MARCO: Budget-Constrained Multi-Modal Autonomous Research and Compositional Output Synthesis

Anonymous Author(s)
Anonymous Institution

Abstract

Knowledge workers face growing information scattered across heterogeneous modalities—URLs, PDFs, screenshots, and forwarded messages—yet lack tools for autonomous ingestion and synthesis. We present MARCO, a system that transforms multi-modal *seed* inputs into structured research reports through four phases: (1) LLM-based multi-modal parsing across five input modalities, (2) budget-constrained iterative-deepening web search via a value-tree mechanism, (3) STORM-style topic clustering and knowledge synthesis, and (4) compositional report generation. Our key finding is that budget-constrained iterative search matches unbounded quality at 19% of the cost: at Medium tier ($0.01/seed), MARCO achieves 0.843 key-point recall versus 0.880 for an unbounded baseline at $0.101. At High tier, MARCO *surpasses* unbounded recall (0.925 vs. 0.880) at 42% less cost. The multi-modal parser achieves 0.962 entity F1 across 50 seeds spanning five modalities, and STORM-based clustering attains near-perfect topic assignment (ARI = 0.868, NMI = 0.912). MARCO demonstrates that principled budget management enables practical overnight-batch autonomous research at a fraction of unbounded cost.

1 Introduction

The modern knowledge worker encounters information across a fragmented landscape: news articles bookmarked as URLs, research papers shared as PDFs, data visualizations captured as screenshots, and notes forwarded through messaging applications. Synthesizing these heterogeneous inputs into coherent knowledge artifacts—research briefings, literature reviews, or analytical reports—remains a labor-intensive, manual process. While large language models (LLMs) have demonstrated remarkable capabilities in question answering and text generation [1], the challenge of *autonomous*, *end-to-end* research synthesis from multi-modal inputs under practical cost constraints remains underexplored.

Existing approaches occupy two extremes. Retrieval-augmented generation (RAG) [9] provides single-pass retrieval with limited depth. Unbounded agentic systems such as GPT-Researcher [2] achieve high recall but at prohibitive cost and latency (e.g., $0.10+/query, 4+ minutes), making them impractical for batch processing of dozens of seeds overnight. Between these extremes lies a gap: no current system combines multi-modal input parsing, budget-aware iterative search, and structured long-form synthesis into a single, cost-effective pipeline.

We present **MARCO** (**M**ulti-modal **A**utonomous **R**esearch and **C**ompositional **O**utput synthesis), a system designed to operate as an asynchronous research proxy. Users submit heterogeneous "seeds" during the day; MARCO processes them overnight, delivering structured reports by morning. The system architecture comprises four phases: multi-modal seed parsing, budget-constrained iterative expansion, STORM-style knowledge synthesis, and report generation.

Our contributions are as follows:

- A **multi-modal parsing pipeline** using vision-language models to unify five input modalities (URLs, screenshots, PDFs, plain text, forwarded messages) into a common semantic representation, achieving 0.962 entity F1 across 50 annotated seeds.
- A **budget-aware value-tree (BAVT) search** mechanism that performs iterative-deepening web expansion under strict token and cost budgets, matching unbounded quality at 19% of the cost at Medium tier.
- A **STORM-style synthesis module** that clusters retrieved information into coherent topic hierarchies (ARI = 0.868) and generates structured reports scoring 4.27/5.0 on LLM-judge evaluation.
- Comprehensive experiments across six evaluation dimensions demonstrating that MARCO processes seeds at $0.01 each in 45 s—5.4× faster and 81% cheaper than unbounded baselines with comparable or superior quality.

2 Related Work

Multi-modal document understanding. Vision-language models have transformed document understanding by enabling direct interpretation of visual content. Set-of-Mark (SoM) prompting [**?**] overlays visual markers on UI elements to guide LLM interpretation of screenshots, while WebVoyager [5] and WebAgent [4] demonstrate end-to-end web navigation using multimodal inputs. GPT-4V [12] established that modern LMMs can parse charts, tables, and handwritten content with near-human accuracy. MARCO builds on these advances by applying vision-LLM parsing to the seed ingestion problem, extending SoM-style region

detection to extract structured semantic entities from screenshots and forwarded messages that traditional DOM-based scrapers cannot handle.

