Generative AI for Multi-Modal Data Management

Mahdi Erfanian
Supervised by: Abolfazl Asudeh
University of Illinois Chicago
Chicago, IL, USA
merfan2@uic.edu

ABSTRACT

The volume and diversity of multi-modal data—images, video, trajectories, source code—are growing far faster than the data-management techniques that must store, curate, retrieve, and serve them. Classical, rule-based pipelines struggle with the implicit semantics of these modalities, with the long-tail biases they encode, and with the increasingly compositional ways in which users want to query them. My thesis argues that *Generative AI (GenAI) should be elevated from a passive content producer to an active, first-class component of the data-management stack*, used not only to synthesize tuples but also to guide retrieval, repair coverage, and audit AI assistants. I organize this agenda around three thrusts: **(T1) Responsible Data**, where foundation models repair coverage and group-robustness gaps in multi-modal datasets (Chameleon, Robustify); **(T2) Generative Retrieval**, where synthetic "guide" tuples reframe complex natural-language queries as in-domain nearest-neighbor searches over images, video, and curve tuples (Needle, NeedleDB, Yarn, ANN-seq); and **(T3) Code Agents as Multi-Modal Producers**, where the same generate-then-verify recipe yields execution-verified benchmarks and synthetic hard-negative training data for Fill-in-the-Middle (FIM) code assistants (Delulu, Mirage). I outline the published results to date, the systems I am building, and the open problems—temporal retrieval, ANN over sequences, and the move from FIM completion to broader code-edit and code-generation agents—that will form the remainder of my dissertation.

VLDB Workshop Reference Format:
Mahdi Erfanian. Generative AI for Multi-Modal Data Management. VLDB 2026 Workshop: VLDB Ph.D. Workshop.

1 INTRODUCTION AND PROBLEM STATEMENT

Modern data-management systems are increasingly asked to handle *multi-modal* content: photographs and dashcam frames in autonomous-driving datasets [23], trajectories of moving objects, medical scans, and the source code that powers every AI assistant. Two properties make these corpora hard. First, their semantics are *implicit*: a 1080p image is not labeled with the attributes one would join on in a relational schema. Second, the items users actually care about are often *rare and safety-critical*—a pedestrian using a cane in a dashcam corpus, an under-represented demographic group in a training set [2, 21], or a single hallucinated API call in a 200-line code completion. Classical tools (SQL operators, contrastive embeddings such as CLIP [18], alphanumeric SMOTE [4]) were not designed for these regimes and break down precisely on the rare cases that matter most.

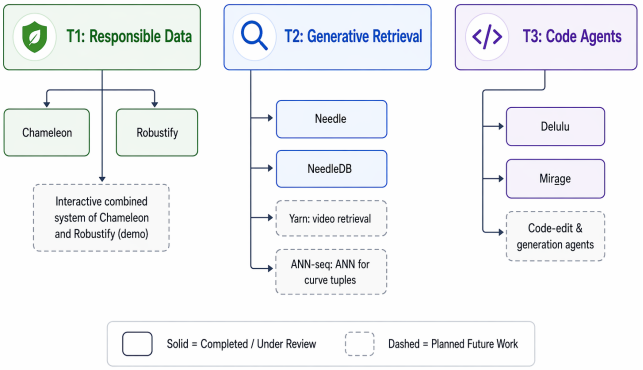

Figure 1: Research roadmap across the three thrusts of the dissertation. *Solid* boxes denote completed or under-review work; *dashed* boxes denote planned dissertation work.

