

Mathematical Programming in Machine Learning and Artificial Intelligence: A Unified Taxonomy of Models and Applications

CHAOSHENG DONG

Mathematical programming provides a common language for many decisions embedded in modern machine-learning (ML) and artificial-intelligence (AI) systems: selecting retrieval context, routing tokens, allocating inference compute, fitting structured predictors, protecting against distribution shift, and balancing competing objectives. However, the relevant literature is fragmented across optimization, information retrieval, recommendation, natural-language processing, computer vision, and learning theory. This paper organizes various applications under common mathematical programming paradigms: linear, quadratic, binary and mixed-integer, conic, bilevel, multi-objective, inverse, distributionally robust, submodular, and min-max optimization. We normalize the models with a mostly unified notation and, for every application, identify inputs, decision variables, a principal formulation, structural properties, solution strategies, and limitations. Across paradigms, we compare tractability, relaxation quality, decomposition, approximation guarantees, and scalability bottlenecks. The paper shows that mathematical programming is most useful not as a claim that all learning is LP or MIP, but as a disciplined interface between predictions and constrained decisions.

CCS Concepts: • **Theory of computation** → **Mathematical optimization**; • **Computing methodologies** → **Machine learning**; • **Information systems** → *Information retrieval*.

Additional Key Words and Phrases: mathematical programming, machine learning, artificial intelligence

ACM Reference Format:

Chaosheng Dong. 2026. Mathematical Programming in Machine Learning and Artificial Intelligence: A Unified Taxonomy of Models and Applications. *ACM Trans. Inf. Syst.* 0, 0, Article 0 (August 2026), 32 pages. <https://doi.org/10.0000/0000000>

1 Introduction

The output of an ML model is often an input to a decision. A retrieval score must be converted into a context that fits a token window; a mixture-of-experts gate must respect capacity; a recommender must construct a list rather than score items independently; and an alignment procedure must remain useful under shifts in preference data. These decisions couple predictions through budgets, assignments, diversity, dynamics, or adversaries. Mathematical programming (MP) makes those couplings explicit and exposes which guarantees survive when learned quantities enter an optimization model.

This perspective is especially natural for information systems. Search, recommendation, retrieval-augmented generation (RAG), notification delivery, and large language model (LLM) inference all combine statistical estimates with resource-constrained actions. The same modeling vocabulary also clarifies classical learning procedures, including sparse recovery, manifold learning, grouped regression, structured model selection, and adversarial generation. The point is not that every

Author's Contact Information: Chaosheng Dong, ensteindcs@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1558-2868/2026/8-ART0

<https://doi.org/10.0000/0000000>

algorithm is implemented by a generic solver. Rather, an MP formulation can serve as an exact computational model, an equivalent representation that reveals structure, or a normative reformulation useful for analysis and system design.

This paper makes four contributions. First, it provides a taxonomy connecting MP paradigms to ML/AI applications. Second, it gives normalized formulations using a mostly unified notation, so that similarities such as budgets, simplex constraints, and nested objectives are visible across domains. Third, it distinguishes *explicit*, *equivalent*, and *natural* MP relationships to the cited primary literature. Fourth, it compares tractability, relaxations, decomposition, approximation, and scalability, including where solver certificates are practical and where a formulation primarily guides a heuristic.

The paper is organized as follows. Sections 2–12 develop the paradigms and their application models; Section 13 compares them; and Section 14 identifies research directions.

2 Scope, Taxonomy, and Mathematical Programming Preliminaries

In operations research, MP denotes finite-dimensional optimization with objectives and constraints, especially linear programming (LP), quadratic programming (QP), mixed-integer programming (MIP), and conic programming. Our scope in this paper is deliberately broader: it also includes bilevel, multi-objective, inverse, distributionally robust optimization (DRO), submodular optimization, and min-max games. These classes retain the central MP discipline of explicit decisions, feasible sets, and objectives, even when they are nonsmooth, combinatorial, infinite-dimensional before dualization, or game-theoretic. This broader scope is necessary to describe modern ML/AI faithfully; it does not imply that every class reduces efficiently to LP or MIP.

We use three relationship labels. *Explicit formulation* means that the cited work directly formulates the application in the stated paradigm or an unmistakably matching form. *Equivalent formulation* means that the displayed program is mathematically equivalent to the cited objective but may expose auxiliary variables or a lifted representation not emphasized by the source. *Natural MP reformulation* means that we translate a source’s algorithmic or system description into a useful MP model; we do not make the claim that the paper solved that displayed program.

For a generic continuous model, let the decision be $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, the objective be $f(\mathbf{x})$, inequality functions be $g_u(\mathbf{x})$ for u in a finite index set $\mathcal{U}$, and equality functions be $h_v(\mathbf{x})$ for v in a finite index set $\mathcal{W}$. These symbols are local to the next display:

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad g_u(\mathbf{x}) \leq 0 \ (u \in \mathcal{U}), \quad h_v(\mathbf{x}) = 0 \ (v \in \mathcal{W}). \quad (1)$$

Convexity of f , the functions g_u , and the set $\mathcal{X}$, together with linear equalities, makes local optima global. Binary variables replace a continuous feasible region by a discrete one, usually making worst-case solution time exponential unless special structure applies. Relaxations enlarge a feasible set to produce bounds; rounding or primal heuristics recover implementable decisions. Decomposition separates weakly coupled blocks, while approximation algorithms trade exactness for a provable or empirical quality guarantee.

2.1 Taxonomy at a Glance

We report the taxonomy of mathematical programming applications in machine learning and artificial intelligence in Table 1.

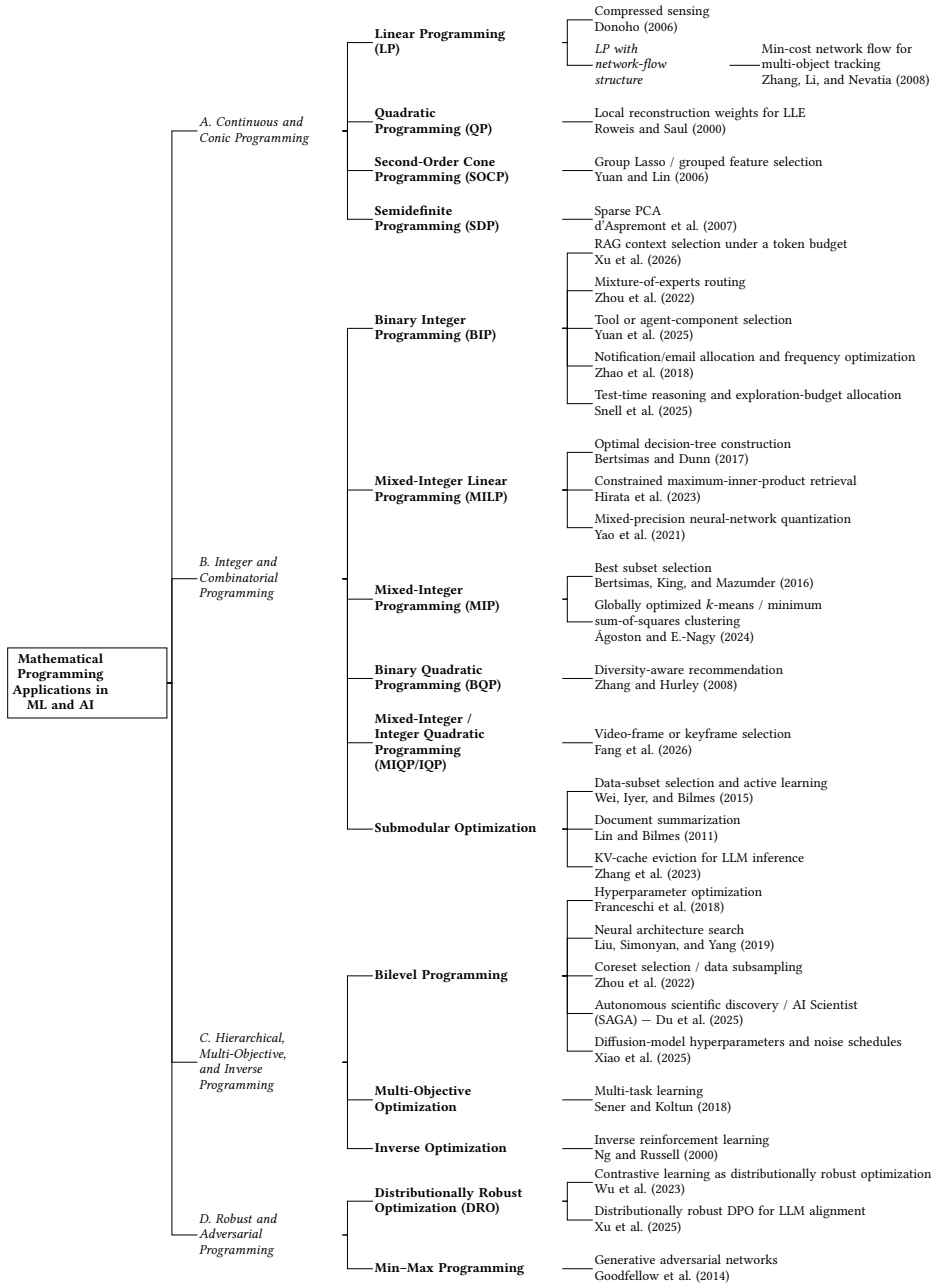


Fig. 1. Taxonomy of mathematical programming applications in machine learning and artificial intelligence.

Table 1. Unified taxonomy.

