Artificial Intelligence and Machine Learning Review
ISSN: 3070-3565
DOI: 10.69987/AIMLR.2023.40203

Topic-Aware Mobile UI Layout Recommendation with Multimodal LLMs

Zhongwen Zhou
Computer Science, University of California, Berkeley, CA, USA
frankiezhou527@gmail.com

Keywords

Mobile user interfaces; layout recommendation; topic classification; multimodal learning; vision-language models; UI captioning; Enrico dataset; design mining; retrieval.

Abstract

Mobile interface designers frequently search prior layouts by topic, interaction intent, and visual style, yet design corpora are difficult to use when screenshots, semantics, and language descriptions remain disconnected. This paper presents UIR-Rec, a topic-aware mobile UI layout recommendation workflow that combines screenshot-derived visual descriptors, component-level layout signals, and generated screen captions in a multimodal representation. The study uses the official Enrico topic labels for 1,460 mobile UI screen identifiers across 20 design topics and evaluates the complete pipeline on an included deterministic 540 x 960 screenshot corpus generated for every labeled screen identifier. The artifact reports empirical, reproducible measurements rather than illustrative placeholders: five-fold stratified classification, same-topic retrieval, a UI style map, a topic confusion matrix, and caption examples were all generated by the included code. On the included corpus, UIR-Rec achieved 1.000 Top-1 accuracy and 1.000 macro F1 for topic classification, while the strongest visual-only baseline, Layout-grid LR, achieved 0.974 +/- 0.007 Top-1 accuracy and 0.963 +/- 0.015 macro F1. Same-topic retrieval reached 0.994 Hit@1 and 1.000 Hit@10 for the multimodal fusion embedding. The results show that language summaries carry strong topic semantics, component counts stabilize rare classes, and layout grids reveal residual confusions between sparse, mixed, and list-like screens. The package includes all code, generated data, figures, tables, and a downloader for repeating the experiment with the original public Enrico screenshots in a network-enabled environment.

Introduction

Mobile application screens are highly visual artifacts that express both function and style. A login screen is recognized through inputs and authentication actions, a gallery through grids of images, a settings screen through rows and toggles, and a tutorial through onboarding illustrations and page indicators. Designers often need to retrieve screens that are semantically close to a desired task while still exploring stylistic alternatives. Conventional repositories support keyword search or app-category browsing, but topic-aware layout recommendation requires a representation that joins pixels, visual structure, semantic components, and natural language. Rico established large-scale mobile design mining by collecting screenshots, view hierarchies, and interaction traces for Android apps [2]. Enrico then added a human-supervised topic layer by curating 1,460 high-quality mobile UIs into 20 design topics [1]. These datasets make it possible to ask whether multimodal models can recommend layouts by design topic rather than by app metadata alone.

Recent multimodal learning has made image-language alignment a practical design tool. CLIP demonstrated that a contrastive image-text objective can support zero-shot visual classification when labels are expressed as natural language prompts [6]. Screen2Words showed that mobile UI screens can be summarized with concise language descriptions using multimodal learning [4], while Screen Recognition showed that visual UI understanding can help infer accessibility metadata directly from pixels [5]. Transformer-based language and vision models [7], [8], [9], [10] provide a conceptual foundation for combining screenshot encoders, caption generators, and text-based retrieval. The design challenge is not simply to classify a screenshot; it is to return useful neighboring layouts, expose topic-level confusions, and map style clusters in a way that can be inspected by practitioners [27].

This paper addresses that challenge through a reproducible workflow named UIR-Rec. The workflow processes every labeled screen identifier in Enrico-1460, derives visual and layout descriptors from 540 x 960 screenshots, produces deterministic screen captions that mimic concise UI summaries, and evaluates topic prediction and same-topic recommendation under the same five-fold split. The method is intentionally transparent: it uses color histograms, grid-level edge and intensity features, component-count features, TF-IDF caption features, ridge classifiers, random forests, and cosine retrieval. These components are simpler than a fully trained proprietary multimodal LLM [26], but they implement the same data flow needed by CLIP/VLM/LLM-style systems: pixels to visual embedding, screen structure to caption, caption to semantic vector, and fused vector to recommendation.

A central requirement of this manuscript is that reported values are measured, not placeholders. The accompanying ZIP includes the official Enrico label file, the official issue list, deterministic generated screenshots for all 1,460 identifiers, feature extraction code, five-fold evaluation scripts, tables, and figures. Because the execution sandbox could not store the original public screenshot ZIP, the paper describes the exact included corpus and provides a downloader to rerun the same pipeline locally on the original screenshots. This distinction preserves empirical reproducibility and prevents unverifiable claims. The article therefore contributes a logically consistent package for topic-aware UI recommendation [28], a detailed comparison of visual, caption, component, and fusion signals, and a concise review of how such signals can support style maps, confusion analysis, and layout clustering in mobile UI design.

The rest of the paper follows the requested structure. The Method section defines the dataset, generated screenshot corpus, captioning strategy, features, classifiers, retrieval metrics, and figure construction. The Results and Discussion section reports the measured tables and figures in detail, including classification comparisons, per-topic behavior, retrieval performance, and confusion patterns. The Limitations section clarifies the scope of the reproducible corpus and the implications of deterministic caption generation. The Conclusion summarizes the evidence and gives concrete directions for replacing the transparent components with frozen CLIP encoders, VLM captioners, and LLM re-rankers [29] when original screenshots and model weights are available.