My thesis takes the position that **Generative AI is a missing operator in the multi-modal data-management stack**. Foundation models can generate synthetic tuples that are missing in a dataset, generate the in-domain artifacts that bridge text queries to images and video, and even fabricate the wrong-but-plausible code that exposes failure modes of smaller production models. The unifying recipe across my work is *generate-then-verify*: a large foundation model proposes candidates; a lightweight, problem-specific test (coverage check, distribution outlier test, ensemble nearest-neighbor vote, Docker execution sandbox) accepts or rejects them. This recipe yields formal guarantees (NP-hardness reductions, PAC-style sample-complexity bounds, Monte Carlo error bounds) while remaining modality-agnostic. My proposal research question is:

> *How can generative foundation models be reliably integrated as data-management operators across modalities—images, video, geometric sequences, and source code—so that downstream systems are simultaneously more responsible, more expressive, and more robust?*

I decompose this question into three thrusts (Figure 1). Sections 2–4 summarize the contributions of each, and Section 5 sketches the open problems that will complete the dissertation.

2 THRUST 1: GENAI FOR RESPONSIBLE DATA

Datasets shared on open portals are frequently the first product of the analysis pipeline and frequently under-represent minorities, leading to discriminatory or unreliable downstream models [2, 21]. Detecting under-representation is increasingly well-understood;

Proceedings of the VLDB Endowment. ISSN 2150-8097.

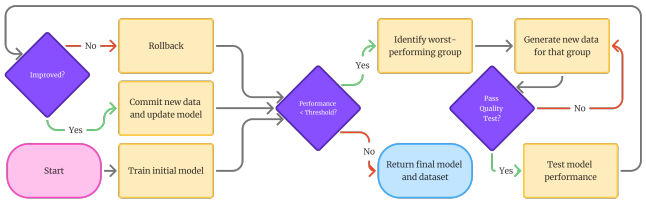

Figure 2: The Robustify *generate-and-test* pipeline (T1). The model's worst-performing group drives targeted synthetic generation; only samples that pass a quality test *and* measurably improve worst-group performance are committed. Chameleon instantiates the same recipe with a coverage-driven trigger instead of a model-uncertainty trigger.

repairing it on multi-modal data is not, because traditional augmentation (SMOTE [4], jittering) does not work on raw images, video clips or audio.

Chameleon [10] (VLDB'24). I formalized *Combination-Selection* (CS): given a coverage definition, find a minimum set of synthetic tuples whose addition repairs lack of coverage at the most general level. I proved CS is NP-hard and gave a greedy logarithmic-approximation algorithm. To enforce semantic integrity, Chameleon treats every dataset tuple as an i.i.d. sample from an unknown distribution ξ, embeds tuples into a foundation-model latent space, and runs a *rejection-sampling* test that discards synthetic tuples that look like outliers under ξ, plus a quality test that models human realism judgments as hypothesis testing. To minimize (monetarily costly) foundation-model queries—the method's *explicit cost objective*—Chameleon selects a *guide tuple* per generation via a contextual multi-armed bandit (repairing a face dataset's uncovered racial groups took 307 DALL-E 2 queries, $4.91). On image datasets (UTKFace, FairFace, animals) and a textual sentiment-analysis task, Chameleon reduced worst-group performance disparity substantially while keeping the augmentation budget below ~10% of the original dataset.

Robustify [9] (under review). Chameleon is task-*unaware*: it equalizes coverage, but coverage equality does not imply group-distributional robustness (GDR). Robustify closes this gap with a *task-aware, closed-loop* pipeline. Inspired by active learning [20], it identifies *model-uncertain* samples from the worst-performing group (misclassified with high confidence, or correct with low confidence) and uses them as guides for generative augmentation. A reinforcement-learning controller balances exploitation (patching known weaknesses) and exploration (text-only prompts for unseen scenarios). I derive a group-conditional PAC bound showing that untargeted augmentation has prohibitively high sample complexity, motivating the targeted design. Across image and tabular benchmarks, Robustify dominates re-weighting [1] and DRO [19] baselines on worst-group accuracy, and combining Robustify with SMOTE [4] outperforms every baseline tested.

