well-separated candidate rankings, low-fidelity validation can be decision-preserving – the exact property that resource-aware selection methods assume but rarely verify empirically against a locked, honest test set.

The results also expose an important boundary that a narrower reading of the headline numbers would miss. T0, the uncompressed sparse TF-IDF representation, outperformed T128 on every test repeat by approximately 0.024–0.028 macro-F1. Because T0 was deliberately excluded from the selectable set, the low-budget selector achieved zero regret while still missing the best-performing representation available

overall. This distinction matters for how the result should be used: the study demonstrates that 10% of the training data was adequate for choosing among four compressed candidates, not that T128 was globally optimal, and the two claims should not be conflated. In deployments where high-dimensional sparse features are acceptable and predictive quality dominates fitting time, T0 is the better default; where compact, dense inputs, lower final-fit time, or downstream storage constraints matter, T128 offers a well-characterized and cheaper alternative with a known, small quality cost.

The lexical advantage of T0 and T128 over the MiniLM-based candidates is plausible for recommendation-bearing review text. Explicit evaluative language, negation, fit descriptions, and product-specific phrasing may be retained directly by word and bigram features, whereas a generic, frozen sentence embedding trained for broad semantic similarity may smooth over exactly the task-specific lexical cues that drive this label. Compression to 128 dimensions preserved substantially more of the TF-IDF signal than compression to 32 dimensions, whereas reducing MiniLM from 384 to 128 dimensions cost little on average and reducing it further to 32 dimensions was noticeably more damaging. These patterns are consistent with broader evidence that text pre-processing and the choice of downstream learner materially shape performance on tabular-with-text tasks [2–4], and they should not be read as a general claim that neural encoders are inferior: only one frozen checkpoint, one classical learner, one task, and two compression levels were tested here.

Several design choices strengthen confidence in these findings. The Clothing-ID-disjoint grouping prevented products with many reviews from crossing partitions and inflating apparent generalization. The pre-test lock separated the selection decision from the final assessment, directly addressing a well-documented source of selection bias in model comparison [12]. Reporting all split-level outcomes, cost assumptions, predictions, and file checksums makes the result independently auditable. At the same time, three repeated splits are too few for reliable inferential statistics, so we report observed ranges and descriptive paired differences rather than p-values, and the three repeats should not be treated as statistically independent samples.

Several limitations bound the generality of these conclusions. The dataset represents a single retailer, a single historical period, and a heavily positive target distribution. Predicting a recommendation from an already-written review is a retrospective language-understanding task, not a forward model of future purchasing behaviour. Reported timings are specific to a Kaggle environment with four CPU threads and a Tesla T4 GPU used for embedding extraction. The classifier and decision threshold were fixed throughout, so an alternative learner, calibration procedure, or threshold-selection method could change the observed ordering. Partitions were random and group-disjoint rather than temporal or cross-retailer, and the confidence-stratified analyses were descriptive, based on repeated – not independent – observations. Finally, the selectable candidate set contained only four representations; a substantially larger candidate space could make low-budget rankings less stable and change the relative value of Successive Halving.

The most informative next step is external replication across several text-rich tabular datasets with pre-specified grouping or temporal splits, including sparse high-dimensional references inside the selectable set itself, more than one downstream learner, additional frozen encoders, and uncertainty intervals estimated from independent groups or datasets. Within the boundaries established here, however, the result supports a concrete and immediately usable workflow: run a small, leakage-controlled validation screen; confirm that the winning margin exceeds a pre-specified tolerance; lock the decision before any test-set access; and fit the selected representation once on the full training partition. When validation margins are narrow, or candidate costs differ sharply, expanding the validation budget or retaining an adaptive elimination schedule such as Successive Halving remains the safer choice.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Mujahid Shahid at mshahid1@st.knust.edu.gh.

Funding

This research received no external funding.

Author Contribution

M.S. conceived the study, designed the methodology, developed the software, conducted the experiments and formal analysis, curated the data and artifacts, created the visualizations, and wrote the original manuscript draft. K.O.A and O.K supervised the study, reviewed the methodology and interpretation. All authors approved the final manuscript.

Data Availability

The source Women's E-Commerce Clothing Reviews dataset is publicly available from Kaggle (https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews).

Code availability

The four executable Kaggle notebooks and their saved outputs are available in the public collection at https://www.kaggle.com/work/collections/19100771. The collection contains data-cleaning and split manifests, representation-generation code, locked validation and test experiments, and final analysis artifacts.

References

1. Shi, X., Mueller, J., Erickson, N., Li, M. & Smola, A. J. Benchmarking multimodal AutoML for tabular data with text fields. In Adv. Neural Inf. Process. Syst., Datasets and Benchmarks Track. (2021). https://doi.org/10.48550/arXiv.2111.02705
2. Mráz, M., Das, B., Gupta, A., Purucker, L. & Hutter, F. Towards benchmarking foundation models for tabular data with text. *Preprint at.* https://doi.org/10.48550/arXiv.2507.07829 (2025).
3. Blayer, G. et al. STRABLE: benchmarking tabular machine learning with strings. *Preprint in* (2026).
4. Purucker, L. et al. Beyond IID: how general are tabular foundation models. *really? Preprint at.* https://doi.org/10.48550/arXiv.2606.30410 (2026).
5. Agarap, A. F. Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network (RNN). Preprint at (2020). https://doi.org/10.48550/arXiv.1805.03687
6. Salton, G. & Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag*. **24**, 513–523 (1988).
7. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* **41**, 391–407 (1990).
8. Reimers, N., Gurevych, I. & Sentence -BERT: sentence embeddings using Siamese BERT-networks. In Proc. 2019 Conf. Empirical Methods in Natural Language Processing 3982–3992. (2019). https://doi.org/10.18653/v1/D19-1410
9. Sentence Transformers. all-MiniLM-L6-v2 model card. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (accessed 7 September 2026).
10. Jamieson, K. & Talwalkar, A. Non-stochastic best arm identification and hyperparameter optimization. Proc. Mach. Learn. Res. 51, 240–248 (2016).
11. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. & Talwalkar, A. Hyperband: a novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **18**, 1–52 (2018).
12. Cawley, G. C. & Talbot, N. L. C. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* **11**, 2079–2107 (2010).
13. Kaggle. Women's E-Commerce Clothing Reviews. https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews
14. Ke, G. et al. LightGBM: a highly efficient gradient boosting decision tree. In *Adv. Neural Inf. Process. Syst.* 30 (2017).