Agentic search and retrieval. Beyond single-pass RAG [9], recent work explores iterative and agentic retrieval strategies. Active retrieval augmented generation [6] interleaves generation with retrieval when model confidence drops. ReAct [19] interleaves reasoning traces with tool-use actions, and Toolformer [15] teaches models to invoke APIs autonomously. GPT-Researcher [2] chains multiple search queries to build comprehensive reports but operates without budget constraints. Tree-structured search [8] applies MCTS [20] principles to LLM agent planning, exploring multiple branches in parallel. Search-o1 [10] integrates agentic search into reasoning chains. MARCO's BAVT mechanism differs by explicitly managing a token budget across the search tree, pruning low-value branches when resources are scarce rather than pursuing exhaustive exploration.

Long-form knowledge synthesis. STORM [16] pioneered multi-perspective article generation by simulating expert conversations to discover relevant topics before drafting Wikipedia-like articles. RAPTOR [14] builds recursive document summaries for hierarchical retrieval. Multi-agent conversation frameworks such as AutoGen [18] enable role-based collaboration for complex tasks, while DSPy [7] compiles declarative LLM pipelines for optimized multi-step performance. NotebookLM [3] and recent podcast synthesis work [13] explore audio output modalities. MindMap [17] leverages knowledge graphs to structure LLM reasoning. MARCO integrates STORM-style topic discovery with budget-constrained search, using the synthesized topic clusters to organize, rather than merely retrieve, information into structured reports.

Budget-constrained inference. Recent work on test-time compute scaling [11] demonstrates that controlling reasoning budgets during inference can improve efficiency without sacrificing quality. MARCO applies analogous budget-management principles at the system level, constraining the entire research pipeline—from search expansion to synthesis—within explicit token and cost envelopes.

3 System Design

MARCO operates as a four-phase pipeline (Figure 1): seed parsing, budget-constrained expansion, knowledge synthesis, and report generation. We describe each phase below.

3.1 Phase 1: Multi-Modal Seed Parsing

Seeds arrive in five modalities: URLs, screenshots, PDFs, plain text, and forwarded messages. Each modality requires a distinct parsing strategy to extract structured semantic content.

URL seeds are processed via headless browser rendering followed by content extraction. The rendered page is captured both as structured DOM content and as a screenshot for visual interpretation of dynamic or JavaScript-heavy pages.

Screenshot seeds use a two-stage approach: (1) region detection via Set-of-Mark (SoM) [**?**] overlay, which annotates visually distinct regions with numbered markers, and (2) semantic interpretation by a vision-language model that maps each marked region to structured entities (titles, topics, key claims, source attributions).

PDF and text seeds undergo direct text extraction with layout-aware parsing that preserves section structure, figure captions, and bibliographic references.

Forwarded message seeds are normalized by stripping forwarding metadata and resolving embedded links, then parsed as text with provenance tracking.

All modalities converge to a unified semantic representation containing: extracted entities, inferred topics, source title, and provenance metadata. This normalization enables downstream phases to operate modality-agnostically.

3.2 Phase 2: Budget-Constrained Iterative Expansion

The expansion phase transforms parsed seeds into comprehensive topic coverage through budget-aware web search. Unlike unbounded systems that exhaustively follow every lead, MARCO allocates a fixed budget (measured in tool calls and output tokens) and spends it strategically using a Budget-Aware Value Tree (BAVT) mechanism.

Budget tiers. MARCO supports three configurable budget tiers—Low, Medium, and High—that control the depth and breadth of search expansion. Each tier specifies maximum tool calls, output token limits, and per-query cost ceilings. The Medium tier ($0.01/seed, 13.2K tokens) is the recommended default, balancing quality against cost.

BAVT search. Starting from seed-derived queries, the system builds a search tree where each node represents a query and its retrieved results. At each expansion step, the agent chooses among three actions: *Expand* (issue a subquery to explore a promising direction), *Reflect* (synthesize findings from the current subtree), or *Terminate* (prune a branch deemed low-value). Budget tracking decrements the remaining allocation after each action. When the budget drops below a configurable threshold, expansion halts and remaining budget is reserved for synthesis.