No.	Paradigm	ML/AI application	Main decision variable	Principal constraint or structure	Representative citation	MP relationship
1	LP	Compressed sensing	Sparse signal $\mathbf{x}$, absolute-value epigraph	Linear measurements; ℓ_1 epigraph	Donoho [8]	Equivalent formulation
2	LP (network flow)	Min-cost network flow for multi-object tracking	Detection and transition flows	Flow conservation, capacities, source/sink	Zhang et al. [30]	Explicit formulation
3	QP	Local reconstruction weights in LLE	Neighbor weights w_{ij}	linear sum-to-one; neighborhood support	Roweis and Saul [19]	Explicit formulation
4	BIP/MMKP	RAG context selection	Passage indicators z_i	Token/resource budgets; group choice	Xu et al. [25]	Explicit formulation
5	BIP	Mixture-of-experts routing	Token-expert assignments z_{ik}	Per-token assignment; expert capacity	Zhou et al. [36]	Natural MP reformulation
6	BIP	Tool or agent-component selection	Component indicators z_i	Budget and incompatibility constraints	Yuan et al. [29]	Explicit formulation
7	BIP	Notification/email allocation and frequency	User-message-time indicators	Frequency, channel, quota, and global budgets	Zhang et al. [32], Zhao et al. [34]	Natural MP reformulation
8	BIP/MCKP	Test-time reasoning and exploration budget	Query-compute-level choices	Exactly one level; global compute	Snell et al. [21]	Natural MP reformulation
9	MILP	Optimal decision-tree construction	Splits, leaf assignments, predictions	Tree logic and misclassification linearization	Bertsimas and Dunn [2]	Explicit formulation
10	MIQP	Best subset selection	Coefficients θ_j , indicators z_j	Cardinality and coefficient linking	Bertsimas et al. [3]	Explicit formulation
11	MILP/MIQP	Constrained maximum-inner-product retrieval	Retrieved-item indicators z_i	Cardinality, coverage/fairness; pairwise diversity	Hirata et al. [13]	Natural MP reformulation
12	BIP/ILP	Mixed-precision neural-network quantization	Layer-bit-width assignments	Exactly one width; size or latency budget	Yao et al. [27]	Explicit formulation
13	BQP/MIQP	Diversity-aware recommendation	List indicators z_i	Cardinality; pairwise similarity penalty	Zhang and Hurley [31]	Natural MP reformulation
14	MIQP/MINLP/MILP	Globally optimized k -means clustering	Assignments z_{ik} , centroids μ_k	Exactly one cluster; assignment-centroid coupling	Ágoston and Eisenberg-Nagy [1], Burgard et al. [4]	Explicit formulation
15	BQP/IQP	Video frame or keyframe selection	Frame indicators z_i	Cardinality/temporal budget; pairwise redundancy	Fang et al. [10]	Explicit formulation
16	SOCP	Group Lasso	Coefficients β_g , epigraphs t_g	Group-norm cones	Yuan and Lin [28]	Equivalent formulation
17	SDP	Sparse PCA	Lifted matrix $\mathbf{X} \succeq 0$	Trace normalization; rank relaxation; sparsity	d’Aspremont et al. [5]	Explicit formulation
18	Bilevel	Hyperparameter optimization	Hyperparameters λ , weights θ	Validation outer/training inner	Franceschi et al. [11]	Explicit formulation
19	Bilevel	Neural architecture search	Architecture λ , weights θ	Relaxed architecture outer/weight-training inner	Liu et al. [15]	Explicit formulation
20	Bilevel	Coreset selection and data subsampling	Selection probabilities/indicators and weights	Coreset-size outer; selected-data training inner	Zhou et al. [35]	Explicit formulation

Continued on next page

Table 1 continued

No.	Paradigm	ML/AI application	Main decision variable	Principal constraint or structure	Representative citation	MP relationship
21	Bilevel	Autonomous scientific discovery	Objective design λ , candidate design $\mathbf{x}$	Goal-evolution outer; scientific search inner	Du et al. [9], Ma et al. [16]	Equivalent formulation
22	Bilevel	Diffusion hyperparameter/noise-schedule optimization	Schedule λ , diffusion parameters θ	Quality outer; training/inference inner	Xiao et al. [24]	Explicit formulation
23	Multi-objective	Multi-task learning	Shared/task model parameters θ	Pareto order; simplex gradient QP	Sener and Koltun [20]	Explicit formulation
24	Inverse optimization	Inverse reinforcement learning	Reward weights θ	Expert policy optimality/Bellman inequalities	Ng and Russell [18]	Explicit formulation
25	DRO	Contrastive learning as DRO	Encoder parameters θ , negative law Q	Divergence ambiguity around empirical negatives	Wu et al. [23]	Explicit formulation
26	DRO	Distributionally robust DPO	Policy parameters θ , preference law Q	Wasserstein or KL ambiguity set	Xu et al. [26]	Explicit formulation
27	Submodular	Data subset selection and active learning	Selected set S	Cardinality; diminishing returns	Wei et al. [22]	Explicit formulation
28	Submodular	Document summarization	Selected sentence set S	Word budget; coverage and diversity	Lin and Bilmes [14]	Explicit formulation
29	Submodular	KV-cache eviction for LLM inference	Retained token set S_r	Dynamic cache budget; heavy hitters plus recency	Zhang et al. [33]	Explicit formulation
30	Min-max	Generative adversarial networks	Generator G , discriminator D	Two-player saddle objective	Goodfellow et al. [12]	Explicit formulation

2.2 Unified Notation

Bold lowercase letters denote vectors, bold uppercase letters matrices, calligraphic letters sets, and ordinary lowercase letters scalars. A symbol may acquire a narrowly specialized meaning only when that meaning is defined next to its equation. Table 2 lists the notations.

3 Linear Programming

LP optimizes an objective over linear equalities and inequalities. Its global optimality certificates and mature decomposition methods make it attractive when a learning-system decision is continuous or when an integral combinatorial model has an exact LP relaxation.

3.1 Compressed Sensing

Context. Compressed sensing reconstructs a high-dimensional signal from fewer linear measurements than ambient dimension by exploiting sparsity. Basis pursuit chooses, among signals consistent with the measurements, one with minimum ℓ_1 norm.

Inputs and decision variables. Let $\mathbf{A} \in \mathbb{R}^{p \times d}$ be a known sensing matrix with $p < d$, let $\mathbf{b} \in \mathbb{R}^p$ be the measured vector, and let the continuous decision $\mathbf{x} \in \mathbb{R}^d$ be the reconstructed signal. With auxiliary decisions $\mathbf{t} \in \mathbb{R}_+^d$, the canonical and LP forms are

$$\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{Ax} = \mathbf{b}, \quad (2)$$

$$\min_{\mathbf{x} \in \mathbb{R}^d, \mathbf{t} \in \mathbb{R}_+^d} \sum_{j=1}^d t_j \quad \text{s.t.} \quad \mathbf{Ax} = \mathbf{b}, \quad -t_j \leq x_j \leq t_j \quad (j = 1, \dots, d). \quad (3)$$

Interpretation. Measurement consistency is enforced exactly, while the epigraph inequalities make $t_j \geq |x_j|$; minimization makes them tight. Thus (3) is equivalent to (2). Sparsity is encouraged because the ℓ_1 unit ball has coordinate-aligned extreme points.

MP classification and structure. Equation (3) is an LP and hence convex. Recovery of the sparsest feasible signal is exact only under conditions on $\mathbf{A}$; LP optimality alone does not establish statistical identifiability.

Solution approaches. Interior-point or simplex methods apply directly; first-order primal–dual, operator-splitting, and homotopy methods exploit large sparse sensing operators.

MP relationship and citation. Equivalent formulation. Donoho [8] establishes the compressed-sensing role of ℓ_1 recovery. Our auxiliary-variable LP is exact; we do not attribute the elementary epigraph conversion as the paper’s contribution.

Limitations. The model abstracts measurement noise, structured sparsity, model mismatch, and the computational cost of applying $\mathbf{A}$; it also treats sparsity as the relevant inductive bias without validating it for a particular signal source.

3.2 LP with Network-Flow Structure

Network-flow optimization is a structured subclass of LP.

3.2.1 Min-Cost Network Flow for Multi-Object Tracking. Context. Given detections across video frames, global data association chooses which detections belong to the same trajectories, where trajectories start and end, and which observations are false positives.

Inputs and decision variables. Let $\mathcal{I}$ index detections ordered in time, and let $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{I}$ contain admissible forward-time transitions. Known costs are γ_i^{det} , γ_{ij}^{tr} , γ_i^{in} , and γ_i^{out} for using detection i , transition (i, j) , starting at i , and ending at i . For a specified number M of tracks, let continuous

Table 2. List of notation.

Symbol	Domain or type	Meaning	First section used
i, j	indices	Objects, candidates, samples, or items	Linear Programming
q	index	Query or request	Binary and Mixed-Integer Programming
g	index	Group	Binary and Mixed-Integer Programming
k	index	Cluster, expert, component, or discrete option	Binary and Mixed-Integer Programming
τ	nonnegative integer	Time step	Linear Programming
d, n, K, T	positive integers	Feature dimension, sample count, cluster/option count, and task/time horizon	Linear Programming
$\mathcal{I}, \mathcal{J}, \mathcal{G}$	finite sets	Corresponding object, candidate, and group index sets	Linear Programming
$\mathcal{K}, \mathcal{Q}, \mathcal{T}$	finite sets	Options/experts, queries, and tasks/time steps	Binary and Mixed-Integer Programming
$\mathcal{V}$	finite set	Ground set for subset selection	Submodular Optimization
$\mathbf{a}_i$	$\mathbb{R}^d$	Observed input or feature vector	Quadratic Programming
y_i	label/response space	Observed label or response	Binary and Mixed-Integer Programming
$\mathbf{A}, \mathbf{b}$	matrix/vector	Linear operator and observed right-hand side	Linear Programming
θ	parameter space Θ	Learned model parameters	Binary and Mixed-Integer Programming
λ	scalar or vector	Hyperparameters or trade-off parameters	Binary and Mixed-Integer Programming
$\mathbf{x}$	$\mathbb{R}^d$	Continuous decision or latent state	Linear Programming
z_i, z_{ik}	$\{0, 1\}$	Selection or assignment decisions	Binary and Mixed-Integer Programming
w_i, w_{ij}	$\mathbb{R}$ or $\mathbb{R}_+$	Continuous weights	Quadratic Programming
r_i	$\mathbb{R}$	Relevance, reward, or utility	Binary and Mixed-Integer Programming
c_i	$\mathbb{R}_+$	Resource cost	Binary and Mixed-Integer Programming
s_{ij}	$\mathbb{R}$	Similarity or pairwise interaction	Binary and Mixed-Integer Programming
B	$\mathbb{R}_+$	Resource budget	Binary and Mixed-Integer Programming
m	positive integer	Selection-cardinality limit	Binary and Mixed-Integer Programming
$S \subseteq \mathcal{V}$	set decision	Selected subset of a ground set	Submodular Optimization
L_{tr}	real-valued function	Training loss	Bilevel Programming
L_{val}	real-valued function	Validation loss	Bilevel Programming
$\hat{P}$	probability measure	Empirical distribution	Distributionally Robust Optimization
Q	probability measure	Adversarial or alternative distribution	Distributionally Robust Optimization
$\mathcal{U}(\hat{P})$	set of measures	Ambiguity set around $\hat{P}$	Distributionally Robust Optimization
Φ, Ψ	real-valued functions	Upper- and lower-level objectives	Bilevel Programming
$F(S)$	$\mathbb{R}$	Submodular set function	Submodular Optimization
$\ell(\theta; \xi)$	$\mathbb{R}$	Loss on random observation ξ	Distributionally Robust Optimization
$\mathbf{1}, \mathbf{I}$	vector/matrix	All-ones vector and identity matrix of contextual size	Linear Programming
$\ \cdot\ _p$	norm	ℓ_p vector norm	Linear Programming
$\langle \cdot, \cdot \rangle$	inner product	Euclidean/Frobenius inner product	Binary and Mixed-Integer Programming
$\mathbf{X} \succeq 0$	symmetric matrix	Positive-semidefinite constraint	Conic Programming
ϵ	random vector	Exogenous noise	Bilevel Programming

relaxation variables $z_i, w_{ij}, u_i, v_i \in [0, 1]$ denote detection use, transition flow, source flow, and sink

flow. The min-cost flow model is

$$\begin{aligned}
\min \quad & \sum_{i \in \mathcal{I}} \gamma_i^{\text{det}} z_i + \sum_{(i,j) \in \mathcal{E}} \gamma_{ij}^{\text{tr}} w_{ij} + \sum_{i \in \mathcal{I}} (\gamma_i^{\text{in}} u_i + \gamma_i^{\text{out}} v_i) \\
\text{s.t.} \quad & u_i + \sum_{j: (j,i) \in \mathcal{E}} w_{ji} = z_i = v_i + \sum_{j: (i,j) \in \mathcal{E}} w_{ij} \quad (i \in \mathcal{I}), \\
& \sum_{i \in \mathcal{I}} u_i = M, \quad \sum_{i \in \mathcal{I}} v_i = M, \\
& 0 \leq z_i, u_i, v_i \leq 1 \quad (i \in \mathcal{I}), \\
& 0 \leq w_{ij} \leq 1 \quad ((i,j) \in \mathcal{E}).
\end{aligned} \tag{4}$$

Interpretation. The two equalities at each detection enforce flow conservation: a selected detection has exactly one incoming source/transition unit and one outgoing sink/transition unit. Source and sink constraints create M complete trajectories, and unit capacities prevent one detection from serving two trajectories. Costs are commonly negative log-likelihood components.