15. Sokolova, M. & Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag*. **45**, 427–437 (2009).

Figures

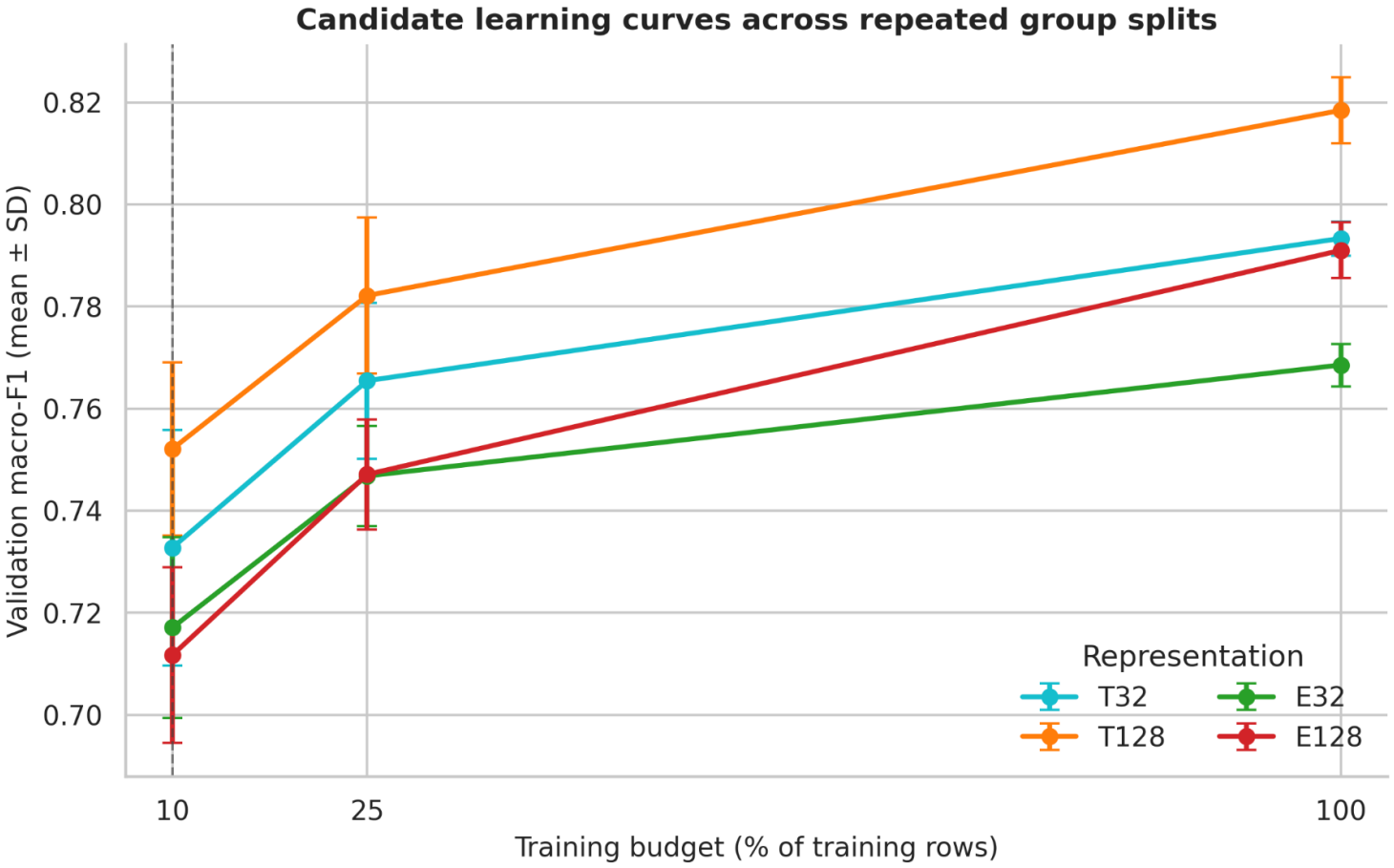


Figure 1

Candidate learning curves across nested training budgets. Mean validation macro-F1 for T32, T128, E32, and E128 is shown at 10%, 25%, and 100% of each repeat's training partition. Error bars show the observed range across the three Clothing-ID-disjoint repeats. T128 (orange) ranked first at every budget and every repeat, and its margin over the next-best candidate already exceeded the pre-specified 0.01 selection tolerance at the smallest (10%) budget.

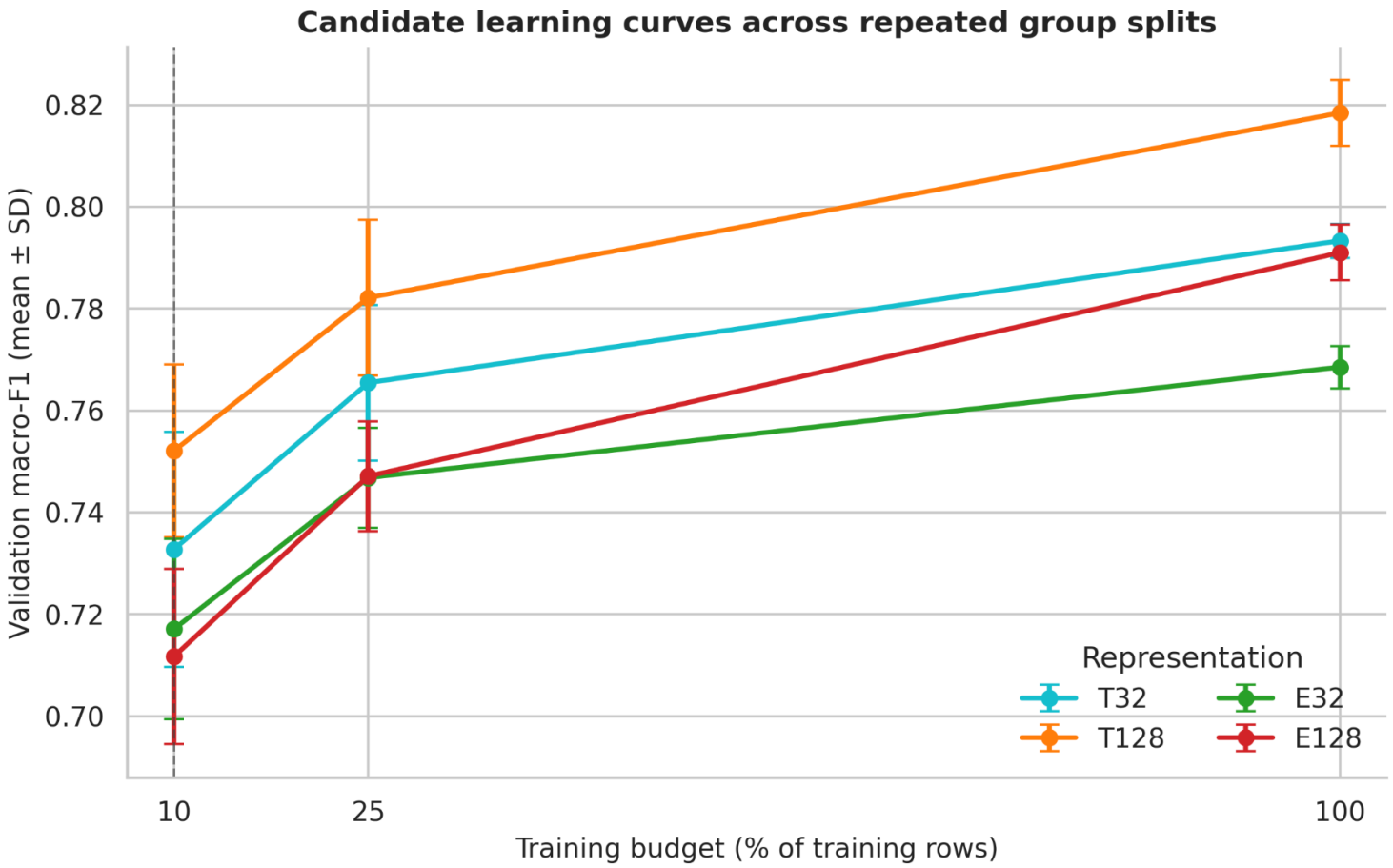


Figure 1

Candidate learning curves across nested training budgets. Mean validation macro-F1 for T32, T128, E32, and E128 is shown at 10%, 25%, and 100% of each repeat's training partition. Error bars show the observed range across the three Clothing-ID-disjoint repeats. T128 (orange) ranked first at every budget and every repeat, and its margin over the next-best candidate already exceeded the pre-specified 0.01 selection tolerance at the smallest (10%) budget.

