Computational Materials Today 12 (2026) 100068

Contents lists available at ScienceDirect

Computational Materials Today

journal homepage: www.sciencedirect.com/journal/computational-materials-today

QMOF-Rec: A multi-objective recommender-system framework for quantum-chemical MOF candidate screening

Mehrdad Jalali [a,*], Soumya Sekhar Banerjee [a], Arnd Koeppe [b,c]

[a] Applied Data Science and Artificial Intelligence , SRH University of Applied Sciences Heidelberg, Heidelberg, Germany
[b] Karlsruhe Institute of Technology (KIT), Institute for Applied Materials - Microstructure Modeling and Simulation (IAM-MMS), Karlsruhe, Germany
[c] Karlsruhe Institute of Technology (KIT), Institute of Nanotechnology - Microstructure Simulation (INT-MSS), Karlsruhe, Germany

ARTICLE INFO

Keywords:
QMOF
Candidate screening
Incomplete metadata
Recommendation
Diversity
Differential evolution

ABSTRACT

Screening metal–organic frameworks from incomplete metadata requires explicit separation of available evidence, missing descriptors, and application preferences. QMOF-Rec provides a query-conditioned recommendation framework that separates masked utility, material-space diversity, candidate generation, and auxiliary explanation. Evaluation uses 20,372 QMOF records, five predefined application scenarios, ten seeds, shared pools of 100, and top-five output. LEA and WeightedSum achieve Rel@5 = 0.9004 and NDCG@5 = 1.0000; hybrid diversity is 0.0188 for WeightedSum, 0.0204 for LEA, and 0.0215 for MMR. Differential evolution yields diversity 0.0213. MMR is faster than LEA under matched top-five work. Relevance metrics use the optimized utility and therefore measure utility preservation rather than independent relevance. Descriptor checks characterize internal screening consistency; absent adsorption, porosity, experimental, and expert-usefulness evidence prevents claims of validated application performance.

1. Introduction

Materials discovery increasingly depends on computational screening, machine learning, and interactive decision support because experimentally accessible materials spaces are too large to explore by manual intuition alone. Metal–organic frameworks (MOFs) are a representative example: their modular chemistry, tunable pore environments, and broad electronic-property landscape make them attractive for gas adsorption, storage, catalysis, separations, sensing, drug delivery, and electronic applications. Yet the same chemical diversity creates a practical recommendation problem.

We present QMOF-Rec, a multi-objective scientific recommender system for quantum-chemical MOF candidate screening. Screening existing QMOF records can be formulated as a scientific top-*K* recommendation task: given a user query and an incomplete set of material descriptors, the system must return a ranked and diverse shortlist of candidate MOFs that are relevant, scientifically plausible, and useful for follow-up investigation.

In this manuscript, *quantum-chemical MOF recommendation* means recommending experimentally derived QMOF records using query intent together with available quantum-chemical and metadata-derived signals such as band gap, density, stability proxies, and engineered metadata-proxy signals. The formulation extends conventional single-target screening by modelling recommendation as a multi-objective ranking problem over relevance, property suitability, diversity, and engineered metadata-proxy signals.

The Quantum MOF (QMOF) database introduced by Rosen et al. provides quantum-chemical properties for experimentally derived MOF structures and has become an important resource for machine-learning-assisted MOF discovery [1,2]. However, using QMOF as a recommendation substrate is different from using it as a static table. A scientific recommender must interpret user intent, retrieve a candidate pool, score property suitability, avoid redundant candidates, explain trade-offs, and communicate uncertainty when descriptors are missing. This is especially important for queries such as "CO_2 adsorption MOFs" or "photocatalytic MOFs", where semantic relevance alone is not equivalent to scientific suitability.

QMOF recommendation is a query-conditioned candidate-prioritization problem rather than a database lookup. A user rarely asks only for the nearest textual match to a query; instead, they seek a scientifically plausible portfolio of candidates that balances several objectives under incomplete evidence. Vector retrieval can identify materials whose metadata are textually or semantically related to a query, but semantic similarity alone is not a sufficient scientific objective. A material may

* Corresponding author.
E-mail address: mehrdad.jalali@srh.de (M. Jalali).

https://doi.org/10.1016/j.commt.2026.100068
Received 7 August 2026; Received in revised form 13 October 2026; Accepted 21 October 2026
Available online 23 October 2026
2950-4635/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

match a query textually while lacking the available band-gap, density, stability-proxy, or metadata-proxy signals needed for a specific use case. Conversely, a material with sparse metadata may remain worth inspecting when its observed descriptors or formula-derived metadata proxies resemble high-value candidates. QMOF recommendation is therefore naturally multi-objective and partially evidence-limited.

The central idea of QMOF-Rec is to use retrieval for candidate generation, property and metadata-proxy scoring for scientific item representation, a reranking policy for multi-objective top-K portfolio construction, and candidate-conditioned interaction for explainable recommendation. LEA-guided optimization is one evaluated reranking policy among eight. The implemented evaluation compares these policies on shared candidate pools and characterizes deterministic explanation templates. The broader architecture includes prospective graph-aware representation learning and property prediction, together with retrieval-augmented assistance and user feedback.

The methodological contribution is the explicit formulation of query-conditioned screening under incomplete metadata: utility scoring uses observation masks, diversity is measured in a separate full-precision material representation, and explanation is separated from selection. These choices make assumptions and trade-offs auditable. The constituent retrieval, masking, evolutionary search, and graph-learning ideas are established methods; LEA is one evaluated reranking policy, not the core novelty. Its diversity lies between WeightedSum and MMR in this evaluation, but the matched top-five runtime comparison does not support preferring LEA over MMR.

The contributions are:

1. A query-conditioned screening formulation for existing QMOF records that distinguishes observed, missing, and heuristic information.
2. A separation of masked recommendation utility from full-precision material diversity, exposing the loss of resolution caused by binned score distances.
3. An empirical characterization of reranking trade-offs using shared pools, scenario-level analysis, component ablations, coefficient sensitivity, and a differential-evolution portfolio comparator.
4. An auditable separation of ranking evidence from auxiliary explanation and prospective learned representations, with explicit limits on physical validation.

2. QMOF-Rec as a scientific recommender system

QMOF-Rec is designed as a query-driven scientific recommender system. The user is a materials scientist, chemist, or researcher seeking a practical shortlist of candidate MOFs for a target application. The items are QMOF material records. The user intent is expressed as a scientific query such as "CO_2 adsorption MOFs", "photocatalytic MOFs", "lightweight storage frameworks", or "wide-band-gap insulating frameworks". The recommendation task is to map this query context to a top-K list of QMOF candidates that balances relevance, scientific suitability, diversity, and available engineered metadata proxies.

The architecture follows a common two-stage recommender-system design. Candidate generation first narrows the searchable item universe to a query-relevant candidate pool using vector, semantic, or metadata retrieval. Ranking and reranking then operate on this smaller candidate set. Deterministic rankers such as WeightedSum, Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), ParetoCrowding, SemanticOnly, Random, and maximal marginal relevance (MMR) provide baseline recommendation policies, while LEA-guided multi-objective reranking optimizes the final user-facing top-K candidate portfolio. Explanation is provided through candidate-conditioned recommendation rationale, uncertainty communication, and evidence-grounded response generation.

Table 1 makes explicit that QMOF-Rec is a hybrid scientific recommender with a strong content-based focus rather than a collaborative-filtering system. No large-scale user–MOF interaction matrix is currently available. Instead, recommendation is driven by query context and item content, including metadata, property scores, semantic retrieval signals, and engineered metadata-proxy representations.

3. Related work

3.1. QMOF databases and machine-learning-assisted MOF discovery

High-throughput computational databases have reshaped materials informatics by making electronic, structural, and thermodynamic properties available at scales that support statistical learning. QMOF is especially relevant because it focuses on quantum-chemical properties of MOFs and coordination polymers [1]. Rosen et al. later expanded the electronic-property analysis of QMOF and benchmarked graph neural networks for high-throughput electronic-property prediction [2]. These studies establish QMOF as a resource for property prediction, screening, and data exploration.

Machine learning has also been used broadly for materials and MOF screening, including adsorption-property prediction, band-gap prediction, and data-driven discovery workflows. Seko et al. demonstrated that matrix- and tensor-based recommender systems can prioritize currently unknown inorganic compounds [3]. Transformer-based MOF representations such as MOFormer illustrate the increasing role of learned representations for QMOF and hypothetical MOF tasks [4]. Recent natural-language-guided materials discovery work further motivates query-conditioned retrieval and ranking interfaces for scientific users [5]. In the present work, QMOF records are used as the retrieval and recommendation substrate. Because the locally available metadata are incomplete for some descriptors, the manuscript avoids strong adsorption-specific claims and focuses on ranking behavior under the descriptors actually present.

Table 1
Mapping classical recommender-system components to QMOF-Rec.

RS Concept	QMOF-Rec Implementation	Role in Candidate Screening
User	Materials scientist, chemist, or researcher	Defines scientific intent and evaluates usefulness
Item	QMOF material candidate	Candidate MOF for follow-up investigation
Query/context	Application-oriented scientific query	Specifies target use case and objective weights
Candidate generation	Vector, semantic, or metadata retrieval	Narrows QMOF search space to C_q
Item representation	Metadata, formula, density, band gap, stability proxy, metadata proxies, semantic metadata	Encodes scientific signals for ranking
Scoring function	Query-dependent objective utility	Estimates relevance and suitability
Multi-objective ranking	WeightedSum, TOPSIS, ParetoCrowding, SemanticOnly, Random, MMR	Provides deterministic and diversity-aware baselines
Reranking	LEA-guided multi-objective reranking	Optimizes final top-K portfolio
Top-K recommendation	Ranked shortlist $S_K(q)$	Produces practical candidate list for expert inspection
Diversity	Intra-list diversity and future novelty metrics	Reduces redundant candidate portfolios
Explanation	Candidate-conditioned rationale and uncertainty explanation	Supports interpretable scientific decision support
Feedback	User/expert feedback, logging, analytics	Enables future preference-aware improvement

3.2. Graph-based and structure-aware MOF learning

Graph neural networks (GNNs) are natural for molecular and crystalline materials because atoms, bonds, distances, and higher-level material relationships can be represented as nodes, edges, and attributes. Crystal Graph Convolutional Neural Networks (CGCNN) demonstrated interpretable graph-based prediction of crystalline properties [6]. Graph networks have also been framed as a general machine-learning architecture for molecules and crystals [7]. Foundational GNN methods include the graph convolutional network (GCN) [8], GraphSAGE [9], and Graph Attention Networks (GAT) [10].