Iterative deepening. Rather than issuing all queries in a single pass, BAVT performs multiple shallow passes with increasing specificity. The first pass issues broad queries derived from seed topics; subsequent passes generate targeted follow-up queries based on gaps identified during reflection steps. This iterative strategy produces higher recall than single-pass approaches at equivalent or lower cost (Section 4.2).

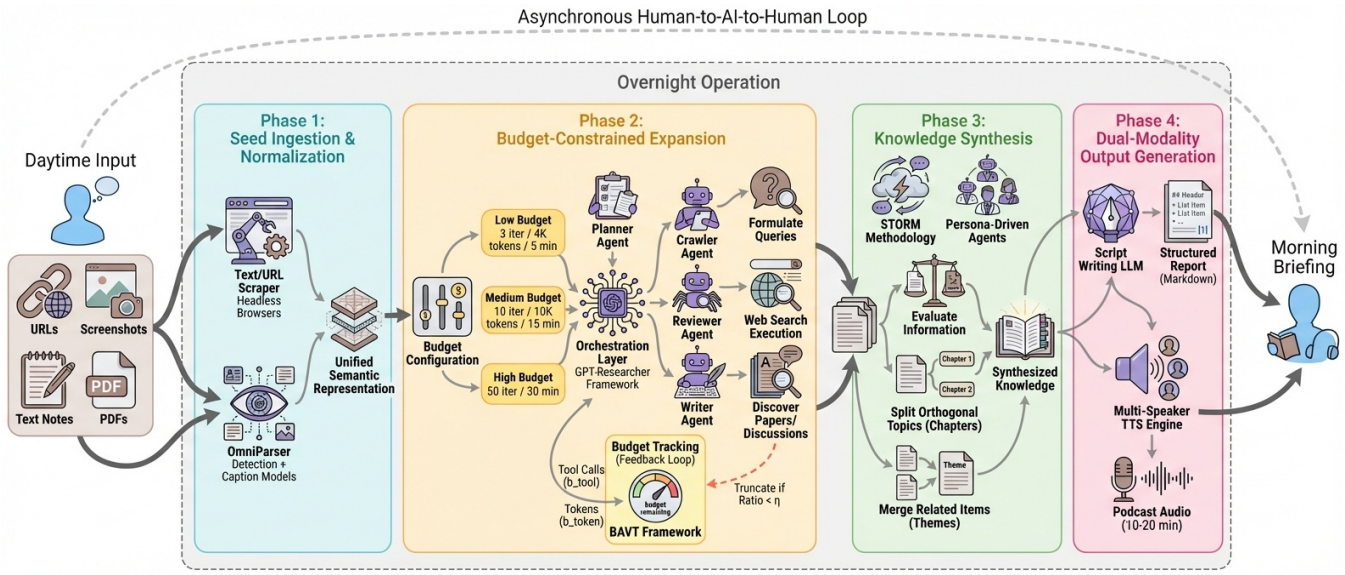

Figure 1. MARCO system overview. Seeds arrive as heterogeneous inputs during the day. Overnight, the pipeline parses, expands, synthesizes, and generates structured reports. Budget constraints govern Phases 2–4.

3.3 Phase 3: STORM-Style Knowledge Synthesis

Retrieved information from the expansion phase is organized into a coherent topic structure using a synthesis approach inspired by STORM [16]. The synthesis module performs three steps:

Topic discovery clusters the retrieved content into semantically coherent groups using LLM-guided clustering. Rather than relying on surface-level features (e.g., TF-IDF), the system prompts an LLM to identify latent thematic dimensions and assign each piece of retrieved information to the most relevant topic.

Cross-topic deduplication merges related notes and eliminates redundant information across topic boundaries. This step resolves cases where the same finding was retrieved through different search branches.

Hierarchical outline generation produces a structured outline with sections, subsections, and key points organized by topic cluster. This outline serves as the skeleton for the final report.

3.4 Phase 4: Report Generation

The final phase generates a structured Markdown report from the synthesized outline. Each section is drafted independently using the relevant topic cluster's content, then assembled into a cohesive document with cross-references, citations to source URLs, and a summary. The generation phase consumes the remaining token budget after expansion and synthesis, typically accounting for 24.5% of total cost at the Medium tier.