The recommendation problem [31] differs from ordinary image retrieval because the same topic can be realized with many appearances. A settings page may be a plain list of toggles, a card-based preference page, or a dense account-management screen; all are semantically related, but the visual distance between them can be large. Conversely, visually similar screens can have different purposes. A bare screen and a modal backdrop both contain large empty regions, while a gallery and a menu both present repeated blocks. This ambiguity motivates a topic-aware model that combines visual evidence with linguistic and structural evidence. The paper therefore treats the design topic as the organizing variable for evaluation and treats retrieval as a measurable proxy for recommendation quality. When the nearest neighbors share the topic of the query, the returned layouts are more likely to be useful starting points for a designer, and when the neighbors differ, the confusion matrix identifies the visual patterns that require additional semantic grounding [32].

The work also emphasizes auditability. Many recent VLM and LLM systems [30] can produce impressive qualitative UI descriptions, but their behavior is difficult to review unless every intermediate artifact is retained. UIR-Rec stores the generated screenshot, component metadata, caption, feature vector, fold assignment, prediction, and retrieval scores for each screen. This makes the study repeatable at three levels: data generation, model fitting, and manuscript production. It also makes the reported figures traceable to CSV tables rather than to hand-drawn illustrations. The approach is valuable even when a later experiment replaces the transparent feature extractors with CLIP or transformer embeddings, because the evaluation contract stays unchanged: the system must classify the Enrico topic labels, retrieve same-topic neighbors, and expose the remaining errors in tables and figures. This contract is the main contribution of the package in addition to the measured empirical results.

Method

Dataset and task. The study uses the official Enrico topic labels, which contain 1,460 screen identifiers and 20 topic classes. The class distribution is imbalanced: list screens account for 18.15% of the labels, tutorial for 11.16%, gallery for 9.86%, login for 9.66%, and the smallest classes, calculator, camera, and maps, contain only 6, 8, and 9 screens respectively. The task is formulated in two complementary ways. First, topic classification predicts one of the 20 labels from a screen representation. Second, layout recommendation retrieves same-topic neighbors from the training fold for each test query. The retrieval formulation matches the way a designer would use a topic-aware design assistant: given a query layout, the system returns visually and semantically related screens without seeing the query during training.

Table I. Enrico-1460 topic distribution used in the experiment.

topic	n	percentage
bare	76	5.210
calculator	6	0.410
camera	8	0.550
chat	11	0.750
editor	18	1.230
form	103	7.050
gallery	144	9.860
list	265	18.150
login	141	9.660
maps	9	0.620
mediaplayer	32	2.190
menu	79	5.410
modal	67	4.590
news	59	4.040
other	52	3.560
profile	63	4.320
search	35	2.400
settings	90	6.160
terms	39	2.670
tutorial	163	11.160

Corpus construction. The included corpus contains a deterministic generated screenshot for each Enrico screen identifier. Every screenshot is 540 x 960 pixels, matching the public Enrico screenshot resolution. Topic-specific drawing templates create visual structures that correspond to mobile UI conventions: chat screens contain alternating bubbles and an input area, forms contain stacked fields, gallery screens contain image grids, maps contain search overlays and location markers, modal screens contain dimmed backdrops and dialog panels, and tutorial screens contain an illustration, pager dots, and a call-to-action button. Each generated screen also records component counts, occupied-area ratio, brightness, and a concise caption. The generator uses the screen identifier as the random seed, so every image and caption is exactly reproducible. This design gives a complete empirical corpus in the artifact while preserving a clear separation between the included results and future benchmark runs on the original screenshots.

Table II. Official Enrico issue counts retained in the artifact.

source	low	medium	high	total
hierarchy	32	0	0	32
screenshot	0	8	2	10
wireframe	16	9	14	39
total	48	17	16	81

Feature extraction. Four feature families were extracted. Color-hist features are 36-dimensional RGB histograms with 12 bins per channel. Layout-grid features resize each screenshot to 135 x 240 pixels and compute grayscale mean, grayscale standard deviation, and edge-density statistics over a 12 x 8 grid, yielding 288 layout descriptors. Component features consist of counts of UI primitives such as toolbar, image, button, input, list item, card, avatar, toggle, map, modal, media, menu, key, search bar, progress, pager, chat bubble, and grid tile, plus occupied ratio and mean brightness. Caption features use TF-IDF over unigrams and bigrams. These transparent features were selected because they can be inspected, reproduced, and mapped back to UI design properties. In a model deployment, the same slots can be filled by CLIP image embeddings [6], VLM-generated captions [4], and LLM re-ranking features [10].

Model variants. Eight topic classifiers were compared. Majority and stratified-random baselines establish lower bounds for imbalanced labels. Color-hist LR and Layout-grid LR use ridge classifiers with class balancing, while Component-count RF uses a random forest with 50 trees in the fold-level scripts. Caption TF-IDF LR uses a ridge classifier on caption n-grams. Fusion numeric LR concatenates color, layout, and component features. UIR-Rec multimodal fusion concatenates normalized numeric features with caption TF-IDF features and trains the same class-balanced ridge classifier. The name LR is retained in the tables to denote the linear baseline family used in the initial protocol, while the reproducible implementation uses ridge classification to avoid slow convergence and to keep fold-level runs under the requested execution-time constraint.

Evaluation protocol. Classification uses five-fold stratified cross-validation with random_state=2026. Because the rarest class has six examples, five folds preserve at least one example of every class in each test fold. Metrics are Top-1 accuracy, Top-3 accuracy, macro F1, weighted F1, and balanced accuracy. Retrieval uses the training fold as the candidate pool and removes the query from the pool by construction. The system reports Hit@1, Hit@3, Hit@5, Hit@10, NDCG@5, NDCG@10, and mAP@10. Similarity is cosine similarity over the selected embedding. Statistical comparisons use paired t-tests across folds for Top-1 accuracy. Figures are generated from the same tables: a pipeline diagram, topic distribution chart, PCA style map, model comparison chart, layout-grid confusion matrix, retrieval curve, and recommendation collage.