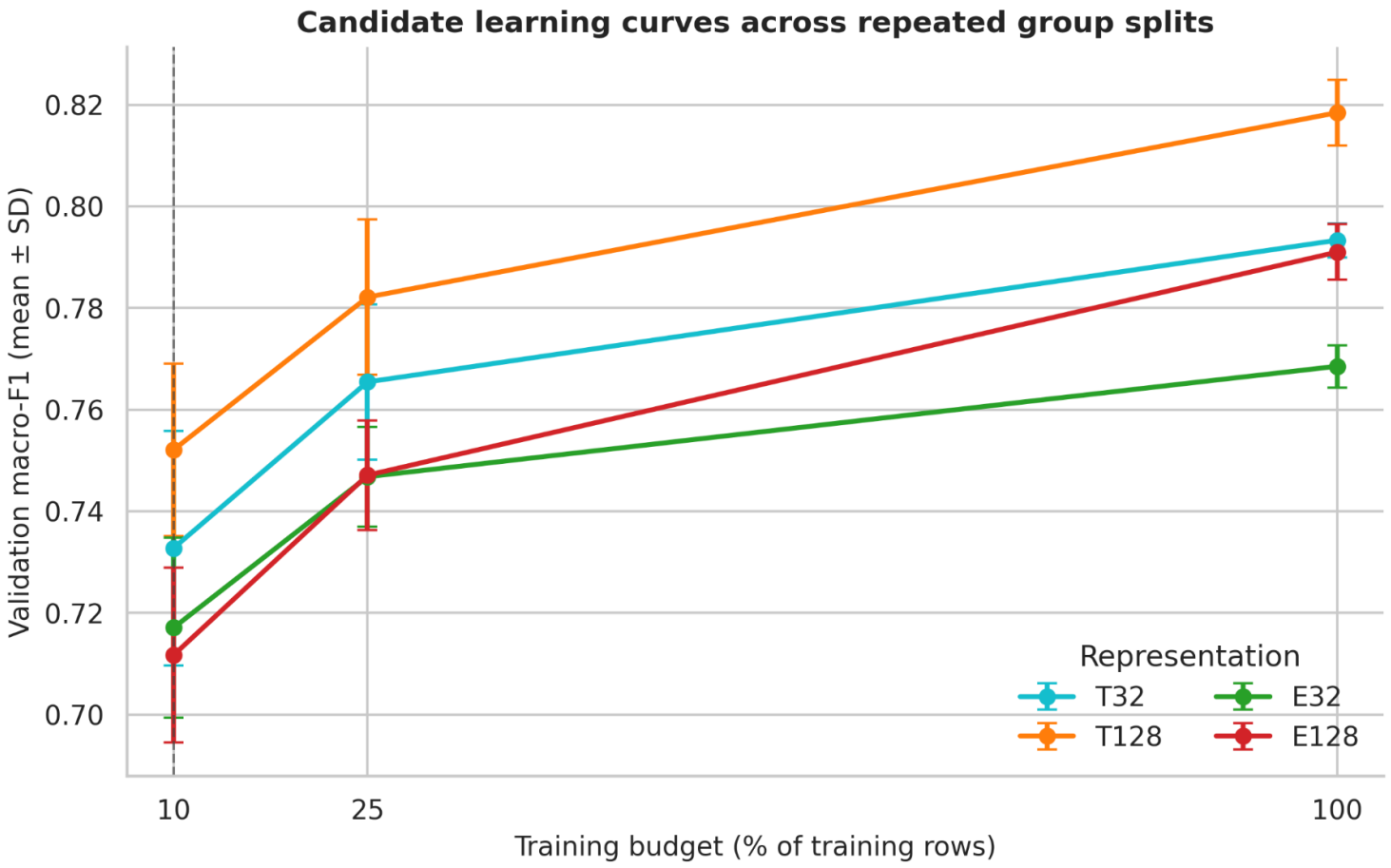


Figure 1

Candidate learning curves across nested training budgets. Mean validation macro-F1 for T32, T128, E32, and E128 is shown at 10%, 25%, and 100% of each repeat's training partition. Error bars show the observed range across the three Clothing-ID-disjoint repeats. T128 (orange) ranked first at every budget and every repeat, and its margin over the next-best candidate already exceeded the pre-specified 0.01 selection tolerance at the smallest (10%) budget.

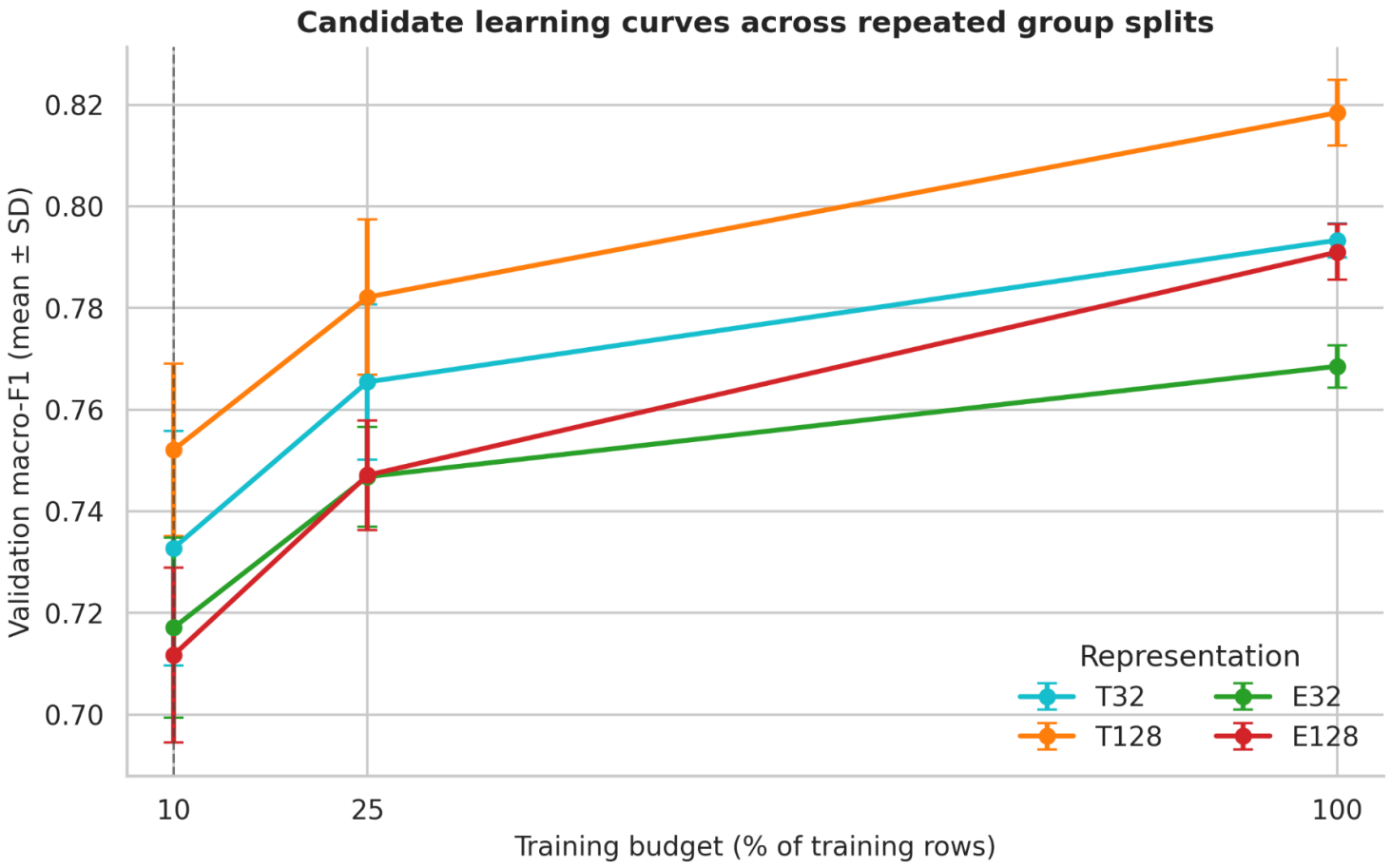


Figure 1

Candidate learning curves across nested training budgets. Mean validation macro-F1 for T32, T128, E32, and E128 is shown at 10%, 25%, and 100% of each repeat's training partition. Error bars show the observed range across the three Clothing-ID-disjoint repeats. T128 (orange) ranked first at every budget and every repeat, and its margin over the next-best candidate already exceeded the pre-specified 0.01 selection tolerance at the smallest (10%) budget.