MP classification and structure. This is an LP with a node–arc incidence matrix. With integral supplies and capacities, the incidence matrix is totally unimodular; therefore an extreme optimal solution is integral even though variables are declared continuous. Additional non-network coupling, such as general pairwise exclusion or higher-order occlusion constraints, can destroy this property.

Solution approaches. Successive shortest paths, network simplex, and specialized dynamic graph algorithms exploit the structure; generic LP solvers also certify optimality.

MP relationship and citation. Explicit formulation. Zhang et al. [30] map maximum-a-posteriori global association to min-cost network flow and extend the network for occlusion. Equation (4) is a normalized core of that explicit construction.

Limitations. The formulation presumes calibrated local costs, a chosen transition graph, and a fixed track count M . Long-range interactions, identity ambiguity, and errors in the detection stage are only represented through the supplied arcs and costs.

4 Quadratic Programming

QP has a quadratic objective and linear constraints. A positive-semidefinite Hessian yields a convex problem, while indefinite curvature yields nonconvexity even without integrality.

4.1 Local Reconstruction Weights in Locally Linear Embedding

Context. Locally Linear Embedding (LLE) estimates, for each observation, weights that reconstruct it from nearby observations. The low-dimensional embedding is subsequently asked to preserve these local linear relations.

Inputs and decision variables. For observation $i \in \mathcal{I}$, let $\mathbf{a}_i \in \mathbb{R}^d$ be observed, let $\mathcal{N}_i \subseteq \mathcal{I} \setminus \{i\}$ be its fixed neighborhood, and let $w_{ij} \in \mathbb{R}$ be the reconstruction weight on neighbor j . The local program is

$$\begin{aligned}
\min_{(w_{ij})_{j \in \mathcal{I}}} \quad & \left\| \mathbf{a}_i - \sum_{j \in \mathcal{I}} w_{ij} \mathbf{a}_j \right\|_2^2 \\
\text{s.t.} \quad & \sum_{j \in \mathcal{I}} w_{ij} = 1, \quad w_{ij} = 0 \quad (j \notin \mathcal{N}_i).
\end{aligned} \tag{5}$$

Interpretation. The objective measures local reconstruction error. The support constraint prohibits non-neighbor influence, and the linear sum-to-one condition makes the weights invariant to translating all inputs by a common vector.

MP classification and structure. Equation (5) is an equality-constrained convex QP because its Hessian is positive semidefinite. The problems decompose by i . A singular neighborhood covariance can make weights nonunique, though the objective remains convex.

Solution approaches. Each small QP can be solved through a KKT linear system; diagonal regularization stabilizes nearly singular neighborhoods. Parallelism is immediate across observations. Neighborhood construction, not the QP alone, often dominates large-scale cost.

MP relationship and citation. Explicit formulation. The constrained reconstruction problem is a defining step of LLE in Roweis and Saul [19]; (5) retains its neighborhood and linear-invariance structure.

Limitations. The model treats neighborhoods as correct and fixed, is sensitive to sampling density and noise, and does not itself determine the final embedding dimension or guarantee preservation of global geometry.

5 Binary and Mixed-Integer Programming

Binary integer programming (BIP) represents yes/no and assignment choices. Mixed-integer linear programming (MILP) adds continuous variables with linear algebra; mixed-integer quadratic programming (MIQP) permits quadratic objectives or constraints. Branch-and-bound, cutting planes, presolve, relaxations, and primal heuristics are the main exact-solver ingredients, but problem-specific decomposition is usually decisive at ML scale.

5.1 RAG Context Selection under a Token Budget

Context. After retrieval, a RAG system must choose a compact, nonredundant collection of passages that supports a query without exceeding the generator's context and auxiliary resource limits.

Inputs and decision variables. For a fixed query q , let groups $\mathcal{G}$ partition candidate passages, let $\mathcal{I}_g$ be the candidates in group g , and let $\mathcal{H}$ index resources. Candidate i has known faithfulness/relevance utility r_i , resource consumption $c_{ih} \geq 0$ for $h \in \mathcal{H}$ (including tokens), and each resource has budget B_h . The binary decision z_i indicates inclusion:

$$\begin{aligned} & \max_{z_i \in \{0,1\}} \sum_{g \in \mathcal{G}} \sum_{i \in \mathcal{I}_g} r_i z_i \\ & \text{s.t.} \quad \sum_{g \in \mathcal{G}} \sum_{i \in \mathcal{I}_g} c_{ih} z_i \leq B_h \quad (h \in \mathcal{H}), \\ & \quad \sum_{i \in \mathcal{I}_g} z_i \leq 1 \quad (g \in \mathcal{G}). \end{aligned} \tag{6}$$

Equality may replace the final inequality when exactly one representative per mandatory group is required.

Interpretation. The objective aggregates estimated evidential value. Resource rows enforce the token window and any redundancy or latency limits, while group-choice constraints avoid selecting near-duplicate alternatives from the same semantic group.

MP classification and structure. Equation (6) is a multidimensional multiple-choice knapsack problem (MMKP), hence a BIP and NP-hard in general. Its LP relaxation gives an upper bound; a single resource and no group coupling reduce it to 0–1 knapsack.

Solution approaches. Dynamic programming with Pareto-state pruning, branch-and-bound, Lagrangian relaxation of budgets, and value-density heuristics are representative. Learned utilities can be recalibrated online, but prediction uncertainty is separate from combinatorial optimality.

MP relationship and citation. Explicit formulation. Xu et al. [25] explicitly formulate context selection as MMKP and solve it with Pareto-pruned dynamic programming before an NLI-guided tree-search generation stage. The verifiable record is the 2026 arXiv preprint; no proceedings metadata is assumed here.

Limitations. Additive utilities imperfectly represent interactions among passages, group construction may be brittle, and token counts do not capture attention dilution or ordering. The downstream generator can still ignore selected evidence.

5.2 Mixture-of-Experts Routing

Context. A sparse mixture-of-experts layer assigns tokens to a small number of expert subnetworks. Routing should exploit affinity scores while preventing expert overload.

Inputs and decision variables. Let $\mathcal{I}$ index tokens, $\mathcal{K}$ experts, r_{ik} a known gate score, h_i the required number of assignments for token i , and C_k expert k 's capacity. Binary z_{ik} assigns token i to expert k :

$$\begin{aligned} \max_{z_{ik} \in \{0,1\}} \quad & \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} r_{ik} z_{ik} \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}} z_{ik} = h_i \quad (i \in \mathcal{I}), \\ & \sum_{i \in \mathcal{I}} z_{ik} \leq C_k \quad (k \in \mathcal{K}). \end{aligned} \tag{7}$$

Interpretation. The first constraints provide the requested sparse computation per token; the second implement finite expert buckets. Expert-choice routing reverses the conventional viewpoint by having experts choose high-scoring tokens, naturally fixing the second constraint while allowing heterogeneous token multiplicity.

MP classification and structure. The normalized model is a capacitated bipartite BIP. When h_i , C_k , and the constraint matrix have pure transportation structure, the LP relaxation is integral; side constraints or nonlinear gate effects may remove that property.

Solution approaches. Exact min-cost flow is available for the pure assignment form. Production routers commonly use parallel top- k , sorting, capacity dropping, or local balancing because solving a global assignment at every layer and batch can be too expensive.

MP relationship and citation. Natural MP reformulation. Zhou et al. [36] introduce expert-choice routing with fixed expert bucket sizes and variable experts per token. Equation (7) formalizes the capacity-aware assignment principle, but we do not claim their router invokes a BIP solver.

Limitations. Scores are treated as fixed and additive, communication topology and expert specialization dynamics are omitted, and a locally optimal route need not optimize end-to-end training loss.

5.3 Tool or Agent-Component Selection

Context. An agent composer selects which tools, models, memories, or specialized agents to activate for a task, balancing expected capability against monetary, latency, and token costs.

Inputs and decision variables. Let $\mathcal{I}$ index components, r_i be tested or predicted task utility, c_i cost, and B a total budget. Let $\mathcal{E}^-$ be known incompatible unordered pairs and $\mathcal{D}$ be ordered

dependency pairs, where $(i, j) \in \mathcal{D}$ means component i requires j . Binary z_i selects component i :

$$\begin{aligned}
 & \max_{z_i \in \{0,1\}} \quad \sum_{i \in \mathcal{I}} r_i z_i \\
 & \text{s.t.} \quad \sum_{i \in \mathcal{I}} c_i z_i \leq B, \\
 & \quad z_i + z_j \leq 1 \quad (\{i, j\} \in \mathcal{E}^-), \\
 & \quad z_i \leq z_j \quad ((i, j) \in \mathcal{D}).
 \end{aligned} \tag{8}$$

Interpretation. The knapsack row enforces the operating budget; conflict rows prevent unusable combinations; dependency rows make a selected component bring its prerequisite. Utilities can be updated after lightweight tests of candidates.

MP and structure. Equation (8) is a knapsack-style BIP with side constraints and is NP-hard. Dependencies can sometimes be contracted; conflicts induce a packing structure. Additive utility neglects complementarity unless interaction variables are added.

Solution approaches. MIP solvers provide bounds for moderate inventories. Online knapsack policies, greedy marginal-utility selection, Lagrangian prices, and cached evaluations support dynamic inventories and tight latency.

MP relationship and citation. Explicit formulation. Yuan et al. [29] explicitly frame automated agentic component selection through a knapsack approach that jointly accounts for utility, cost, and compatibility. Equation (8) is a compact deterministic version of that design problem.

Limitations. Component utility is task- and order-dependent, tests may be noisy, and budget feasibility does not guarantee safe tool interaction. Execution planning after selection is outside the model.

5.4 Notification/Email Allocation and Frequency Optimization

Context. A communication platform decides which message to send to which user and when, seeking engagement or long-term value without overwhelming users or violating operational quotas.

