Mind Reader: Latent User Demand-Guided Content Optimization for Generative Search Engine

Tong Chen[1] **Jiawei Guo**[1] **Yuxi Li**[1] **Baiming Chen**[1] **Houxing Ren**[2,*]
Zhiwei Zhang[1] **Yunxiang Zhang**[1] **Hanyang Xia**[1] **Kun Liang**[1] **Zhaoran Fan**[1,*]
[1]SenseTime Research [2]CUHK MMLab
shawn98chen@gmail.com renhouxing@gmail.com fanzhaoran@outlook.com

Abstract

Generative Search Engines (GSEs) have reshaped information retrieval, and Generative Engine Optimization (GEO) emerges to improve the content visibility in GSEs' responses. Previous methods mainly rely on empirical strategies or query-dependent preferences of GSEs for content optimization. However, they remain limited in effectiveness as they overlook the latent user search demands in queries that drive content retrieval and response generation of GSEs. To address this, we propose Mind Reader, a novel GEO method to effectively improve the content visibility within the generated responses of GSEs through content optimization guided by the extracted latent demands of user search. Specifically, we propose a decomposition-recombination query augmentation module, which enriches the query with latent semantic information by decomposing it into diverse perspectives, capturing underlying semantic information, and recombining them into variants to support subsequent optimization. Then, we propose a reasoning coverage content optimization module. By optimizing content to cover critical reasoning information of GSEs, we align the content with the user search demands, effectively improving the content visibility. Extensive experiments on widely used GEO-Bench and our proposed PC-GEO show that our method significantly outperforms baselines and effectively improves content visibility (with up to 2.44× objective metrics and 1.23× subjective metrics on average).

1 Introduction

Generative Search Engines (GSEs) (Sharma et al., 2024; Krasovitskiy et al., 2025), such as BingChat[1], Google's SGE[2], and perplexity.ai[3], have transformed information retrieval by leveraging Large Language Models (LLMs) to understand user search demands and directly generate relevant responses. Benefiting from their provided more efficient and user-friendly search experience, they have been rapidly applied in various everyday scenarios, *e.g.*, daily search (Metzler et al., 2021; Li et al., 2024) and shopping guides (Amazon, 2023; Tang et al., 2025a). Although greatly enhancing retrieval efficiency for users, they introduce non-trivial challenges for content creators. Unlike traditional search engines, GSEs shift the user experience from link retrieval to answer synthesis, making it harder for creators to understand how and whether their content is incorporated into generated responses. As a result, a large amount of content is overlooked or misrepresented by GSEs and drowned out in the information space, making the improvement of content visibility for GSEs a pressing concern for creators.

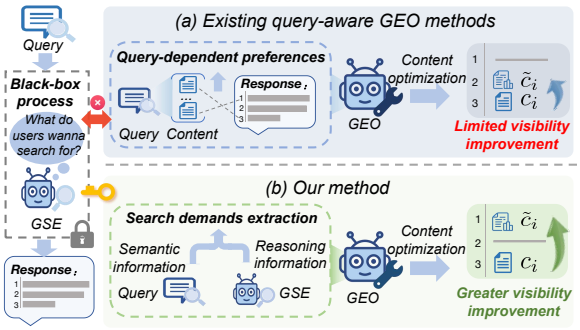

Figure 1: (a) Existing query-aware GEO methods (*e.g.*, (Wu et al., 2025)) vs. (b) our Mind Reader. Mind Reader aims to improve content visibility in the generated responses of GSEs through content optimization guided by the extracted latent user search demands.

To tackle the above challenges, several pioneering studies have explored Generative Engine Optimization (GEO). Specifically, query-agnostic GEO methods typically rely on empirical or generally heuristic strategies to optimize content. Despite showing some progress, their effectiveness is limited since they optimize content without tailoring it to user queries. More recently, query-aware GEO

*Corresponding author.
[1]https://chat.bing.com
[2]https://labs.google.com/search/
[3]https://www.perplexity.ai

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 40832–40848
July 2-7, 2026

methods have emerged that further improve the content visibility by aligning the content with the query-dependent preferences of GSEs.

Although query-aware GEO methods achieve some improvements, as shown in Figure 1, they focus on aligning content with explicit query tokens, rather than modeling the underlying user demand or task-level information needs in queries. Such demands are explicitly considered by GSEs during content retrieval and response generation (Chen et al., 2025a), and overlooking them results in unsatisfactory performance of existing methods. Specifically, we identify two key sources of this oversight: ❶ User queries are typically short and underspecified, yet implicitly encode rich latent semantic information that reflects underlying user search demands (Jansen et al., 2000; Chen et al., 2012), which is crucial for GSEs' query understanding and response generation. However, existing GEO methods primarily rely on these explicit and limited query texts, leaving the latent semantic information largely unexploited, thus significantly limiting their effectiveness. ❷ GSEs typically infer user search demands through multi-step reasoning and generate responses to satisfy these demands (Wei et al., 2022). Nevertheless, existing methods overlook such reasoning information, preventing the optimized content from aligning with the inferred user search demands and consequently resulting in suboptimal optimization outcomes.

To this end, we propose a novel GEO method, namely Mind Reader, to effectively improve the content visibility within the generated responses of GSEs through content optimization guided by the extracted latent demands of user search. Specifically, to enrich queries with their implicitly encoded latent semantics, we propose a Decomposition-Recombination Query Augmentation (**DRQA**) module that decomposes each query into diverse semantic perspectives, captures underlying semantic information, and recombines them to generate augmented queries. To achieve this, we first construct an entity graph for each input query, perform weighted random walks to extract diverse sub-graphs in a fine-grained manner, and carefully design prompts to infer their corresponding latent intents and linguistic styles. Consequently, we can effectively generate diverse query variants that explicitly expose underlying user search demands, making the demand information more accessible and thereby facilitating more effective subsequent content optimization. Then, we propose a Reasoning Coverage Content Optimization (**RCCO**) module to optimize content toward covering critical reasoning information. Specifically, we adopt Reinforcement Learning (RL) (Christiano et al., 2017; Ouyang et al., 2022) to update the optimization model and introduce a reasoning-coverage reward into the RL objective. By encouraging the optimized content to cover both query-specific personalized reasoning information and cross-query shared reasoning information distilled across the corresponding augmented query set, we align the content with user search demands in a comprehensive and stable manner. Consequently, our method effectively improves content visibility in the generated responses of GSEs.

Our contributions are summarized as follows:

- We propose a novel GEO framework, namely Mind Reader, that effectively improves the content visibility within the generated responses of GSEs by optimizing content based on the extracted latent demands of user search.
- Our framework is realized by two modules: DRQA and RCCO. The former enriches queries with latent semantic information via decomposition and recombination, while the latter optimizes the content by encouraging the GSEs' reasoning information coverage.
- We conduct extensive experiments on widely used GEO-Bench and our proposed PC-GEO, which demonstrate that our method significantly outperforms baselines and effectively improves the content visibility with up to 2.44× objective metrics and 1.23× subjective metrics on average.

2 Related Work

2.1 SEO Methods

Search Engine Optimization (SEO) aims to improve the visibility of web content in traditional search engines, and we refer to (Aryani et al., 2023; Vinutha and Prajwal, 2023) and divide it into on-page-based and off-page-based methods. Specifically, ❶ **on-page-based SEO methods** focuses on optimizing elements within the content itself. For example, Kanara *et al.* (Kanara et al., 2024) employed the keywords that were both semantically relevant to the content theme and search queries of users. An *et al.* (An and Jung, 2021) selected influential metadata features based on their frequency and weight, and incorporated them into

webpage descriptions, thereby increasing website traffic. More recently, Chodak et al. (Chodak and Błażyczek, 2023) proposed leveraging LLMs to generate optimized content, thereby effectively enhancing content visibility. In contrast, ❷ **off-page-based SEO methods** seek to improve content's authority and reputation through external signals. For instance, Google's SEO Starter Guide (Google, 2025) recommends employing the strategy of link building to enhance a website's perceived authority and credibility by search engines, thereby improving the visibility of the created content. However, GSEs employ LLMs for query understanding, content retrieval, and response generation, resulting in a fundamentally different search paradigm from the traditional search, which in turn limits the effectiveness of these conventional SEO methods.