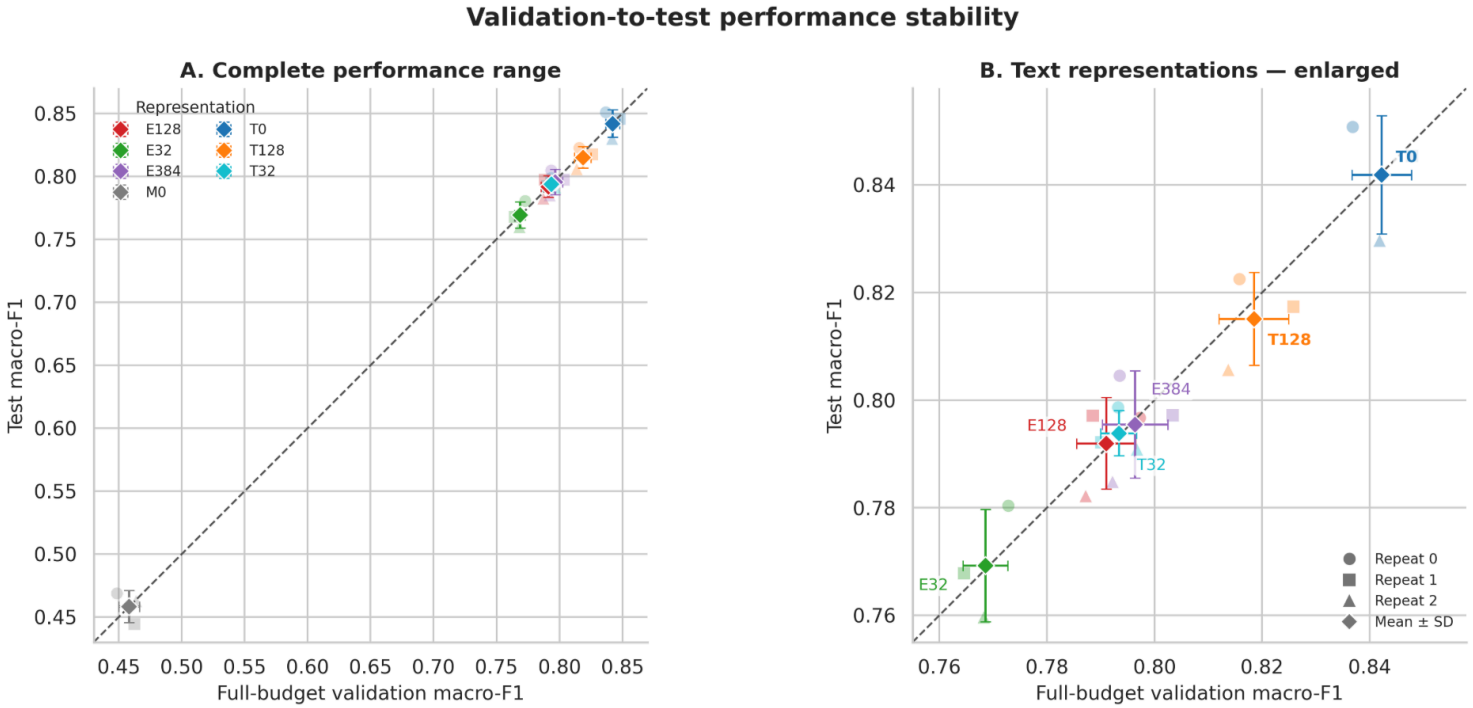


Figure 2

Held-out representation quality versus observed final-fit time. Each point summarizes mean test macro-F1 and mean final-fit time (log-scaled, seconds) across the three repeats; the black outline marks representations on the cold-cost Pareto frontier, circles denote selection candidates, and squares denote fixed references. T0 (uncompressed TF-IDF) gives the highest predictive quality, whereas T128, the selected compact representation, achieves a comparable position on the frontier at a fraction of the final-fit time.

Validation-to-test performance stability
A. Complete performance range
B. Text representations — enlarged
Representation
E128
E32
E384
M0
T0
T128
T32
Test macro-F1
Full-budget validation macro-F1
Repeat 0
Repeat 1
Repeat 2
Mean ± SD

Figure 2

Held-out representation quality versus observed final-fit time. Each point summarizes mean test macro-F1 and mean final-fit time (log-scaled, seconds) across the three repeats; the black outline marks representations on the cold-cost Pareto frontier, circles denote selection candidates, and squares denote fixed references. T0 (uncompressed TF-IDF) gives the highest predictive quality, whereas T128, the selected compact representation, achieves a comparable position on the frontier at a fraction of the final-fit time.

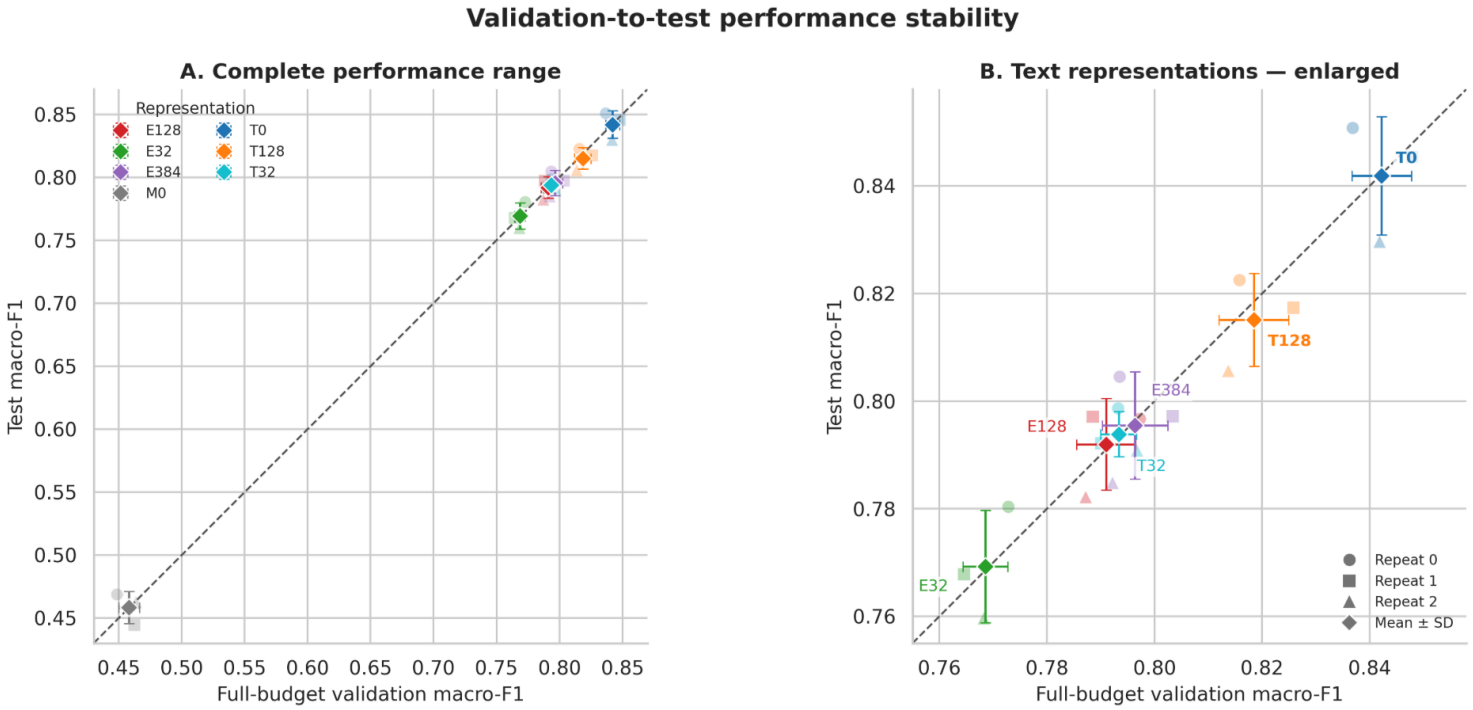


Figure 2

Held-out representation quality versus observed final-fit time. Each point summarizes mean test macro-F1 and mean final-fit time (log-scaled, seconds) across the three repeats; the black outline marks representations on the cold-cost Pareto frontier, circles denote selection candidates, and squares denote fixed references. T0 (uncompressed TF-IDF) gives the highest predictive quality, whereas T128, the selected compact representation, achieves a comparable position on the frontier at a fraction of the final-fit time.