Inputs and decision variables. Let $\mathcal{I}$ index users, $\mathcal{J}$ campaigns/messages, $\mathcal{T}$ eligible times, and $\mathcal{H}$ channels. Message j uses channel $h(j) \in \mathcal{H}$. Known values are predicted utility $r_{ij\tau}$, cost $c_{ij\tau}$, user cap f_i , channel cap f_{ih} , campaign lower/upper quotas L_j, U_j , and budget B . Binary $z_{ij\tau}$ denotes a send:

$$\begin{aligned}
 & \max_{z_{ij\tau} \in \{0,1\}} \quad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{\tau \in \mathcal{T}} r_{ij\tau} z_{ij\tau} \\
 & \text{s.t.} \quad \sum_{j \in \mathcal{J}} \sum_{\tau \in \mathcal{T}} z_{ij\tau} \leq f_i \quad (i \in \mathcal{I}), \\
 & \quad \sum_{j: h(j)=h} \sum_{\tau \in \mathcal{T}} z_{ij\tau} \leq f_{ih} \quad (i \in \mathcal{I}, h \in \mathcal{H}), \\
 & \quad L_j \leq \sum_{i \in \mathcal{I}} \sum_{\tau \in \mathcal{T}} z_{ij\tau} \leq U_j \quad (j \in \mathcal{J}), \\
 & \quad \sum_{i,j,\tau} c_{ij\tau} z_{ij\tau} \leq B.
 \end{aligned} \tag{9}$$

Interpretation. Predicted long-term value is aggregated across sends. User and channel rows control frequency, campaign rows implement delivery commitments, and the last row limits shared resources.

MP and structure. This generalized allocation is a large BIP with knapsack and quota coupling. Without coupling, choices decompose by user; dual prices for shared constraints restore approximate decomposition. Lower quotas can make the model infeasible and should be monitored explicitly.

Solution approaches. Distributed Lagrangian methods, primal–dual pricing, LP relaxation plus repair, and partitioned heuristics are natural at billion-variable scale. Zhang et al. [32] demonstrate distributed near-optimal methods for generalized industrial knapsack models, but such methods are not universally exact.

MP relationship and citation. Natural MP reformulation. Zhao et al. [34] describe a deployed ML system for per-user notification-volume control; Zhang et al. [32] explicitly study massive knapsack optimization in a different production setting. Equation (9) is the paper’s joint user–message–time BIP and should not be attributed verbatim to either source.

Limitations. The model treats predicted effects as additive, abstracts causal carryover and fatigue, and does not encode content safety or consent semantics. Long-term value estimates can shift after the policy changes exposure.

5.5 Test-Time Reasoning and Exploration-Budget Allocation

Context. An inference service allocates a finite test-time-compute pool across heterogeneous queries. Harder queries may benefit from more samples, search depth, or verification, but allocating the largest level to every query is infeasible.

Inputs and decision variables. Let Q index queries and $\mathcal{K}_q$ the permitted compute levels for query q . Offline evaluation or a predictor supplies expected accuracy/utility r_{qk} and compute cost c_{qk} for level $k \in \mathcal{K}_q$. Binary z_{qk} chooses a level and B is the batch-wide compute budget:

$$\begin{aligned}
 & \max_{z_{qk} \in \{0,1\}} \quad \sum_{q \in Q} \sum_{k \in \mathcal{K}_q} r_{qk} z_{qk} \\
 & \text{s.t.} \quad \sum_{k \in \mathcal{K}_q} z_{qk} = 1 \quad (q \in Q), \\
 & \quad \sum_{q \in Q} \sum_{k \in \mathcal{K}_q} c_{qk} z_{qk} \leq B.
 \end{aligned} \tag{10}$$

Interpretation. Each query receives exactly one inference configuration; the single coupling row makes compute fungible across the batch. The utilities may represent pass probability, verifier score, or downstream value, but must be calibrated at each compute level.

MP and structure. Equation (10) is a multiple-choice knapsack BIP. It is NP-hard, but admits pseudopolynomial dynamic programming for integer budgets and a strong LP relaxation when level curves are well behaved.

Solution approaches. Dynamic programming, Lagrangian pricing of compute, greedy marginal-gain allocation, and batch-level MIP are viable. Within a query, the selected level can govern tree search or repeated sampling; that execution policy is not a second application.

MP relationship and citation. Natural MP reformulation. Snell et al. [21] show that test-time compute should be allocated adaptively according to problem difficulty and inference strategy. They do not present the across-query BIP in (10); it is our resource-allocation abstraction of their compute-optimal principle.

Limitations. Expected gains can be nonstationary and correlated across samples, compute levels may be interruptible rather than discrete, and verifier error can make measured utility diverge from answer correctness.

5.6 Optimal Decision-Tree Construction

Context. Optimal-tree learning selects all splits and leaf predictions jointly, trading training errors against tree complexity rather than greedily committing to one split at a time.

Inputs and decision variables. Let $\mathcal{I}$ index labeled observations $(\mathbf{a}_i, y_i)$ with $\mathbf{a}_i \in \mathbb{R}^d$ and class $y_i \in C$. A fixed tree template has branch nodes $\mathcal{B}$ and candidate leaves $\mathcal{L}$. For leaf l , sets $\mathcal{A}_l^L, \mathcal{A}_l^R \subseteq \mathcal{B}$ identify left and right ancestors. Binary d_t activates split t , p_{jt} selects feature j , z_{il} assigns sample i to leaf l , u_l activates leaf l , q_{lc} predicts class c , and e_i records error; $b_t \in \mathbb{R}$ is a threshold. With valid feature bounds U_t , big- M value M_t , strict-separation constant $\varepsilon > 0$, and complexity price $\lambda \geq 0$, a compact MILP is

$$\begin{aligned}
 \min \quad & \sum_{i \in \mathcal{I}} e_i + \lambda \sum_{t \in \mathcal{B}} d_t & (11) \\
 \text{s.t.} \quad & \sum_{j=1}^d p_{jt} = d_t, \quad 0 \leq b_t \leq U_t d_t & (t \in \mathcal{B}), \\
 & \sum_{l \in \mathcal{L}} z_{il} = 1 & (i \in \mathcal{I}), \\
 & \sum_{c \in C} q_{lc} = u_l, \quad z_{il} \leq u_l & (i \in \mathcal{I}, l \in \mathcal{L}), \\
 & z_{il} \leq d_t & (i \in \mathcal{I}, l \in \mathcal{L}, t \in \mathcal{A}_l^L \cup \mathcal{A}_l^R), \\
 & \sum_{j=1}^d a_{ij} p_{jt} \leq b_t + M_t(1 - z_{il}) & (i, l, t \in \mathcal{A}_l^L), \\
 & \sum_{j=1}^d a_{ij} p_{jt} \geq b_t + \varepsilon - M_t(1 - z_{il}) & (i, l, t \in \mathcal{A}_l^R), \\
 & e_i \geq z_{il} - q_{l, y_i} & (i \in \mathcal{I}, l \in \mathcal{L}), \\
 & d_t, p_{jt}, z_{il}, u_l, q_{lc}, e_i \in \{0, 1\}.
 \end{aligned}$$

Here and below, a quantifier such as $(i, l, t \in \mathcal{A}_l^L)$ abbreviates all $i \in \mathcal{I}$, $l \in \mathcal{L}$, and $t \in \mathcal{A}_l^L$.

Interpretation. Feature-selection and threshold rows define univariate splits. Assignment and path rows route every sample consistently; leaf rows choose one prediction at each used leaf; the last inequalities detect misclassification. Complexity regularization favors fewer active splits. Standard hierarchy constraints, omitted only to save display width, prevent an inactive parent from having an active descendant.

MP and structure. The model is an MILP. Integrality represents the tree topology and routing, while big- M path constraints can weaken the relaxation. The problem is combinatorial and globally optimizing even shallow trees can be expensive.

Solution approaches. Branch-and-cut with feature/threshold preprocessing, warm starts from greedy trees, symmetry breaking, valid inequalities, and decomposition by leaves are representative. Early termination yields a feasible tree plus an optimality gap rather than a false claim of exactness.

MP relationship and citation. Explicit formulation. Bertsimas and Dunn [2] formulate optimal classification trees using mixed-integer optimization and jointly optimize accuracy and complexity. Equation (11) is a normalized compact template, not a line-for-line reproduction.

Limitations. Big- M constants require careful scaling, training accuracy may not predict calibration or robustness, and exact solution is generally restricted to moderate data or shallow templates.

5.7 Best Subset Selection

Context. Best subset regression chooses at most m features while fitting coefficients by squared error, producing a sparse predictive model with an explicit cardinality budget.

Inputs and decision variables. Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be the design matrix and $\mathbf{y} \in \mathbb{R}^n$ the response. Continuous $\theta_j \in \mathbb{R}$ are regression coefficients, binary z_j selects feature j , and valid bounds $M_j > 0$ link the two:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^d, z \in \{0,1\}^d} \quad & \frac{1}{2} \|\mathbf{y} - \mathbf{A}\theta\|_2^2 \\ \text{s.t.} \quad & -M_j z_j \leq \theta_j \leq M_j z_j \quad (j = 1, \dots, d), \\ & \sum_{j=1}^d z_j \leq m. \end{aligned} \tag{12}$$

Interpretation. If $z_j = 0$, the linking bounds force coefficient θ_j to zero; if $z_j = 1$, it may vary within a valid range. The final row enforces the feature budget and the objective measures residual fit.

MP and structure. Equation (12) is a convex MIQP: its continuous quadratic objective has a positive-semidefinite Hessian, but binary selection makes the full problem nonconvex. Loose M_j values degrade the relaxation; perspective or indicator formulations can strengthen it.

Solution approaches. Modern MIP solvers combine branch-and-bound, cutting planes, and warm starts from sparse heuristics. Discrete first-order methods, screening, and local swaps rapidly produce incumbents; only a solver bound certifies their global gap.

MP relationship and citation. Explicit formulation. Bertsimas et al. [3] develop a modern mixed-integer optimization treatment of best subset selection, including warm starts and certifiable solutions.

Limitations. The formulation assumes squared-error adequacy and valid coefficient bounds, can be unstable under collinearity, and does not automatically account for post-selection uncertainty.

5.8 Constrained Maximum-Inner-Product Retrieval

Context. Constrained maximum-inner-product search (MIPS) retrieves a top- m set for a query while enforcing category, fairness, or coverage requirements. Pairwise diversity is an extension of the same retrieval application, not a separate application.

Inputs and decision variables. For query q , let $\mathbf{v}_q \in \mathbb{R}^d$ be its embedding and $\mathbf{a}_i \in \mathbb{R}^d$ item i 's embedding; the known score is $r_i = \langle \mathbf{v}_q, \mathbf{a}_i \rangle$. Matrix $\mathbf{C} \in \mathbb{R}^{h \times |I|}$ and vectors $\mathbf{l}, \mathbf{u} \in \mathbb{R}^h$ encode h lower/upper category, fairness, or coverage bounds. Binary z_i selects an item. The constrained linear case and pairwise-diverse case are respectively

$$\begin{aligned} \max_{z \in \{0,1\}^{|I|}} \quad & \sum_{i \in I} r_i z_i & \text{s.t.} \quad & \sum_i z_i = m, \quad \mathbf{l} \leq \mathbf{C}z \leq \mathbf{u}, \end{aligned} \tag{13}$$

$$\begin{aligned} \max_{z \in \{0,1\}^{|I|}} \quad & \sum_{i \in I} r_i z_i - \lambda \sum_{i < j} s_{ij} z_i z_j & \text{s.t.} \quad & \sum_i z_i = m, \quad \mathbf{l} \leq \mathbf{C}z \leq \mathbf{u}, \end{aligned} \tag{14}$$

where $s_{ij} \geq 0$ is pairwise similarity and $\lambda \geq 0$ its penalty weight.

Interpretation. Equation (13) chooses the highest total inner-product score among feasible lists. Equation (14) subtracts redundancy: two similar items incur a penalty only when both are selected.