MOF-specific graph learning has developed rapidly. MOFSocialNet represented relationships among MOFs as a network built from descriptor similarity and analyzed it with social-network methods [11]. MOFGalaxyNet used social-network-style graph analysis and graph convolutional learning to predict guest accessibility in MOFs [12]. The Black Hole Strategy later introduced gravity-based representative sampling for frugal graph learning on MOF networks, including GraphSAGE, GCN, and GAT experiments [13]. These works motivate the graph-aware extension in QMOF-Rec: learned graph embeddings can provide relational signals in future extensions; the evaluated ranking signals below are engineered metadata proxies.

3.3. Retrieval-augmented systems and vector search

Large-scale vector retrieval enables fast nearest-neighbor search over embedded documents, materials, and metadata. FAISS is a widely used library for billion-scale similarity search [14]. Sentence-BERT showed that Siamese sentence-transformer architectures can produce effective dense sentence embeddings for semantic search [15]. Retrieval-Augmented Generation (RAG) combines parametric generation with retrieved evidence to support knowledge-intensive tasks [16].

For QMOF-Rec, retrieval serves two functions. First, it provides a candidate pool for recommendation. Second, it supplies evidence for the scientific assistant. The LLM is treated as an explanation and interaction layer rather than as a source of ground-truth material properties. This distinction is important because generated text can be fluent while still requiring verification against QMOF metadata, retrieved literature, and candidate records.

The deployed repository separates semantic retrieval from numerical reranking. Retrieval uses SentenceTransformer embeddings of metadata text templates stored in a FAISS L2 index; it does not search directly over the normalized numerical objective vector used by WeightedSum or LEA. During vector-index construction, unavailable fields are serialized as unavailable metadata rather than numerical zeros. Reranking then constructs explicit objective vectors and availability masks for semantic score, band-gap score, density score, and stability score. The reproducible offline benchmark tables were generated with a deterministic semantic proxy over metadata text and formula tokens rather than by rerunning a live sentence-embedding search.

Research-data assistants such as KadiAssistant provide a related but distinct direction: privacy-preserving access to heterogeneous research repositories through semantic search and local or self-hosted language-model components [17]. QMOF-Rec instead focuses on MOF candidate retrieval, objective-aware reranking, and recommendation evaluation over QMOF-derived metadata.

3.4. Scientific, content-based, and hybrid recommender systems

Scientific recommender systems extend classical recommendation from consumer preference prediction to expert decision support, candidate discovery, and prioritization under incomplete evidence. In materials discovery, Seko et al. used matrix- and tensor-based recommender systems to identify currently unknown inorganic compounds [3], and Qu et al. used language representations for materials exploration and discovery [5]. In QMOF-Rec, users are scientists and items are material candidates rather than products, movies, or documents. The absence of large user–item interaction histories makes the problem closer to query-driven, content-based, context-aware recommendation than to collaborative filtering. Content-based recommenders rank items using item attributes and user/query profiles, while hybrid recommenders combine multiple signals such as metadata, semantic retrieval, graph representations, and user feedback when available [18–20].

QMOF-Rec is therefore a query-driven hybrid recommender with a strong content-based focus. Candidate items are represented by QMOF metadata, formula-derived information, property scores, engineered metadata proxies, and semantic retrieval signals. This differs from matrix/tensor completion over composition spaces [3] and complements language-representation-based material ranking [5] by adding descriptor masks, metadata-proxy augmentation, and LEA list-level reranking. Future user feedback can support preference-aware or online recommendation, but the current offline evaluation uses query-dependent objective weights rather than click logs or explicit ratings.

3.5. Multi-objective recommender systems and diversity-aware reranking

Scientific recommendation is typically a multi-objective problem. Candidate portfolios should include relevant, diverse, and non-redundant materials rather than a list of nearly identical top-scoring records. Multi-criteria recommender systems explicitly model multiple utility dimensions [21,22], while explainable recommendation emphasizes transparency and user trust [23]. Diversity-aware reranking has a long history in information retrieval, including maximal marginal relevance (MMR) [24].

Classical decision methods also remain useful baselines. TOPSIS ranks alternatives by closeness to ideal and anti-ideal solutions [25]. Pareto ranking and crowding-distance strategies provide diversity-aware approximations of multi-objective selection, and Non-dominated Sorting Genetic Algorithm II (NSGA-II) is a standard evolutionary reference method for multi-objective optimization [26]. QMOF-Rec evaluates weighted-sum, TOPSIS, ParetoCrowding, random, semantic-only, and LEA-based reranking under a common top-K protocol.

Unlike similarity-only material recommendation, QMOF-Rec formulates MOF candidate screening as a multi-objective top-K recommendation problem. It combines retrieval augmentation, property-aware scoring, engineered metadata proxies, LEA-guided reranking, and explanation support to recommend diverse candidate portfolios rather than only nearest-neighbor materials.

3.6. Explainable recommendation and scientific assistance

Explainable recommender systems aim to clarify why items are recommended, how scores are produced, and where uncertainty remains [23]. For scientific recommendation, explanation is especially important because candidate lists may influence costly follow-up experiments, simulations, or synthesis decisions. In QMOF-Rec, candidate-conditioned explanations are framed as an explainable-recommendation layer that can summarize retrieved evidence, material metadata, objective trade-offs, and uncertainty flags. The current manuscript does not claim that LLM explanation improves recommendation accuracy, and live-LLM faithfulness remains a separate future evaluation target.

3.7. Nature-inspired and evolutionary optimization for scientific discovery

Nature-inspired and evolutionary optimization methods are often used when search spaces are continuous, nonconvex, or multi-objective. Particle swarm optimization, genetic algorithms, NSGA-II, and related evolutionary or swarm-based methods have been applied across scientific discovery and engineering optimization [26]. Bayesian

optimization is better described separately as model-based sequential optimization for expensive black-box objectives [27]. The Lotus Effect Algorithm (LEA) is a lotus-nature-inspired optimization algorithm originally proposed for engineering design optimization [28]. MOF-LENS later used LEA-inspired optimization for accelerated discovery of MOF nanocarriers for doxorubicin delivery [29].

The present work distinguishes the original LEA algorithm from the implemented LEA-rec adaptation. LEA-rec uses lotus-inspired mutation and self-cleaning replacement in a post-retrieval normalized objective space, then maps candidates back to valid QMOF records. The method constructs recommendation portfolios from a retrieved candidate pool and leaves physics-based validation, DFT, adsorption simulations, and expert assessment as downstream steps.

3.8. Positioning of the present work

This manuscript positions QMOF-Rec between materials informatics, recommender systems, graph learning, and scientific-assistant systems. It models user intent and top-K recommendation, evaluates candidates using objective scores and diversity, and grounds explanations in retrieved evidence and material metadata. The evaluated graph-named variants are engineered metadata proxies; trained metadata-graph, CIF-derived, and atomistic GNN comparisons remain future work.

4. End-to-End QMOF-Rec recommendation workflow

QMOF-Rec is framed as an end-to-end workflow for screening existing material records. Fig. 1 summarizes the workflow in three panels. Panel a shows the six stages of the executed offline benchmark: (1) the query and the catalog of 20,372 QMOF records, (2) generation of a shared pool of 100 candidates, (3) masked utility and representation, (4) ranking and reranking by eight policies on the same pool, (5) the top-five shortlist, and (6) auxiliary explanation templates that cannot change it. Panel b states the evidence boundary of each descriptor: density is observed for all records, band gap is observed for 53.1% and masked when missing, stability and proxies S/A are derived quantities, and void fraction is absent and excluded. Panel c lists what lies outside the executed protocol: learned graph or CIF-derived representations, live-LLM explanation, and application validation.

4.1. Scientific query and candidate representation

The workflow starts from a scientific query and QMOF metadata. Query text, formula tokens, and available material descriptors are converted into text and metadata representations for semantic candidate retrieval. This retrieval stage generates the candidate pool used by all downstream rankers, so the final comparison is conditioned on a shared set of query-relevant QMOF records rather than the full catalog.

Each retrieved candidate has masked active objectives and masked physical coordinates. The engineered proxy-A coordinate separately retains the missing-gap fallback disclosed in Section 8. Availability masks prevent missing physical coordinates from being interpreted as measured zeros; in particular, void fraction is excluded from numerical ranking because it is unavailable in the evaluated metadata. Objective-space scoring and portfolio-level hybrid material diversity are kept distinct: active objective scores support relevance and suitability