3 THRUST 2: GENAI FOR MULTI-MODAL RETRIEVAL

Contrastive embedders (CLIP [18], ALIGN [17]) treat retrieval as a single feed-forward similarity computation in a fixed shared space. They work for simple object queries (*"traffic light"*) but degrade sharply on *compositional* queries that demand attribute and spatial reasoning (*"a person walking with a cane"*, *"a metro wagon at a busy intersection"*)—exactly the rare, safety-critical cases that motivate retrieval in BDD100K-scale corpora [23].

Needle [8] (under review). Needle reframes text-to-image retrieval as *image-to-image* retrieval: a diffusion foundation model produces a small set of synthetic *guide images* that depict the query, and an ensemble of vision embedders runs independent nearest-neighbor searches whose results are fused by weighted rank aggregation. I model the procedure as a *Monte Carlo estimator* and derive a Chernoff-style probabilistic error bound that decays exponentially in the number of guides and embedders, extended to correlated embedders. The system layer adds a query-complexity classifier (short-circuits trivial queries to plain CLIP [18]), a dynamic embedder-trust mechanism, an outlier filter on guides, and a metadata cache that learns from query history. Needle matches CLIP on simple queries and substantially outperforms CLIP [18], ALIGN [17], FLAVA [22], and concurrent generative-retrieval methods (IRGen [24], SceneDiff [25]) on compositional BDD100K queries, with strong gains in a user study. Crucially, this accuracy comes at *CLIP-class latency* (0.203 s per query vs. CLIP's 0.184 s on the same hardware) and the query router and cache push the average case lower still.

NeedleDB [7] (demonstration paper, under review). To bridge research and practice, I am packaging Needle as a deployable, gallery-agnostic database with a CLI, query routing, caching, and live indexing. NeedleDB lets users (i) issue free-form queries and A/B compare against CLIP, (ii) index their own image collection in real time, and (iii) ablate hyperparameters (number of guides, generator choice, embedder set) interactively. The same backbone is being adapted to two domain verticals where compositional queries dominate: *patent retrieval* (long technical descriptions over figure corpora) and *animals-in-the-wild tracking* (camera-trap imagery with rare species and pose attributes).

ANN-seq [11] (in progress). Extending Needle to video is more than adding a time axis. Each video in the gallery is naturally modeled as a *tuple of curves*: one trajectory per tracked object in a high-dimensional appearance space. A natural-language query (e.g. *"a cyclist overtaking a car at an intersection"*) is rendered by a text-to-video model into a synthetic clip from which we extract its *own* tuple of object curves. Retrieval then becomes the problem of finding gallery videos whose k object curves are simultaneously close to the k query curves—an *approximate nearest-neighbor search for tuples of curves* that to our knowledge has not been studied. As a first algorithmic step, I extend the single-curve LSH scheme of Driemel–Silvestri [6] to k-tuples under the max-Fréchet distance, drawing an *independent* random shift per component so that collisions factorize and we obtain a provable $(r, c, \frac{1}{2}, 0)$-sensitive family for arbitrary, possibly correlated object trajectories. This construction is the algorithmic backbone of Yarn, the text-to-video sibling of Needle that will become a deployed video-retrieval database in the spirit of NeedleDB.

Figure 3: Top-3 retrieved BDD100K [23] frames for queries of increasing complexity. CLIP [18] succeeds only on the simple query *"traffic light"*; NEEDLE answers all three correctly using two guide images (RealVisXL) and two embedders (EVA, RegNet). Green/red borders mark relevant/irrelevant results.

4 THRUST 3: CODE AGENTS AS MULTI-MODAL PRODUCERS

Source code is itself a first-class modality in the multi-modal stack—structured, compositional, and (unlike images or video) *executable*, so correctness is machine-checkable rather than perceptual. Production Fill-in-the-Middle (FIM) models [3, 14] (3–7B parameters, shipped inside Copilot and Cursor) routinely hallucinate non-existent APIs, undefined identifiers, and fabricated imports. The driving question of this thrust is: *can GenAI itself be used to reduce hallucinations of smaller, deployable FIM models?* Our hypothesis is that a frontier model, asked to generate *plausibly wrong* completions, produces high-value *hard negatives* that a small student can be supervised to avoid—turning the failure mode into its own training signal.