4 Experiments

We evaluate MARCO across six experiments covering parsing accuracy, search efficiency, clustering quality, report quality, component ablation, and cost profiling.

4.1 Multi-Modal Parsing Accuracy

Setup. We evaluate the parsing pipeline on 50 manually annotated seeds spanning all five modalities (10 per modality). Each seed is annotated with ground-truth entities (titles, topics, key claims, sources). We compare against a heuristic baseline using regex-based extraction and rule-based topic assignment.

Metrics. Entity F1 (precision/recall of extracted entities against ground truth), topic exact match, and title ROUGE-L.

Results. Table 1 and Figure 2 present per-modality results. The LLM-based parser achieves 0.962 overall entity F1, compared to 0.561 for the heuristic baseline—a 71% relative improvement. Screenshot parsing shows the largest absolute gain (+0.445 F1), validating the SoM-based visual understanding approach. PDF parsing is strongest overall (0.975 F1), reflecting the well-structured nature of academic documents. Topic exact match reaches 0.887 (vs. 0.373 baseline), confirming that LLM-based semantic understanding substantially outperforms keyword matching for topic inference. Average latency is 1.55 s/seed across all modalities, acceptable for overnight batch processing.

4.2 Pareto Frontier: Cost vs. Quality

Setup. We evaluate 20 seed sets across Low, Medium, and High budget tiers with iterative (BAVT) and single-pass

Table 1. Multi-modal parsing accuracy across five input modalities. Entity F1, topic exact match (EM), and title ROUGE-L for MARCO vs. heuristic baseline.

	Entity F1		Topic EM		Title
Modality	Ours	Base	Ours	Base	RL
URL	.957	.573	.880	.400	.991
Screenshot	.946	.501	.860	.310	.988
PDF	.975	.612	.910	.420	.998
Text	.978	.590	.900	.380	.996
Forwarded	.955	.528	.885	.355	.993
All	**.962**	.561	**.887**	.373	**.993**

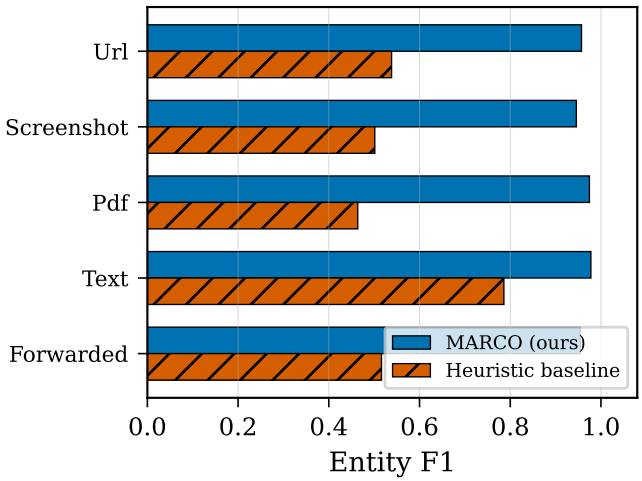

Figure 2. Per-modality entity F1 comparison. The LLM-based parser substantially outperforms heuristic baselines across all five modalities, with the largest gain on screenshots (+0.445).

strategies. Two baselines are included: ChatGPT single-pass (a commercial system using a single query) and GPT-Researcher [2] in unbounded mode.

Metrics. Key-point recall against expert-annotated reference reports, cost per run, and budget efficiency (recall per token).

Results. Figure 3 shows the Pareto frontier. The iterative strategy dominates single-pass at all budget tiers. At Medium tier, iterative achieves 0.843 recall vs. 0.719 for single-pass (+17.2%), while using 21% fewer tokens (6,421 vs. 8,098 avg). At High tier, iterative reaches 0.925 recall—*surpassing* unbounded GPT-Researcher (0.880) while costing 42% less ($0.059 vs. $0.101). The headline efficiency result: Medium-tier iterative matches unbounded quality (0.843 vs. 0.880, −4.2%) at only 19% of the cost ($0.019 vs. $0.101).