2.2 GEO Methods

To improve the visibility of the created content within the generated responses of GSEs, some pioneering GEO works have been proposed recently, which can be mainly categorized into two types: ❶ **Query-agnostic GEO methods**: The GEO methods in this category typically rely on empirical or generally heuristic strategies to optimize content without conditioning on any specific user query. Specifically, Aggarwal *et al.* (Aggarwal et al., 2024) proposed several typical strategies, such as authoritative, statistics addition, and keyword stuffing, *etc.*, improving the visibility of the optimized content to some extent. Nestaas *et al.* (Nestaas et al.) proposed preference manipulation attacks by injecting a malicious prompt into the content of competitors, effectively lowering the competitors' ranking and improving their own ranking. Additionally, Chen *et al.* (Chen et al., 2025b) designed a multi-agent system that automated the strategic refinement of content through a collaborative analyze-revise-evaluate workflow. Moreover, Chen *et al.* (Chen et al., 2025c) proposed RAID that enabled targeted content enhancement by modeling search intent through reflective refinement across diverse informational roles. Despite showing some effectiveness, their visibility gains are limited because they optimize content in a query-agnostic manner, without tailoring it to user queries.

❷ **Query-aware GEO methods**: In contrast, a line of work has emerged that explicitly conditions content optimization on user queries, aiming to enhance content visibility by aligning the content with the query-dependent preferences of GSEs. For instance, Kumar *et al.* (Kumar and Lakkaraju, 2024) employed a greedy coordinate gradient algorithm to improve the visibility of their content for the given user query. To further improve visibility while preserving textual fluency, Yiming *et al.* (Tang et al., 2025b) proposed StealthRank, an energy-based optimization framework combined with Langevin dynamics to optimize the content with respect to the user query, which subtly yet effectively manipulated the visibility. More recently, Wu *et al.* (Wu et al., 2025) introduced AutoGEO, which explicitly analyzed GSE preferences over content given user queries and incorporates these preferences as rewards for content optimization.

Although these query-aware GEO methods achieve improvements, they remain limited due to the reliance on surface-level query tokens. In contrast, our method extracts user search demands in queries and leverages them to guide content optimization, achieving more effective results.

3 Method

Overview. We propose Mind Reader, as shown in Figure 2, to optimize content for enhancing its visibility in the generated responses of GSEs through content optimization guided by the extracted latent demands of user search, which consists of a DRQA module and a RCCO module. Specifically, the DRQA module enriches the query with latent semantic information by decomposing each query into diverse semantic perspectives, capturing underlying semantic information, and recombining them. Moreover, based on the augmented queries, the RCCO module optimizes content to cover reasoning information of GSEs, aligning content with user search demands comprehensively and stably, thereby effectively improving content visibility.

3.1 Decomposition-Recombination Query Augmentation

User queries are often short and underspecified, yet they implicitly convey rich latent semantics that capture underlying search demands. However, the existing methods largely overlook such latent semantic information, thus significantly limiting their effectiveness. In response to this, we suggest decomposing each query into diverse semantic perspectives, capturing their latent semantic information, and recombining them to generate diverse query variants that explicitly expose underlying user search demands, thereby making latent user

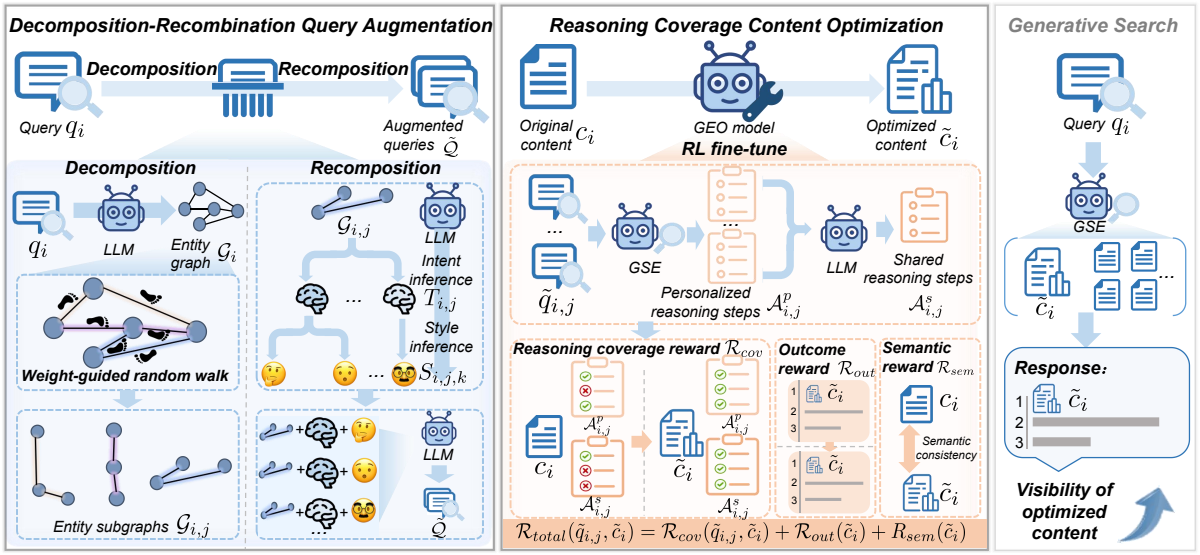

Figure 2: An overview of our Mind Reader. We first augment queries with latent semantic information via decomposition and recombination, and then optimize content by encouraging coverage of both query-specific personalized and cross-query shared reasoning information. Benefiting from these modules, our Mind Reader effectively improve the visibility of optimized content within the generated responses of GSEs.

search demand information more accessible for downstream content optimization.

Specifically, we begin by decomposing each query into diverse semantic perspectives. Given a query $q_i \in \mathcal{Q}$, we first construct an entity graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ to support structured exploration of its semantic space, which is achieved by the LLM with carefully designed prompt. The node set $\mathcal{V}_i$ consists of entity nodes derived from q_i, while the edge set $\mathcal{E}_i$ encodes semantic associations among entities, with edge weights indicating their association strengths. After constructing $\mathcal{G}_i$, we randomly sample starting nodes and perform weight-guided random walks (Nikolentzos and Vazirgiannis, 2020; Perozzi et al., 2014) to obtain entity subgraphs $\mathcal{G}_{i,j} = (\mathcal{V}_{i,j}, \mathcal{E}_{i,j})$, where the transition probability to a neighboring node is proportional to the corresponding edge weight. Repeating this process a maximum of w times yields a set of subgraphs that reflect diverse semantic perspectives of q_i.

Then, motivated by that user intent and linguistic style encode rich semantic information (Shen et al., 2014), we design a two-stage prompt to infer them, thereby capturing the latent semantic information of each entity subgraph. Since linguistic styles are typically conditioned on intents (Broder, 2002; Jansen et al., 2008), we first infer intents $T_{i,j}$ associated with $\mathcal{G}_{i,j}$ as: $T_{i,j} = \text{LLM}(p_{\text{intent}}(\mathcal{G}_{i,j}))$, where $T_{i,j} = \{t_{i,j,0}, \ldots, t_{i,j,K-1}\}$ denotes the set of inferred intents, K is the number of intents, and p_{intent} is a prompt designed for intent inference based on $\mathcal{G}_{i,j}$. Subsequently, for each intent $t_{i,j,k} \in T_{i,j}$, we infer corresponding linguistic styles $S_{i,j,k}$ based on both $\mathcal{G}_{i,j}$ and $t_{i,j,k}$ as: $S_{i,j,k} = \text{LLM}(p_{\text{style}}(\mathcal{G}_{i,j}, t_{i,j,k}))$, where $S_{i,j,k} = \{s_{i,j,k,0}, \ldots, s_{i,j,k,M-1}\}$ represents the set of inferred styles, M is the number of inferred styles, and p_{style} denotes the prompt for style inference conditioned on $\mathcal{G}_{i,j}$ and $t_{i,j,k}$.

Consequently, we recombine the entity subgraphs that capture diverse semantic perspectives of q_i with their corresponding inferred intents and linguistic styles to generate query variants as:

$$\hat{\mathcal{Q}}_i = \bigcup_{j=0}^{J-1} \bigcup_{k=0}^{K-1} \bigcup_{m=0}^{M-1} \text{LLM}\left(p_{\text{query}}(\mathcal{G}_{i,j}, t_{i,j,k}, s_{i,j,k,m})\right), \tag{1}$$

where $\hat{\mathcal{Q}}_i$ denotes the query variant set derived from q_i, and p_{query} is a prompt that synthesizes $\mathcal{G}_{i,j}$, $t_{i,j,k}$, and $s_{i,j,k,m}$ into a query. As a result, we can combine each query q_i with its corresponding query variant $\hat{\mathcal{Q}}_i$ to form an augmented query group $\tilde{\mathcal{Q}}_i$ as: $\tilde{\mathcal{Q}}_i = \{q_i\} \cup \hat{\mathcal{Q}}_i$, and ultimately construct the augmented query set $\tilde{\mathcal{Q}} = \{\hat{\mathcal{Q}}_0, \cdots, \hat{\mathcal{Q}}_{|\mathcal{Q}|-1}\}$.