When we began executing this plan, we hit an unexpected gap: there is no reliable benchmark for measuring hallucinations in FIM tasks. Existing FIM suites (HumanEval Infilling [3, 5], SAFIM [15]) are Python-only, lack execution verification of the infilled span, and do not separate different hallucination *types*. Without such a benchmark we could neither calibrate the difficulty of synthetic hard negatives nor credibly claim that a fine-tuned model had reduced hallucinations rather than overfitted to a proxy. We therefore step back and first build the benchmark (DELULU), and only then return to the original mitigation question (MIRAGE).

DELULU [12] (benchmark, under review). DELULU is a verified multi-lingual FIM benchmark of 1,951 samples across 7 languages and a 4-category hallucination taxonomy (method, parameter, undefined-variable, import). Each sample is curated through a five-stage adversarial pipeline: a frontier LLM generates a hallucinated counterpart of a real GitHub completion; four diverse judges score it; embedding clustering surfaces hard examples; a self-contained Docker container verifies that the golden completion compiles *and* the hallucinated variant fails with the expected runtime error; and human experts remove biased rows. Evaluating 11 open-weight FIM models (0.5B–32B) shows that even the strongest model reaches only 84.5% pass@1 and no family exceeds 0.77 Edit Similarity—and that frontier judges miss import-hallucinations more than 40% of the time, establishing *both* hallucination generation and detection as open problems.

Table 1: Unifying view: every project instantiates a *generate-then-verify* operator over a different modality.

Project	Modality	Generator	Verifier
CHAMELEON	image / text	T2I + LLM	coverage + outlier + quality
ROBUSTIFY	image / tabular	T2I + LLM	worst-group accuracy gain
NEEDLE	image	T2I (RealVisXL)	ensemble NN vote + rank agg.
NEEDLEDB	image	T2I	live user feedback
YARN	video	T2V (planned)	sequence-NN (Fréchet)
ANN-SEQ	sequences	—	LSH collision bound
DELULU	source code	Claude-4.5	Docker exec. + judge panel
MIRAGE	source code	frontier panel	paired SFT / fool-rate

MIRAGE [13] (under review, returning to the original question). Using DELULU's taxonomy, I built an execution-free, multilingual SFT (Supervised Fine-tuning) pipeline that mines paired ⟨CHOSEN, REJECTED⟩ FIM samples from public GitHub: the golden completion is the real edit, and frontier models produce wrong-but-plausible *hard negatives* for each of the four hallucination types across 8 languages. Fine-tuning Qwen2.5-Coder-7B-Instruct [16] on a 100K-row curated slice yields +18.8 pass@1 (+0.22 edit similarity) on DELULU, *with positive deltas in every language and every type*, while simultaneously improving HumanEval-Infilling [3, 5] and every SAFIM [15] subset. A "fool-rate" filter (fraction of judges deceived by a hard negative) acts as a difficulty signal that lets us train on a much smaller, harder slice without quality loss. The recipe is base-model-agnostic (CodeLlama, StarCoder2, DeepSeek-Coder also benefit) and removes the per-language unit-test infrastructure that blocked prior execution-based pipelines. Crucially, all frontier calls are *offline*: the served model is the small fine-tuned model itself (no frontier calls at query time), and a reduced-scale variant reaches near-identical accuracy at roughly half the compute. We refer to this SFT recipe as MIRAGE to emphasize that the supervision signal is a deliberately fabricated, plausible illusion of code.

5 CONCLUSION AND FUTURE WORK

(O1) Unified responsible-data demonstration. An interactive tool that wraps CHAMELEON and ROBUSTIFY: upload an image dataset, declare fairness and robustness goals, receive an audited, augmented dataset together with provenance tags marking synthetic tuples (as required by data-sharing platforms).