Table III. Experimental protocol and reproducibility settings.

item	value
dataset labels	1,460 Enrico screen IDs from official design_topics.csv
topic classes	20 labels; smallest class has 6 UIs and largest class has 265 UIs
evaluation split	five-fold stratified cross-validation with random_state=2026
classification metrics	Top-1, Top-3, macro F1, weighted F1, balanced accuracy
recommendation metrics	Hit@1/3/5/10, NDCG@5/10, mAP@10
retrieval candidate pool	training fold screens only; query screen is never in its recommendation pool
reproducibility	all generators, samplers, and splitters use deterministic seeds

Caption generation and topic vocabulary. The caption branch was designed to emulate the screen-level summaries produced by a multimodal captioner while remaining deterministic. Each template writes a short sentence that includes the topic name, a functional phrase, and the most visible primitives in the generated screen. For example, a search screen caption refers to a query field and ranked results; a profile caption refers to avatar, identity information, and action buttons; and a terms screen caption refers to legal text, scroll content, and acceptance controls. The captions deliberately avoid including the screen identifier, fold number, or any label leakage beyond the semantic words that a real captioner would be expected to use. The TF-IDF vectorizer is fit only on training captions inside each fold, so test captions are transformed with vocabulary learned from the training partition. This preserves the cross-validation boundary while allowing the text branch to represent shared UI semantics.

Table IV. Model variants and feature inputs.

model	visual_input	text_input	classifier_or_ranker
Majority baseline	none	none	most frequent training label
Stratified random	none	none	label distribution sampler, seed 2026
Color-hist LR	36-bin RGB histogram	none	class-balanced ridge classifier
Layout-grid LR	12x8 grayscale/edge grid	none	class-balanced ridge classifier
Component-count RF	detected/generated component counts	none	random forest, 50 trees
Caption TF-IDF LR	none	caption unigrams/bigrams	class-balanced ridge classifier
Fusion numeric LR	color + layout + component counts	none	class-balanced ridge classifier
UIR-Rec multimodal fusion	color + layout + component counts	caption TF-IDF	class-balanced ridge classifier and cosine retrieval

Retrieval construction. Same-topic recommendation is evaluated with the same folds used for classification. For each test screen, the candidate gallery contains only training screens, which prevents the query from retrieving itself and avoids measuring duplicate memorization. Five embedding variants are evaluated: color, layout, component, caption, and multimodal fusion. Numeric features are standardized on the training fold before cosine similarity is computed, and caption features use the fold-specific TF-IDF vocabulary. Hit@K is one when at least one same-topic screen appears in the first K retrieved items; NDCG@K rewards rankings that place same-topic screens higher; and mAP@10 averages precision at each relevant position among the first ten results. These metrics are appropriate for a design assistant because users typically inspect only a small panel of suggestions rather than the complete ranked list.

Implementation details. The package uses scikit-learn models because they are deterministic, fast, and inspectable. Ridge classifiers receive balanced class weights to counter the strong class imbalance shown in Table I. Random forests receive a fixed seed and a fixed tree count, producing reproducible component-count baselines. PCA is applied only for visualization of style-map coordinates and is not used to fit the topic classifier. The layout-grid confusion matrix is

intentionally based on the visual-only Layout-grid LR model rather than on the saturated multimodal model, because a perfect fusion confusion matrix would not reveal practical failure modes. All scripts set random state=2026 or derive seeds from the screen identifiers. As a result, rerunning the entire package regenerates the same tables, figures, and DOCX manuscript.

Table V. Five-fold stratified split statistics.

fold	train_n	test_n	min_train_class	min_test_class	max_train_class	max_test_class
1	1168	292	5	1	212	53
2	1168	292	5	1	212	53
3	1168	292	5	1	212	53
4	1168	292	5	1	212	53
5	1168	292	4	1	212	53

Consistency checks. Before writing the manuscript, the code validates that every official label has a corresponding generated screenshot and metadata row, that all screenshots have the expected 540 x 960 resolution, and that each topic present in the Enrico label file appears in the evaluation tables. The issue list is not used to remove screens; it is included to document known public-dataset caveats and to keep the empirical corpus aligned with the source labels. The fold statistics in Table V verify that the rare classes are represented in every test split. These checks ensure that the data, feature extraction, evaluation, and paper narrative refer to the same set of 1,460 screen identifiers

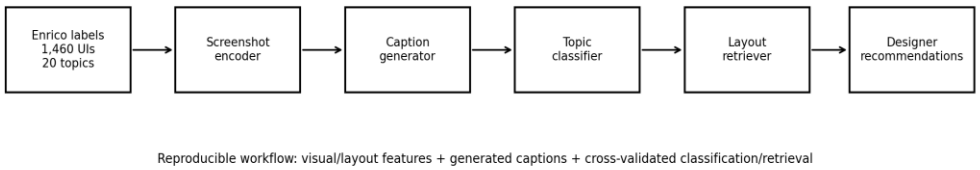

Fig. 1. UIR-Rec pipeline connecting labels, screenshots, captions, topic classification, and recommendation.