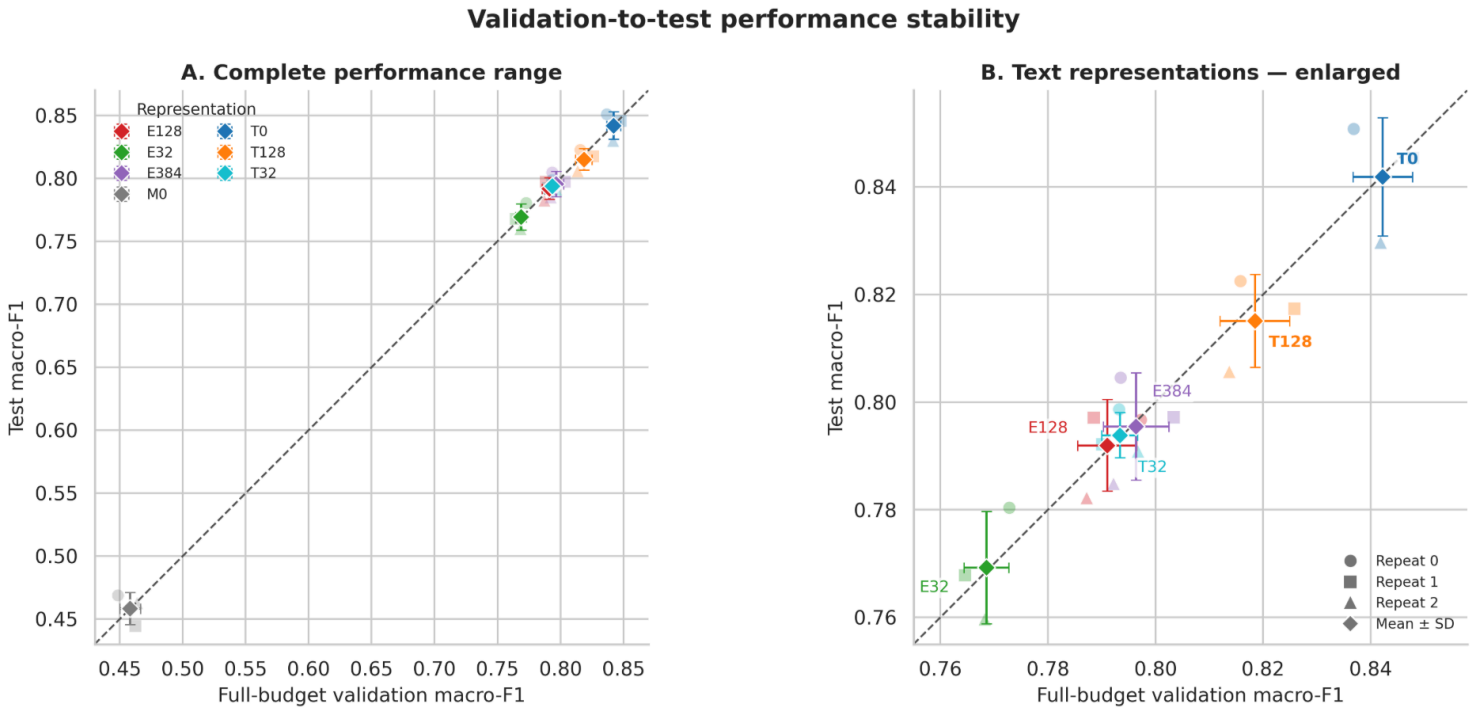


Figure 2

Held-out representation quality versus observed final-fit time. Each point summarizes mean test macro-F1 and mean final-fit time (log-scaled, seconds) across the three repeats; the black outline marks representations on the cold-cost Pareto frontier, circles denote selection candidates, and squares denote fixed references. T0 (uncompressed TF-IDF) gives the highest predictive quality, whereas T128, the selected compact representation, achieves a comparable position on the frontier at a fraction of the final-fit time.

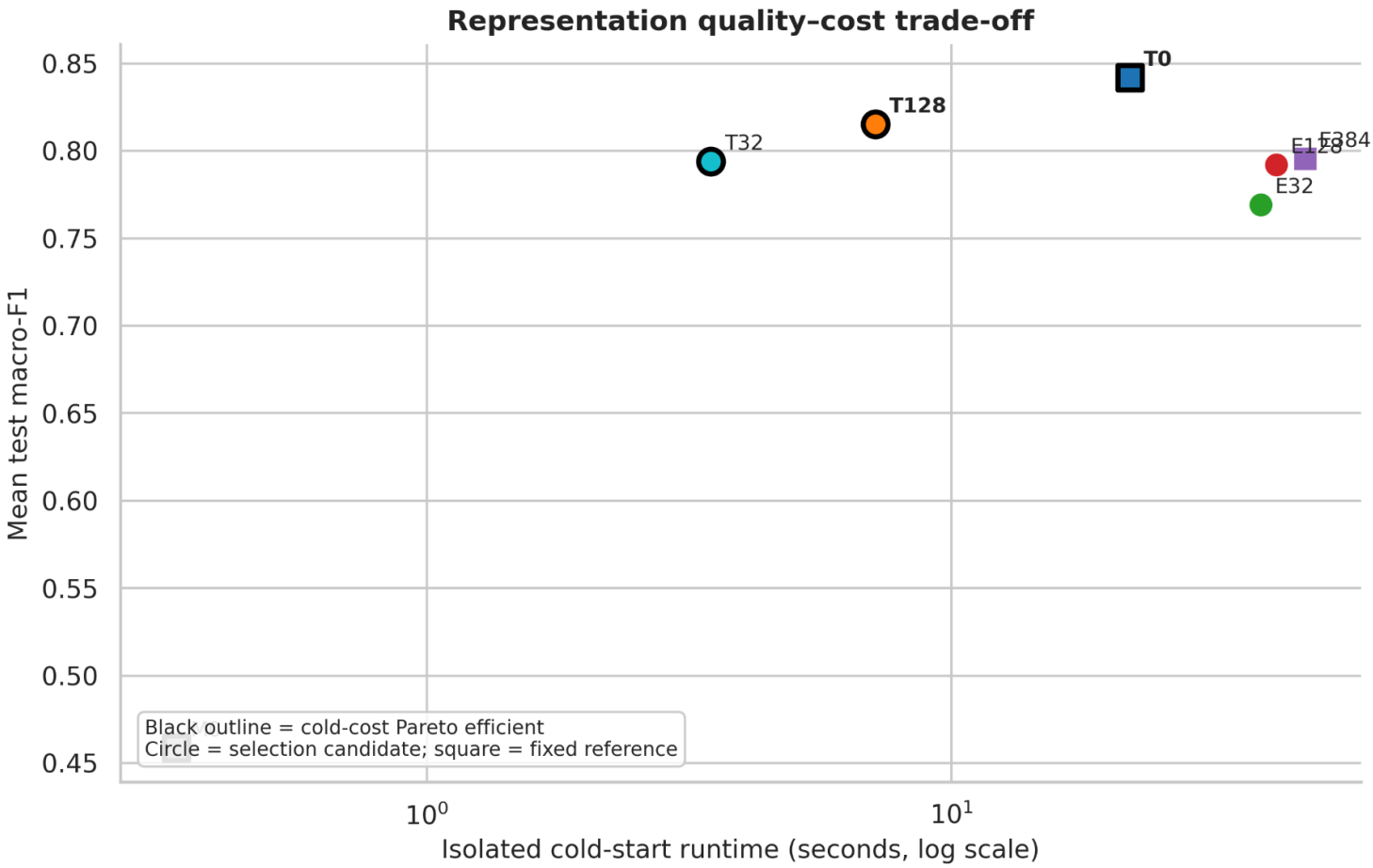


Figure 3

Policy-level cost comparison. Left: mean end-to-end workflow time per repeat under cached-feature and shared-cache cold-start accounting for fixed 10%, fixed 25%, Successive Halving, and full 100% candidate selection. Right: percentage reduction in cached time, cold time, and candidate row-evaluations for each lower-cost policy relative to full-budget selection. All policies selected T128 in all three repeats, so observed held-out selection regret was zero within the four-candidate selectable set.