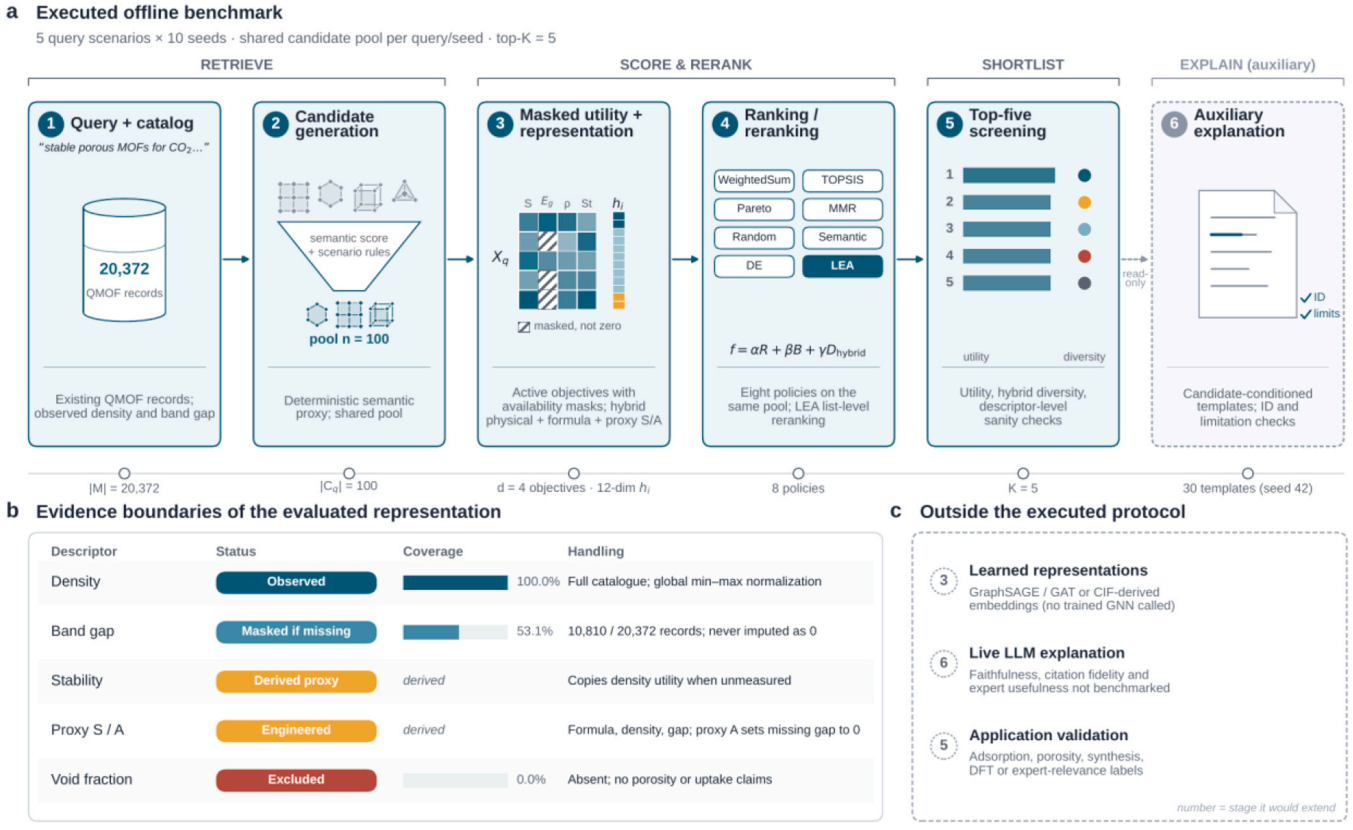


Fig. 1. QMOF-Rec screening workflow and evidence boundaries. (a) Executed offline benchmark (five query scenarios, ten seeds, shared candidate pool per query/seed, $K = 5$): query and catalog, candidate generation by the deterministic semantic proxy, masked utility and hybrid representation h_i, ranking and reranking by eight policies on the same pool, top-five shortlist, and auxiliary explanation templates that read the shortlist and do not change rankings. Hatched cells denote missing descriptors that are masked, not set to zero. (b) Status, catalog coverage, and handling of each descriptor; proxies S and A are engineered metadata scores, not trained graph embeddings, and proxy A retains the disclosed missing-gap fallback. (c) Components outside the executed protocol; the number marks the stage each would extend.

scoring, while the hybrid material representation supports list-level diversity comparisons.

The executed representation uses engineered formula/descriptor metadata proxies. Learned metadata graphs or atomistic representations are prospective extensions. The current proxy values do not validate adsorption, synthesizability, or industrial readiness. WeightedSum, TOPSIS, and MMR provide comparison methods, while LEA performs multi-objective reranking over the retrieved candidate pool to produce the final top-K recommendation portfolio.

Candidate-conditioned explanations are generated from retrieved records, available descriptors, scores, and missing-descriptor warnings. They are designed to communicate objective trade-offs and limitations while preserving safeguards against unsupported claims. The deterministic routine below checks only identifier inclusion and limitation phrases; assigned flags do not independently establish safeguard behavior.

4.2. Interface demonstration

The interface prototype was executed locally and inspected through browser-based screenshots (Fig. 2). These views demonstrate how the recommendation workflow can be exposed to a user through query input, scientific chat, recommendation, material-inspection, and analytics views. The screenshots are illustrative interface evidence and do not replace the offline ranking, metadata-proxy, ablation, or deterministic candidate-conditioned explanation evaluations.

5. End-to-End recommendation scenario

This section illustrates how QMOF-Rec operates as a scientific recommender system from user request to final recommendation. Consider a materials scientist who enters the benchmark query: “stable porous MOFs for CO_2 adsorption.” QMOF-Rec treats this input as a recommendation context rather than only as a text-search string. The offline benchmark maps this application intent to predefined density and descriptor-completeness preferences, with a density-derived stability proxy and material-space diversity. No adsorption-specific metadata evidence is available. Because void fraction is unavailable in the current metadata, porosity is not used as an active numerical objective.

The first stage is candidate generation. QMOF-Rec retrieves an initial candidate pool C_q from the QMOF metadata/vector store. This stage is analogous to candidate generation in a two-stage recommender-system pipeline: it narrows the full QMOF item catalog to a smaller set of query-relevant materials. In the current reproducible evaluation, this pool contains 100 candidate QMOF records for each query scenario.

Each retrieved material is then represented as an item feature vector. Available descriptors and scores, including formula-derived signals, density, band gap where available, semantic metadata score, stability proxy, and metadata-derived objective scores, are normalized before ranking. Missing descriptors are handled explicitly. In particular, void fraction is absent in the current metadata, so the system avoids strong adsorption or porosity claims and communicates that limitation in the recommendation rationale.

The current benchmark can augment scoring with engineered metadata proxies S and A. A separate prospective extension could use learned GraphSAGE/GAT or CIF-derived representations; none is trained or evaluated in the final ranking protocol.

The retrieved pool can be ranked by multiple baselines. WeightedSum provides the strongest relevance-oriented scoring baseline. TOPSIS ranks candidates by distance to ideal and anti-ideal solutions. ParetoCrowding emphasizes trade-offs and diversity. MMR provides a diversity-aware reranking reference. SemanticOnly and Random provide retrieval-only and chance-level controls. LEA is the active multi-objective reranker. In recommender-system terms, LEA acts as a multi-objective reranker whose diversity effect depends on the resolution and geometry of the selected material representation.

Table 2 summarizes the complete workflow. The final output is a ranked top-K list in which each recommendation can include the QMOF identifier, formula, available property values, method score, a short rationale, and a warning when relevant descriptors are missing. The

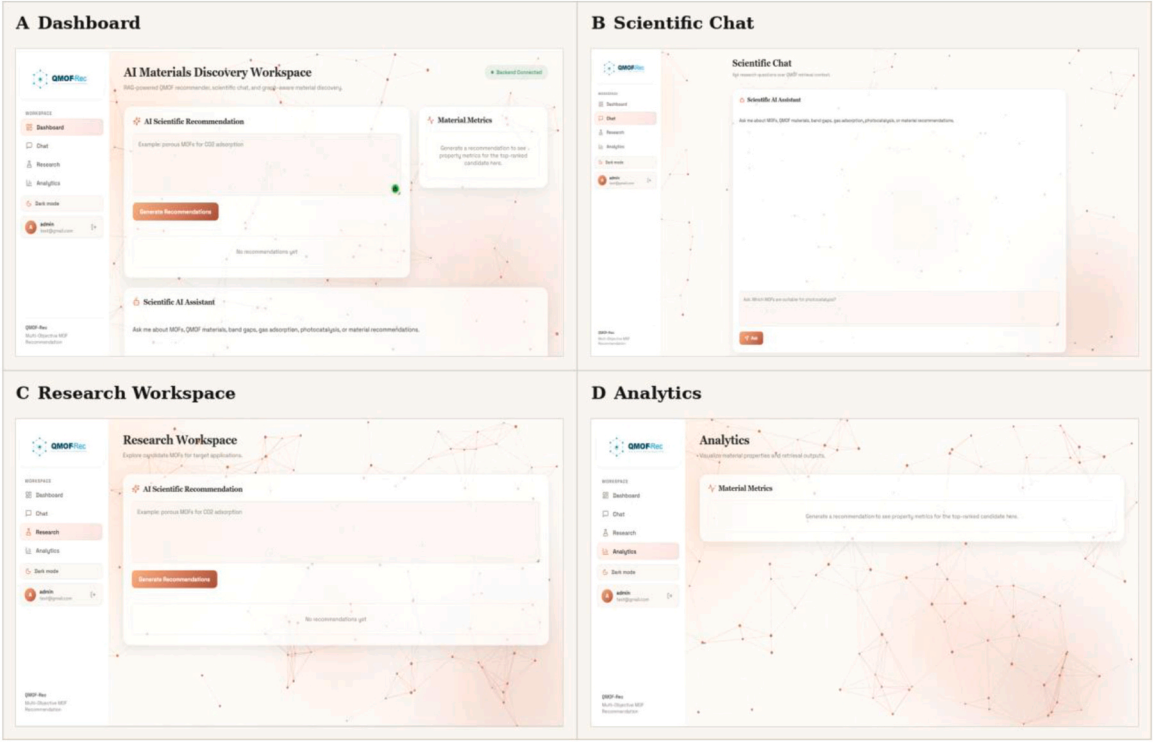


Fig. 2. Compact QMOF-Rec application screenshot montage. Panel A shows the recommendation workspace, Panel B the scientific chat interface, Panel C the research workspace, and Panel D the analytics view. Screenshots demonstrate the implemented interface and are not part of the offline ranking validation.

Table 2
Example end-to-end QMOF-Rec recommendation scenario.

Step	System Action	RS Role	QMOF-Rec Output
User query	Interpret "CO_2 adsorption with low density and good stability"	Query/context capture	Weights for relevance, density, stability, diversity, and metadata-proxy signal; porosity excluded when void fraction is unavailable
Candidate generation	Retrieve 100 query-relevant QMOF records	First-stage candidate generation	Candidate pool C_q
Objective scoring	Normalize available descriptors and metadata scores	Item representation and scoring	Objective matrix X_q with missing-descriptor flags
Metadata proxies	Engineered formula/descriptor metadata proxies	Current benchmark	Proxy S and proxy A augmentation
Prospective learning	Learned GraphSAGE/GAT or CIF-derived representations	Future extension	Not executed in the benchmark
Baseline ranking	Apply deterministic and diversity-aware policies	Baseline top-K ranking	Comparable ranked lists for offline evaluation
LEA reranking	Optimize relevance, balance, and diversity	List-level reranking	LEA-guided top-K portfolio
Top-K output	Return ranked QMOF candidates	User-facing recommendation	IDs, formulas, values, scores, and warnings
Explanation	Generate evidence-aware rationale	Explainable recommendation	Trade-off and uncertainty notes
Feedback	Record expert judgments	Future preference learning	Logs for review and later preference modeling

auxiliary explanation routine is characterized in Section 18; assigned scores are not independent quality evidence, and explanation does not alter ranking. After inspecting candidate records or 3D/CIF structures where available, the user can mark materials as useful, uncertain, or irrelevant. Such feedback is stored for future expert review and possible preference-aware recommendation.