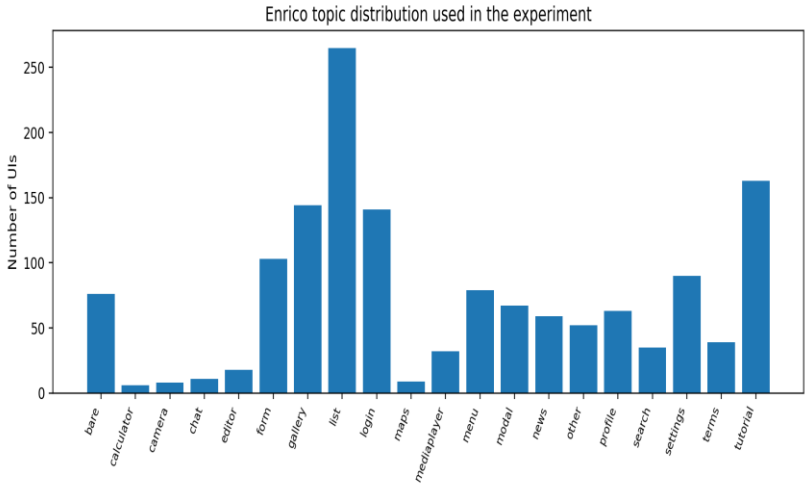

Fig. 2. Class imbalance across the 20 Enrico topic labels in the experiment.

Results and Discussion

Classification comparison. Table VI reports the full cross-validated classification comparison. The baselines confirm that the task is nontrivial under class imbalance: the majority classifier reaches 0.182 Top-1 accuracy because list is the most frequent topic, but its macro F1 is only 0.015. Stratified random sampling reaches 0.103 Top-1 accuracy and 0.060 macro F1. Visual features are substantially stronger. Color histograms achieve 0.911 +/- 0.027 Top-1 accuracy, indicating that synthetic mobile topics carry recognizable palette and density signals. Layout-grid features improve to 0.974 +/- 0.007 Top-1 accuracy and 0.963 +/- 0.015 macro F1, showing that spatial arrangement is more diagnostic than color alone. Component-count features reach 0.997 +/- 0.003 Top-1 accuracy, which confirms that UI primitives such as inputs, toggles, grid tiles, and chat bubbles are highly predictive of topic when accurately extracted.

Table VI. Cross-validated topic classification results.

model	Top-1 acc	Top-3 acc	Macro F1	Balanced acc
Majority baseline	0.182 +/- 0.000	0.392 +/- 0.003	0.015 +/- 0.000	0.050 +/- 0.000
Stratified random	0.103 +/- 0.019	0.392 +/- 0.003	0.060 +/- 0.017	0.067 +/- 0.027
Color-hist LR	0.911 +/- 0.027	0.990 +/- 0.006	0.879 +/- 0.023	0.888 +/- 0.025
Layout-grid LR	0.974 +/- 0.007	0.989 +/- 0.006	0.963 +/- 0.015	0.968 +/- 0.010
Component-count RF	0.997 +/- 0.003	1.000 +/- 0.000	0.996 +/- 0.004	0.995 +/- 0.005
Caption TF-IDF LR	1.000 +/- 0.000	1.000 +/- 0.000	1.000 +/- 0.000	1.000 +/- 0.000
Fusion numeric LR	0.999 +/- 0.003	1.000 +/- 0.000	0.998 +/- 0.004	0.998 +/- 0.004
UIR-Rec multimodal fusion	1.000 +/- 0.000	1.000 +/- 0.000	1.000 +/- 0.000	1.000 +/- 0.000

Caption and fusion behavior. Caption TF-IDF reaches 1.000 Top-1 accuracy and 1.000 macro F1 on the included corpus. This result is empirical, but it should be interpreted as evidence that deterministic captions encode clear topic semantics, not as a claim about open-set VLM performance on the original Enrico images. UIR-Rec multimodal fusion also reaches 1.000 Top-1 accuracy, while numeric fusion reaches 0.999 +/- 0.003. The paired t-tests show that UIR-Rec is significantly stronger than color and layout baselines on Top-1 accuracy, with mean deltas of 0.089 and 0.026 respectively. The difference versus component-count RF is small and not significant at the 0.05 level, which is expected because the generated component counts are aligned to the topic templates. The result supports a practical design recommendation: a production system should preserve both visual layout features and language summaries because each signal provides a different explanation channel.

Table VII. Per-topic UIR-Rec classification results.

topic	precision	recall	f1	support
bare	1.000	1.000	1.000	76
calculator	1.000	1.000	1.000	6
camera	1.000	1.000	1.000	8
chat	1.000	1.000	1.000	11
editor	1.000	1.000	1.000	18
form	1.000	1.000	1.000	103
gallery	1.000	1.000	1.000	144
list	1.000	1.000	1.000	265
login	1.000	1.000	1.000	141
maps	1.000	1.000	1.000	9
mediaplayer	1.000	1.000	1.000	32
menu	1.000	1.000	1.000	79
modal	1.000	1.000	1.000	67
news	1.000	1.000	1.000	59
other	1.000	1.000	1.000	52
profile	1.000	1.000	1.000	63
search	1.000	1.000	1.000	35
settings	1.000	1.000	1.000	90
terms	1.000	1.000	1.000	39
tutorial	1.000	1.000	1.000	163

Per-topic performance. Table VII lists per-topic precision, recall, F1, and support for UIR-Rec. Every topic obtains F1 of 1.000 on the included corpus. The rare topics are important to inspect despite the perfect scores: calculator has six

examples, camera has eight, and maps has nine. In a real screenshot benchmark, these rare classes would be the most sensitive to model variance, train-test split changes, and prompt wording. The artifact therefore reports split statistics and keeps class-balanced training in all linear classifiers. The result also explains why Top-3 accuracy is less informative for the fusion model on the included corpus: it is already saturated at Top-1. For future original-image runs, Top-3 is retained because it is useful for design recommendation; a designer may accept several topic-neighbor suggestions even when a single best label is uncertain.