Therefore, through query decomposition and recombination, we expose underlying user search demands in the tokens of augmented queries, making these demand information more accessible and thereby facilitating more effective subsequent content optimization.

3.2 Reasoning Coverage Content Optimization

Considering that existing methods are prevented from aligning content with user search demands

and result in their suboptimal optimization, we propose to optimize content to cover the reasoning information of GSEs. By introducing a reasoning-coverage reward into the RL objective, we guide the model to optimize content to cover both query-specific personalized reasoning information, which is typically provided by GSEs (Mo et al., 2025; He et al., 2025), and cross-query shared reasoning information distilled across the corresponding augmented query set. This design aligns the optimized content and user search demands comprehensively and stably, thereby effectively improving content visibility within the generated responses of GSEs.

Specifically, given a $\tilde{q}_{i,j} \in \tilde{\mathcal{Q}}_i$, to encourage the optimized content $\tilde{c}_i$ to cover the personalized reasoning information that is specific to $\tilde{q}_{i,j}$, we first design a personalized-reasoning coverage reward term $\mathcal{R}_{cov}^{per}$, which is formulated as follows:

$$\mathcal{R}_{cov}^{per}(\tilde{q}_{i,j}, \tilde{c}_i) = \frac{\sum_{n=0}^{|\mathcal{A}_{i,j}^p|-1} \text{LLM}(p_{\text{cov}}(a_{i,j,n}^p, \tilde{c}_i))}{|\mathcal{A}_{i,j}^p|}, \quad (2)$$

where $\mathcal{A}_{i,j}^p = a_{i,j,0}^p, \cdots, a_{i,j,N-1}^p$ denotes the set of personalized reasoning steps extracted by LLM using our designed prompt p_{per} from GSEs' generated responses to $\tilde{q}_{i,j}$. N is the number of steps. p_{cov} is the prompt that instructs LLM to compute the coverage score of $\tilde{c}_i$ for reasoning steps.

Moreover, since optimizing content based on diverse personalized reasoning information of different query variants may result in unstable optimization, we design a shared-reasoning coverage reward term $\mathcal{R}_{cov}^{share}$ to encourage $\tilde{c}_i$ to cover cross-query shared reasoning information distilled from personalized reasoning information of corresponding augmented-query set, which is defined as:

$$\mathcal{R}_{cov}^{share}(\tilde{q}_{i,j}, \tilde{c}_i) = \frac{\sum_{n=0}^{|\mathcal{A}_{i,j}^s|-1} \text{LLM}(p_{\text{cov}}(a_{i,n}^s, \tilde{c}_i))}{|\mathcal{A}_{i,j}^s|}, \quad (3)$$

where $\mathcal{A}_{i,j}^s = \{a_{i,j,0}^s, \cdots, a_{i,j,N-1}^s\}$ denotes the set of cross-query shared reasoning steps, and $\mathcal{A}_{i,\cdot}^s$ is distilled as: $\mathcal{A}_{i,\cdot}^s = \text{LLM}(p_{\text{share}}(\sum_{j=0}^{|\tilde{\mathcal{Q}}_i|-1} \mathcal{A}_{i,j}^p))$, and p_{share} is the prompt used by the LLM to distill the shared reasoning steps from the personalized reasoning steps of the corresponding augmented query set. Consequently, the reasoning-coverage reward $\mathcal{R}_{cov}$ is defined as follows:

$$\mathcal{R}_{cov}(\tilde{q}_{i,j}, \tilde{c}_i) = \mathcal{R}_{cov}^{per}(\tilde{q}_{i,j}, \tilde{c}_i) + \mathcal{R}_{cov}^{share}(\tilde{q}_{i,j}, \tilde{c}_i). \quad (4)$$

In addition, referring to (Wu et al., 2025), we also introduce its outcome reward $\mathcal{R}_{out}(\tilde{c}_i)$ and semantic reward $\mathcal{R}_{sem}(\tilde{c}_i)$ to encourage improving the visibility of $\tilde{c}_i$ while preserving its semantic consistency to the original content c_i. Consequently, the final reward is defined as follows:

$$\mathcal{R}_{total}(\tilde{q}_{i,j}, \tilde{c}_i) = \mathcal{R}_{cov}(\tilde{q}_{i,j}, \tilde{c}_i) + \mathcal{R}_{out}(\tilde{c}_i) + R_{sem}(\tilde{c}_i). \quad (5)$$

In particular, we follow (Wu et al., 2025) to utilize GRPO (Shao et al., 2024) to train the optimization model via RL, and the overall objective function of GRPO can be formulated as:

$$\begin{aligned} &\mathcal{L}_{GRPO}(\theta) = -\mathbb{E}_{c,i}\Big[\min\Big(r_i(\theta)A_i,\ clip\big(r_i(\theta), \\ &1-\epsilon, 1+\epsilon\big)A_i\Big)\Big] + \beta D_{KL}\big[\pi_{\theta_{old}} \parallel \pi_\theta\big], \\ &where \quad r_i(\theta) = \frac{\pi_\theta(\tilde{c}_i \mid c)}{\pi_{\theta_{old}}(\tilde{c}_i \mid c)}, A_i = \frac{\mathcal{R}_{total}(\tilde{c}_i) - \mu}{\sigma}, \end{aligned} \quad (6)$$

where π_θ and $\pi_{\theta_{old}}$ are current and old policies, ϵ and β are the clipping range and KL regularization strength, D_{KL} prevents large policy deviations, μ and σ denote the mean and standard deviation of rewards in the group, and A_i is the standardized group-relative advantage.

Therefore, the optimization model can effectively optimize content to cover both query-specific personalized and cross-query shared reasoning information, effectively improving its visibility in the generated responses of GSEs.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our methods on two datasets: a typical dataset for GEO, *i.e.*, GEO-Bench (Aggarwal et al., 2024), and our proposed dataset on the personal computer domain, *i.e.*, PC-GEO. Specifically, GEO-Bench is a widely used benchmark for GEO that includes 8000 training queries, 1000 validation queries, and 1000 test queries, each paired with relevant web content to support the response generation for GSEs. Moreover, we propose PC-GEO, which contains 3533 training and 424 test queries. The associated web content is sourced from technical forums, product review sites, and e-commerce platforms.

Baselines. To assess the effectiveness of our method, we compare it with several typical baselines. Specifically, we compare Mind Reader with representative query-agnostic GEO methods, including all nine optimization methods from (Aggarwal et al., 2024) (*i.e.*, technical terms, cite sources, keyword stuffing, unique words, authoritative, easy-to-understand, statistics addition, quotation addi-

Table 1: Comparisons in terms of objective and subjective metrics on GEO-Bench and PC-GEO for Qwen3 GSE.