Table 3 reports LEA top-five output for seed 42. Density and band-gap values are observed, but adsorption-specific descriptors are unavailable. Ties in utility are resolved by history-dependent LEA fitness and seed-dependent pool order.

The CO_2 scenario uses a density threshold of 1.8 g/cm^3 and band-gap availability in candidate generation, followed by density, band-gap, and density-derived stability scores. No explicit CO_2 metadata tokens occur in its seed-42 candidate pool. Thus these recommendations are preliminary metadata-based prioritization, not adsorption-performance screening; neither uptake nor void fraction is available.

Table 3
Seed-42 top-K recommendations for the CO_2 scenario. Density is in g/cm^3 and band gap in eV; void fraction is unavailable. Scores are LEA optimization utilities, not probabilities.

Rank	QMOF ID	Formula	Density	Band gap	LEA score
1	qmof-13a76dd	$Zn_4C_{40}H_{28}N_{12}O_8$	0.983	3.309	0.939643
2	qmof-ac6a77a	$Zn_4C_{76}H_{44}N_8O_{16}$	0.829	2.401	0.894619
3	qmof-235f291	$Ni_2C_{24}H_{16}N_8$	0.871	3.336	0.894456
4	qmof-a1a5849	$Fe_2C_{24}H_{16}N_8$	0.925	2.753	0.893872
5	qmof-cef46c3	$Zn_2C_{48}H_{32}N_8O_8$	0.790	2.913	0.893735

6. Problem formulation

Let $M = \{m_1, \ldots, m_{|M|}\}$ denote the item catalog of QMOF material records and let q denote a scientific query. In recommender-system terms, q is the query context, m_i is an item, and the goal is to return a ranked top-K recommendation list. The initial candidate-generation stage returns a query-dependent candidate pool

$$C_q = \{m_1, \ldots, m_n\} \subset M \tag{1}$$

Each candidate item m_i is represented by a normalized active objective vector $x_i \in [0,1]^d$ and an availability mask $a_i \in \{0,1\}^d$. The mask entry $a_{ij} = 1$ only when objective j is observed or defensibly derived for material m_i; unsupported or missing descriptors have $a_{ij} = 0$ and are excluded from active numerical scoring. This distinction prevents missing descriptors from being interpreted as physical zero values. The query-specific weight vector is

$$w_q = (w_{q1}, \ldots, w_{qd}), w_{qj} \geq 0, \sum_{j=1}^{d} w_{qj} = 1. \tag{2}$$

The objective matrix for the candidate pool is

$$X_q = \begin{pmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{pmatrix}, x_{ij} \in [0,1]. \tag{3}$$

The corresponding availability matrix is

$$A_q = \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nd} \end{pmatrix}, a_{ij} \in \{0,1\}. \tag{4}$$

The recommender outputs a top-K list $S_K \subset C_q$. Let $f(q, m_i, S)$ be the context-aware ranking or reranking utility of item m_i for query q, optionally conditioned on the partial recommendation list S. The top-K recommendation list is

$$S_K(q) = \underset{m_i \in C_q}{\mathrm{TopK}} f(q, m_i, S). \tag{5}$$

For deterministic pointwise rankers, S is ignored. For list-aware reranking, S captures the candidates already selected and enables diversity-aware portfolio construction. The masked weighted relevance score is

$$R(m_i, q) = \frac{\sum_{j=1}^{d} a_{ij} w_{qj} x_{ij}}{\sum_{j=1}^{d} a_{ij} w_{qj}}. \tag{6}$$

The implementation returns zero when the active-weight denominator is at most 10^{-8}; otherwise it divides by that denominator without an added epsilon. For a target property p_i and query-specific target τ, a normalized target score can be defined as

$$s_{\mathrm{target}}(p_i; \tau) = \max\left(0, 1 - \frac{|p_i - \tau|}{r + \epsilon}\right), \tag{7}$$

where r is a normalization range and ϵ avoids division by zero. The masked balance score is

$$B(m_i) = \begin{cases} 1 - \sqrt{\dfrac{1}{\sum_j a_{ij}} \sum_{j: a_{ij}=1} \left(x_{ij} - \bar{x}_{i,\mathrm{obs}}\right)^2}, & \sum_j a_{ij} > 0, \\ 0, & \sum_j a_{ij} = 0. \end{cases} \tag{8}$$

where $\bar{x}_{i,\mathrm{obs}}$ is the mean of the observed active objectives for candidate m_i; the implementation clips this score to [0, 1]. For two candidates, the joint availability indicator is $a_{i \wedge j} = a_{ij} a_{\wedge j}$. Masked cosine similarity is computed only over jointly observed dimensions and is used for auxiliary material-similarity scoring:

$$\cos_{\text{mask}}(x_i, x_{\prime}) = \frac{\sum_{j:a_{i\prime j}=1} x_{ij} x_{\prime j}}{\sqrt{\sum_{j:a_{i\prime j}=1} x_{ij}^2} \sqrt{\sum_{j:a_{i\prime j}=1} x_{\prime j}^2} + \epsilon}. \tag{9}$$

If no jointly observed dimension exists, or if the denominator is zero, the implemented fallback returns zero similarity. With one jointly observed dimension, the same one-dimensional cosine expression is used. The final implementation separates candidate remapping in active objective space from list diversity in hybrid material space.

The objective-space distance used for LEA candidate remapping is

$$d_{\text{obj}}(x_i, x_{\prime}) = \begin{cases} \sqrt{\dfrac{\sum_{j:a_{i\prime j}=1} (x_{ij} - x_{\prime j})^2}{\sum_j a_{i\prime j}}}, & \sum_j a_{i\prime j} > 0, \\ 0, & \sum_j a_{i\prime j} = 0. \end{cases} \tag{10}$$

The final diversity representation h_i is distinct from x_i. It concatenates a two-dimensional physical vector p_i containing full-catalog min–max-normalized density and band gap, an eight-dimensional formula vector f_i containing C, H, N, O, metal, and heteroatom fractions plus scaled atom and element counts, and a two-dimensional engineered metadata-proxy vector g_i containing proxy-S and proxy-A engineered metadata scores:

$$h_i = \left[\left(\sqrt{\frac{0.55}{2}} |p_i| \middle| \sqrt{\frac{0.30}{8}} |f_i| \middle| \sqrt{\frac{0.15}{2}} \middle| g_i \right) \right]. \tag{11}$$

Density and band gap are normalized globally over the 20,372-record catalog rather than within each candidate pool. Missing band gap is masked and excluded from pairwise distance; density is available for all records; formula and metadata-proxy features are available for all formula-bearing records; void fraction is excluded. The proxy coordinates are engineered metadata scores, not trained CIF-derived embeddings. Full precision is used in calculations, and rounding is applied only for display.

The hybrid material distance is the same masked root-mean-square form applied to h_i:

$$d_{\text{hybrid}}(h_i, h_{\prime}) = \begin{cases} \sqrt{\dfrac{\sum_{j:b_{i\prime j}=1} (h_{ij} - h_{\prime j})^2}{\sum_j b_{i\prime j}}}, & \sum_j b_{i\prime j} > 0, \\ 0, & \sum_j b_{i\prime j} = 0, \end{cases} \tag{12}$$

where $b_{i\prime j}$ is the joint availability indicator for the hybrid feature dimension. The diversity score relative to a partial list S is

$$D_{\text{hybrid}}(m_i, S) = \begin{cases} 0, & |S| = 0, \\ \dfrac{1}{|S|} \sum_{m_j \in S} d_{\text{hybrid}}(h_i, h_j), & |S| > 0. \end{cases} \tag{13}$$

QMOF-Rec defines the list-aware multi-objective utility as

$$f(q, m_i, S) = \alpha R(m_i, q) + \beta B(m_i) + \gamma D_{\text{hybrid}}(m_i, S) + \eta G(m_i, q) + \delta N(m_i), \tag{14}$$

where R is query relevance, B is objective balance, D_{hybrid} is diversity relative to the partial list, G is a prospective learned-representation term (not evaluated here), and N is optional novelty (also not evaluated). The current repository implementation uses $\alpha = 1.0$, $\beta = 0.10$, $\gamma = 0.10$, $\eta = 0$ for baseline reranking, and a redundancy penalty of $0.05\exp[-D_{\text{hybrid}}(m_i, S)]$ inside LEA fitness, applied only when the partial list S is non-empty. The revision retains this predefined representation and these baseline coefficients rather than tuning them to the new results. Both relevance and diversity remain metadata-derived diagnostics; they are not independent validation targets.

7. Retrieval-augmented scientific assistance

Let $D = \{d_1, \ldots, d_M\}$ be the document and metadata corpus. Each document chunk is embedded as $e_i = f_\theta(d_i)$, and the query is embedded as $e_q = f_\theta(q)$. The implemented FAISS retrieval index uses squared L2 distance, so the retrieved evidence set is

$$R_K(q) = \underset{d_i \in D}{\text{TopK}}[- \| e_q - e_i \|_2^2], \tag{15}$$

where smaller FAISS L2 distance corresponds to higher retrieval rank. This equation describes the live retrieval store; the offline ranking tables use the deterministic semantic proxy described above. The augmented LLM input is

$$A(q) = \text{Template}(q, R_K(q), C_q, X_q, w_q), \tag{16}$$

and the generated scientific response is $y = \text{LLM}_\phi(A(q))$.

The scientific-assistant component generates explanations from retrieved evidence, candidate records, and QMOF metadata. Authoritative material-property values are taken from the underlying records rather than generated text. The explanation layer is auxiliary and does not change candidate selection or ranking. Section 18 characterizes deterministic template outputs; several legacy scores and flags are directly assigned by the script, so they are not independent quality measurements. Expert human evaluation, citation-fidelity assessment, and user-usefulness studies remain future work.

8. Implemented metadata proxies and prospective graph learning

A prospective material graph can link records using descriptor or formula similarity and can be processed by GraphSAGE or GAT [9,10]. This architectural possibility must be distinguished from the executable ranking benchmark. Its metadata_proxy function computes two scalar engineered metadata scores; it neither constructs a neighborhood graph nor loads trained GNN weights. The result tables identify these engineered features as proxy S and proxy A.