MP and structure. The linear model is a BIP/MILP. The pairwise term makes the extension a binary quadratic program (BQP), commonly handled as MIQP after retaining the product or as MILP after exact product linearization. With nonnegative s_{ij} and maximization of its negative, the continuous quadratic curvature need not make a standard convex MIQP representation; discreteness remains central.

Solution approaches. Candidate generation by approximate MIPS followed by exact constrained reranking is common. Branch-and-bound, Lagrangian relaxation of group bounds, local swaps, and diversity-aware approximate indexes trade candidate recall against optimization cost.

MP relationship and citation. Natural MP reformulation. Hirata et al. [13] define categorical diversity-aware inner-product search and give a sublinear approximation with a probabilistic success guarantee. Equations (13)–(14) are a general MP normalization, particularly for linear fairness/coverage and pairwise diversity; they are not attributed as the paper’s solver model.

Limitations. The model optimizes only over the candidate universe supplied to it, similarity can be a poor proxy for semantic redundancy, and list constraints may conflict. Approximate indexing can miss the globally optimal feasible set before optimization starts.

5.9 Mixed-Precision Neural-Network Quantization

Context. Mixed-precision quantization assigns a bit-width to each network layer, minimizing predicted accuracy degradation under memory or latency limits.

Inputs and decision variables. Let $\mathcal{I}$ index layers and $\mathcal{K}_i$ the allowed bit-width options for layer i . Known $d_{ik} \geq 0$ is a sensitivity-based loss proxy and $c_{ikh} \geq 0$ is consumption of resource $h \in \mathcal{H}$, whose budget is B_h . Binary z_{ik} selects option k :

$$\begin{aligned} \min_{z_{ik} \in \{0,1\}} \quad & \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} d_{ik} z_{ik} \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}_i} z_{ik} = 1 \quad (i \in \mathcal{I}), \\ & \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} c_{ikh} z_{ik} \leq B_h \quad (h \in \mathcal{H}). \end{aligned} \tag{15}$$

Interpretation. Exactly one precision is chosen per layer. Sensitivity scores approximate model perturbation, and resource rows enforce size, energy, or measured latency limits.

MP and structure. This is a multiple-choice multidimensional BIP/ILP and is NP-hard. It has strong separability except for the few budget rows, making Lagrangian relaxation attractive.

Solution approaches. ILP solvers, dynamic programming for a single discretized budget, dual pricing, and sensitivity-guided rounding are representative. Hardware measurements should replace nominal bit counts when latency is the actual constraint.

MP relationship and citation. Explicit formulation. Yao et al. [27] explicitly solve an integer linear program that balances Hessian-based perturbation against hardware-aware memory and latency constraints.

Limitations. Layerwise sensitivities neglect cross-layer interactions, latency is device- and kernel-specific, and accuracy after fine-tuning may not follow the static proxy.

5.10 Diversity-Aware Recommendation

Context. Recommendation-list construction chooses a fixed-size set whose members are individually relevant yet collectively nonmonotonous.

Inputs and decision variables. For a user, candidate item $i \in \mathcal{I}$ has relevance r_i , pairwise similarity $s_{ij} \geq 0$, and binary selection z_i . Optional linear feasibility data $\mathbf{C}, \mathbf{u}$ are known and locally defined. With trade-off $\lambda \geq 0$,

$$\begin{aligned} \max_{z \in \{0,1\}^{|\mathcal{I}|}} \quad & \sum_{i \in \mathcal{I}} r_i z_i - \lambda \sum_{i < j} s_{ij} z_i z_j \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} z_i = m, \quad \mathbf{C}z \leq \mathbf{u}. \end{aligned} \quad (16)$$

Interpretation. The first term rewards predicted preference; the second penalizes similar co-selected pairs. Cardinality fixes list length, and optional linear rows can represent inventory or policy feasibility.

MP and structure. Equation (16) is a BQP/MIQP-style model. Binary products can be linearized with auxiliary variables and McCormick inequalities, producing an MILP at quadratic size. Depending on the similarity matrix and sign convention, the continuous relaxation may be nonconcave for maximization.

Solution approaches. Exact MIQP/MILP is possible after candidate pruning; greedy marginal-gain selection, local search, and maximal-marginal-relevance-style reranking scale farther but do not automatically certify optimality.

MP relationship and citation. Natural MP reformulation. Zhang and Hurley [31] formalize the accuracy–diversity trade-off for recommendation lists and propose diversification methods. Equation (16) is a canonical binary quadratic rendering of that trade-off, not a claim about the exact solver used in the paper.

Limitations. Pairwise similarity is only one notion of diversity, the scalar λ hides user-specific trade-offs, and relevance predictions are affected by exposure and position not modeled here.

5.11 Globally Optimized k -Means or Minimum Sum-of-Squares Clustering

Context. Minimum sum-of-squares clustering jointly assigns observations to K clusters and estimates centroids, seeking a global solution rather than a local fixed point of alternating updates.

Inputs and decision variables. Let $\mathbf{a}_i \in \mathbb{R}^d$ be observed for $i \in \mathcal{I}$, and let $\mathcal{K} = \{1, \dots, K\}$. Binary z_{ik} assigns observation i to cluster k , continuous $\boldsymbol{\mu}_k \in \mathbb{R}^d$ is its centroid, and $t_{ik} \in \mathbb{R}_+$ is an assignment cost. Suppose all feasible centroids lie in known bounds that yield valid constants M_{ik} . An exact mixed-integer quadratic-constraint model is

$$\begin{aligned} \min_{z, \boldsymbol{\mu}, t} \quad & \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} t_{ik} \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}} z_{ik} = 1 & (i \in \mathcal{I}), \\ & \sum_{i \in \mathcal{I}} z_{ik} \geq 1 & (k \in \mathcal{K}), \\ & t_{ik} \geq \|\mathbf{a}_i - \boldsymbol{\mu}_k\|_2^2 - M_{ik}(1 - z_{ik}) & (i \in \mathcal{I}, k \in \mathcal{K}), \\ & z_{ik} \in \{0, 1\}, \quad t_{ik} \geq 0, \quad \boldsymbol{\mu}_k \in \mathbb{R}^d. \end{aligned} \quad (17)$$

Interpretation. Every observation joins exactly one nonempty cluster. If $z_{ik} = 1$, the epigraph constraint charges its squared distance to centroid k ; a valid M_{ik} deactivates the constraint otherwise.

MP and structure. Directly writing $\sum_{ik} z_{ik} \|a_i - \mu_k\|_2^2$ gives a nonconvex mixed-integer nonlinear program (MINLP), not a BQP. Equation (17) is a mixed-integer convex quadratic-constraint reformulation when bounds are valid. Other exact developments produce MIQP or MILP models through disjunctions, distance variables, and linearization. Cluster-label symmetry and weak big- M bounds are major obstacles.

Solution approaches. Spatial branch-and-bound, branch-and-cut, symmetry breaking, bound tightening, semidefinite or assignment relaxations, and incumbents from repeated local clustering are representative. Tailored propagation and cuts improve general-purpose MINLP solvers, but global certification remains difficult.

MP relationship and citation. Explicit formulation. Burgard et al. [4] develop MIP techniques for the minimum sum-of-squares problem, while Ágoston and Eisenberg-Nagy [1] give MILP formulations. These sources support the distinctions among MINLP, quadratic, and reformulated linear models.

Limitations. The squared Euclidean objective favors spherical clusters, K is fixed, outliers can dominate, and global methods remain limited to far smaller instances than local heuristics.

5.12 Video Frame or Keyframe Selection

Context. Long-video systems select a small set of query-relevant frames that covers distinct events and fits a multimodal model's visual-token budget.

Inputs and decision variables. Let $\mathcal{I}$ index frames in temporal order, r_i be query relevance, $s_{ij} \geq 0$ visual/semantic redundancy, m the frame limit, and $\mathcal{E}_\Delta$ pairs that are too close in time to co-select. Binary z_i selects a frame and $\lambda \geq 0$ controls redundancy:

$$\begin{aligned} \max_{z \in \{0,1\}^{|\mathcal{I}|}} \quad & \sum_{i \in \mathcal{I}} r_i z_i - \lambda \sum_{i < j} s_{ij} z_i z_j \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} z_i \leq m, \\ & z_i + z_j \leq 1 \quad ((i, j) \in \mathcal{E}_\Delta). \end{aligned} \tag{18}$$

Interpretation. The objective balances query relevance and content diversity. The first row limits multimodal compute; optional temporal-separation rows prevent near-duplicate adjacent frames.

MP and structure. This is an integer quadratic program/BQP. Pairwise terms make exact optimization NP-hard; an MILP linearization uses one auxiliary product variable per retained pair. Sparsifying the similarity graph reduces both memory and solve time.

Solution approaches. MIQP/MILP handles short candidate lists; greedy marginal-gain search, local swaps, temporal segmentation, and learned candidate pruning support long videos. A heuristic should be reported as such even when the objective is exact.

MP relationship and citation. Explicit formulation. Fang et al. [10] explicitly formulate query-relevant, diversity-aware keyframe choice as integer quadratic programming and use a customized greedy alternative for efficiency.

Limitations. Framewise relevance can miss actions requiring temporal context, pairwise redundancy does not ensure narrative continuity, and downstream generated narratives can introduce errors not represented by (18).

6 Conic Programming

Conic programs optimize a linear objective over linear constraints and membership in convex cones. Second-order-cone programs (SOCPs) capture norm epigraphs; semidefinite programs (SDPs) optimize over the positive-semidefinite cone. Conic duality supplies strong bounds, but generic SDP scaling is substantially less favorable than LP or SOCP scaling.

6.1 Group Lasso as SOCP

Context. Group Lasso selects or removes predefined blocks of coefficients together, matching settings in which predictors form meaningful factors.

Inputs and decision variables. Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ and $\mathbf{y} \in \mathbb{R}^n$ be observed, and let the feature partition be $\{\mathcal{J}_g : g \in \mathcal{G}\}$. Coefficients in group g form $\boldsymbol{\beta}_g \in \mathbb{R}^{|\mathcal{J}_g|}$, with known weight $\omega_g > 0$ and penalty $\lambda \geq 0$. The Group Lasso objective is

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \quad \frac{1}{2} \|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{g \in \mathcal{G}} \omega_g \|\boldsymbol{\beta}_g\|_2. \quad (19)$$

Define the rotated cone $\mathcal{Q}_r^3 = \{(a, b, c) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R} : 2ab \geq c^2\}$. With epigraph variables $u, v, t_g \geq 0$, an equivalent SOCP is

$$\begin{aligned} \min_{\boldsymbol{\beta}, u, v, t} \quad & v + \lambda \sum_{g \in \mathcal{G}} \omega_g t_g \\ \text{s.t.} \quad & \|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2 \leq u, \\ & \|\boldsymbol{\beta}_g\|_2 \leq t_g \quad (g \in \mathcal{G}), \\ & (v, 1, u) \in \mathcal{Q}_r^3. \end{aligned} \quad (20)$$

Interpretation. The first cone bounds residual norm, group cones bound coefficient norms, and the rotated cone enforces $v \geq u^2/2$. At optimum all relevant epigraph bounds are tight, recovering (19).

MP and structure. Equation (20) is a convex SOCP. Nondifferentiability at a zero group produces block sparsity without integrality. Strict convexity, and therefore coefficient uniqueness, depends on the design and loss.