Dataset	Method	Objective metrics			Subjective metrics						
		Word	**Position**	**Overall**	**Rel.**	**Infl.**	**Unique**	**Div.**	**FollowUp**	**Pos.**	**Count**
GEO-Bench	Vanilla	19.92	19.72	19.85	19.84	19.84	19.86	19.89	19.86	19.85	19.83
	Technical terms	20.55	20.44	20.51	20.07	20.10	20.10	20.11	20.08	20.13	20.09
	Cite sources	22.31	22.24	22.28	20.58	20.78	20.72	20.61	20.53	20.79	20.75
	Keyword stuffing	17.84	17.25	17.13	19.10	18.60	18.69	19.03	18.98	18.53	18.70
	Unique words	19.27	19.19	19.18	18.59	18.00	18.16	18.52	18.53	17.96	18.11
	Authoritative	19.37	19.25	19.32	19.75	19.66	19.64	19.69	19.70	19.69	19.66
	Easy-to-understand	16.19	16.15	16.20	19.29	18.89	18.75	19.14	19.14	18.84	18.94
	Statistics addition	22.85	22.71	22.53	20.01	20.19	20.34	19.92	20.10	20.12	20.20
	Quotation addition	22.81	22.72	22.74	20.63	20.84	20.84	20.58	20.49	20.89	20.78
	Fluency optimization	19.68	19.62	19.65	18.96	18.45	18.47	18.84	18.87	18.43	18.52
	RAID	20.73	20.43	20.46	20.43	20.61	20.49	20.46	20.38	20.57	20.54
	$\text{AutoGEO}_{\text{API}}$	37.69	37.87	37.89	22.47	23.91	24.81	22.92	22.80	24.04	23.59
	$\text{AutoGEO}_{\text{Mini}}$	24.69	24.80	24.91	20.93	21.26	21.11	20.94	20.82	21.29	21.17
	Ours	**53.72**	**53.13**	**53.39**	**22.87**	**25.30**	**26.75**	**23.74**	**23.33**	**25.50**	**24.78**
PC-GEO	Vanilla	26.57	26.62	26.42	27.14	26.85	26.93	27.12	27.11	26.88	27.12
	Technical terms	26.78	26.70	26.62	27.26	27.13	27.11	27.22	27.08	27.09	27.27
	Cite sources	28.47	28.46	28.32	27.53	27.49	27.57	27.44	27.32	27.57	27.51
	Keyword stuffing	19.42	19.06	19.16	25.01	23.88	24.38	25.11	25.50	23.89	24.67
	Unique words	24.43	24.26	24.21	26.46	26.02	26.11	26.46	26.61	26.03	26.36
	Authoritative	27.27	27.22	27.06	27.31	27.11	27.05	27.23	27.13	27.10	27.22
	Easy-to-understand	25.81	25.88	25.62	26.89	26.64	26.69	26.89	26.86	26.66	26.85
	Statistics addition	29.80	29.23	29.80	27.40	27.37	27.18	27.25	27.28	27.52	27.40
	Quotation addition	29.62	29.72	29.60	27.79	27.66	27.68	27.65	27.52	27.82	27.77
	Fluency optimization	25.57	25.61	25.46	26.20	25.67	25.84	26.26	26.49	25.65	26.11
	RAID	27.43	27.57	27.35	27.40	27.25	27.29	27.31	27.16	27.31	27.31
	$\text{AutoGEO}_{\text{API}}$	37.47	38.91	38.94	30.94	33.14	33.75	31.09	30.06	34.06	31.82
	$\text{AutoGEO}_{\text{Mini}}$	29.02	29.04	29.06	27.22	27.17	27.11	27.20	27.06	27.20	27.17
	Ours	**57.62**	**57.84**	**57.95**	**31.62**	**34.08**	**34.76**	**34.10**	**30.49**	**34.73**	**32.70**

tion, and fluency optimization) and RAID (Chen et al., 2025c). In addition, we also compare Mind Reader with the state-of-the-art query-aware GEO method AutoGEO (Wu et al., 2025), including two variants $\text{AutoGEO}_{\text{API}}$ and $\text{AutoGEO}_{\text{Mini}}$.

Evaluation metrics. Following (Aggarwal et al., 2024), we employ three objective metrics, *i.e.*, word, position, and overall. Specifically, word measures the weight of citation in the text of the response, position reflects the isolated impact of citation position based on exponential decay, and overall integrates both word count and position weighting to provide a comprehensive visibility measure in generative engine responses. Moreover, we also follow (Aggarwal et al., 2024) to adopt subjective metrics: relevance (Rel.), influence (Infl.), uniqueness (Unique), diversity (Div.), click likelihood (FollowUp), subjective positional prominence (Pos.), and subjective content volume (Count). All results are reported as average percentage values (%) computed over the test dataset.

Implementation details. We employ Qwen2.5-7B-Instruct (Yang et al., 2025a) as the optimization model for baselines and our method. Unless otherwise specified, all experiments are conducted against GSE based on Qwen3-30B (*i.e.*, Qwen GSE). Moreover, we set maximum subgraph extraction times w to 3. The optimization model is fine-tuned for 1 epoch with the batch size of 128 using the Adam optimizer (learning rate of $1e{-5}$, warm-up ratio of 0.03) on 8 NVIDIA A800 80GB GPUs, which takes 15 and 7.5 hours on GEO-Bench and PC-GEO, respectively.

4.2 Overall Performance

To evaluate our Mind Reader, we compare our proposed method with typical query-agnostic GEO methods and query-aware GEO methods. The results are shown in Table 1, which demonstrates that **our method outperforms baselines and ef-**

Table 2: Ablation study on GEO-Bench.

Method					Word	Position	Overall
DRQA	$\mathcal{R}_{out}$	$\mathcal{R}_{sem}$	$\mathcal{R}_{cov}^{per}$	$\mathcal{R}_{cov}^{share}$			
	✓	✓	✓	✓	29.45	29.92	29.64
✓		✓	✓	✓	40.61	41.30	40.96
✓	✓		✓	✓	52.38	52.84	52.60
✓	✓	✓			36.97	36.84	36.95
✓	✓	✓		✓	44.54	44.93	44.81
✓	✓	✓	✓		47.81	47.34	47.52
✓	✓	✓	✓	✓	**53.72**	**53.13**	**53.39**

fectively improves the content visibility within the generated responses of GSEs. Moreover, we provide some further insights and discussions as follows: ❶ **Query-agnostic GEO methods exhibit some performance.** Specifically, these methods outperform the vanilla baseline, achieving average word, position, and overall metrics of 23.31, 23.19, and 23.16, and average subjective metrics of 23.22 on two datasets. ❷ **Query-aware GEO methods achieve better effectiveness than query-agnostic ones.** Query-aware GEO methods outperform query-agnostic GEO methods, achieving average word, position, and overall metrics of 32.22, 32.66, and 32.70, and average subjective metrics of 25.97 across two datasets. We attribute these improvements to the fact that query-aware methods explicitly tailor the optimization to user queries, enabling more effective content optimization. ❸ **Our method outperforms all baselines.** Our proposed method significantly outperforms the baselines, achieving average word, position, and overall of 55.67, 55.49, and 55.67, and average semantic metrics of 28.91 across two datasets. We attribute this superiority to the fact that our method can effectively extract latent user search demands and optimize content accordingly.

4.3 Ablation Study

We perform ablation studies on GEO-Bench to analyze the effectiveness of each crucial component, thereby providing a comprehensive understanding of our method. The results are shown in Table 2, and we have several observations as follows: ❶ DRQA significantly improves the content visibility (row 1 vs. row 7), which illustrates that enriching queries with latent semantic information is critical for subsequent content optimization. ❷ The proposed $\mathcal{R}_{cov}$ is also effective in enhancing content visibility (row 4 vs. row 7), demonstrating that covering reasoning information facilitates the content optimization to be aligned with the user search

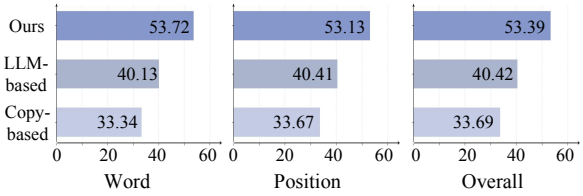

Figure 3: Evaluation of different query augmentation strategies.

demand, thus the optimized content is prioritized by the GSEs in the generated responses. ❸ $\mathcal{R}_{cov}^{share}$ helps to cover different information of diverse reasoning steps more stably, thereby achieving better optimization results. (row 6 vs. row 7) ❹ Our method yields the highest word, position, and overall, which illustrates that our proposed modules are complementary, facilitating a more comprehensive extraction of latent search demands and thereby enabling more effective content optimization. ❺ Notably, $\mathcal{R}_{sem}$ also helps to improve the content visibility during optimization (row 3 vs. row 7). This is mainly because that preserving semantic consistency is crucial for preventing semantic drift and maintaining content fluency during optimization, which in turn increases the likelihood that the optimized content is adopted by GSEs.

4.4 Discussions

To evaluate our method more comprehensively, we conduct more experiments on GEO-Bench and discuss the experimental results in this part.

Effect of query augmentation strategies. To further evaluate the effectiveness of our designed decomposition–recombination query augmentation, we compare our method with two variants equipped with different augmentation strategies: ❶ Copy-based augmentation, which simply duplicates the original query without modification; ❷ LLM-augmentation, which utilizes the LLM to generate diverse query variants based on the original query. As shown in Figure 3, although the LLM-based augmentation strategy offers modest gains over the simple copying strategy, our method, which is equipped with the designed DRQA module, significantly outperforms these two methods. We attribute this to the fact that our DRQA module enables fine-grained extraction of latent semantic information from the original query, thereby generating queries that cover diverse semantic information beyond surface-level information.

Robustness to surrogate reasoning. Although most GSEs provide the reasoning steps during the response generation, in certain scenarios, such as when interacting with proprietary or closed-source

Table 3: Performance comparisons under different reasoning chain sources. The training LLM generates CoT for optimization, while the target GSE uses a different LLM to generate responses.