Let f_M and f_H denote the heuristic metal and heteroatom fractions extracted from the formula, ρ density, and E_g band gap. The executable proxy definitions are

$$g_S = \text{clip}_{[0,1]}[0.45 f_M + 0.25 f_H + 0.30 \max(0, 1 - \rho/4)], \tag{17}$$

$$g_A = \text{clip}_{[0,1]}[0.45 f_H + 0.30 \min(1, E_g/5) + 0.25 \max(0, 1 - \rho/4)]. \tag{18}$$

The proxy code falls back to zero for a missing band gap. Thus the observation mask on the physical band-gap coordinate does not remove this derived proxy coordinate; this is a limitation of the retained representation. Formula fractions are heuristic compositional features, not evidence of topology, oxidation state, pore geometry, or atomic connectivity. CIF files are present in the repository, but the final ranking protocol does not consume them. The reproduced results do not establish performance of an atomistic or learned metadata-graph model.

9. Integration of metadata proxies into recommendation

The proxy-scoring variants use $0.85R(m_i, q) + 0.15 g_S(m_i)$ or $0.85R(m_i, q) + 0.15 g_A(m_i)$, followed by deterministic tie-breaking. These variants retain the relevance term. LEA-plus-proxy variants append the corresponding scalar to the objective vector, multiply the original weights by 0.85, and give the appended coordinate weight 0.15. Evaluation always uses the original masked relevance R and the same hybrid diversity. These variants probe engineered descriptor augmentation and cannot be interpreted as comparisons with trained GraphSAGE or GAT.

10. LEA-guided multi-objective reranking

The original LEA algorithm is a lotus-nature-inspired metaheuristic introduced for engineering design optimization [28]. The LEA-rec adaptation implemented here uses the same broad intuition of adaptive exploration and self-cleaning but operates as a post-retrieval reranker over normalized recommendation objectives. Individuals move in continuous normalized objective space, but final recommendations must always map back to valid QMOF records.

The candidate-validity mapping uses the active objective-space masked distance, whereas final reported list diversity uses the hybrid material distance:

$$\pi(z) = \arg\min_{m_i \in C_q} d_{\text{obj}}(z, x_i), \tag{19}$$

where the candidate mask a_i determines which active objectives participate in the remapping distance. The implementation uses iteration-dependent Gaussian exploration and exploitation toward the current best solution:

$$z_i^{t+1} = \text{clip}_{[0,1]}\left(z_i^t + \epsilon_t + \frac{t}{T}(g^t - z_i^t)\right), \tag{20}$$

where t is the current iteration, T is the maximum iteration count, g^t is the current best solution, and $\epsilon_t \sim N(0, (0.20(1 - t/T))^2 I)$. The clipping operation keeps solutions inside the normalized objective domain.

The self-cleaning replacement step is applied every $\tau = 10$ iterations in the implemented code. The weakest fraction of the population is replaced by random normalized vectors:

$$P_{\text{weak}}^{t+1} \leftarrow \{u_1, \ldots, u_{\lfloor \rho N \rfloor}\}, u_j \sim U(0,1)^d, \tag{21}$$

with $\rho = 0.15$ in the current implementation. The implementation replaces $\lfloor 0.15N \rfloor$ individuals (at least one), including iteration zero. It stores the best fitness ever observed for each mapped candidate, even as the selected context changes, and scores unseen candidates at the end. The redundancy penalty is applied only when the partial list S is nonempty, so candidates scored in the first iteration carry no penalty. These stored values are history-dependent and are not evaluations of a single fixed final-list objective. The no-self-cleaning ablation tests this particular implementation.

Algorithm 1 records the executable ordering that matters for reproduction. The partial selected context is the top-K candidates in the accumulated best-ever map from the preceding iteration. After the last iteration, previously unseen pool members receive one score in that final context; all candidates are then sorted by stored fitness. Thus the reported list is a ranking of candidate-specific best-ever values, not a greedy replay under one frozen context.

Algorithm 1. Implemented LEA reranking control flow

Require: candidate objective vectors and masks, hybrid vectors and masks, seed; $N = 30, T = 60, K = 5$

initialize N population vectors by cycling through candidate objective vectors
$H \leftarrow \emptyset$ (candidate ↦ best fitness and index); $S \leftarrow \emptyset$
for $t = 0, \ldots, T - 1$ **do**
for each population vector z **do**
map z to the nearest valid candidate using masked d_{obj}
compute $R + 0.10B + \gamma D_{\text{hybrid}}(m, S) - \lambda 1[|S| > 0] e^{-D_{\text{hybrid}}(m,S)}$
update $H[m]$ only if this fitness exceeds its stored best-ever value
end for
move all vectors toward the current best with annealed Gaussian noise and clip to [0, 1]
if $t \bmod 10 = 0$ and self-cleaning is enabled **then**
replace the lowest-fitness $\max(1, \lfloor 0.15N \rfloor)$ vectors by uniform random vectors
end if
$S \leftarrow$ indices of the K largest stored values in H
end for
score every candidate absent from H once using the final S
return all candidates sorted by stored fitness; take the first K

11. Workflow summary

The computational workflow has three stages. First, QMOF-Rec retrieves a query-dependent candidate pool from metadata and text representations. Second, it converts candidate records into active objective vectors with descriptor-availability masks, excluding unsupported descriptors from numerical relevance and balance while using a richer masked material representation for diversity. Third, LEA-guided reranking constructs a top-K portfolio and the candidate-conditioned explanation layer generates an evidence-grounded explanation. The supplementary material gives the implementation scope and exact reproduction commands; executable control flow is preserved in the versioned runner.

12. Experimental setup

The current evaluation uses 20,372 QMOF vector metadata records. Density is available for all materials. Band gap is available for 10,810

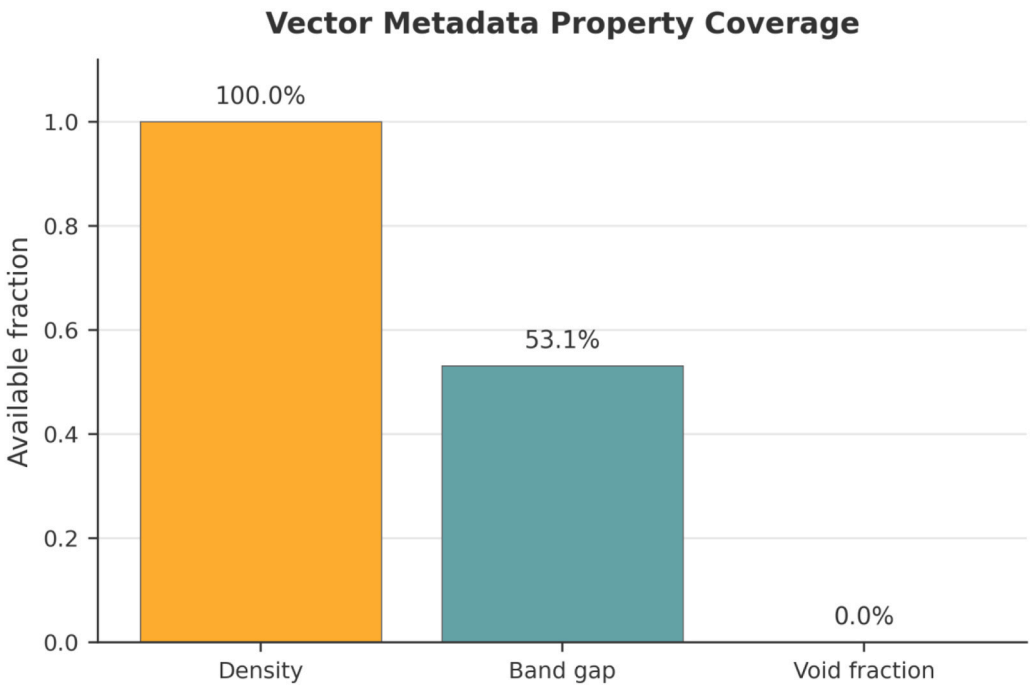


Fig. 3. Coverage of metadata fields used in the current reproducible QMOF-Rec evaluation. Density is available for all records, band gap is partially available, and void fraction is absent.

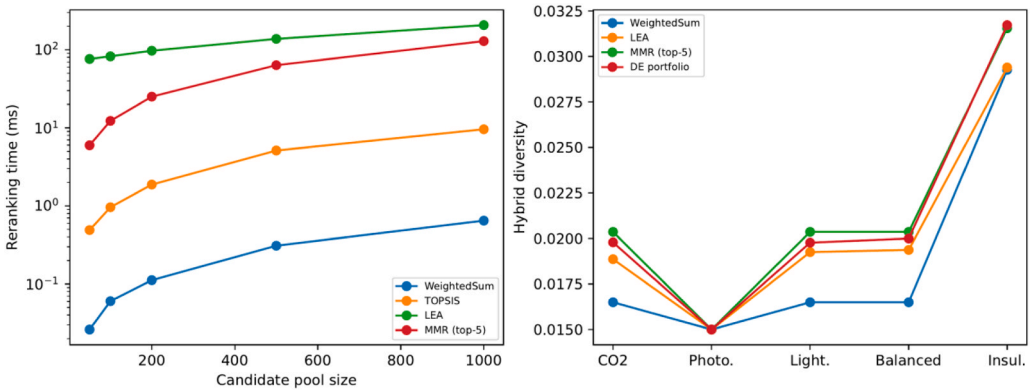


Fig. 4. Reranking scalability at fixed $K = 5$ and scenario-dependent hybrid diversity. Timing uses five scenarios, three seeds, and three repeats per setting; it excludes catalog scoring, embedding retrieval, network calls, and generation.

materials, corresponding to approximately 53.1% coverage. Void fraction is absent in the current metadata. (Figs. 3–5) summarizes this descriptor coverage.

The missing void-fraction field materially affects interpretation. The current evaluation supports density-aware and partial band-gap-aware ranking. It does not support strong porosity-aware, adsorption-specific, or CO_2 uptake claims. CO_2 adsorption recommendations should be read as preliminary recommendation behavior under incomplete metadata, not as validated adsorption screening.

Five query scenarios are evaluated: CO_2 adsorption, photocatalysis, lightweight storage, balanced screening, and insulating frameworks. Each query/seed uses a shared pool of 100 records and $K = 5$. Seeds 42–51 vary tie-breaking and stochastic search within five scenarios. The offline semantic score starts at 0.35, adds 0.20 times the fraction of query tokens of length greater than two found as exact case-folded alphanumeric tokens in metadata text, and then applies the documented scenario heuristic. Formula strings do not contribute to token overlap. Candidate generation sorts by semantic score, then masked relevance, then a seeded random tie-break. This deterministic proxy is distinct from the deployed SentenceTransformer/FAISS retriever and does not validate live retrieval.