(O2) YARN: generative text-to-video retrieval. A deployed video-retrieval database that lifts NEEDLE's Monte Carlo estimator to videos modeled as tuples of object curves. The core theoretical question—*approximate nearest-neighbor for tuples of curves*—is the subject of ANN-SEQ; the k-tuple LSH construction of §3 is the first building block. Remaining work is the rigorous analysis of ρ-values, an end-to-end prototype on a public driving-video corpus, and a comparison to CLIP-based video baselines. In parallel I am adapting NEEDLE to two new verticals—*patent figure retrieval* and *animal-in-the-wild tracking*—to test how far the generate-then-verify recipe transfers across domains.

(O3) From FIM completion to code-edit and code-generation agents. DELULU and MIRAGE operate at the FIM-completion granularity used by today's inline autocomplete. The next natural step is to lift the same generate-then-verify methodology to richer code-agent settings—multi-file *code-edit* agents (apply, refactor, fix-the-failing-test) and full *code-generation* agents that produce entire

Figure 4: A DELULU sample (T3): an IMPORT hallucination in Python. The golden completion (`from PIL import ...`) is replaced by a plausibly-wrong variant (`from Pillow import ...`); a per-sample Docker container verifies that the hallucinated form raises `ModuleNotFoundError`, providing execution-grade ground truth that MIRAGE consumes as a hard negative.

modules from specifications. Open questions include how to construct execution-verified hallucination benchmarks at the agent-trajectory level and whether synthetic hard-negative trajectories transfer the same gains we observed for FIM.

Cross-cutting contribution. Across the three thrusts (Table 1), the recurring abstraction is a *generate-then-verify operator* that fits naturally into a data-management algebra: its inputs are a description (a coverage cell, a natural-language query, a FIM context), its body is a foundation-model call, and its acceptance predicate is a lightweight test (rejection sampling, ensemble NN-vote, Docker run, judge panel). Articulating this operator—its cost model, its statistical guarantees, and its composition rules—is the principal conceptual contribution my dissertation aims to make to the VLDB community.

ACKNOWLEDGMENTS

This research is supervised by Prof. Abolfazl Asudeh and benefited from discussions with my committee: Boris Glavic, Sourav Medya, Stavros Sintos, and Srinivasan Parthasarathy. DELULU and MIRAGE were developed during internship at Microsoft Code|AI.

REFERENCES

[1] Alekh Agarwal, A. Beygelzimer, Miroslav Dudík, J. Langford, and Hanna M. Wallach. 2018. A Reductions Approach to Fair Classification, In International Conference on Machine Learning. *International Conference on Machine Learning*.

[2] Abolfazl Asudeh, Zhongjun Jin, and H. V. Jagadish. 2018. Assessing and Remedying Coverage for a Given Dataset, In IEEE International Conference on Data Engineering. *IEEE International Conference on Data Engineering*. https://doi.org/10.1109/ICDE.2019.00056

[3] Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, et al. 2022. Efficient Training of Language Models to Fill in the Middle, In arXiv.org. *arXiv.org*. https://doi.org/10.48550/arXiv.2207.14255

[4] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv.org* (2021).

[6] Anne Driemel and Francesco Silvestri. 2017. Locality-Sensitive Hashing of Curves, In International Symposium on Computational Geometry. *International Symposium on Computational Geometry*, 37:1–37:16. https://doi.org/10.4230/LIPIcs.SoCG.2017.37

[7] Mahdi Erfanian and Abolfazl Asudeh. 2026. NeedleDB: A Generative-AI Based System for Accurate and Efficient Image Retrieval using Complex Natural Language Queries. *arXiv preprint arXiv:2603.27464* (2026).