Surrogate model	Word	Position	Overall
Qwen2.5-7B	43.81	44.00	44.07
Qwen3-14B	52.97	53.40	53.27
Deepseek-R1-7B	43.53	44.21	43.93
Deepseek-R1-14B	49.24	49.43	49.34

systems, the reasoning steps may not be exposed. To evaluate the robustness of our method when the ground-truth reasoning steps from the GSE are unaccessible, we employ two open-source LLMs (*i.e.*, Qwen2.5-7B (Yang et al., 2025a), Qwen3-14B (Yang et al., 2025b), Deepseek-R1-7B (Guo et al., 2025), and Deepseek-R1-14B (Guo et al., 2025)) as the surrogate model to obtain the reasoning steps utilized by RCCO module of our method. As the results in Table 3 show, although the effectiveness is slightly lower than that obtained using the ground-truth reasoning steps of GSEs, content optimized with reasoning steps generated by surrogate models still attains considerable visibility. This suggests our method is applicable even in scenarios where reasoning steps are inaccessible. Moreover, we observe that utilizing larger surrogate models yields better results due to their stronger reasoning capability. Notably, Qwen-based models outperform Deepseek ones, possibly because they share a similar architecture with the LLM of the target GSE, leading to more aligned reasoning patterns.

Effectiveness on different LLM-based GSEs. We further evaluate our method and query-aware GEO baselines for more different representative GSEs on GEO-bench, including the GSEs based on GPT (Achiam et al., 2023) and Gemini (Team et al., 2023) (*i.e.*, GPT GSE and Gemini GSE). As shown in Figure 4, our method consistently outperforms baselines across all objective metrics. These results demonstrate the effectiveness of our method across diverse LLM-based GSEs, underscoring its practical applicability in real-world settings where multiple GSEs coexist.

Evaluation of the optimization model transferability across dataset domains. To assess our transferability across dataset domains, we evaluate Mind Reader trained on GEO-Bench (general domain) using PC-GEO test dataset (personal computer domain), and conversely evaluate Mind Reader trained on PC-GEO using GEO-Bench test

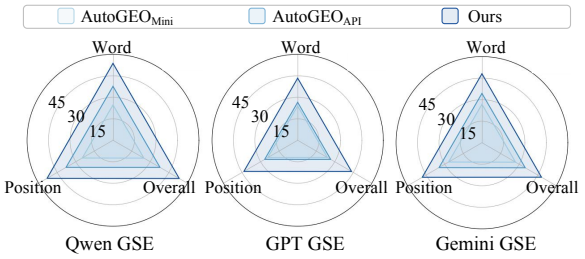

Figure 4: Effectiveness on different LLM-based GSEs.

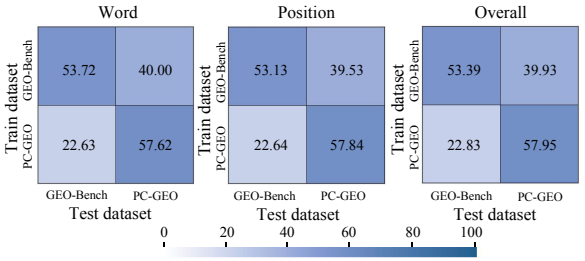

Figure 5: Cross-domain effectiveness of the optimization model.

dataset. As shown in Figure 5, Mind Reader exhibits considerable transferability when transferring from the general to the personal computer domain, whereas its performance degrades markedly in the reverse direction. This asymmetry arises because the optimization strategy learned from the general-domain dataset is more generalizable than that learned from the domain-specific dataset.

4.5 Case Study

Here, we present a case to show the effectiveness of the proposed Mind Reader, as shown in Table 4, demonstrating how our framework optimizes content to effectively improve its visibility within the generated responses of GSEs.

In the DRQA module, we construct an entity graph for the input query and extract diverse subgraphs to extract diverse semantic perspectives. Then, we infer the corresponding latent intents and linguistic styles based on theses subgraphs. Consequently, we can generate diverse query variants that incorporate the extracted latent semantic information, thereby facilitating more effective subsequent content optimization.

In the RCCO module, the expanded query is processed to derive the personalized reasoning information, alongside the shared reasoning information distilled cross the corresponding augmented query set. By comparing the original and optimized content, the case study demonstrates that our framework effectively guides content optimization: the optimized content covers the personalized reason-

Table 4: Case Study. Illustration of how our framework process the query by capturing latent semantics information (Module 1) and guiding content to cover reasoning information (Module 2).

Module 1: Decomposition-Recombination Query Augmentation

Original Query	*What are the moral obligations of the biographer?*
Expanded Queries	1. How do biographers navigate the ethical considerations when presenting complex or controversial aspects of a subject's life in their biographies? 2. How should the moral obligations of a biographer be defined in professional writing practice?

Module 2: Reasoning Coverage Content Optimization

Case Query: *"How do biographers navigate the ethical considerations when presenting complex or controversial aspects of a subject's life in their biographies?"*

Personalized Reasoning Steps ($\mathcal{A}^p$) for the Case Query:
[Step 1]: Identify core ethical principles (accuracy, fairness, privacy). → **[Step 2]**: Assess public interest and historical significance of controversial details. → **[Step 3]**: Rely on verified sources to ensure factual integrity. → **[Step 4]**: Balance transparency, harm mitigation, and contextualized narration.

Shared Reasoning Steps ($\mathcal{A}^s$) Distilled cross the Corresponding Augmented Query Set:
[Step 1]: Ensure accuracy and verifiability of biographical claims. → **[Step 2]**: Respect personal dignity and justify disclosure through legitimate public interest. → **[Step 3]**: Avoid bias, sensationalism, and undisclosed conflicts of interest. → **[Step 4]**: Situate actions within appropriate historical, cultural, and social contexts.

Original Content	**Optimized Content (Ours)**
Biographers have a moral obligation to consider the ethics of telling someone's life story. Writers, in general, have a moral responsibility to express their opinions and accept that there is more at stake than the number of copies sold. Editors also have a moral obligation to maintain editorial integrity with their writers.	Biographers have a moral obligation to represent their subjects truthfully while respecting personal dignity. This requires careful selection of verifiable facts, avoidance of speculation or sensationalism, and fair treatment of both positive and negative aspects of a subject's life. In fulfilling this responsibility, biographers must weigh the public value of disclosure against potential harm to individuals involved, and situate actions within their historical and social contexts to prevent misinterpretation. More broadly, ethical biographical writing demands accuracy, fairness, and contextual sensitivity, ensuring that moral responsibility extends beyond narrative appeal or commercial considerations.

Highlighting Convention. Blue text indicates coverage of personalized reasoning information, while purple text represents coverage of shared reasoning information.

ing information while covering shared reasoning information.

5 Conclusion

In this paper, we introduce a novel GEO method, Mind Reader, that enhances content visibility in GSE-generated responses by optimizing content under the guidance of extracted latent user search demands in queries. Specifically, Mind Reader consists of a decomposition-recombination query augmentation module and a reasoning coverage content optimization module. The former augments the query with the latent semantics via decomposition and recombination to expose underlying user search demands and facilitate more effective subsequent content optimization, and the latter optimizes content by encouraging both query-specific personalized and cross-query shared reasoning information coverage, aligning the optimized content with the user search demands, and thereby effectively improving the content visibility within the generated responses of GSEs. Extensive experiments demonstrate that Mind Reader significantly outperforms baselines, achieving remarkably improved visibility (with up to 2.44× objective metrics and 1.23× subjective metrics on average). In the future, we plan to extend Mind Reader to cross-modal generative search scenarios, thereby supporting more real-world applications.

Limitations

The primary limitation of our method is the increased computational overhead due to the query augmentation, reasoning step obtainment, and fine-tuning of the optimization model on more data. However, it is important to note that the query augmentation and reasoning step obtainment can be performed offline, and thus the only non-negligible overhead stems from training on the augmented dataset. Considering the considerable effectiveness of our method and naively increasing training data without our designed DRQA module only yields marginal gains, the increased computational overhead is acceptable. Moreover, as distributed computing resources become more accessible and cost-effective, this increased overhead can be efficiently mitigated through standard data-parallel strategies (*e.g.*, data parallelism across multiple devices). We anticipate that this limitation will diminish in practice as scalable training infrastructure becomes widely available.