Query weights start at (0.35, 0.25, 0.20, 0.20) in the order semantic, band gap, density, stability. Controlled token sequences recognize CO_2, carbon dioxide, or gas adsorption; photocatalysis, photocatalytic, or photocatalyst; lightweight, low density, or gas storage; balanced/multi-objective; and insulating, insulator, wide band gap, or wide-gap. Hyphens separate tokens; case normalization preserves the distinction between the chemical token Co_2 and CO_2. Formula fields and serialized formula lines are excluded from application-text overlap. Matching operates on whole tokens or consecutive token sequences. Photocatalysis and insulating rules add 0.25 to band gap; CO_2 and lightweight rules add 0.15 to density; stable/stability adds 0.20 to stability. Balanced screening activates the base profile without an additional increment. Each objective increment applies at most once, then weights are normalized without rounding. These rules and queries were fixed, and their configuration and source files hashed, before the final evaluation was run; the corrections relative to the original submission were

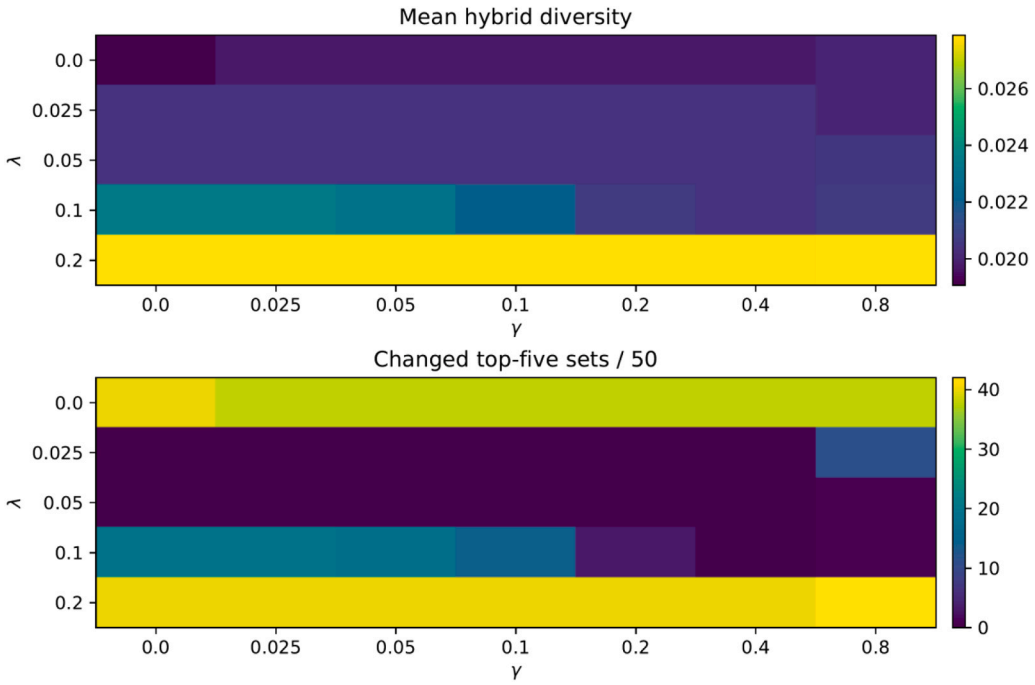


Fig. 5. Joint coefficient sensitivity. The upper panel reports diversity; the lower panel reports changed top-five sets relative to baseline. Changing the selected set is distinguished from changing the reported mean utility.

identified by auditing which rules were triggered, not by tuning metrics. They are transparent preferences, not fitted physical models.

1. stable porous MOFs for CO_2 adsorption:
2. (0.259259, 0.185185, 0.259259, 0.296296).
3. photocatalytic MOFs with suitable band gaps:
4. (0.280000, 0.400000, 0.160000, 0.160000).
5. lightweight gas storage MOFs with low density:
6. (0.304348, 0.217391, 0.304348, 0.173913).
7. balanced stable MOF candidates for general materials discovery:
8. (0.291667, 0.208333, 0.166667, 0.333333).
9. wide band gap insulating MOFs with low density:
10. (0.250000, 0.357143, 0.250000, 0.142857).

Band-gap utility is 1.0 at $E_g \geq 3.5$ eV and 0.5 otherwise for insulating queries; for other scenarios it is 1.0 within 1.0–3.5 eV and 0.5 otherwise. The moderate-gap preference is a preliminary photocatalysis heuristic: absorption, band edges, charge transport, and catalytic sites are not evaluated. Wide-gap preference is an electronic insulation heuristic, not a dielectric or conductivity measurement. Density utility is 1.0 below 1.0 g/cm^3, 0.7 from 1.0 to below 2.0, and 0.4 otherwise. Low density cannot substitute for porosity or adsorption uptake. If measured stability is absent, stability copies density utility; missing band gap is masked and void fraction is excluded.

Candidate-generation bonuses added to $0.35 + 0.20 \times$ token overlap are: CO_2, 0.08 for density below 1.8 and 0.04 for observed gap; photocatalysis, 0.22 for a gap in 1.0–3.5 eV and 0.04 for formula-parsed Ti, Zn, Cd, or Cu; lightweight, 0.25 below density 1.0 plus 0.10 below 1.5; balanced, 0.10 for observed gap, 0.10 below density 2.0, and 0.05 for at least four elements; insulating, 0.25 for a gap at least 3.5 and 0.08 below density 2.0. Density units are g/cm^3. Formula parsing supplies explicit compositional heuristics only, not text overlap or proof of activity. Scores are clipped to [0, 1]. These unchanged candidate-generation preferences are evaluated separately from reranking. Full scenario results and coefficient settings are in the supplement.

13. Recommendation evaluation metrics

13.1. Offline recommendation protocol without user histories

Because no historical user-click or rating log exists for QMOF materials, the evaluation follows an offline query-driven recommendation protocol. QMOF-Rec is a hybrid scientific recommender with a strong content-based focus: it does not rely on collaborative filtering because no large-scale user–MOF interaction log is available. The task is similar to session-based or context-aware recommendation, where the query supplies the current user intent and item content supplies the ranking evidence. User feedback is included architecturally for future online learning, but the present evaluation uses query-dependent objective weights rather than user ratings.

13.2. Ranking, diversity, and coverage metrics

Mean relevance is computed as

$$\mathrm{Rel@}K(S_K) = \frac{1}{K}\sum_{m_i \in S_K} R(m_i, q) \tag{22}$$

If binary or graded expert labels become available, Precision@K can be computed as $\mathrm{Precision@}K(S_K) = \frac{1}{K}\sum_{m_i \in S_K} 1[\mathrm{rel}(m_i, q) \geq \theta]$, where θ is a relevance threshold. In the current offline protocol, no expert-labeled relevant set is available, so Precision@K is defined as a future supervised recommender-system metric. Discounted cumulative gain is

$$\mathrm{DCG@}K = \sum_{r=1}^{K} \frac{2^{\mathrm{rel}_r} - 1}{\log_2(r+1)}, \tag{23}$$

where $\mathrm{rel}_r = R(m_{(r)}, q)$ is the same masked utility used by WeightedSum. The ideal list sorts that utility within the identical candidate pool. Therefore WeightedSum maximizes Rel@K and this NDCG by construction; a tie indicates utility preservation, not independent relevance accuracy. Normalized discounted cumulative gain is $\mathrm{NDCG@}K = \mathrm{DCG@}K/\mathrm{IDCG@}K$. The final rerun keeps relevance scoring on the mask-aware active utility vector, but computes intra-list diversity (ILD) over a full-precision hybrid material representation containing observed

density, observed band gap, formula-derived descriptors, and formula-derived metadata proxies:

$$\mathrm{ILD}_{\mathrm{hybrid}}(S_K) = \frac{2}{K(K-1)} \sum_{i<j} d_{\mathrm{hybrid}}(h_i, h_j) \tag{24}$$

The hybrid distance masks missing physical dimensions, excludes void fraction, and uses full precision before any manuscript rounding. This representation is used for final diversity, ParetoCrowding crowding distance, MMR redundancy, and LEA list-level diversity. Unique@K is the fraction of unique items within a top-K list and is distinct from catalog coverage. Catalog coverage measures how much of the item space is recommended across a query set Q:

$$\mathrm{Coverage} = \frac{\left|\bigcup_{q \in Q} S_K(q)\right|}{|M|} \tag{25}$$

The ablation rerun also reports Unique@K within each recommendation list. Runtime is reported as mean runtime in milliseconds. The reported hypervolume value is a hypervolume proxy rather than exact Pareto hypervolume, computed per list as Rel@5 × ILD × $\bar{s}_{\mathrm{gap}}$ × $\bar{s}_{\mathrm{density}}$ × $\bar{s}_{\mathrm{stability}}$, then averaged over runs; it is a comparative proxy only.

Ranking uses the executable configuration configs/final_scenario_rules.json, shared candidate pools, and full-precision masked scoring. Machine-readable computational results are stored in artifacts/final_scenario_rules. The frozen configuration and source fingerprints identify the evaluated implementation.

14. Recommendation performance: ranking quality and portfolio diversity

LEA preserves WeightedSum utility and has modestly higher aggregate diversity (0.0204 versus 0.0188). Table 4 reports all primary comparators, including differential evolution (DE). These values describe this fixed query suite and representation, without a claim of universal superiority.

The DE comparator uses the differential-evolution method of Storn and Price [30], implemented in SciPy 1.13.1 [31], with 30 random-key vectors over the 100 candidates and 1800 list-objective evaluations (initial population plus 59 generations). The five largest keys select five distinct records. One initial member encodes WeightedSum's top five and the other 29 are random. The fixed portfolio utility is $\overline{R} + 0.10\overline{B} + 0.10\mathrm{ILD} - 0.05\exp(-\mathrm{ILD})$. Selected items are ordered by masked relevance. This is an additional portfolio-search baseline, not an exact implementation of LEA's changing candidate-level fitness; equal evaluation counts do not imply equal computational work. Distance preparation is included in its reported runtime. No tuning against external outcomes is performed.

MMR stops after selecting five records and takes 12.64 ms, compared with 85.21 ms for LEA. Its aggregate diversity is 0.0215; the matched workload provides no runtime rationale for preferring LEA.