Solution approaches. Interior-point SOCP solvers are effective at moderate scale. Block coordinate descent, proximal gradient, and screening exploit separability for larger designs; they solve the original composite objective without explicitly forming cones.

MP relationship and citation. Equivalent formulation. Yuan and Lin [28] introduce grouped variable selection through the Group Lasso. The explicit rotated-cone representation in (20) is mathematically equivalent but is not attributed as their implementation.

Limitations. Groups must be specified, the penalty can over-shrink large or correlated groups, and the basic model does not capture overlapping hierarchy unless additional variables are introduced.

6.2 Sparse PCA as SDP

Context. Sparse principal component analysis (PCA) seeks a direction explaining high variance while using few coordinates, improving interpretability at the cost of a combinatorial sparsity constraint.

Inputs and decision variables. Let $\Sigma \in \mathbb{R}^{d \times d}$ be a known sample covariance matrix with $\Sigma \succeq 0$. A loading $\mathbf{v} \in \mathbb{R}^d$ induces the lifted decision $\mathbf{X} = \mathbf{v}\mathbf{v}^\top$. For sparsity budget $\kappa > 0$, the rank-one

model and its convex relaxation are

$$\max_{\mathbf{X}} \quad \langle \Sigma, \mathbf{X} \rangle \quad \text{s.t.} \quad \text{tr}(\mathbf{X}) = 1, \|\mathbf{X}\|_1 \leq \kappa, \mathbf{X} \succeq 0, \text{rank}(\mathbf{X}) = 1, \quad (21)$$

$$\max_{\mathbf{X}} \quad \langle \Sigma, \mathbf{X} \rangle \quad \text{s.t.} \quad \text{tr}(\mathbf{X}) = 1, \|\mathbf{X}\|_1 \leq \kappa, \mathbf{X} \succeq 0, \quad (22)$$

where $\|\mathbf{X}\|_1 = \sum_{i=1}^d \sum_{j=1}^d |X_{ij}|$ is the entrywise ℓ_1 norm.

Interpretation. Trace normalization corresponds to $\|\mathbf{v}\|_2 = 1$ in the rank-one model; explained variance is $\mathbf{v}^\top \Sigma \mathbf{v} = \langle \Sigma, \mathbf{X} \rangle$. The entrywise budget promotes a loading supported on few coordinates. Dropping rank one enlarges the feasible set and gives an upper bound.

MP and structure. Equation (21) is nonconvex because of rank. Equation (22) is an SDP after linearizing absolute values. Its solution need not be rank one, so a loading recovered from its leading eigenvector is generally approximate.

Solution approaches. Interior-point SDP works at small to medium dimension; first-order spectral methods, smoothing, low-rank factorization, and screening scale farther. Randomized or leading-eigenvector rounding converts a higher-rank relaxation into a sparse loading but may lose the SDP bound's value.

MP relationship and citation. Explicit formulation. d'Aspremont et al. [5] directly derive an SDP relaxation for sparse PCA from a rank-constrained lifted formulation.

Limitations. The $d \times d$ lift has quadratic storage, relaxation solutions can have high rank, and a covariance estimate contaminated by noise can dominate any optimization guarantee.

7 Bilevel Programming

Bilevel programming nests one optimization problem inside another. With outer feasible set Λ , inner parameter set Θ , upper objective Φ , and lower objective Ψ , the normalized form is

$$\min_{\lambda \in \Lambda} \Phi(\lambda, \theta^*(\lambda)) \quad \text{s.t.} \quad \theta^*(\lambda) \in \arg \min_{\theta \in \Theta} \Psi(\theta, \lambda). \quad (23)$$

When the inner minimizer is not unique, optimistic and pessimistic conventions differ. Hypergradients can be obtained by unrolling, implicit differentiation, or approximation, but nonconvex inner learning means these procedures need not solve the global bilevel problem.

7.1 Hyperparameter Optimization

Context. Hyperparameter optimization selects regularization, data-processing, or training-control parameters for validation performance after the model has been trained with those choices.

Inputs and decision variables. Training and validation samples define losses $L_{\text{tr}}(\theta, \lambda)$ and $L_{\text{val}}(\theta, \lambda)$. The outer decision is $\lambda \in \Lambda$ and the learned inner decision is $\theta \in \Theta$:

$$\min_{\lambda \in \Lambda} L_{\text{val}}(\theta^*(\lambda), \lambda) \quad \text{s.t.} \quad \theta^*(\lambda) \in \arg \min_{\theta \in \Theta} L_{\text{tr}}(\theta, \lambda). \quad (24)$$

Interpretation. The inner problem trains a model for the proposed hyperparameters; the outer objective measures generalization on held-out data. Separating the data prevents the outer level from merely choosing settings that overfit the training loss.

MP and structure. This is bilevel programming. It is smooth only under strong differentiability and solution-regularity conditions; deep-network instances are nonconvex at both practical and theoretical levels.

Solution approaches. Unrolled differentiation backpropagates through a finite training trajectory; implicit differentiation solves a linear system involving the inner Hessian; truncated or first-order

hypergradients reduce memory. Black-box search is useful for discrete or nonsmooth λ but does not exploit the bilevel derivative.

MP relationship and citation. Explicit formulation. Franceschi et al. [11] explicitly unify gradient-based hyperparameter optimization and meta-learning through bilevel programming and analyze finite inner approximations.

Limitations. Validation reuse can overfit, inner training is rarely solved exactly, and hypergradient quality depends on optimizer dynamics, stopping rules, and Hessian approximations.

7.2 Neural Architecture Search

Context. Neural architecture search (NAS) chooses operations and connectivity using validation performance, while network weights are learned on training data.

Inputs and decision variables. Let λ collect architecture weights and let Λ be a product of simplices: for every edge e in a fixed search graph and operation $k \in \mathcal{K}_e$, $\lambda_{ek} \geq 0$ and $\sum_{k \in \mathcal{K}_e} \lambda_{ek} = 1$. Let θ be network weights. With architecture-dependent losses,

$$\begin{aligned} \min_{\lambda \in \Lambda} \quad & L_{\text{val}}(\theta^*(\lambda), \lambda) \\ \text{s.t.} \quad & \theta^*(\lambda) \in \arg \min_{\theta \in \Theta} L_{\text{tr}}(\theta, \lambda), \end{aligned} \tag{25}$$

$$\Lambda = \left\{ \lambda : \lambda_{ek} \geq 0, \sum_{k \in \mathcal{K}_e} \lambda_{ek} = 1 \text{ for every } e \right\}.$$

Interpretation. Soft architecture weights mix candidate operations so gradients can update architecture and model parameters. A discrete architecture is derived after search, typically by retaining the largest weight on each edge.

MP and structure. Equation (25) is a nonconvex bilevel program with a continuous relaxation of an underlying discrete design. The relaxed optimum and the discretized architecture need not agree.

Solution approaches. Alternating inner/outer gradient steps, one-step unrolling, implicit gradients, and weight sharing reduce search cost. Retraining the selected discrete architecture is necessary to evaluate the final design independently of shared weights.

MP relationship and citation. Explicit formulation. Liu et al. [15] explicitly formulate differentiable architecture search as a bilevel problem and introduce the continuous operation relaxation represented in (25).

Limitations. Weight sharing biases architecture comparisons, short inner optimization changes the intended objective, discretization creates a gap, and search results can be sensitive to random initialization and regularization.

7.3 Coreset Selection and Data Subsampling

Context. Coreset selection chooses a small weighted or probabilistic training subset whose trained model performs well on the full distribution or a validation set.

Inputs and decision variables. Let training samples $(\mathbf{a}_i, y_i)$ be indexed by $\mathcal{I}$. Outer decisions $w_i \in [0, 1]$ are selection probabilities (binary indicators are an exact-subset alternative), with

expected size at most m . For per-sample loss $\ell_i(\theta) = \ell(\theta; (\mathbf{a}_i, y_i))$, define

$$\begin{aligned} \min_{\mathbf{w} \in [0,1]^{|I|}} \quad & L_{\text{val}}(\theta^*(\mathbf{w})) \\ \text{s.t.} \quad & \theta^*(\mathbf{w}) \in \arg \min_{\theta \in \Theta} \frac{1}{\sum_i w_i} \sum_{i \in I} w_i \ell_i(\theta), \\ & \sum_{i \in I} w_i \leq m, \quad \sum_i w_i > 0. \end{aligned} \quad (26)$$

Interpretation. Selection weights define the data seen by inner training; outer validation evaluates the resulting model. The size row limits expected coreset mass. Probabilistic sampling turns a discrete outer search into stochastic continuous optimization.

MP and structure. The model is bilevel and generally nonconvex. Binary selection is combinatorial and nondifferentiable; probabilistic weights permit unbiased score-function gradients but can have high variance.

Solution approaches. Policy-gradient estimators, implicit or unrolled hypergradients for continuous weights, warm-started inner training, and class-balanced sampling are representative. Rounding or sampling produces the final subset.

MP relationship and citation. Explicit formulation. Zhou et al. [35] explicitly propose probabilistic bilevel coreset selection and an unbiased policy-gradient solver, including the size-controlled sampling interpretation.

Limitations. Repeated inner training is expensive, validation choice controls what the coreset preserves, probabilities do not directly guarantee one realized subset's quality, and nonconvex learning complicates reproducibility.

7.4 Autonomous Scientific Discovery and AI Scientist Systems

Context. An autonomous discovery system may optimize not only candidate hypotheses or designs, but also the scientific objective or computable evaluation criteria used to judge them.

Inputs and decision variables. Let outer decision $\lambda \in \Lambda$ encode a computable scientific criterion or its weights, let candidate design $\mathbf{x} \in \mathcal{X}$ include discrete hypotheses and continuous physical parameters, let $\Psi(\mathbf{x}, \lambda)$ be the criterion-driven search score, and let $\Phi(\lambda, \mathbf{x})$ be an external scientific evaluation obtained from simulation, experiment, or a broader goal. A normative bilevel abstraction is

$$\min_{\lambda \in \Lambda} \Phi(\lambda, \mathbf{x}^*(\lambda)) \quad \text{s.t.} \quad \mathbf{x}^*(\lambda) \in \arg \min_{\mathbf{x} \in \mathcal{X}} \Psi(\mathbf{x}, \lambda). \quad (27)$$

Interpretation. The inner loop searches for a candidate under the current scientific goal; the outer loop analyzes outcomes and revises the goal or scoring function. In simulation-grounded variants, language models may propose discrete structures while differentiable simulation tunes continuous parameters.

MP and structure. Equation (27) is a bilevel representation, but its components may be discrete, black-box, stochastic, and nonstationary. An agentic outer loop that writes new objectives is not automatically equivalent to conventional differentiable bilevel programming.

Solution approaches. Iterative propose–simulate–analyze loops, surrogate modeling, derivative-based inner simulation, discrete search, and human checkpoints can be combined. Solver convergence claims apply only to the mathematical subproblem actually solved, not to open-ended scientific validity.

MP relationship and citation. Equivalent formulation. Ma et al. [16] explicitly call their LLM-simulation Scientific Generative Agent a bilevel optimization framework. Du et al. [9] describe SAGA’s outer goal-evolving agents and inner optimizer as a bi-level architecture. Equation (27) faithfully normalizes the architecture, while avoiding a claim that SAGA is a smooth or globally solved bilevel program.