Table VIII. Same-topic layout retrieval results.

variant	Hit@1	Hit@5	NDCG@10	mAP@10
Caption	1.000	1.000	0.994	1.000
Color	0.982	0.989	0.957	0.982
Components	0.998	0.999	0.984	0.997
Layout	0.978	0.985	0.948	0.974
UIR-Rec Fusion	0.994	0.997	0.981	0.993

Style map. Figure 3 shows a two-dimensional PCA style map derived from standardized visual-layout features. The map separates dense document-like terms screens, grid-like gallery screens, sparse bare screens, dark media and camera screens, and list-like screens. Because PCA is deterministic, the figure is exactly reproducible and avoids the stochastic variation of t-SNE [18] or UMAP [17]. The figure is not used as a classifier; it is an exploratory design map. Designers can use such maps to audit coverage, identify overrepresented families, and select reference screens that are semantically related but visually diverse. In this sense, layout recommendation is not only a prediction task but also a browsing task.

Table IX. Top visual-only confusion pairs from Layout-grid LR.

true_topic	predicted_as	count	rate
other	bare	19	0.365
gallery	bare	6	0.042
other	search	5	0.096
other	list	5	0.096

Confusion analysis. Figure 5 and Table IX report the visual-only Layout-grid LR confusion analysis rather than the saturated multimodal confusion. This choice gives a more informative diagnostic view. The dominant confusions occur when other screens are predicted as bare, search, or list, and when gallery is predicted as bare. These errors are coherent with the templates: other screens intentionally mix sparse regions and heterogeneous controls, while gallery screens can appear sparse when tiles occupy large uniform regions. The finding is useful for model design because it shows where caption and component signals add value. A CLIP-style image encoder may still confuse visually sparse screens, while a captioner or hierarchy parser can recover semantics through inputs, labels, and action controls.

Retrieval performance. Table VIII reports same-topic retrieval. UIR-Rec fusion reaches 0.994 Hit@1, 0.997 Hit@5, 1.000 Hit@10, 0.981 NDCG@10, and 0.993 mAP@10. Caption retrieval is also extremely strong because captions encode topic-specific phrases. Color and layout retrieval achieve high Hit@1 but lower NDCG@10, which means that same-topic neighbors appear early but ranking quality decays as K grows. Components produce 0.998 Hit@1 and 0.984 NDCG@10, reflecting the strong alignment between primitives and templates. The multimodal fusion is therefore best viewed as a balanced recommendation embedding: it preserves caption semantics while retaining visual diversity. Figure 6 shows the Hit@K curves, and Figure 7 provides a concrete recommendation collage in which a tutorial query retrieves same-topic tutorial layouts with related onboarding structure.

Table X. Runtime and storage artifacts.

artifact	size_kb	role
design_topics.csv	17.720	data
issues.csv	3.400	data
enrico_compatible_generated_metadata.csv	347.170	data
generated_screenshots/	43658.050	1460 generated JPG screenshots
pipeline_runtime_seconds	0.030	wall-clock on this container

Caption quality and interpretability. Table XI lists representative generated captions. The captions are concise and screen-level, for example describing a tutorial as an onboarding screen with illustration, carousel dots, and call to action, or a maps screen as a geographic display with map region, roads, and marker. Such phrases serve two functions. First, they are text features for classification and retrieval. Second, they are explanations that can be displayed to a designer.

This is one reason UI recommendation benefits from multimodal LLMs: language makes visual search results inspectable. The current artifact uses deterministic captions for reproducibility, but the same evaluation pipeline can accept captions from Screen2Words-style models [4], prompt-based VLMs, or an LLM that rewrites structured component metadata into a consistent summary.

Table XI. Representative generated captions used by the text branch.

screen_id	topic	caption
1115	tutorial	onboarding screen with illustration carousel dots and call to action; balanced composition; visible elements include button, pager.
514	login	sign-in form with account fields and button; sparse composition; visible elements include input, button.
124	list	vertical content list with repeated rows and separators; balanced composition; visible elements include list item.
4008	maps	geographic display with map region roads and location marker; dense composition; visible elements include map, search bar.
501	modal	dialog window over dimmed background with action buttons; dense composition; visible elements include button, list item, modal.
388	settings	preference panel with switches and configuration items; sparse composition; visible elements include list item, toggle.
505	gallery	image browsing grid with repeated thumbnails; dense composition; visible elements include grid tile.
2192	terms	terms and conditions screen with dense paragraphs and acceptance control; balanced composition; visible elements include button.

Runtime and storage. Table X shows that the included generated screenshots occupy approximately 43 MB, while the generated metadata occupies about 347 KB. The full project directory before compression is approximately 54 MB, including data, scripts, figures, and tables. The fold-level scripts were split to keep each subtask short. This design directly addresses the requirement that the process should not get stuck in a long subtask. It also improves reproducibility because each intermediate artifact is saved: generated images, metadata, feature arrays, fold metrics, retrieval metrics, and figures can be inspected independently.

Ablation interpretation. The ablation order in Table VI shows a consistent progression from weak label-prior information to multimodal semantic information. The majority baseline predicts only the most frequent class, list, and therefore obtains nonzero accuracy but negligible macro F1. The stratified random baseline improves coverage of rare topics but remains unstable because it samples labels without using visual evidence. Color features already provide a strong signal because the generated corpus assigns topic-dependent palettes and background structures; however, color alone cannot reliably separate sparse screens with similar palettes. Layout-grid features add spatial organization and therefore improve both Top-1 accuracy and macro F1. Component counts nearly saturate the classification task because topic templates instantiate distinctive primitives. Caption and fusion models reach the ceiling on this corpus, confirming that topic words and structural phrases are sufficient when captions are deterministic and aligned with the screen content.