Table 4
Ranking results; five scenarios, ten seeds, shared pools of 100, $K = 5$. Runtime includes method-specific preprocessing and excludes catalog scoring and retrieval. The product-based HV proxy is not Pareto hypervolume.

Method	Rel@5	Diversity	NDCG@5	HV proxy	ms
WeightedSum	0.9004	0.0188	1.0000	0.0170	0.05
LEA	0.9004	0.0204	1.0000	0.0184	85.21
TOPSIS	0.9004	0.0188	1.0000	0.0170	1.07
ParetoCrowding	0.8972	0.0240	0.9952	0.0168	219.84
MMR (top-5)	0.9004	0.0215	1.0000	0.0195	12.64
Random	0.8019	0.0246	0.8552	0.0087	0.05
SemanticOnly	0.7928	0.0240	0.8452	0.0073	0.03
DE portfolio	0.9004	0.0213	1.0000	0.0192	67.14

14.1. Example recommendation scenarios

(Tables 5–8) reports the highest-ranked LEA recommendation for each evaluated query scenario from the predefined offline protocol. These examples illustrate the recommendation use case and should be read as offline recommendation outputs, not as experimentally validated application-performance results.

CO_2 recommendations use the density and completeness evidence specified in Methods; they are not adsorption predictions. Photocatalysis increases band-gap weight and prefers the moderate-gap interval. Lightweight storage increases density weight. Balanced screening retains the base profile plus the explicit stability modifier. Insulating frameworks increase both wide-gap and low-density preference. None of these descriptions implies external application validation.

Table 5 contains a top candidate repeated in 4 of five scenarios. The first four scenarios share the moderate-gap utility and low-density preference (also used as a stability fallback); binned utilities create ties, while LEA's best-ever fitness favors particular records within those ties. Different weights therefore need not produce different top candidates. Insulating scoring changes the preferred gap regime. Pool-overlap and shortlist-overlap summaries in the supplement quantify the resulting scenario differentiation.

15. Engineered metadata representations

The evaluated protocol uses engineered proxy scores. It does not call a trained GNN, consume atomistic CIF graphs, or supply a matched learned ranking experiment. No conclusion about learned graph-model effectiveness follows from these proxy comparisons.

16. Component ablations and metadata-proxy augmentation

Disabling both diversity-dependent terms changes mean diversity from 0.0204 to 0.0191. Setting only gamma to zero retains the exponential redundancy term and yields 0.0204; no self-cleaning yields 0.0204. These results do not establish independent benefits for every mechanism. Zero-imputed WeightedSum is a same-pool scoring diagnostic evaluated by original masked utility. Its null selected-list effect is specific to pools whose top records have observed gaps. Uniform-pool LEA changes the feasible set and is not a same-pool optimization comparison.

17. Scenario-level analysis and coefficient sensitivity

The independent descriptive units are five query scenarios; ten seeds measure repeated algorithm behavior, not 50 independent scientific tasks. LEA–WeightedSum diversity differences are 0.0024, 0.0000, 0.0028, 0.0029, 0.0001 in the listed scenario order. CO_2 contributes 29.1% of the summed difference. Complete scenario results are in Supplementary Table S1; these descriptive differences do not establish statistical superiority.

The joint sweep uses gamma values 0, 0.025, 0.05, 0.10, 0.20, 0.40, and 0.80, and redundancy coefficients 0, 0.025, 0.05, 0.10, and 0.20 on all 50 query/seed pools: 1750 runs. We define observable activation by a changed top-five set, order, or unrounded metric; a nonzero coefficient alone is not evidence of an output effect. Coefficients are retained at their predefined baseline, not selected from the sweep. The sweep shows that the redundancy weight λ, not γ, is the active coefficient. At $\gamma = 0.10$, $\lambda = 0$ changes 38/50 top-five sets (diversity 0.0198), $\lambda = 0.10$ changes 14/50 (diversity 0.0220, Rel@5 0.8951), and $\lambda = 0.20$ changes 40/50 (diversity 0.0278, Rel@5 0.8713), so higher diversity is obtained only at a utility cost. γ has no effect at the baseline $\lambda = 0.05$ for $\gamma \leq 0.40$ but interacts with λ at 0.10, where $\gamma = 0$, 0.10, and 0.40 change 20/50, 14/50, and 0/50 sets (Supplementary Table S2). This interaction follows from the best-ever history: candidates scored in the first iteration, when the selected list is still empty, receive no redundancy penalty, so a larger λ lowers every later-scored candidate relative to them and a larger γ offsets that shift. The

Table 5
Example seed-42 LEA top-1 recommendations. Density is in g/cm^3 and band gap in eV. Void fraction is unavailable. The LEA utility is a weighted optimization score and is not a probability or a quantity constrained to [0, 1].

Scenario	QMOF ID	Formula	Band gap	Density	LEA score	Rationale
CO_2	qmof-13a76dd	$Zn_4C_{40}H_{28}N_{12}O_8$	3.309	0.983	0.9396	Density; observed gap
Photocatalysis	qmof-13a76dd	$Zn_4C_{40}H_{28}N_{12}O_8$	3.309	0.983	0.9847	Moderate band gap
Lightweight	qmof-13a76dd	$Zn_4C_{40}H_{28}N_{12}O_8$	3.309	0.983	1.0056	Density below 1
Balanced	qmof-13a76dd	$Zn_4C_{40}H_{28}N_{12}O_8$	3.309	0.983	0.9660	Shared utility profile
Insulating	qmof-4050181	$Li_4C_{28}H_{32}N_4O_8$	5.066	0.936	1.0281	Wide band gap

Table 6
Scope of engineered representation evidence.

Component	Executable evidence
Proxy S/A	Formula, density, and band-gap heuristics
GNN ranking	No trained GNN is called by the protocol
CIF structures	Available separately; unused in this ranking protocol
Learned models	No matched targets, training splits, or evaluated checkpoints

Table 7
Component diagnostics. LEA is repeated as the reference; principal baselines appear only in Table 4. True no diversity sets both $\gamma = 0$ and redundancy coefficient $\lambda = 0$. Proxy variants are engineered metadata augmentation. The uniform-pool diagnostic changes candidate generation and is not a same-pool ranking comparison.

Variant	Rel@5	NDCG@5	Diversity	ms
LEA	0.9004	1.0000	0.0204	85.21
LEA $\gamma = 0$ only	0.9004	1.0000	0.0204	84.76
LEA true no diversity	0.9004	1.0000	0.0191	85.84
LEA no self-clean	0.9004	1.0000	0.0204	84.72
Proxy-S scoring	0.9004	1.0000	0.0199	0.05
Proxy-A scoring	0.9004	1.0000	0.0151	0.04
LEA + proxy S	0.9004	1.0000	0.0199	84.67
LEA + proxy A	0.9004	1.0000	0.0151	84.38
Zero-imputed WeightedSum	0.9004	1.0000	0.0188	0.10
LEA uniform pool	0.8298	0.9987	0.0276	83.00

Table 8
Gamma sensitivity with redundancy coefficient fixed at 0.05. Changed sets are relative to baseline gamma= 0.10.

γ	Rel@5	Diversity	Changed sets
0.0	0.9004	0.0204	0/50
0.025	0.9004	0.0204	0/50
0.05	0.9004	0.0204	0/50
0.1	0.9004	0.0204	0/50
0.2	0.9004	0.0204	0/50
0.4	0.9004	0.0204	0/50
0.8	0.9003	0.0205	1/50

Table 9
Actual scope of the deterministic explanation routine.

Output/check	Interpretation
30 candidate-conditioned templates	Reproducible local output; no live LLM call
Identifier inclusion	String-presence check against selected IDs
Limitation wording	Template/phrase check
Metadata and violation scores	Assigned in code; not independent measurements
Expert usefulness and faithfulness	Not evaluated

diversity gain at $\lambda \geq 0.10$ is therefore at least partly an initialization effect, not purely redundancy avoidance. The supplement reports every coefficient pair and scenario-level evidence, including null effects. Numerical fitness histories are also hashed in the artifacts.

18. Auxiliary explanation and evidence limits

Explanation is an auxiliary interface module; it has no effect on candidate generation, objective weights, or rankings in this benchmark. The runner emits 30 deterministic templates conditioned on seed-42 LEA candidate identifiers. It checks identifier presence and limitation phrases, but directly assigns metadata consistency, the nominal explanation-quality score, and all unsupported-claim flags. Those assigned values cannot demonstrate factual accuracy or a zero hallucination rate. Table 9 reports the scope of these checks.

A future explanation benchmark requires independent claim checking, source attribution, expert ratings, and adversarial prompts. Interface behavior is not evaluated by these deterministic checks.

19. Runtime, scalability, and descriptor-level screening

Reranking time was measured for WeightedSum, TOPSIS, LEA, and top-five MMR at pool sizes 50, 100, 200, 500, and 1000, using five scenarios, seeds 42–44, and three repeats (45 measurements per method/size). These are serial wall-clock measurements on a MacBook Pro (Mac14,9; Apple M2 Pro, 10 CPU cores, 32 GB RAM; macOS 26.6.2; Python 3.9.6). Thread-control environment variables were not set, and warm-up/cache effects were not separately controlled. Timings exclude shared catalog scoring, embedding retrieval, network calls, and explanation generation. Mean and standard deviation, plus raw measurements, are supplied in the supplement and artifacts. At $n = 100$, mean times for WeightedSum, TOPSIS, MMR, and LEA are 0.060, 0.962, 12.290, 82.436 ms, respectively; at $n = 1,000$ they are 0.647, 9.589, 129.118, 207.488 ms. At fixed K, top-five MMR uses $O(nK^2)$ distance work. LEA repeatedly remaps population vectors against the pool, with dominant work proportional to population size, iterations, pool size, and objective dimension. These measurements establish local post-retrieval scaling only.

Descriptor checks compare the catalog, candidate pool, Random, WeightedSum, and LEA, including missingness. These descriptors also influence selection, so the checks measure internal consistency rather than independent physical validation. For lightweight storage, mean densities in that order are 1.696, 0.688, 0.683, 0.859, 0.890 g/cm^3. Thus reranking raises density relative to the pool and Random even though it remains below the catalog mean; every pooled and selected record is below 1 g/cm^3. Only 7.0% of pool records have observed gaps, whereas WeightedSum and LEA select records with observed gaps and higher combined utility. Candidate generation and reranking therefore have different density effects. Selected LEA mean gaps are 2.804 eV for photocatalysis and 4.311 eV for insulating frameworks; all observed insulating selections have gaps at least 3.5 eV. Neither density nor gap alone establishes adsorption, photocatalysis, or insulation performance.