Limitations. Scientific objectives are incomplete proxies, simulators can be misspecified, experiments can be costly, and autonomous criterion revision creates auditability and safety challenges beyond optimization.

7.5 Diffusion-Model Hyperparameter and Noise-Schedule Optimization

Context. Diffusion systems choose a noise schedule or fine-tuning hyperparameters whose quality can be assessed only after training or running an inner generative process.

Inputs and decision variables. Let $\lambda \in \Lambda$ parameterize a feasible noise schedule, let θ be denoiser parameters, let $\tau \in \{1, \dots, T\}$ be a sampled diffusion time, and let $\epsilon \sim P_\epsilon$ be noise. A clean datum is $\mathbf{a} \sim P_{\text{data}}$, the schedule-generated noisy state is $\mathbf{x}_\tau(\mathbf{a}, \epsilon; \lambda)$, and the denoiser is $\epsilon_\theta(\cdot, \tau)$. For an outer generation-quality loss L_{gen} , the bilevel model is

$$\begin{aligned} \min_{\lambda \in \Lambda} \quad & L_{\text{gen}}(\theta^*(\lambda), \lambda) \\ \text{s.t.} \quad & \theta^*(\lambda) \in \arg \min_{\theta \in \Theta} \mathbb{E}_{\mathbf{a} \sim P_{\text{data}}, \tau, \epsilon \sim P_\epsilon} [\|\epsilon - \epsilon_\theta(\mathbf{x}_\tau(\mathbf{a}, \epsilon; \lambda), \tau)\|_2^2]. \end{aligned} \quad (28)$$

Interpretation. The inner level trains the denoiser for the proposed schedule; the outer level scores generated validation samples or a differentiable quality surrogate. In an inference-only fine-tuning regime, the inner map can instead be a sampling or adaptation solver.

MP and structure. This is stochastic generative bilevel optimization. The inner problem is nonconvex and its output is a distribution induced by iterative sampling, so standard finite-dimensional implicit differentiation can be computationally prohibitive.

Solution approaches. First-order hypergradient estimators, reparameterization of training dynamics, sample-efficient outer estimators, truncated unrolling, and schedule parameterization reduce cost. Each method optimizes an approximation determined by finite samples and finite inner work.

MP relationship and citation. Explicit formulation. Xiao et al. [24] explicitly formalize diffusion fine-tuning and noise-schedule training as generative bilevel optimization and derive first-order estimators for both regimes.

Limitations. Generation metrics are imperfect and costly, stochastic gradient variance is high, and a schedule optimized for one data distribution, sampler, or compute budget may not transfer.

8 Multi-Objective Optimization

Multi-objective optimization retains several objectives rather than collapsing them prematurely to one scalar. Dominance and Pareto stationarity replace a unique total order, while scalarization or common-descent subproblems provide computational steps.

8.1 Multi-Task Learning

Context. Multi-task learning shares representations across tasks whose losses can conflict. The decision is a common parameterization that is not improvable on one task without degrading at least one other task.

Inputs and decision variables. Let $\mathcal{T} = \{1, \dots, T\}$ index tasks, let $L_t(\theta)$ be differentiable task t 's loss, and let $\theta \in \Theta$ contain shared and task-specific model parameters. The vector problem is

$$\min_{\theta \in \Theta} (L_1(\theta), \dots, L_T(\theta)). \quad (29)$$

At a current θ , let $\mathbf{g}_t = \nabla_{\theta} L_t(\theta) \in \mathbb{R}^p$. Simplex weights α_t solve the common-descent QP

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^T} \quad & \frac{1}{2} \left\| \sum_{t=1}^T \alpha_t \mathbf{g}_t \right\|_2^2 \\ \text{s.t.} \quad & \alpha_t \geq 0 \ (t = 1, \dots, T), \quad \sum_{t=1}^T \alpha_t = 1. \end{aligned} \quad (30)$$

Interpretation. A parameter is Pareto optimal if no feasible parameter weakly improves every L_t and strictly improves one. The minimum-norm convex combination of task gradients gives a common descent direction when nonzero; zero indicates first-order Pareto stationarity under standard regularity.

MP and structure. Equation (29) is multi-objective and generally nonconvex for neural models. Equation (30) is a small convex QP in T variables even when p , the parameter dimension, is large.

Solution approaches. Gradient-based multiple-gradient descent, weighted scalarizations, preference-conditioned training Momma et al. [17], and approximate low-dimensional gradient representations are representative. Solving one QP produces one Pareto-stationary trajectory, not the entire Pareto front.

MP relationship and citation. Explicit formulation. Sener and Koltun [20] explicitly cast multi-task learning as multi-objective optimization and use a simplex-constrained minimum-norm subproblem, with a scalable upper-bound construction for deep networks.

Limitations. Pareto stationarity is not global Pareto optimality in nonconvex models, gradient scales influence trade-offs, and a technically Pareto solution may still be unacceptable on a critical task.

9 Inverse Optimization

Inverse optimization observes decisions and infers objective parameters under which those decisions are optimal or nearly optimal. Identifiability is central: many objectives can rationalize the same behavior.

9.1 Inverse Reinforcement Learning

Context. Inverse reinforcement learning (IRL) infers a reward from expert behavior rather than learning a policy from a supplied reward.

Inputs and decision variables. Consider a known finite discounted Markov decision process with states $\mathcal{S}$, actions $\mathcal{A}$, transition probabilities $P(s' | s, a)$, discount $\gamma \in [0, 1)$, and demonstrated deterministic policy $\pi_E(s)$. Known features $\phi(s, a) \in \mathbb{R}^d$ define reward $r_{\theta}(s, a) = \theta^{\top} \phi(s, a)$. Decisions are reward weights $\theta \in \mathbb{R}^d$, expert-policy values $V_s \in \mathbb{R}$, and slacks $\xi_{sa} \geq 0$. Define the continuation operator $(\mathcal{P}_a V)_s = \sum_{s' \in \mathcal{S}} P(s' | s, a) V_{s'}$. For regularization $C > 0$, a maximum-margin

inverse-optimality QP is

$$\begin{aligned}
 \min_{\theta, V, \xi} \quad & \frac{1}{2} \|\theta\|_2^2 + C \sum_{s \in \mathcal{S}} \sum_{a \neq \pi_E(s)} \xi_{sa} \\
 \text{s.t.} \quad & V_s = r_\theta(s, \pi_E(s)) + \gamma(\mathcal{P}_{\pi_E(s)} V)_s \quad (s \in \mathcal{S}), \\
 & V_s \geq r_\theta(s, a) + \gamma(\mathcal{P}_a V)_s + 1 - \xi_{sa} \quad (s \in \mathcal{S}, a \neq \pi_E(s)), \\
 & \xi_{sa} \geq 0 \quad (s \in \mathcal{S}, a \neq \pi_E(s)).
 \end{aligned} \tag{31}$$

Interpretation. The first row evaluates the demonstrated policy. The second asks its action to beat each alternative by unit margin, softened by ξ_{sa} . Regularization selects one explanation among otherwise scale-ambiguous rewards.

MP and structure. Equation (31) is a convex QP with linear Bellman inverse-optimality constraints. Alternative IRL formulations use feature-expectation separation or a bilevel condition that places the expert policy in an inner control argmax.

Solution approaches. QP/LP solvers apply for known finite dynamics. Constraint generation avoids enumerating all policies, while forward control oracles find violated optimality conditions. Approximate dynamics and large state spaces require sampling or function approximation.

MP relationship and citation. Explicit formulation. Dong et al. [6], Dong and Wang [7], Ng and Russell [18] formulate IRL through linear inequalities expressing policy optimality and develop algorithms that resolve reward degeneracy through margins and priors. Equation (31) is a normalized soft-margin Bellman version of that explicit inverse-optimization idea.

Limitations. Rewards are generally nonidentifiable up to transformations, demonstrations may be suboptimal, dynamics may be unknown, and inferred intent need not correspond to ethically acceptable goals.

10 Distributionally Robust Optimization

DRO protects against probability laws near an empirical distribution. For parameter $\theta \in \Theta$, random observation ξ , loss $\ell(\theta; \xi)$, empirical law $\hat{P}$, and ambiguity set $\mathcal{U}(\hat{P})$, the common structure is

$$\min_{\theta \in \Theta} \max_{Q \in \mathcal{U}(\hat{P})} \mathbb{E}_{\xi \sim Q} [\ell(\theta; \xi)]. \tag{32}$$

The ambiguity geometry determines which shifts are protected: reweighting under a divergence ball differs from transporting probability mass under a Wasserstein ball. Duality often replaces the inner distribution problem by a finite-dimensional regularized loss.

10.1 Contrastive Learning as DRO

Context. Contrastive learning pulls an anchor toward a positive view while separating it from negatives. Negative sampling can be biased, so robustness is taken over plausible negative distributions.

Inputs and decision variables. Let $(\mathbf{a}, \mathbf{a}^+)$ be an anchor–positive pair drawn from known empirical law $\hat{P}_+$, let $\hat{P}_-$ be the empirical negative law, and let encoder f_θ have parameters $\theta \in \Theta$. Define similarity $s_\theta(\mathbf{u}, \mathbf{v}) = \langle f_\theta(\mathbf{u}), f_\theta(\mathbf{v}) \rangle$, temperature $\eta > 0$, divergence D_φ , and radius $\rho \geq 0$. With $\mathcal{U}_\varphi(\hat{P}_-) = \{Q : D_\varphi(Q \| \hat{P}_-) \leq \rho\}$, a robust pairwise contrastive model is

$$\min_{\theta \in \Theta} \mathbb{E}_{(\mathbf{a}, \mathbf{a}^+) \sim \hat{P}_+} \left[\max_{Q \in \mathcal{U}_\varphi(\hat{P}_-)} \mathbb{E}_{\mathbf{a}^- \sim Q} \log \left(1 + \exp \frac{s_\theta(\mathbf{a}, \mathbf{a}^-) - s_\theta(\mathbf{a}, \mathbf{a}^+)}{\eta} \right) \right]. \tag{33}$$

Interpretation. The inner distribution upweights negatives that most challenge the current representation while remaining in a divergence neighborhood of observed negatives. Radius ρ controls conservatism, and temperature η has a dual role related to robustness regularization.

MP and structure. Equation (33) is DRO/min-max optimization. Convex duality may solve the distributional inner problem for fixed losses, but neural dependence on θ makes the outer problem nonconvex. Overly large ambiguity sets can emphasize false negatives or outliers.

Solution approaches. Divergence dualization, adversarial reweighting of minibatch negatives, stochastic gradient descent-ascent, and adjusted contrastive losses are representative. Finite negative batches approximate both the empirical law and its adversary.

MP relationship and citation. Explicit formulation. Wu et al. [23] explicitly interpret contrastive learning as DRO over the negative-sampling distribution and relate temperature to ambiguity-set control.

Limitations. The ambiguity set may not match semantic shift, adversarial emphasis can amplify mislabeled negatives, and finite-batch dual estimates can be biased or unstable.

10.2 Distributionally Robust DPO for LLM Alignment

Context. Direct preference optimization (DPO) aligns an LLM from preferred/rejected response pairs. DRO protects against demographic, linguistic, geographic, or temporal shifts in the preference distribution.