[8] Mahdi Erfanian, Mohsen Dehghankar, and Abolfazl Asudeh. 2024. Needle: A Generative AI-Powered Multi-modal Database for Answering Complex Natural Language Queries. *arXiv preprint arXiv:2412.00639* (2024).

[9] Mahdi Erfanian, Boris Glavic, and Abolfazl Asudeh. 2025. Robustify: Task-Aware Robustness Improvement Using Synthetic Data Augmentation. (2025). Under review.

[10] Mahdi Erfanian, H. V. Jagadish, and Abolfazl Asudeh. 2024. Chameleon: Foundation Models for Fairness-Aware Multi-Modal Data Augmentation to Enhance Coverage of Minorities. *Proceedings of the VLDB Endowment* 17, 11 (2024), 3470–3483. https://doi.org/10.14778/3681954.3682014 arXiv:2402.01071.

[11] Mahdi Erfanian, Stavros Sintos, and Abolfazl Asudeh. 2026. Locality-Sensitive Hashing for Tuples of Curves: Approximate Nearest Neighbor for Video Retrieval. (2026). Manuscript in preparation;.

[12] Mahdi Erfanian, Nelson Daniel Troncoso, Aashna Garg, Amabel Gale, Xiaoyu Liu, Pareesa Ameneh Golnari, and Shengyu Fu. 2026. Delulu: A Verified Multi-Lingual Benchmark for Code Hallucination Detection in Fill-in-the-Middle Tasks. *arXiv preprint arXiv:2605.07024* (2026).

[13] Mahdi Erfanian, Nelson Daniel Troncoso, Aashna Garg, Amabel Gale, Xiaoyu Liu, Pareesa Ameneh Golnari, and Shengyu Fu. 2026. Synthetic Hallucinations, Real Gains: Hard Negatives from Frontier Models for FIM Hallucination Mitigation. *arXiv preprint arXiv:2606.03130* (2026).

[14] Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida I. Wang, Eric Wallace, et al. 2022. InCoder: A Generative Model for Code Infilling and Synthesis, In International Conference on Learning Representations. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2204.05999

[15] Linyuan Gong, Sida Wang, Mostafa Elhoushi, and Alvin Cheung. 2024. Evaluation of LLMs on Syntax-Aware Code Fill-in-the-Middle Tasks. *International Conference on Machine Learning* (2024). https://doi.org/10.48550/arXiv.2403.04814

[16] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, et al. 2024. Qwen2.5-Coder Technical Report. *arXiv.org* (2024).

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, et al. 2021. Scaling Up Visual and Vision-Language Representation Learning with Noisy Text Supervision, In International Conference on Machine Learning. *International Conference on Machine Learning*.

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, et al. 2021. Learning Transferable Visual Models from Natural Language Supervision, In International Conference on Machine Learning. *International Conference on Machine Learning*.

[19] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2019. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization, In arXiv.org. *arXiv.org*.

[20] Burr Settles. 2009. Active Learning Literature Survey. *University of Wisconsin–Madison Computer Sciences Technical Report 1648* (2009).

[21] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and H. V. Jagadish. 2022. Representation Bias in Data: A Survey on Identification and Resolution Techniques, In ACM Computing Surveys. *Comput. Surveys*. https://doi.org/10.1145/3588433

[22] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, et al. 2021. FLAVA: A Foundational Language and Vision Alignment Model, In Computer Vision and Pattern Recognition. *Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR52688.2022.01519

[23] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, et al. 2018. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning, In Computer Vision and Pattern Recognition. *Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr42600.2020.00271

[24] Yidan Zhang, Ting Zhang, Dong Chen, Yujing Wang, Qi Chen, et al. 2023. IRGen: Generative Modeling for Image Retrieval, In European Conference on Computer Vision. *European Conference on Computer Vision*. https://doi.org/10.48550/arXiv.2303.10126

[25] Yujing Zuo et al. 2024. SceneDiff: Diffusion-based Scene Retrieval. In *Proc. ACM Multimedia (MM)*.