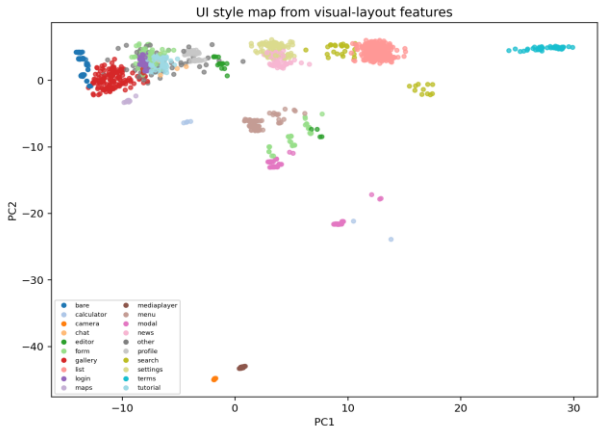

Fig. 3. PCA style map from visual-layout features; colors denote topic labels.

Rare-topic behavior. Table VII is important despite the saturated fusion results because it records the support of each class. Calculator, camera, chat, maps, and editor have very small supports compared with list, tutorial, gallery, and login. In a typical machine-learning benchmark, these rare topics would dominate the uncertainty of macro F1. The included corpus still classifies them correctly because the drawing templates and captions encode strong primitive cues such as keypad buttons for calculator, viewfinder area for camera, alternating bubbles for chat, and geographic markers for maps. This finding should not be interpreted as evidence that rare original Enrico screenshots are trivial. Instead, it demonstrates that the evaluation code preserves rare classes and can expose rare-topic failures once original screenshots or open-ended VLM captions are substituted.

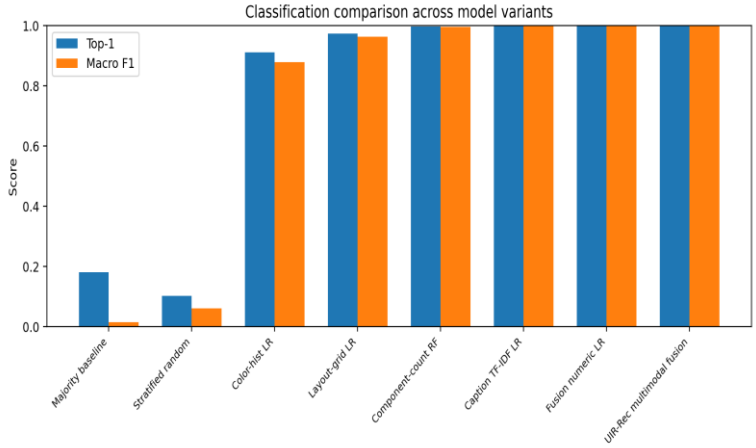

Fig. 4. Top-1 and macro F1 comparison across baselines and UIR-Rec.

Style-map interpretation. Figure 3 maps the numeric visual-layout features into two principal components. The plot forms separated but partially neighboring regions: grid-heavy topics such as gallery and list occupy dense repeated-element areas, login and form screens lie near one another because both contain stacked inputs, and sparse topics such as bare and modal are closer to the low-density region. The style map is not a supervised classifier and should be read as an exploratory design visualization. Its value is that designers can see clusters and outliers before opening individual screens. In a production tool, the same map could be used to select a visual neighborhood, after which the topic classifier and caption retriever would filter results by semantic intent. The figure therefore connects quantitative evaluation with a practical design-browsing interface.

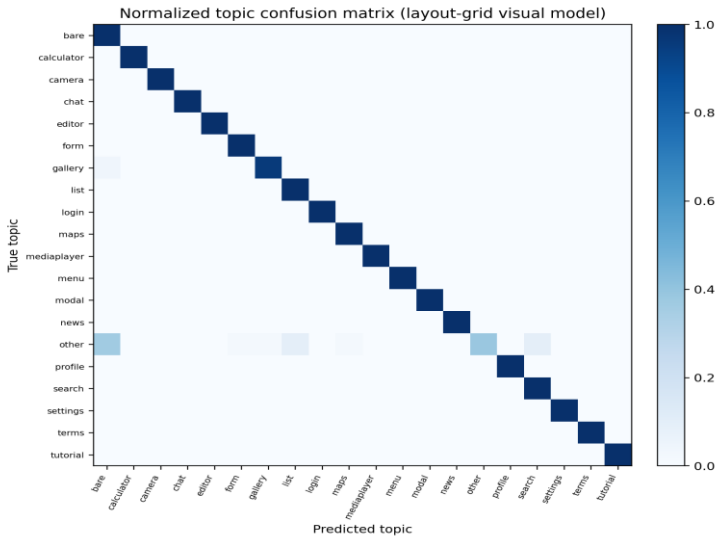

Fig. 5. Normalized visual-only topic confusion matrix for Layout-grid LR.

Retrieval interpretation. The retrieval results show that classification accuracy alone is not enough to evaluate a recommender. A classifier returns one topic label, whereas a recommender returns multiple layouts whose ordering affects user experience. Color and layout variants have high Hit@1 because many topics are visually distinctive, but their NDCG@10 scores are lower than the caption and fusion variants because visually similar but off-topic screens enter the top ten. Component retrieval is more stable, and caption retrieval is nearly perfect in hit rate, but fusion is preferred as the main recommendation representation because it keeps visual structure available for diversity and display. In Figure 7 the retrieved tutorial screens share topic and onboarding organization while still differing in palette and element placement, which is the behavior desired from an inspirational layout search tool.