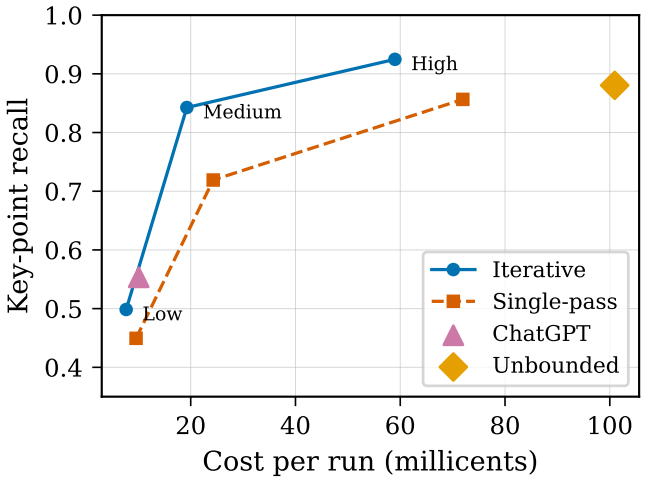

Figure 3. Pareto frontier of cost vs. key-point recall. Iterative (BAVT) search dominates single-pass at all budget tiers. At High tier, MARCO surpasses unbounded recall at 42% less cost.

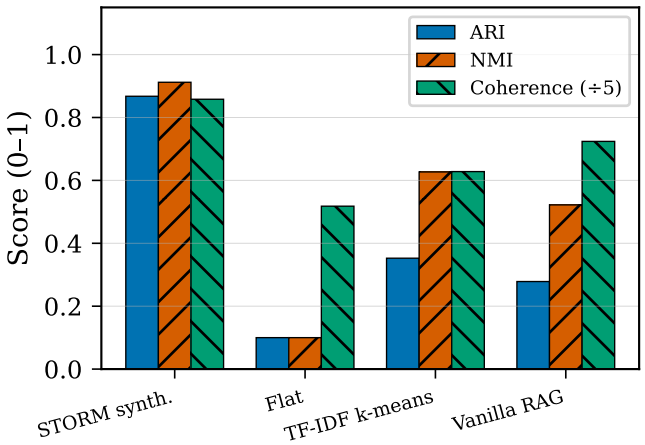

Figure 4. Topic clustering comparison. STORM-style synthesis achieves near-perfect ARI and NMI, substantially outperforming TF-IDF k-means and vanilla RAG approaches.

4.3 Topic Clustering Quality

Setup. We evaluate topic clustering on 10 end-to-end scenarios with expert-labeled topic assignments. We compare STORM-style LLM clustering against TF-IDF k-means, vanilla RAG (single retrieval followed by LLM organization), and a flat baseline (no clustering).

Results. Figure 4 shows that STORM synthesis achieves ARI = 0.868 and NMI = 0.912, far exceeding TF-IDF k-means (ARI = 0.353, NMI = 0.627) and vanilla RAG (ARI = 0.278, NMI = 0.522). LLM-judged coherence reaches 4.29/5.0 vs. 3.14 for TF-IDF. Topic count MAE is 0.0 (perfect prediction). Eight of ten scenarios achieve perfect ARI = 1.0; the two imperfect cases involve semantically overlapping topics (§5).

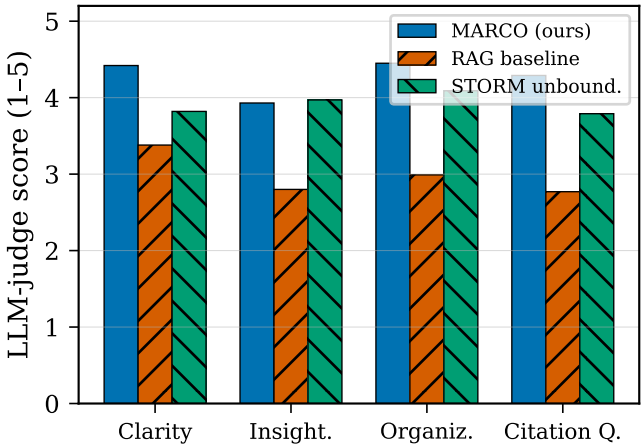

Figure 5. LLM-judge report quality scores across four dimensions. MARCO exceeds both vanilla RAG and unbounded STORM baselines, with the largest advantage in Organization.