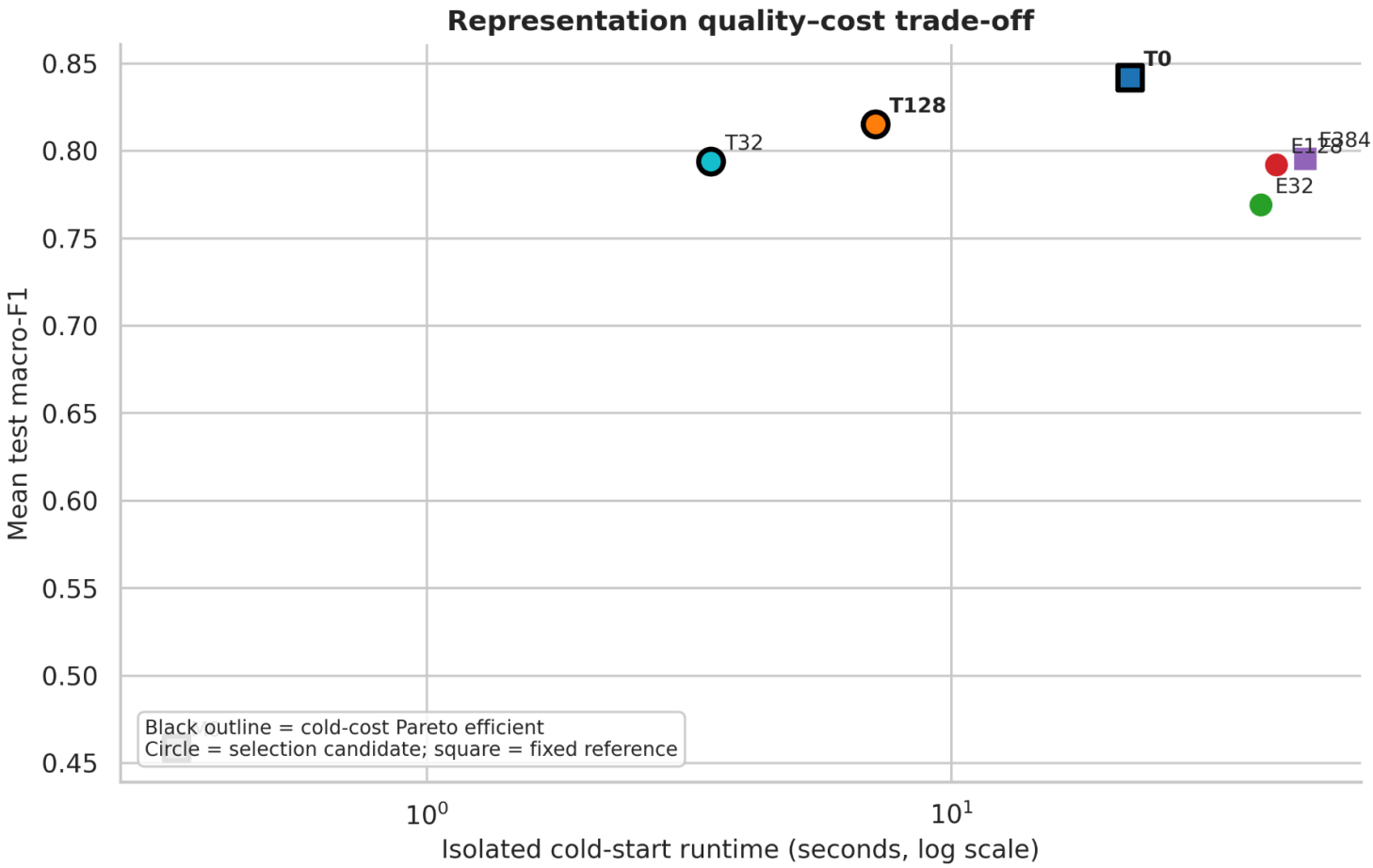


Figure 3

Policy-level cost comparison. Left: mean end-to-end workflow time per repeat under cached-feature and shared-cache cold-start accounting for fixed 10%, fixed 25%, Successive Halving, and full 100% candidate selection. Right: percentage reduction in cached time, cold time, and candidate row-evaluations for each lower-cost policy relative to full-budget selection. All policies selected T128 in all three repeats, so observed held-out selection regret was zero within the four-candidate selectable set.

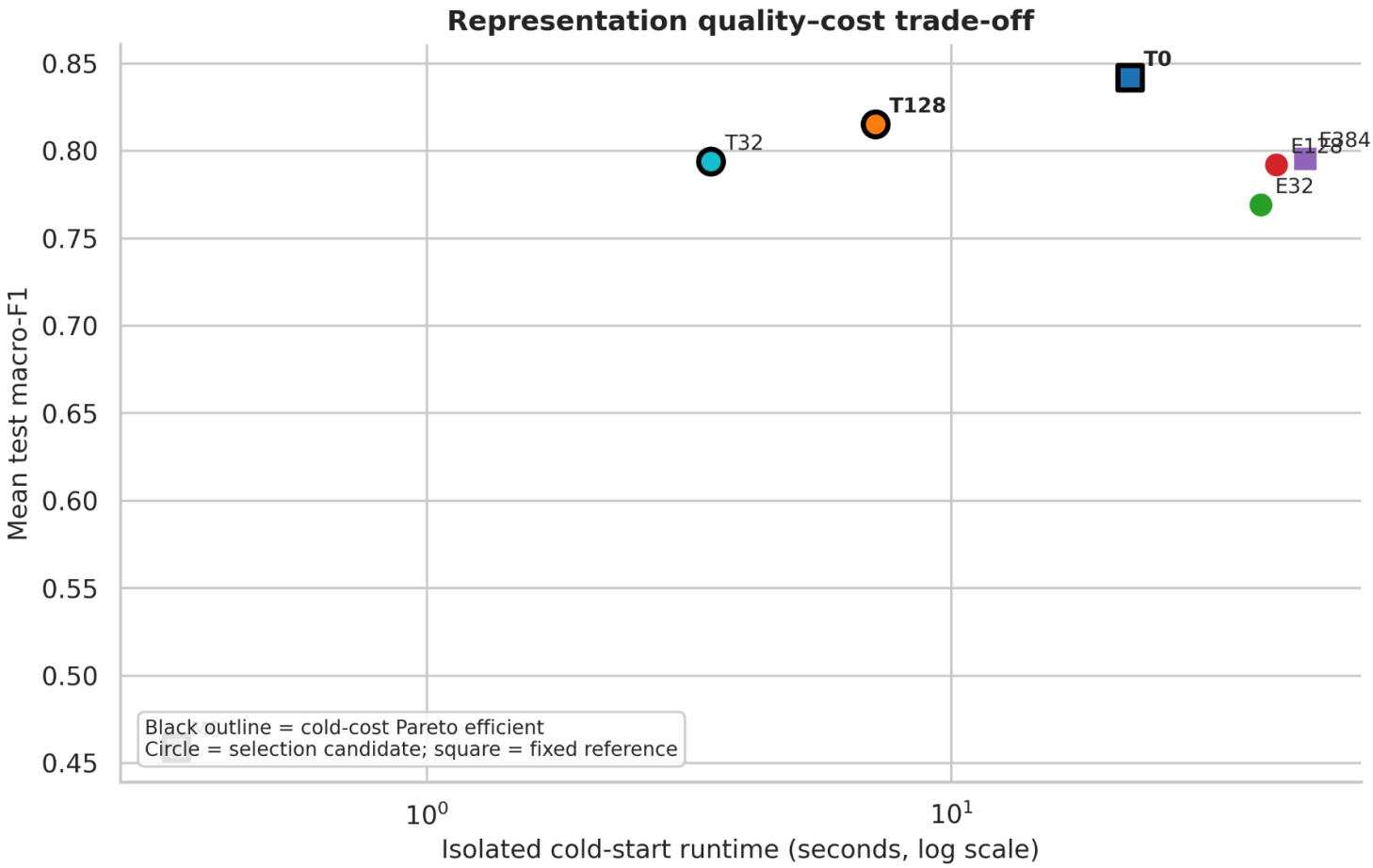


Figure 3

Policy-level cost comparison. Left: mean end-to-end workflow time per repeat under cached-feature and shared-cache cold-start accounting for fixed 10%, fixed 25%, Successive Halving, and full 100% candidate selection. Right: percentage reduction in cached time, cold time, and candidate row-evaluations for each lower-cost policy relative to full-budget selection. All policies selected T128 in all three repeats, so observed held-out selection regret was zero within the four-candidate selectable set.