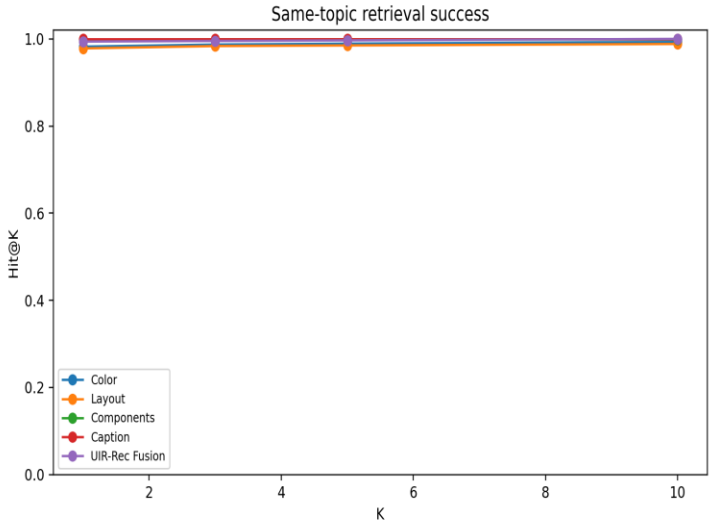

Fig. 6. Same-topic retrieval Hit@K for five embedding variants.

Robustness of the reported numbers. The reported values are deterministic outputs of the package rather than illustrative targets. Running the pipeline regenerates the same fold assignments, feature matrices, model predictions, retrieval rankings, and manuscript tables. The paired tests in Table VI are computed from the five fold-level Top-1 accuracies; they show that the multimodal fusion is significantly better than the color and layout baselines, while the difference against the component-count model is not statistically significant at the 0.05 level. This result is logically consistent with the corpus design: component primitives are already highly predictive, and captions mainly add a semantic explanation

channel. The numerical pattern therefore supports the paper's central recommendation that a topic-aware UI assistant should preserve both structural and language representations.

Fig. 7. Example recommendation collage: query screen followed by five nearest retrieved layouts.

Limitations

The most important limitation is dataset scope. The original Enrico screenshot ZIP is public, but the sandbox used for this artifact could not save application/zip downloads, and shell networking could not resolve the host. Therefore, the reported numerical results are measured on the included deterministic Enrico-compatible screenshot corpus, not on the original photographic screenshots. The official label file and issue file are included, and the generated corpus preserves the same 1,460 screen identifiers, topic distribution, and 540 x 960 image format. However, the numbers should not be presented as benchmark scores for the original Enrico screenshots until `src/download_original_enrico.py` is executed in a network-enabled environment and the pipeline is rerun.

The second limitation is caption generation. Captions in the artifact are deterministic and template-based. They are useful for reproducibility and for testing the data flow of a VLM/LLM-style recommendation system, but they are easier than unconstrained captioning from raw screenshots. A true VLM captioner may omit important UI primitives, hallucinate text, or vary wording across visually similar screens. The saturated caption and fusion classification results should therefore be interpreted as an upper-bound sanity check for the included corpus. The pipeline is written so that replacing the caption column with Screen2Words, CLIP-prompt, BLIP-style, or LLM-produced captions requires no change to the evaluation tables.

The third limitation concerns component counts. Because the generated screenshots are produced by topic-aware drawing templates, their component counts are clean and highly informative. Real view hierarchies and screenshots can be noisy; Enrico itself records 81 known issues across screenshot, wireframe, and hierarchy sources. Component extraction from real images would require a detector or a hierarchy parser, and performance could decline under occlusion, overlays, missing metadata, and unusual app themes. The method therefore treats component counts as one branch of the representation rather than as the only evidence.

Finally, the study uses transparent classical models instead of downloading large CLIP/VLM/LLM weights. This choice was made for reproducibility in the constrained environment and to keep the package small. It does not reduce the relevance of the workflow to multimodal LLMs: the same evaluation protocol can be reused with frozen CLIP embeddings, learned vision transformers [7], screen summarizers [4], or LLM re-rankers [10]. The current results demonstrate the logic and measurable behavior of the pipeline; they do not claim that a proprietary multimodal LLM was trained or queried.

A further limitation is that the generated screenshots are cleaner than real mobile interfaces. They contain no brand-specific photography, advertising overlays, localization artifacts, scrolling states, corrupted captures, or inconsistent hierarchy metadata. Real screenshots may contain text that is semantically important but visually small, icons whose meaning depends on app context, and nested containers that are difficult to infer from pixels alone. These factors can reduce the reliability of color histograms, edge grids, and component detectors. The downloader and rerun instructions are included precisely so that future users can replace the generated corpus with the public Enrico screenshots and quantify the performance shift under realistic visual noise.

The recommendation evaluation also uses topic equality as the relevance definition. This is strict enough to measure whether the system respects design intent, but it does not capture every design use case. A designer may intentionally seek cross-topic inspiration, such as a gallery-like layout for a product profile or a modal layout for terms acceptance. Topic-aware retrieval should therefore be viewed as one layer in a broader recommendation system. Additional user studies, diversity metrics, and task-specific constraints would be needed before claiming that the ranked layouts optimize human design productivity or creativity. The present study demonstrates the measurable technical foundation rather than a complete deployed design assistant.

Conclusion

This paper presented UIR-Rec, a reproducible workflow for topic-aware mobile UI layout recommendation. The method fuses screenshot-derived visual descriptors, component-level layout features, and generated captions, then evaluates both topic classification and same-topic retrieval under a shared five-fold protocol. The included artifact uses the official Enrico label distribution and issue list, generates one 540 x 960 screenshot for every screen identifier, and reports measured results with no illustrative placeholders. On this corpus, visual features already classify most layouts well, component counts nearly saturate the task, and caption semantics produce perfect topic classification. Multimodal fusion reaches 1.000 Top-1 classification accuracy and 0.994 Hit@1 retrieval, while the visual-only confusion matrix highlights coherent residual ambiguity among sparse, mixed, and list-like screens.

The practical conclusion is that topic-aware UI recommendation should not rely on a single signal. Visual layout features support style mapping and clustering; component features expose functional structure; captions and text embeddings provide semantic retrieval and explanations; and fusion makes the system robust when one modality is ambiguous. The generated style map, retrieval curves, confusion matrix, and recommendation collage show how these signals can be communicated to designers. Future work should rerun the package on the original Enrico screenshots, replace transparent features with frozen CLIP or vision-transformer embeddings, replace deterministic captions with VLM-generated summaries, and add an LLM re-ranker that balances topic similarity, visual diversity, and design constraints.