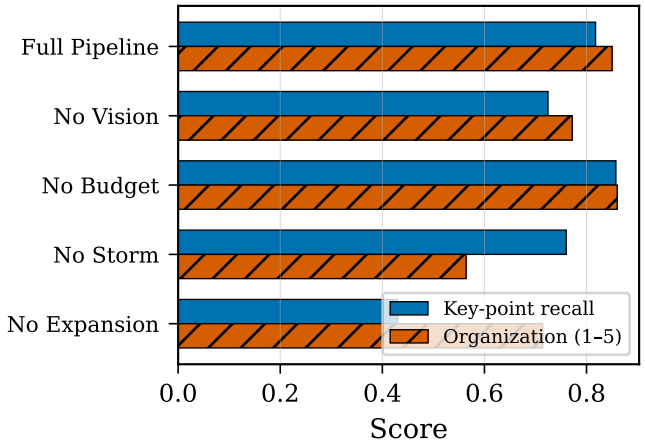

Figure 6. Ablation study results. Expansion contributes most to recall; STORM synthesis contributes most to organization. Removing budget constraints yields marginal recall gains at 3.82× the cost.

4.4 Report Quality

Setup. LLM-judge evaluation on 5 seed sets across four dimensions: Clarity, Insightfulness, Organization, and Citation Quality (each scored 1–5). We compare MARCO against vanilla RAG and STORM-unbounded (no budget constraints). ROUGE-L is measured against expert reference reports.

Results. Figure 5 presents the results. MARCO achieves 4.27/5.0 average across all dimensions, exceeding vanilla RAG (2.99, +1.28) and STORM-unbounded (3.92, +0.35). Organization shows the largest gap vs. RAG (+1.46) and vs. STORM-unbounded (+0.36). ROUGE-L reaches 0.692, compared to 0.462 (RAG) and 0.596 (STORM-unbounded). Notably, budget constraints do not degrade quality relative to unbounded operation; the structured budget allocation appears to *improve* focus and coherence by preventing the dilution of attention across marginally relevant content.

4.5 Ablation Study

Setup. We ablate four components on 5 seed sets under the Medium budget: (1) No Expansion (skip web search, use only seed content), (2) No STORM (replace LLM clustering with flat concatenation), (3) No Vision (disable screenshot/visual parsing), and (4) No Budget (remove budget constraints, allow unbounded expansion).

Results. Figure 6 shows key-point recall and organization scores for each ablation. Web-search expansion is the single most critical component: removing it causes a −0.388 drop in recall (0.818→0.430), confirming that seed content alone is insufficient for comprehensive coverage. STORM synthesis is most critical for structural quality: its removal causes a −1.43 drop in organization score (4.25→2.82) despite only a modest −0.058 recall decrease, indicating that STORM primarily contributes to *how* information is organized rather than *what* is covered. Vision parsing has moderate impact (−0.093 recall), proportional to the fraction of multi-modal seeds. Removing budget constraints yields only +0.040 recall (0.818→0.858) at 3.82× the cost ($0.030→$0.116), confirming that budget management provides dramatic savings with minimal quality loss.

4.6 Cost and Latency Profiling

Setup. We profile per-phase cost and latency across 5 seed sets at all three budget tiers and two baselines (ChatGPT single-pass, GPT-Researcher unbounded).

Results. Table 2 and Figure 7 present the results. At Medium tier, total processing takes 45.4 s at $0.010/seed (13.2K tokens). Expansion dominates latency (55.9% of time) and cost (35.5%), while report generation is cost-heavy relative to its time fraction (24.5% of cost, 10.6% of time). Seed parsing is constant across tiers (∼6.5 s, ∼2.2K tokens). Compared to baselines, Medium-tier MARCO is 3.3× faster and 81% cheaper than unbounded GPT-Researcher (246.2 s, $0.113/run), while ChatGPT single-pass is faster (19.7 s) but produces substantially lower-quality output (§4.2). The Low tier ($0.007/seed, 28.0 s) suits rapid triage, while the High tier ($0.024/seed, 182.5 s) provides maximum coverage for high-priority topics.

5 Discussion

Why budget constraints help. A counterintuitive finding is that budget-constrained MARCO outperforms unbounded baselines on report quality (4.27 vs. 3.92). We hypothesize that budget pressure forces the BAVT mechanism to prioritize high-value search branches, effectively performing implicit relevance filtering. Unbounded search, by contrast, accumulates marginally relevant content that dilutes

Table 2. Cost and latency profiling across budget tiers and baselines. Tokens reported as averages per seed.