Ethics Statement

The models utilized in this paper, including Qwen2.5-7B, Qwen3-14B, Qwen3-30B, Deepseek-R1-7B, Deepseek-R1-14B, GPT, and Gemini, are employed under terms that permit academic research use, and Alibaba, OpenAI, and Google allow non-commercial academic investigation. Furthermore, the data employed in this study comprises two benchmarks: GEO-Bench, which is publicly available under an open license, and PC-GEO, a domain-specific dataset is obtained from open technical forums, product review sites, and e-commerce platforms. And all datasets are used exclusively for academic research purposes.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, and Ameet Deshpande. 2024. Geo: Generative engine optimization. In *ACM KDD*, pages 5–16.

Daria Alexander, Wojciech Kusa, and Arjen P. de Vries. 2022. Orcas-i: queries annotated with intent using weak supervision. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3057–3066.

Amazon. 2023. Amazon Introduces AI Shopping Guides to Help Customers Research Products.

Sojung An and Jason J Jung. 2021. A heuristic approach on metadata recommendation for search engine optimization. *Concurrency and Computation: Practice and Experience*, 33(3):e5407.

Diah Aryani, Shine Pintor Siolemba Patiro, Aji Setiawan, and Budi Tjahjono. 2023. Comparative analysis of on-page and off-page white hat search engine optimization (seo) techniques on website popularity. *International Journal of Science, Technology & Management*, 4(3):527–533.

Andrei Broder. 2002. A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM New York, NY, USA.

ByteDance. 2025. Doubao-seed-1.6. In *Accessed: 2025-0720*.

Jiale Chen, Xuelian Dong, Wenxiu Xie, Ru Peng, Kun Zeng, and Tianyong Hao. 2025a. Llm-enhanced query generation and retrieval preservation for task-oriented dialogue. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14307–14321.

Long Chen, Dell Zhang, and Levene Mark. 2012. Understanding user intent in community question answering. In *Proceedings of the 21st international conference on world wide web*, pages 823–828.

Qiyuan Chen, Jiahe Chen, Hongsen Huang, Qian Shao, Jintai Chen, Renjie Hua, Hongxia Xu, Ruijia Wu, Ren Chuan, and Jian Wu. 2025b. Beyond keywords: Driving generative search engine optimization with content-centric agents. *arXiv preprint arXiv:2509.05607*.

Xiaolu Chen, Haojie Wu, Jie Bao, Zhen Chen, Yong Liao, and Hu Huang. 2025c. Role-augmented intent-driven generative search engine optimization. *arXiv preprint arXiv:2508.11158*.

Grzegorz Chodak and Klaudia Błażyczek. 2023. Large language models for search engine optimization in e-commerce. In *International Advanced Computing Conference*, pages 333–344. Springer.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Ms marco: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1566–1576.

Google. 2025. SEO Starter Guide.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 178 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and Weinan E. 2025. Pasa: An LLM agent for comprehensive academic paper search. In *The 63rd Annual Meeting of the Association for Computational Linguistics: ACL 2025*.

Bernard J Jansen, Danielle L Booth, and Amanda Spink. 2008. Determining the informational, navigational, and transactional intent of web queries. *Information Processing & Management*, 44(3):1251–1266.

Bernard J Jansen, Amanda Spink, and Tefko Saracevic. 2000. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227.

Aditya Pranav Kanara, Priya Kumari, and Boppuru Rudra Prathap. 2024. Python driven keyword analysis for seo optimization. In *2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1170–1176. IEEE.

Konstantin Krasovitskiy, Stelios Christou, and Demetrios Zeinalipour-Yazti. 2025. Llm-ms: A multi-model llm search engine. In *2025 IEEE 41st International Conference on Data Engineering Workshops (ICDEW)*, pages 107–114. IEEE.

Aounon Kumar and Himabindu Lakkaraju. 2024. Manipulating large language models to increase product visibility. *arXiv preprint arXiv:2404.07981*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yongqi Li, Xinyu Lin, Wenjie Wang, Fuli Feng, Liang Pang, Wenjie Li, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2024. A survey of generative search and recommendation in the era of large language models. *arXiv preprint arXiv:2404.16924*.

Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: making domain experts out of dilettantes. In *Acm sigir forum*, volume 55, pages 1–27. ACM New York, NY, USA.

Fengran Mo, Chuan Meng, Mohammad Aliannejadi, and Jian-Yun Nie. 2025. Conversational search: From fundamentals to frontiers in the llm era. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4094–4097.

Fredrik Nestaas, Edoardo Debenedetti, and Florian Tramèr. Adversarial search engine optimization for large language models. In *The Thirteenth International Conference on Learning Representations*.

Giannis Nikolentzos and Michalis Vazirgiannis. 2020. Random walk graph neural networks. *Advances in Neural Information Processing Systems*, 33:16211–16222.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Houxing Ren, Zimu Lu, Weikang Shi, Haotian Hou, Yunqiao Yang, Ke Wang, Aojun Zhou, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2025a. Alignment with fill-in-the-middle for enhancing code generation. *Preprint*, arXiv:2508.19532.

Houxing Ren, Mingjie Zhan, Zhongyuan Wu, and Hongsheng Li. 2024. Empowering character-level text infilling by eliminating sub-tokens. *Preprint*, arXiv:2405.17103.

Houxing Ren, Mingjie Zhan, Zhongyuan Wu, Aojun Zhou, Junting Pan, and Hongsheng Li. 2025b. Reflectioncoder: Learning from reflection sequence for enhanced one-off code generation. *Preprint*, arXiv:2405.17057.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Nikhil Sharma, Q Vera Liao, and Ziang Xiao. 2024. Generative echo chamber? effect of llm-powered search systems on diverse information seeking. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–17.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd international conference on world wide web*, pages 373–374.

Tian Tang, Zhixing Tian, Zhenyu Zhu, Chenyang Wang, Haiqing Hu, Guoyu Tang, Lin Liu, and Sulong Xu. 2025a. Lref: A novel llm-based relevance framework for e-commerce search. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 468–475.

Yiming Tang, Yi Fan, Chenxiao Yu, Tiankai Yang, Yue Zhao, and Xiyang Hu. 2025b. Stealthrank: Llm ranking manipulation via stealthy prompt optimization. *arXiv preprint arXiv:2504.05804*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

MS Vinutha and MR Prajwal. 2023. A survey on search engine optimization-types, techniques and factors. *Int. J. All Res. Educ. Sci. Methods (IJARESM)*, 11(8).

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yujiang Wu, Shanshan Zhong, Yubin Kim, and Chenyan Xiong. 2025. What generative search engines like and how to optimize web content cooperatively. *arXiv preprint arXiv:2510.11438*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025b. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.

Appendix

A Additional Experiments

A.1 Hyperparameter Analysis

To analyze the effect of the maximum entity subgraph extraction times w, we vary $w \in 1, 2, 3, 4, 5$ and conduct experiments on GEO-Bench against Qwen GSE. As shown in Figure 6, both the overall metric and the average subjective metric improve as w increases, since a higher w facilitates capturing more diverse semantic perspectives. Notably, the effectiveness gains from $w = 3$ to $w = 5$ are substantially smaller than those from $w = 1$ to $w = 3$. We attribute this to the fact that when $w = 3$, our method already captures most latent semantic information of the query, and further increasing w yields limited additional benefits. Therefore, we set $w = 3$ in the experiments to balance the effectiveness and computational cost of our method.

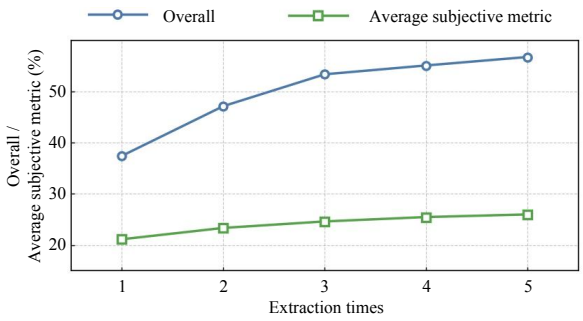

Figure 6: The impact of w in terms of overall and average subjective metrics on GEO-Bench against Qwen GSE.

A.2 Effect of Reversing Inference Order for Intents and Styles

In the DRQA module, we first infer user intents from the entity subgraph and then infer the corresponding linguistic styles conditioned on both the subgraph and the inferred intents. To validate the rationality of this inference order, we conduct experiments by reversing the inference order on GEO-Bench against Qwen GSE, *i.e.*, inferring linguistic styles from the entity subgraph first and then inferring user intents conditioned on the subgraph and styles. As shown in Table 5, reversing the inference order results in consistent performance degradation across word, position, and overall compared to our method. This result indicates that linguistic styles are typically conditioned on user intents. Without first inferring intents, style inference lacks sufficient semantic context and may even distort the

core semantic information.