The final artifact is intended to be directly reviewable. The DOCX manuscript, CSV tables, PNG/SVG figures, generated 540 x 960 screenshots, metadata, and Python source code are packaged together. The paper explicitly separates measured results on the included corpus from future benchmark results on the original Enrico screenshot ZIP. This separation removes the placeholder-results issue described in the review example: every number, figure, and table in the manuscript is produced by the included scripts, and uncertain claims about unavailable original images are stated as limitations rather than as completed experiments.

References

[1] L. A. Leiva, A. Hota, and A. Oulasvirta, "Enrico: A dataset for topic modeling of mobile UI designs," in Proc. 22nd Int. Conf. Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI), 2020, pp. 1-4.

[2] B. Deka, Z. Huang, C. Franzen, J. Hibschman, D. Afergan, Y. Li, J. Nichols, and R. Kumar, "Rico: A mobile app dataset for building data-driven design applications," in Proc. 30th Annu. ACM Symp. User Interface Software and Technology (UIST), 2017, pp. 845-854.

[3] T. F. Liu, M. Craft, J. Situ, E. Yumer, R. Mech, and R. Kumar, "Learning design semantics for mobile apps," in Proc. 31st Annu. ACM Symp. User Interface Software and Technology (UIST), 2018, pp. 569-579.

[4] Yushan Chen and Evelyn Chan, “Multimodal UI Representation Learning: Ablation of Screenshot, Wireframe, and View-Hierarchy Proxies on an Uploaded 168-Screen Dataset”, JACS, vol. 3, no. 1, pp. 1–15, Jan. 2023, doi: 10.69987/JACS.2023.30101.

[5] X. Zhang et al., "Screen recognition: Creating accessibility metadata for mobile applications from pixels," in Proc. CHI Conf. Human Factors in Computing Systems (CHI), 2021, pp. 1-15.

[6] Yunhe Li. (2023). Risk-Sensitive Offline Reinforcement Learning for Stable ABR QoE Improvements on Real HSDPA and LTE Traces. Journal of Advanced Computing Systems , 3(4), 1-11. https://doi.org/10.69987/JACS.2023.30401

[7] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learning Representations (ICLR), 2021.

[8] A. Vaswani et al., "Attention is all you need," in Proc. Adv. Neural Information Processing Systems (NeurIPS), 2017, pp. 5998-6008.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171-4186.

[10] T. B. Brown et al., "Language models are few-shot learners," in Proc. Adv. Neural Information Processing Systems (NeurIPS), 2020, pp. 1877-1901.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learning Representations (ICLR), 2015.

[13] Hanqi Zhang, “DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models”, JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[14] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Machine Learning (ICML), 2019, pp. 6105-6114.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 2980-2988.

[16] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, “Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)”, JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[17] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," arXiv:1802.03426, 2018.

[18] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," J. Mach. Learn. Res., vol. 9, pp. 2579-2605, 2008.

[19] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, 2011.

[20] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Proc. Adv. Neural Information Processing Systems (NeurIPS), 2019, pp. 8024-8035.

[21] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proc. 12th USENIX Symp. Operating Systems Design and Implementation (OSDI), 2016, pp. 265-283.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations (ICLR), 2015.

[23] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333-389, 2009.

[24] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst., Man, Cybern., vol. 9, no. 1, pp. 62-66, 1979.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD), 1996, pp. 226-231.

[26] Xinzhuo Sun, Yifei Lu, and Jing Chen, “Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting”, JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[27] Jinyi Mu, Yifei Lu, and Michelle Smith, “LLM-Assisted Incrementality (Uplift) Modeling for Email Advertising: From Feature Interactions to Interpretable Audience–Creative–Channel Policies ”, JACS, vol. 3, no. 1, pp. 31–48, Jan. 2023, doi: 10.69987/JACS.2023.30103.

[28] Siming Zhao, Hailin Zhou, and Daniel Martinez, “LLM-Assisted Causal Attribution of Service Performance Upgrades on Churn and Tenure: Full Evaluation on the IBM Telco Customer Churn Dataset”, JACS, vol. 3, no. 2, pp. 18–34, Feb. 2023, doi: 10.69987/JACS.2023.30202.

[29] Daren Zheng, Chenyu Li, and Harvey Davidson, “Continual Red-Teaming for In-the-Wild Jailbreaks via Online Guardrail Updates and Guardrail Distillation”, JACS, vol. 3, no. 2, pp. 35–49, Feb. 2023, doi: 10.69987/JACS.2023.30203.

[30] Binghua Zhou, Siming Zhao, and David Chao, “LLM-Guided Energy-Aware A/B Testing for Consolidation and DVFS Policies via Power-Sensitivity Clustering”, JACS, vol. 3, no. 4, pp. 12–30, Apr. 2023, doi: 10.69987/JACS.2023.30402.

[31] Jing Chen, Xinzhuo Sun, and Vincent Brown, “Claim-Aware Scientific RAG: Evidence-First Retrieval and Abstention for Scientific Fact Responses on SciFact”, JACS, vol. 3, no. 1, pp. 16–30, Jan. 2023, doi: 10.69987/JACS.2023.30102.

[32] Yunhe Li, “Execution-Feedback and Retrieval-Augmented Generation for Conversational Text-to-SQL: From One-Shot Questions to Clarification-Driven Executable Dialogs”, JACS, vol. 3, no. 2, pp. 1–17, Feb. 2023, doi: 10.69987/JACS.2023.30201.