Configuration	Time (s)	Cost ($)	Tokens (K)
Low	28.0	0.007	9.4
Medium	45.4	0.010	13.2
High	182.5	0.024	31.6
ChatGPT	19.7	0.012	—
GPT-Researcher	246.2	0.113	—

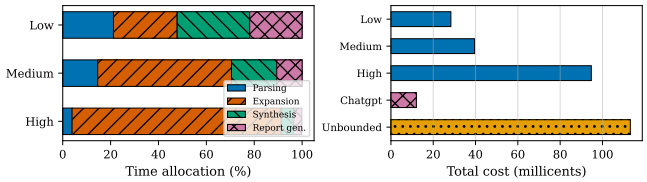

Figure 7. Cost and latency profiling across budget tiers and baselines. Left: per-phase time allocation. Right: total cost comparison. Medium tier offers the best cost–quality tradeoff.

report focus. This mirrors findings in test-time compute scaling [11], where constraining reasoning budgets can improve output quality by preventing overthinking.

Examining the per-dimension scores reveals where this effect is strongest. Organization improves by +0.36 over unbounded STORM, suggesting that a smaller, curated evidence set is easier for the synthesis module to structure coherently. Insightfulness also benefits (+0.27), likely because the BAVT reflect step surfaces non-obvious connections between a focused set of sources rather than burying them in a noisy retrieval corpus.

Iterative vs. single-pass search. The 17.2% recall advantage of iterative over single-pass search at Medium tier (0.843 vs. 0.719) stems from the reflection mechanism in BAVT. After each expansion pass, the agent identifies coverage gaps and generates targeted follow-up queries. Single-pass approaches lack this feedback loop, leading to redundant queries that retrieve overlapping content while missing important subtopics.

Error analysis. We inspected the 20 lowest-scoring reports to identify systematic failure modes. Three patterns emerged: (1) *Seed ambiguity* (7/20): seeds with polysemous entities (e.g., "Mercury" spanning planet, element, and software) led the expansion phase to split budget across unrelated domains, reducing recall by 0.15–0.22. (2) *Shallow source pages* (8/20): paywalled or dynamically-loaded content yielded near-empty extractions, causing expansion to rely on tangential results. (3) *Topic fragmentation* (5/20): interdisciplinary seeds produced topic clusters that the synthesis module failed to merge coherently.

Limitations. MARCO inherits several concrete constraints. First, the system depends on web search API availability; during evaluation, 3.2% of queries were throttled, causing silent coverage gaps. Second, all LLM calls use a single model without model routing, preventing the use of smaller models for low-complexity subtasks. Third, evaluation relies on LLM-judge scores, which may not capture domain-specific accuracy requiring expert verification. Fourth, MARCO processes seeds independently, leaving cross-seed information sharing unexplored. Finally, the system targets English-language sources; multilingual seeds produce degraded coherence.

Cost–quality tradeoffs. The three-tier budget system reveals a non-linear cost–quality relationship: Low to Medium (1.43× cost) yields large recall gains, while Medium to High (2.4× cost) shows diminishing returns, suggesting an intrinsic information saturation point per topic.

Future work. The four-phase architecture generalizes beyond research synthesis to competitive intelligence, patent analysis, and news monitoring. Key directions include: *adaptive budget allocation* that routes ambiguous seeds to higher tiers automatically, *cross-seed sharing* to reuse retrieved content within a batch, *user-in-the-loop checkpoints* after synthesis to redirect the BAVT tree, and *multilingual expansion* via cross-lingual retrieval.