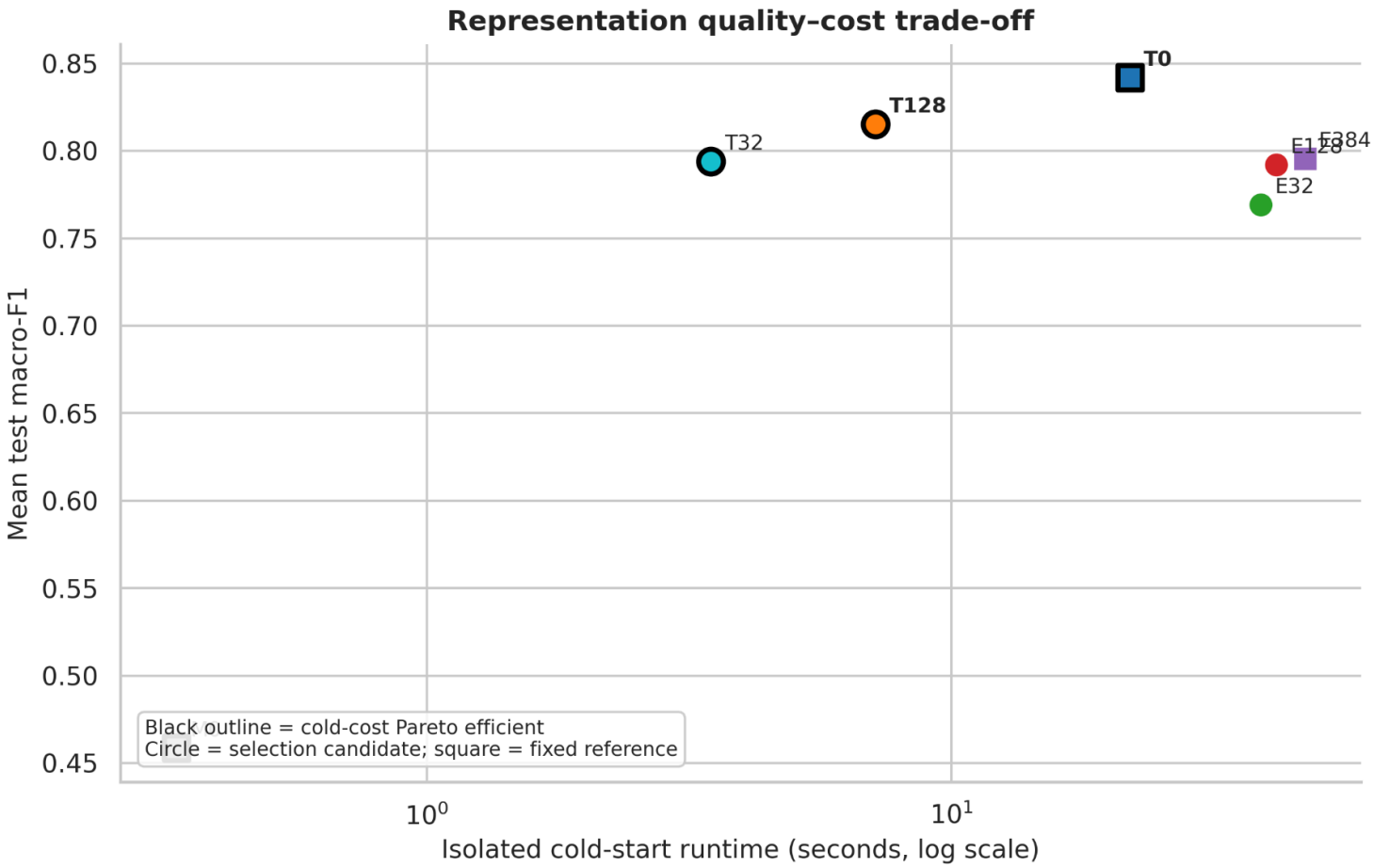


Figure 3

Policy-level cost comparison. Left: mean end-to-end workflow time per repeat under cached-feature and shared-cache cold-start accounting for fixed 10%, fixed 25%, Successive Halving, and full 100% candidate selection. Right: percentage reduction in cached time, cold time, and candidate row-evaluations for each lower-cost policy relative to full-budget selection. All policies selected T128 in all three repeats, so observed held-out selection regret was zero within the four-candidate selectable set.

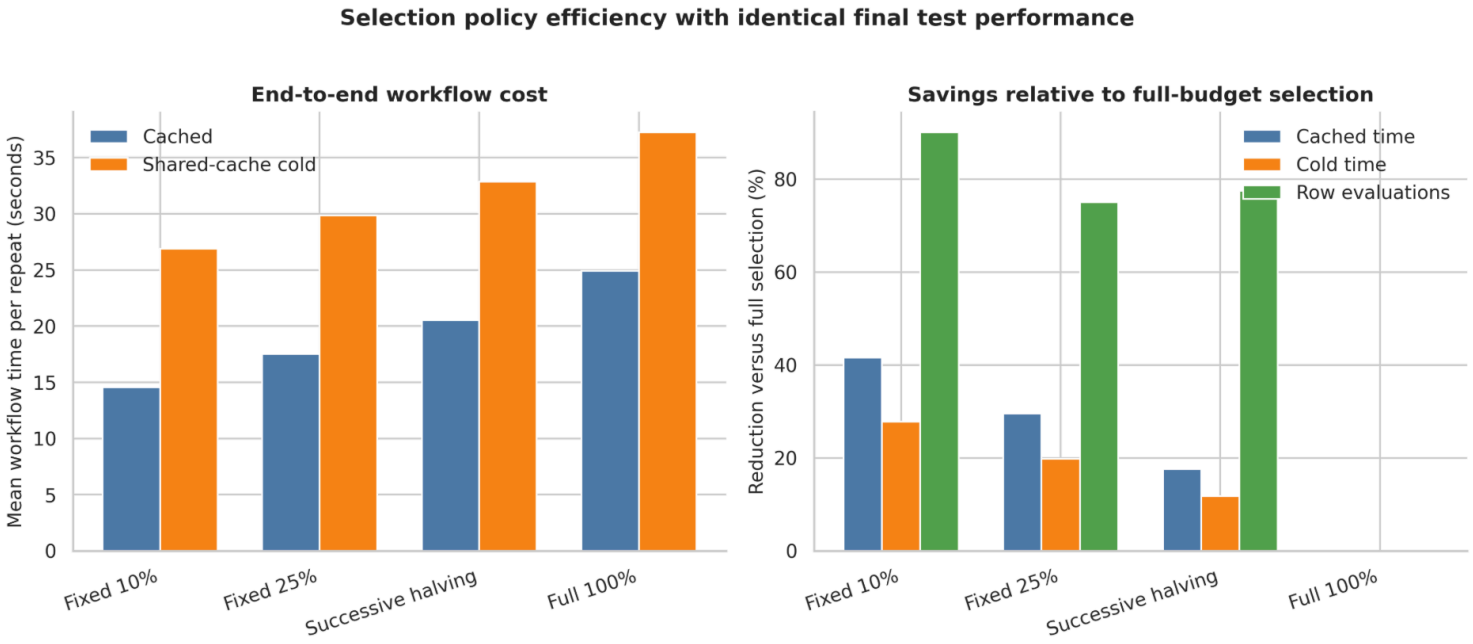


Figure 4

Validation-to-test performance stability. (A) Full performance range and (B) an enlarged view of the six text- and embedding-based representations, plotting mean test macro-F1 against mean full-budget validation macro-F1 for each representation, with repeat-specific markers, means ± s.d., and the identity line. Points close to the identity line indicate small validation-to-test drift. Repeat-level Spearman rank correlations across all seven representations were 0.893, 0.964, and 1.000 for repeats 0, 1, and 2, respectively.