20. Discussion

The evidence supports an auditable screening formulation under incomplete metadata, with established algorithms as components. Relevance and NDCG are aligned with the utility that WeightedSum optimizes, so their ties cannot validate scientific relevance independently. Full-precision hybrid distances resolve variation hidden by

binned objectives, but the resulting diversity remains representation-dependent and partly heuristic.

LEA lies between WeightedSum and MMR in aggregate diversity. CO_2 accounts for 29.1% of the summed scenario differences; the remaining pattern is scenario dependent. At penalty 0.05, gamma= 0 changes 0/50 sets relative to baseline, while gamma= 0.8 changes 1/50. The true no-diversity comparison isolates removal of both diversity-dependent terms, but gamma-only and self-cleaning null effects limit mechanistic claims. DE uses a fixed list objective, whereas LEA retains history-dependent candidate fitness. Neither comparison proves general superiority.

Top-five MMR is faster than LEA and has higher aggregate diversity. This fixed-budget benchmark therefore does not support choosing LEA on runtime grounds. Metric ties reflect the engineered utility rather than independent scientific relevance.

Learned atomistic GNNs and transformers answer a different representation-learning question from the present engineered-proxy benchmark. A fair learned comparison needs matched inputs, training splits, checkpoints, and targets. Although CIFs exist locally, the final protocol supplies none of these trained ranking comparisons. Bayesian optimization and active learning additionally require a sequential acquisition task and an oracle or held-out property target; the current deterministic utility is not such validation. These remain explicit gaps rather than artificial comparison rows.

21. Limitations

Only five engineered query scenarios, $K = 5$, incomplete descriptors, heuristic stability, controlled phrase weighting, and deterministic semantic proxies are evaluated. Band gap and density influence both selection and evaluation. The stability fallback duplicates the density score when measured stability is absent. The engineered metadata-proxy block uses zero for missing band gap and is not fully mask-pure even though the physical coordinate is masked. No independent adsorption, synthesis, porosity, expert relevance, new DFT, online-user, or generative-discovery validation is provided. The metadata-proxy rows are engineered scores, not trained GNN predictions. Explanation templates and assigned flags cannot establish LLM faithfulness. The redundancy penalty is not applied in the first LEA iteration, so coefficient sensitivity partly reflects this initialization effect. DE and LEA optimize differently defined utilities, and the limited search budget and query suite do not establish general optimizer rankings. Timing excludes retrieval, catalog scoring, and generation, and should not be read as production latency.

22. Future work

Future work should rebuild the full QMOF descriptor matrix, add real embedding-based retrieval evaluation, incorporate CIF-derived graph embeddings, evaluate graph-aware recommendation on larger query suites, assess LLM faithfulness and citation fidelity with expert review, expand comparisons with evolutionary and swarm-based baselines such as NSGA-II, PSO, and GA, add Bayesian optimization as a separate model-based sequential optimization baseline, evaluate additional MMR settings, and optimize deployment latency.

Recommender-system extensions should include expert-labeled relevance benchmarks, user studies with materials scientists, active-learning from feedback, preference-aware recommendation for application domains, cold-start analysis for newly added QMOFs, learning-to-rank baselines, and explainable-recommendation evaluation with faithfulness and usefulness metrics. Live RAG/LLM extensions should include multi-turn dialogue evaluation, citation fidelity scoring, expert usefulness ratings, and adversarial robustness testing against fabrication-inducing prompts.

23. Conclusion

QMOF-Rec formulates screening of existing records under incomplete metadata through separate masked utility, material diversity, candidate generation, and explanation. Five predefined scenario profiles support transparent comparisons of ranking policies, component ablations, coefficient sensitivity, and computational scaling. LEA preserves utility with intermediate aggregate diversity, while matched top-five MMR is faster and more diverse. These findings support explicit evidence boundaries and reproducible screening diagnostics, not general optimizer superiority or validated materials performance. Independent application outcomes, learned representations, and expert-assessed explanations remain necessary extensions.

Author contributions (CRediT)

Soumya Sekhar Banerjee: Software, Investigation, Visualization, Writing—original draft. Arnd Koeppe: Methodology, Writing—review & editing, including critical evaluation of the recommendation and validation framework and interpretation of the analyses. Mehrdad Jalali: Conceptualization, Methodology, Supervision, Writing—original draft, Writing—review & editing.

CRediT authorship contribution statement

Mehrdad Jalali: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Soumya Sekhar Banerjee:** Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis. **Arnd Koeppe:** Writing – review & editing, Validation, Methodology, Formal analysis, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

Generative AI tools were used to assist with language editing, consistency checking, code-review support, and figure-preparation support. The authors are responsible for reviewing and approving the final text, code changes, figures, and scientific conclusions before submission.

Code availability

The executable configuration, runners, tests, and machine-readable evidence are in the source repository: https://github.com/MehrdadJalali-AI/qmof-rec. The evaluated configuration is configs/final_scenario_rules.json. The evaluated source is tagged v1.2-COMMT-revision (commit b687056c).

Demo availability

A live interactive demonstration of the QMOF-Rec user interface is available at https://qmofrecsystem.netlify.app/. The deployment is provided solely as an interface demonstration and does not replace the offline ranking experiments, metadata-proxy analyses, coefficient sensitivity studies, or deterministic explanation-template checks reported in this manuscript.

Funding

The authors received no specific funding for this work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors thank SRH University of Applied Sciences Heidelberg for institutional support.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.commt.2026.100068.

Data availability

Research Link Provided

References

[1] A.S. Rosen, et al., Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery, Matter 4 (5) (2021) 1578–1597.
[2] A.S. Rosen, et al., High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration, npj Comput. Mater. 8 (2022) 112.
[3] A. Seko, H. Hayashi, H. Kashima, I. Tanaka, Matrix- and tensor-based recommender systems for the discovery of currently unknown inorganic compounds, Phys. Rev. Mater. 2 (2018) 013805.
[4] Y. Luo, K. Yan, S. Ji, MOFormer: self-supervised transformer model for metal–organic framework property prediction, J. Am. Chem. Soc. 145 (3) (2023) 1615–1624.
[5] J. Qu, Y.R. Xie, K.M. Ciesielski, C.E. Porter, E.S. Toberer, E. Ertekin, Leveraging language representation for materials exploration and discovery, npj Comput. Mater. 10 (2024) 58, https://doi.org/10.1038/s41524-024-01231-8.
[6] T. Xie, J.C. Grossman, Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties, Phys. Rev. Lett. 120 (2018) 145301.
[7] C. Chen, W. Ye, Y. Zuo, C. Zheng, S.P. Ong, Graph networks as a universal machine learning framework for molecules and crystals, Chem. Mater. 31 (9) (2019) 3564–3572.
[8] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, Int. Conf. Learn. Represent. (2017).
[9] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. 30 (2017).
[10] P. Veličković, et al., Graph attention networks, Int. Conf. Learn. Represent. (2018).
[11] M. Jalali, M. Tsotsalas, C. Wöll, MOFSocialNet: Exploiting metal–organic framework relationships via social network analysis, Nanomaterials 12 (4) (2022) 704, https://doi.org/10.3390/nano12040704.
[12] M. Jalali, M. Schmitt, A.D. Dinga Wonanke, C. Wöll, P. Friederich, MOFGalaxyNet: a social network analysis for predicting guest accessibility in metal–organic frameworks utilizing graph convolutional networks, J. Chemin.-. 15 (2023) 99.
[13] M. Jalali, A.D. Dinga Wonanke, P. Friederich, C. Wöll, The black hole strategy: Gravity-based representative sampling for frugal graph learning on metal–organic framework networks, J. Chem. Inf. Model. 65 (20) (2025) 10885–10902.
[14] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs, IEEE Trans. Big Data 7 (3) (2019) 535–547.
[15] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, Proc. EMNLP-IJCNLP (2019) 3982–3992.
[16] P. Lewis, et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, Adv. Neural Inf. Process. Syst. 33 (2020) 9459–9474.
[17] A. Cierpka, M.S. Islam, J. Steinhülb, E.D.S. Domtchoueng, M. Selzer, A. Koeppe, KadiAssistant: a conversational AI agent for information retrieval in Kadi4Mat, arXiv 2605 (2026) 18850.
[18] R. Burke, Hybrid recommender systems: Survey and experiments, Use Model. Use -Adapt. Interact. 12 (2002) 331–370.
[19] P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends. Recommender Systems Handbook, Springer, 2011, pp. 73–105.
[20] F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011.
[21] G. Adomavicius, Y. Kwon, New recommendation techniques for multicriteria rating systems, IEEE Intell. Syst. 26 (5) (2011) 48–55.
[22] D. Jannach, M. Zanker, M. Fuchs, Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations, Inf. Technol. & Tour. 14 (2014) 119–149.
[23] Y. Zhang, X. Chen, Explainable recommendation: a survey and new perspectives, Found. Trends Inf. Retr. 14 (1) (2020) 1–101.
[24] J. Carbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries. Proc. 21st, ACM SIGIR, 1998, pp. 335–336.
[25] C.-L. Hwang, K. Yoon, Multiple Attribute Decision Making: Methods and Applications, Springer, 1981.
[26] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2) (2002) 182–197.
[27] B. Shahriari, et al., Taking the human out of the loop: a review of Bayesian optimization, Proc. IEEE 104 (1) (2016) 148–175.
[28] E. Dalirinia, M. Jalali, M. Yaghoobi, H. Tabatabaee, Lotus effect optimization algorithm (LEA): a lotus nature-inspired algorithm for engineering design optimization, J. Supercomput. 80 (2024) 8543–8597.
[29] M. Jalali, B. Vu, S. Mehraeen, S. Chandna, F. Jalali, MOF-LENS: Bio-inspired lotus effect optimization for accelerated discovery of metal–organic framework nanocarriers for doxorubicin delivery in cancer therapy, Mater. Today Commun. 49 (2025) 114448.
[30] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359, https://doi.org/10.1023/A:1008202821328.
[31] SciPy developers. Differential evolution, SciPy 1.13.1 reference manual. https://docs.scipy.org/doc/scipy-1.13.1/reference/generated/scipy.optimize.differential_evolution.html.