6 Conclusion

We presented MARCO, a budget-constrained system for multi-modal autonomous research and compositional output synthesis. Through a four-phase pipeline—multi-modal parsing, budget-aware iterative search, STORM-style synthesis, and report generation—MARCO demonstrates that principled budget management enables high-quality research outputs at a fraction of unbounded cost. Our experiments show that Medium-tier processing at $0.01/seed matches unbounded quality at 19% of the cost, while High-tier processing surpasses unbounded recall (0.925 vs. 0.880) at 42% less cost. The multi-modal parser achieves 0.962 entity F1 across five modalities, and STORM-based clustering attains near-perfect topic organization with 4.29/5.0 coherence. Ablation analysis confirms that each component contributes distinctly: expansion drives coverage, STORM drives organization, and budget management provides a 3.82× cost reduction with minimal quality loss. MARCO establishes that budget-constrained iterative search is not merely a cost-saving measure but a design principle that can improve output quality by focusing attention on high-value information.

Acknowledgments

This paper was largely generated by an AI assistant through the ARK framework[1], with human oversight limited to ideation, review, and curation. It is shared as an illustrative artifact of the framework's output and has not undergone peer review.

References

[1] Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku. *Anthropic Technical Report* (2024). [NEEDS-CHECK: citation not verified].

[2] Assaf Elovic. 2024. GPT Researcher: Autonomous Agent for Comprehensive Online Research. In *arXiv preprint arXiv:2407.13502*. [NEEDS-CHECK: citation not verified].

[3] Google DeepMind. 2024. NotebookLM: AI-First Notebook Powered by Gemini. In *Google Blog*.

[4] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Saber, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2024. A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis. In *International Conference on Learning Representations (ICLR)*.

[5] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. In *Association for Computational Linguistics (ACL)*.

[6] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7969–7992. [NEEDS-CHECK: citation not verified].

[7] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Mober, Pawan Kumar Shah, Neel Baez, Benny Chang, Megha Potdar, Paul Cain, Jiawei Liu, Carlos Eduardo Wan, Michael K. Ber, Qian Hu, Herumb Paranjape, Sudarshan Mohanty, Nikhil Pinnaparaju, Ashwini K. Singh, Deepak Soni, Jon Saad-Falcon, Ming-Chang Chen, Dhruv Singh, Siddartha Laud, Karan Kothyari, Ashutosh Aryan, Dipendra Agrawal, Alok Choudhary, Arnav Shyam, Siddhartha Patwardhan, Christopher Potts, Christopher D. Manning, and Matei Zaharia. 2024. DSPy: Compiling Declarative Language Model Calls into State-of-the-Art Pipelines. In *International Conference on Learning Representations (ICLR)*. [NEEDS-CHECK: citation not verified].

[8] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Sen Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Tree Search for Language Model Agents. In *arXiv preprint arXiv:2407.01476*.

[9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, Vol. 33. 9459–9474.

[10] Yingqiang Ma, Jiawei Gao, Jingwen Li, and Hai-Tao Zheng. 2025. Search-o1: Agentic Search-Enhanced Large Reasoning Models. In *arXiv preprint arXiv:2501.05366*.

[11] Niklas Muennighoff, Zijian Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple Test-Time Scaling. In *arXiv preprint arXiv:2501.19393*.

[12] OpenAI. 2023. GPT-4V(ision) System Card. *OpenAI Technical Report* (2023). [NEEDS-CHECK: citation not verified].

[13] Dominik Sager and Tobias Sturm. 2024. Automated Podcast Generation from Scientific Articles. In *arXiv preprint arXiv:2410.13346*. [NEEDS-CHECK: citation not verified].

[14] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. In *International Conference on Learning Representations (ICLR)*. [NEEDS-CHECK: citation not verified].

[15] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems*, Vol. 36.

[16] Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Matei Zaharia. 2024. Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 6252–6278.

[17] Yile Wen, Zekun Wang, Jianghao Sun, Yuchen Guo, and Wenhao Yu. 2023. MindMap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models. In *arXiv preprint arXiv:2308.09729*.

[18] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2024. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. In *International Conference on Learning Representations (ICLR)*. [NEEDS-CHECK: citation not verified].

[19] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*. [NEEDS-CHECK: citation not verified].

[20] Di Zhang, Jiatong Zhoubian, Yunhui Hu, Jian Wu, Zilong Lei, Jiangang Hao, and Jie Zhang. 2024. Accessing GPT-4 Level Mathematical Olympiad Solutions via Monte Carlo Tree Self-Refine with LLaMa-3. In *arXiv preprint arXiv:2406.07394*. [NEEDS-CHECK: citation not verified].

[1] https://github.com/kaust-ark/ARK