Table 5: Performance comparisons under different inference order for intents and styles.

Inference order	Word	Position	Overall
Style → Intent	42.52	42.49	42.57
Intent → Style (Ours)	**53.72**	**53.13**	**53.39**

A.3 Robustness of Our Method against Diverse Queries

In real-world scenarios, even with similar search demands, user queries often exhibit diversity due to the varied ways they are expressed. Therefore, users hardly input the exactly same query used during optimization. This raises a critical question: does our method generalize beyond the specific query used during optimization? To evaluate this, we simulate realistic query diversity by generating diverse queries based on the original query with the LLM, and then measure the visibility of our optimized content based on the original query against these diverse queries. As shown in Table 6, our method maintains considerable effectiveness across diverse queries, achieving only average 0.92%, 0.39%, and 0.55% drops in terms of word, position, and overall compared to evaluation on the original query. This robustness stems from the fact that our method optimizes content to cover diverse latent user search demand information rather than overfitting to surface-level token information, enabling our method's consistent performance across diverse user queries.

Table 6: Performance comparisons of our method against original query and diverse queries.

Query type	Word	Position	Overall
Original query	53.72	53.13	53.39
Diverse query	52.80	52.74	52.84

B More Details

In this part, we provide more experimental details for reproducibility and completeness.

B.1 Dataset Details

B.1.1 GEO-Bench

GEO-Bench(Aggarwal et al., 2024) is a typical GEO benchmark that consists of 8000 training queries, 1000 validation queries, and 1000 test queries, which are obtained from nine different sources: MS MARCO (Craswell et al., 2021), ORCAS-1 (Alexander et al., 2022), Natural Questions (Kwiatkowski et al., 2019), AllSouls, LIMA (Zhou et al., 2023), Davinci-Debate (Liu et al., 2023), Perplexity.ai Discover [4], ELI-5 [5], GPT-4 Generated Queries. Each sample in GEO-Bench is organized as structured data consisting of a user query, a set of relevant web content, associated metadata tags, and optimization targets.

B.1.2 PC-GEO

As mentioned in Section 4.1, we propose PC-GEO, a GEO dataset on the personal computer domain. This section details the dataset construction process. PC-GEO is built through a multi-stage pipeline that leverages an LLM (*i.e.*, Doubao-seed-1.6 (ByteDance, 2025)) together with the web retrieval tool provided by the Volcano Ark API, as outlined below:

❶ **Query Generation.** We generate personal computer-related user queries using the LLM to simulate across diverse usage scenarios, budgets, and configuration requirements. The prompt detail is shown below:

PC-GEO: Prompt for Query Generation

Goal:
Your core objective is to continuously generate high-quality, diverse, and non-repetitive questions about laptops from the viewpoints of different user personas.

Persona & Context:
When generating questions, simulate the thinking and needs of the following user personas:

- General Consumer: Seeks value for money, may be less technically savvy, and prioritizes everyday experience and brand reputation.
- Student: Has a limited budget, diverse needs, and may require portability.
- Professional: Values efficiency, stability, battery life, and compatibility with work software.
- Gamer: Prioritizes high performance, high refresh rate displays, and cooling systems.
- Creator: Needs powerful graphics processing, large RAM, and high-resolution displays.
- Business Traveler: Extremely values portability, battery life, and durability.

[4] https://www.perplexity.ai/discover
[5] https://huggingface.co/eli5_category

Question Dimensions & Requirements:
Your questions must cover the following dimensions and must never be repeated:

- Budget & Value:
  - Recommendations for the best value laptops in specific price ranges (*e.g.*, $500–$800, $800–$1200, $1500+).
  - How to balance performance, display quality, and portability within a limited budget.
  - Which brands offer the best value in a given price segment.
- Scenario & Use Case:
  - Laptop recommendations for specific fields of study (*e.g.*, programming, design, data science).
  - Hardware configuration choices for specific software (*e.g.*, Adobe Suite, AutoCAD, gaming).
  - Models suitable for specific lifestyles (*e.g.*, frequent travel, remote work, mobile learning).
- Technical Specifications & Performance:
  - Explanations of specific technical specifications (*e.g.*, CPU models, GPU types, RAM specs, storage interfaces).
  - The impact of different specifications on real-world performance (*e.g.*, i5 vs. i7 for office tasks).
  - How to determine if a laptop's performance will meet future needs (2–3 years).
- User Experience & Concerns:
  - Common issues with laptops (*e.g.*, battery life overestimation, overheating noise, poor display quality).
  - How to choose an eye-friendly display.
  - Pros and cons comparison between mechanical keyboards and chiclet keyboards.
  - The impact of laptop weight and size on portability.

Output Format:

- Generate only one question per turn.
- The question should be clear, specific, and avoid being overly broad.
- Don't precede the question with the persona you are simulating.
- Do not add any extra explanations, analysis, or answers.

❷ Resource Collection. For each retained query, we retrieve candidate web resources from diverse online sources and extract their main textual content. The prompt detail is shown below:

PC-GEO: Prompt for Resource Collection

Role:
AI Personal Shopping Assistant

Goal:
Search for and recommend the most relevant products on mainstream platforms based on user requirements.

Execution Rules (Must be strictly followed):

1. Direct Product Search:
When a user makes an inquiry or recommendation request, search directly on e-commerce platforms using the specific product category or name mentioned by the user as the keyword. For example, if a user asks "Recommend cat food," search for "cat food" directly.

2. Sorting Based on Product Sales Pages:
Recommendation results must be sorted and filtered based on information you searched. Priority should be given to the platform's default sorting logic or adjusted according to the user's implicit priorities.

3. Search Scope and Information Sources
The search scope is primarily based on product sales pages of mainstream e-commerce platforms such as JD.com, Taobao, and Pinduoduo.
A small amount of content from formal and authoritative review websites can be used as supplementary reference.

4. Provide Reference Websites:
In your final answer, provide at least 8 reference website URLs that you used to gather information.

5. Output Format:
Your final answer must be returned strictly in JSON format, as specified below:

```json
{
  "quotations": [
    {
      "url": "xxx",
      "summary": "xxx"
    }
  ]
}
```

❸ Rule-based Filtering. To improve subsequent processing efficiency, we first filter out low-quality content (*e.g.*, placeholder texts indicating JavaScript rendering errors or failed crawls) and irrelevant content. This process entails the removal of empty pages, texts with a length of less than 200 characters, and any content that fails to mention the keywords "computer" or "laptop". Such content is deemed non-referential and is therefore excluded from further analysis.

❹ LLM-based Filtering. Furthermore, we employ the LLM to extract the core textual content by stripping away irrelevant elements such as headers, footers, and navigation bars. The prompt detail is shown below:

PC-GEO: Prompt for LLM-based Filtering

You are a professional web content cleaner and formatter. Your task is to accurately extract and optimize the core readable content from the given web page text:

1. Content Cleaning:
Remove all irrelevant content, including but not limited to headers, footers, sidebars, navigation menus, advertisements, recommended articles, and copyright information. Only retain the main body content of the page.

2. Format Optimization:
Remove all HTML tags, JavaScript code, CSS styles, URL links, special symbols (*e.g.*, `&`, ` `), and other non-text content. Ensure the text format is regular and paragraphs are clear.

3. Content Output:
Present the final, clean text content in Markdown format.
Constraints:

1. Strictly follow the principle of "only retaining core readable content". Do not omit any important information, nor retain any redundant information.

2. The output must be pure Markdown text, without any other content such as explanatory text, thought processes, code block markers, or polite expressions.

Upon completing the above steps, our PC-GEO is partitioned into training and test sets at a 9:1 ratio, comprising 3,533 training queries and 424 test queries.

B.2 Baseline Details

We compare our method with several typical baselines, including nine methods from (Aggarwal et al., 2024), RAID (Chen et al., 2025c), and two variants from AutoGEO (Wu et al., 2025).

Specifically, the details of nine methods from GEO-Bench (Aggarwal et al., 2024) are as follows:

- **Technical Terms:** Inject technical terms and domain-specific factual information to signal depth and expertise to the search engine.
- **Cite Sources:** Introduce relevant citations from credible external sources to convey stronger evidential support.
- **Keyword Stuffing:** Injects relevant keywords to better align the content with standard SEO matching principles.
- **Unique Words:** Introduce rare and distinctive terms to increase diversity in the content.
- **Authoritative:** Rewrite the content using a more assertive and confident tone to convey authority and expertise.
- **Easy-to-Understand:** Rewrite the content with simpler language to enhance clarity.
- **Statistics Addition:** Enrich the content with quantitative statistics and numerical information to support more objective descriptions.
- **Quotation Addition:** Insert quoted statements or expert remarks into the content to enhance rhetorical authority.
- **Fluency Optimization:** Improve the fluency and coherence of content to enhance overall readability and clarity.