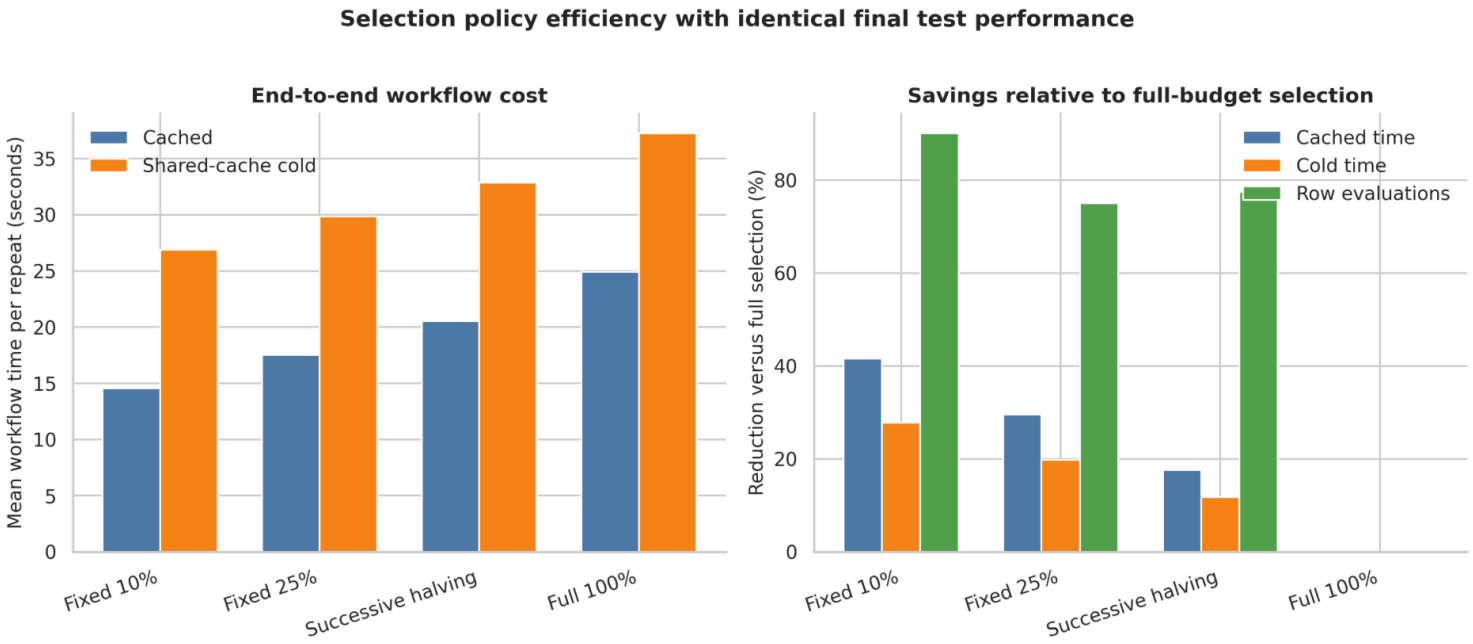


Figure 4

Validation-to-test performance stability. (A) Full performance range and (B) an enlarged view of the six text- and embedding-based representations, plotting mean test macro-F1 against mean full-budget validation macro-F1 for each representation, with repeat-specific markers, means ± s.d., and the identity line. Points close to the identity line indicate small validation-to-test drift. Repeat-level Spearman rank correlations across all seven representations were 0.893, 0.964, and 1.000 for repeats 0, 1, and 2, respectively.

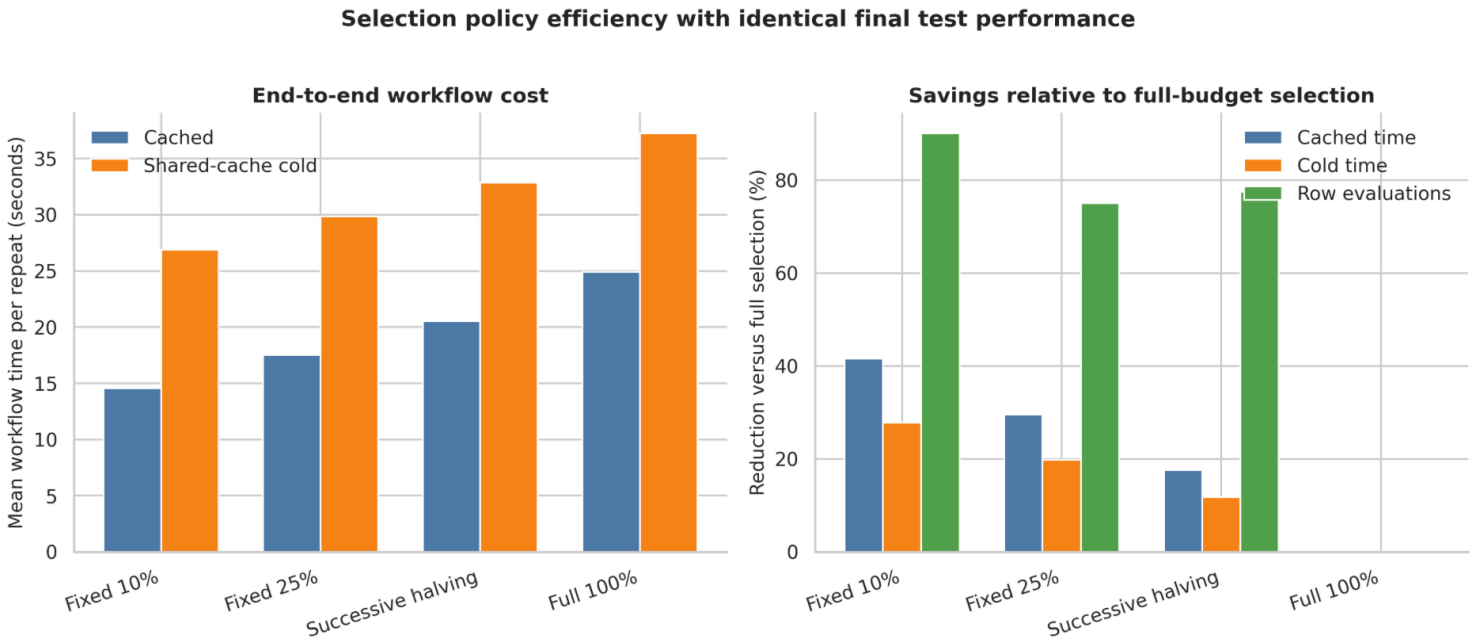


Figure 4

Validation-to-test performance stability. (A) Full performance range and (B) an enlarged view of the six text- and embedding-based representations, plotting mean test macro-F1 against mean full-budget validation macro-F1 for each representation, with repeat-specific markers, means ± s.d., and the identity line. Points close to the identity line indicate small validation-to-test drift. Repeat-level Spearman rank correlations across all seven representations were 0.893, 0.964, and 1.000 for repeats 0, 1, and 2, respectively.

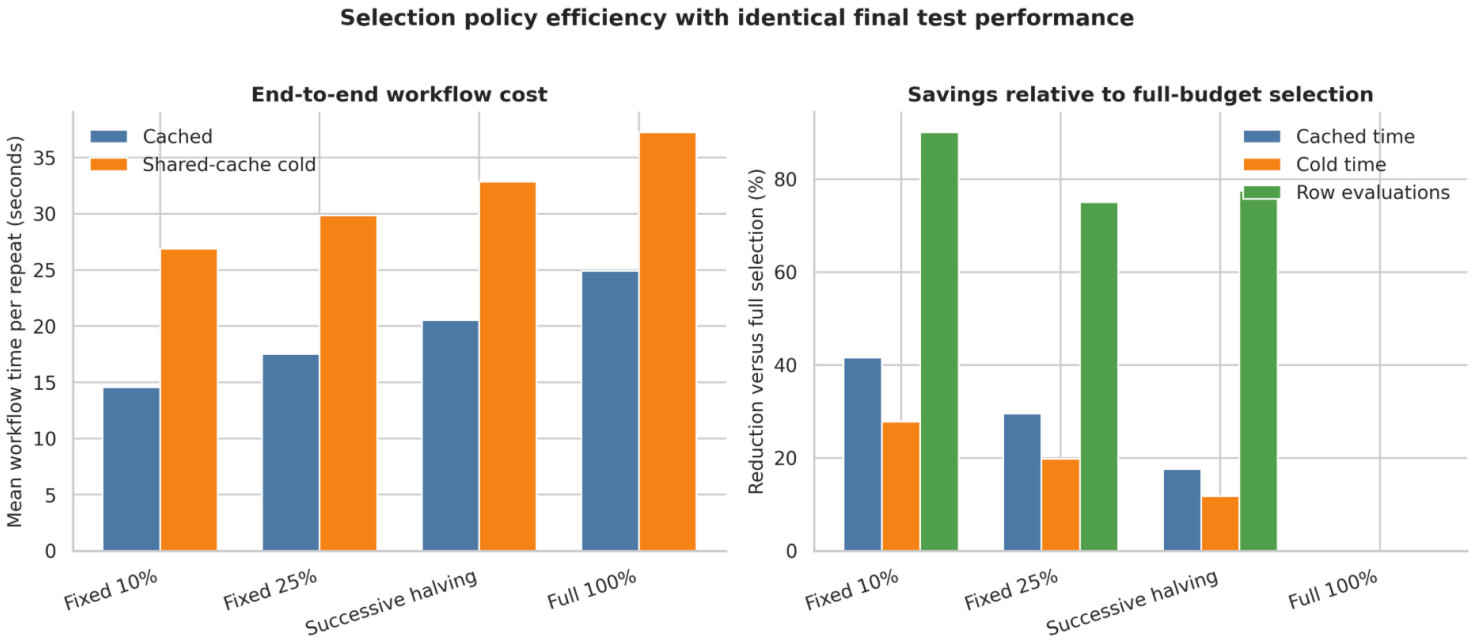


Figure 4

Validation-to-test performance stability. (A) Full performance range and (B) an enlarged view of the six text- and embedding-based representations, plotting mean test macro-F1 against mean full-budget validation macro-F1 for each representation, with repeat-specific markers, means ± s.d., and the identity line. Points close to the identity line indicate small validation-to-test drift. Repeat-level Spearman rank correlations across all seven representations were 0.893, 0.964, and 1.000 for repeats 0, 1, and 2, respectively.