RAID (Chen et al., 2012) executes a four-stage pipeline comprising content summarization, intent inference and refinement, step planning, and content rewriting. A key component of RAID is the *4W Multi-Role Deep Reflection* mechanism, which prompts the LLM to generalize latent search intents by analyzing four dimensions: *Who* (inferring representative user roles), *What* (identifying role-specific retrieval needs), *Why* (analyzing semantic mismatches), and *How* (reconstructing generalized intents). Guided by this refined intent, the model formulates a sequence of actionable optimization steps to rewrite the content, thereby enhancing its visibility in the generated responses of GSEs.

AutoGEO (Wu et al., 2025) is a systematic framework that automatically extracts preference rules from generative engines to optimize web content for higher visibility. The framework provides two variants: a prompt-based API version $\text{AutoGEO}_{\text{API}}$ that serves as a plug-and-play system and a compact Mini version $\text{AutoGEO}_{\text{Mini}}$ designed for cost-efficient deployment. While the API version requires no additional training, the Mini version is optimized through reinforcement learning and operates at a cost approximately 0.0071 times that of the API system. Training for the Mini model begins with a cold start phase using a learning rate of 5e-5 for 5 epochs with the Adam optimizer. This is followed by a Group Relative Policy Optimization phase to align content with visibility and rule adherence rewards.

B.3 Method Prompt

Here, we present the prompts used in our method.

Prompt for Entity Extraction

Extract entities from the following query.

Query: {query}

IMPORTANT:
Even if the query is short, extract ALL meaningful entities and concepts, including:
- Named entities (people, organizations, locations, products)
- Key concepts and technical terms
- Important nouns and noun phrases
- Domain-specific terminology

Output the entities in the following JSON format (ONLY output valid JSON, no other text):
{
"entities": [
"text": "entity name", "type": "entity type",
...
]
}}

Entity type examples (you can use these or create your own):
- PERSON: people, authors, historical figures
- ORG: organizations, companies, institutions
- GPE: geo-political entities (countries, cities, states)
- PRODUCT: products, devices, software, tools
- CONCEPT: abstract concepts, theories, ideas
- TECH: technical terms, technologies, methods
- ACRONYM: abbreviations and acronyms
- EVENT: events, incidents, occurrences
- DATE: dates, time periods
- QUANTITY: numbers, measurements, amounts
- LAW: laws, regulations, policies
- LANGUAGE: programming languages, natural languages
- PROTOCOL: protocols, standards, specifications
- DISEASE: diseases, medical conditions
- CHEMICAL: chemicals, compounds, substances
- BIOLOGICAL: biological entities (genes, proteins, species)
- OBJECT: physical objects, structures, infrastructure
- PROCESS: processes, procedures, activities

Feel free to create more specific entity types if needed to better capture the semantics.

Prompt for Edge Extraction

Compute the association strength matrix for the given entities in texts.

Entities: {texts}

IMPORTANT:
- Shape: (len(texts), len(texts))
- Diagonal: 1.0
- Symmetric
- Values in [0, 1]

Output the edges in the following JSON format (ONLY output valid JSON, no other text):
{
"matrix": []
}

Prompt for Intent Inference

Based on the following entities extracted from the query, infer the query intents.

Entities: {entities}

Output the result in the following JSON format (ONLY output valid JSON, no other text):
{{
"intent_type": "a descriptive intent type that captures the user's question goal",
"keywords": ["keyword1", "keyword2", "keyword3"]
}}

Intent type examples (you can use these or create your own):
- DEFINITION: asking what something is or means
- HOWTO: asking how to do something
- COMPARISON: comparing things
- CAUSATION: asking why something happens
- LOCATION: asking where something is
- TIME: asking when something happens
- QUANTITY: asking how many/much
- QUALITY: asking about characteristics
- EXPLANATION: asking for detailed explanation
- PROCEDURE: asking about step-by-step process
- EVALUATION: asking about pros/cons or assessment
- RECOMMENDATION: asking for suggestions or advice
- TROUBLESHOOTING: asking how to fix or solve a problem
- GENERAL: other types

Feel free to create a more specific intent type if none of these examples fit well.

Prompt for Style Inference

Based on the following entities and intents, infer the linguistic styles of the query.

Entities: {entities}

Intents: {intents}

Output the result in the following JSON format (ONLY output valid JSON, no other text):
{
"style_type": "",
"style_description": ""
}

Style type examples (you can use these or create your own):
- STUDENT: learning the topic
- RESEARCHER: conducting research
- DEVELOPER: building software
- ENGINEER: solving technical problems
- BUSINESS: making business decisions
- ANALYST: analyzing data or trends
- GENERAL_PUBLIC: seeking general information
- ENTHUSIAST: pursuing a hobby or interest
- PATIENT: seeking health information
- CLINICIAN: providing medical care
- EDUCATOR: teaching others

- HOBBYIST: pursuing leisure activities
- PROFESSIONAL: working in the field
- POLICYMAKER: making policy decisions
- INVESTOR: making investment decisions
- CONSUMER: making purchase decisions

Feel free to create more specific linguistic styles if the examples don't fit well.

Prompt for Query Generation

You are a {role_desc} asking a {intent}-type question.

Generate a natural, specific question about the following entities: {entities}

Requirements:
1. The question should be naturally phrased and grammatically correct
2. Use some or all of the provided entities: {entities}
3. The question should match the intent type: {intent}
4. Frame the question from the perspective of a {role_desc}
5. The question should be specific and concrete
6. Output ONLY the question, nothing else

Original context (for reference only): {original_query}

Question:

Prompt for Distilling Shared Reasoning Steps

You are an expert reasoning assistant. Given the personalized reasoning steps of the corresponding augmented-query set, generate cross-query shared reasoning steps.

Personalized reasoning steps: {
"Reasoning steps of query 1": {reasoning_steps_1},
"Reasoning steps of query 2": {reasoning_steps_2},
...
}

Output Format: {
"Shared reasoning steps": []
}

Important:
- Each step should be concise (1-2 sentences)
- Steps should be logically connected
- Focus on factual reasoning, not opinions

Prompt for Reasoning Information Coverage Evaluation

You are an expert evaluator, given a list of reasoning steps and a generated content, evaluate how thoroughly the content covers these reasoning steps.

Reasoning steps: {
"reasoning steps": {reasoning_steps}
}

Content: {
"content": {content}
}

[Evaluation Instructions]
For each reasoning step, assess whether the generated content:
1. Explicitly addresses the step with relevant information
2. Implicitly covers the step through related content
3. Partially touches on the step but lacks depth
4. Does not address the step at all

[Scoring Guidelines]
- 0.9-1.0: Nearly all or all steps are thoroughly covered
- 0.7-0.9: Most steps are well covered, with some minor gaps
- 0.5-0.7: About half to most steps are covered, with notable omissions
- 0.3-0.5: Some steps are covered, but significant gaps remain
- 0.1-0.3: Very few steps are covered
- 0.0-0.1: Content does not meaningfully address the reasoning steps

[Output Format]
Strictly return a valid JSON object with a single key "score" and a precise float value between 0.0 and 1.0.
Do not include any markdown formatting (like ```json), explanations, or extra text. Output only the JSON string.

[Output Example]: {"score": 0.73}

C Notations

Table 7 summarizes the used notations in this paper.

Table 7: Summary of notations.

Notation	Description
q_i	User query
$\mathcal{Q}$	Original query set
$\hat{\mathcal{Q}}_i$	Query variant set
$\tilde{\mathcal{Q}}$	Augmented query set
$\mathcal{G}_i$	Entity graph
$\mathcal{V}_i$	Entity set
$\mathcal{E}_i$	Edge set
$T_{i,j}$	User intents
$S_{i,j,k}$	Linguistic styles
$p_{\texttt{intent}}, p_{\texttt{style}}, p_{\texttt{cov}}$	Prompts
$\mathcal{R}_{cov}/\mathcal{R}_{sem}/\mathcal{R}_{out}/\mathcal{R}_{total}$	Reward
$\mathcal{A}^p_{i,j}, \mathcal{A}^s_{i,j}$	Reasoning step set
c_i	Original content
$\tilde{c}_i$	Optimized content
π_θ	Policy
ϵ	Clipping range
β	Strength of KL regularization
μ	Mean of rewards
σ	Standard deviation of rewards
A_i	Group-relative advantage