Inputs and decision variables. An observation is $\xi = (\mathbf{a}, y^+, y^-)$, containing prompt $\mathbf{a}$, preferred response y^+ , and rejected response y^- . Let $\hat{P}$ be their empirical law, π_θ the learned policy, π_{ref} a fixed reference policy, $\beta > 0$ a scale, and $\sigma(t) = (1 + e^{-t})^{-1}$. Define

$$\ell_{\text{DPO}}(\theta; \xi) = -\log \sigma \left(\beta \left[\log \frac{\pi_\theta(y^+ | \mathbf{a})}{\pi_{\text{ref}}(y^+ | \mathbf{a})} - \log \frac{\pi_\theta(y^- | \mathbf{a})}{\pi_{\text{ref}}(y^- | \mathbf{a})} \right] \right). \quad (34)$$

For radius $\rho \geq 0$ and either divergence/transport discrepancy D , let $\mathcal{U}(\hat{P}) = \{Q : D(Q, \hat{P}) \leq \rho\}$. Robust DPO is

$$\min_{\theta \in \Theta} \max_{Q \in \mathcal{U}(\hat{P})} \mathbb{E}_{\xi \sim Q} [\ell_{\text{DPO}}(\theta; \xi)]. \quad (35)$$

Interpretation. The inner distribution concentrates on preference examples or nearby transported examples with high DPO loss. The outer policy must improve preferred-response odds relative to the reference under this worst plausible law.

MP and structure. Equation (35) is DRO with a nonconvex neural outer problem. Kullback-Leibler (KL) ambiguity produces reweighting; Wasserstein ambiguity permits geometric movement according to a chosen ground cost. The ambiguity set, not only its radius, defines the protected shift.

Solution approaches. Dual approximations convert the inner problem into a regularized loss, followed by stochastic gradients. Alternating adversarial weights and policy steps, clipping, and minibatch approximations support scale but solve approximate minimax objectives.

MP relationship and citation. Explicit formulation. Xu et al. [26] explicitly develop Wasserstein DPO and KL-DPO, characterize sample complexity, and derive scalable approximations for their minimax losses.

Limitations. Preference labels can be systematically wrong, the transport cost may not reflect human similarity, conservatism can reduce average utility, and robustness to represented shifts is not universal alignment.

11 Submodular Optimization

For a finite ground set $\mathcal{V}$, the common model is

$$\max_{S \subseteq \mathcal{V}} F(S) \quad \text{s.t.} \quad S \in \mathcal{F}, \quad (36)$$

where $\mathcal{F}$ is a feasible family. A set function is normalized if $F(\emptyset) = 0$, monotone if $F(A) \leq F(B)$ for $A \subseteq B$, and submodular if $F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B)$ for all $A \subseteq B \subseteq \mathcal{V}$ and $i \notin B$. The last property is diminishing returns. Greedy selection attains $1 - 1/e$ of optimum for normalized monotone submodular maximization under a cardinality constraint; other constraints require different guarantees.

11.1 Data Subset Selection and Active Learning

Context. A learner selects a representative training subset or, after uncertainty filtering, a batch whose labels should be acquired. These are two modes of one approved data-selection application.

Inputs and decision variables. Let $\mathcal{V}$ index data, $s_{ij} \geq 0$ be similarity from item i to representative j , and m be the selection limit. For $S \subseteq \mathcal{V}$, define facility-location value $F(S) = \sum_{i \in \mathcal{V}} \max_{j \in S} s_{ij}$ with $\max_{j \in \emptyset} s_{ij} = 0$. The model is

$$\max_{S \subseteq \mathcal{V}} \sum_{i \in \mathcal{V}} \max_{j \in S} s_{ij} \quad \text{s.t.} \quad |S| \leq m. \quad (37)$$

Interpretation. Every data point is credited according to its most similar selected representative, so additional representatives have diminishing coverage benefit. In active learning, $\mathcal{V}$ may first be restricted to uncertain examples, after which (37) removes redundancy.

MP and structure. The facility-location function is normalized, monotone, and submodular for nonnegative similarities. Cardinality-constrained maximization is NP-hard, but standard greedy has a $1 - 1/e$ guarantee under exactly these assumptions.

Solution approaches. Lazy greedy, stochastic greedy, memoized marginal gains, and distributed partitioning scale selection. The guarantee concerns the chosen set function, not downstream test accuracy.

MP relationship and citation. Explicit formulation. Wei et al. [22] explicitly connect classifier data likelihoods to constrained submodular maximization and combine uncertainty filtering with submodular selection for active learning.

Limitations. Similarity determines what “representative” means, uncertainty estimates may be miscalibrated, and submodular coverage does not preserve rare behavior unless encoded in the ground set or objective.

11.2 Document Summarization

Context. Extractive summarization selects sentences that cover salient concepts and span distinct content while respecting a word budget.

Inputs and decision variables. Let $\mathcal{V}$ index candidate sentences, c_i be sentence length, B the word budget, and $\mathcal{G}$ concepts with weights $\omega_g \geq 0$ and coverage $a_{ig} \geq 0$. Let topic groups $\mathcal{V}_k$ partition sentences, let $r_i \geq 0$ measure salience, and let $\lambda \geq 0$. The selected set S solves

$$\begin{aligned} \max_{S \subseteq \mathcal{V}} \quad & F(S) = \sum_{g \in \mathcal{G}} \omega_g \min \left\{ 1, \sum_{i \in S} a_{ig} \right\} + \lambda \sum_k \sqrt{\sum_{i \in S \cap \mathcal{V}_k} r_i} \\ \text{s.t.} \quad & \sum_{i \in S} c_i \leq B. \end{aligned} \quad (38)$$

Interpretation. The capped coverage term rewards mentioning important concepts without repeatedly crediting them. The concave-over-modular topic term rewards distributing selected salience across topics. Sentence lengths enforce the summary budget.

MP and structure. With nonnegative data, F is normalized, monotone, and submodular. The length constraint is knapsack rather than pure cardinality; modified greedy/enumeration algorithms give constant-factor guarantees under appropriate assumptions.

Solution approaches. Lazy cost-benefit greedy, partial enumeration plus greedy, and streaming approximations are representative. Marginal gains can be cached by concept and topic.

MP relationship and citation. Explicit formulation. Lin and Bilmes [14] design monotone submodular coverage/diversity functions for document summarization and use budgeted maximization with approximation guarantees.

Limitations. Extracted sentences can be incoherent, concept weights may miss factual dependencies, word cost is only a length proxy, and redundancy beyond the chosen features can remain.

11.3 KV-Cache Eviction for LLM Inference

Context. During autoregressive LLM generation, the key–value (KV) cache grows with sequence length. An eviction policy must retain a fixed number of tokens using only information available so far.

Inputs and decision variables. At decoding step τ , let $S_{\tau-1}$ be the retained token indices and $G_\tau = S_{\tau-1} \cup \{\tau\}$ the candidates after the new token arrives. Let $o_{\tau i} \geq 0$ be token i 's accumulated attention-derived heavy-hitter score, let $h : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ be nondecreasing and concave, and define $F_\tau(S) = h(\sum_{i \in S} o_{\tau i})$. Let $R_\tau \subseteq G_\tau$ be a reserved recent window and B the cache budget. The one-step dynamic set problem is

$$\begin{aligned} S_\tau \in \arg \max_{S \subseteq G_\tau} F_\tau(S) \\ \text{s.t. } |S| = B, \quad |S \setminus S_{\tau-1}| \leq 1, \quad R_\tau \subseteq S. \end{aligned} \quad (39)$$

When the cache is not full, $|S| = B$ is replaced by $|S| \leq B$ and the new token is added.

Interpretation. The score favors historical heavy hitters, the reserve protects recent tokens before enough attention evidence accumulates, and the dynamic constraint allows at most the arriving token to enter. Operationally, one evaluates which single eviction from G_τ leaves the greatest score; this is not ordinary static top- k because scores and the feasible ground set evolve with generation.

MP and structure. Concave-over-modular F_τ is monotone submodular. The cited method develops a broader dynamic submodular framework and a greedy eviction guarantee under its assumptions. With the linear choice $h(t) = t$, the one-step score is modular, while the dynamic analysis and score updates remain essential.

Solution approaches. H2O accumulates observed attention, reserves recent entries, and performs low-cost greedy eviction. Exact enumeration of one removal is cheap per step; the systems challenge is obtaining and updating attention scores without offsetting memory savings.

MP relationship and citation. Explicit formulation. Zhang et al. [33] explicitly formulate KV-cache eviction as a dynamic submodular problem, motivate heavy hitters, and retain a balance of heavy and recent tokens. Equation (39) makes that evolving-set structure explicit rather than reducing the method to static top- k .

Limitations. Past attention may not predict future need, heads and layers can disagree about importance, score collection has overhead, and evicted information cannot be recovered without recomputation.

12 Min–Max and Saddle-Point Optimization

Min–max problems model strategic or adversarial interaction. Convex–concave games admit strong saddle theory, but neural games are usually nonconvex–nonconcave; alternating gradients can cycle or converge to points without equilibrium guarantees.

12.1 Generative Adversarial Networks

Context. A generative adversarial network (GAN) trains a generator to imitate data while a discriminator learns to distinguish generated from real samples.

Inputs and decision variables. Let P_{data} be the real-data law, P_ϵ a noise law, generator G_{θ_G} have parameters θ_G , and discriminator $D_{\theta_D} : \mathbb{R}^d \rightarrow (0, 1)$ have parameters θ_D . For real sample $\mathbf{a}$ and noise ϵ , the canonical game is

$$\min_{\theta_G} \max_{\theta_D} \left[\mathbb{E}_{\mathbf{a} \sim P_{\text{data}}} \log D_{\theta_D}(\mathbf{a}) + \mathbb{E}_{\epsilon \sim P_\epsilon} \log(1 - D_{\theta_D}(G_{\theta_G}(\epsilon))) \right]. \quad (40)$$

Interpretation. The discriminator maximizes correct log-likelihood on real and generated samples. The generator minimizes the same value by producing samples that the discriminator accepts. In the ideal unrestricted game, equilibrium occurs when the generated law matches the data law and the discriminator outputs one half.

MP and structure. Equation (40) is a min–max/saddle-point problem. With neural parameterizations it is nonconvex in θ_G and nonconcave in θ_D , so classical minimax interchange and global convergence conditions do not hold.

Solution approaches. Alternating stochastic gradients, multiple discriminator steps, extra-gradient updates, regularization, and alternative generator losses address numerical behavior. These are practical game-optimization methods, not exact saddle solvers.

MP relationship and citation. Explicit formulation. Goodfellow et al. [12] introduce the adversarial framework explicitly as the two-player minimax game in (40) and analyze its ideal equilibrium.

Limitations. Finite models and samples invalidate the ideal analysis, training can cycle or collapse modes, and discriminator loss may correlate poorly with perceptual quality.

13 Cross-Paradigm Comparison and Practical Considerations

Table 3 compares modeling consequences rather than declaring one paradigm universally preferable. A system can cross rows: prediction may be trained by bilevel or DRO methods and then consumed by an MILP; an SDP can bound a discrete model; a Lagrangian dual can decompose a BIP into per-request decisions.

13.1 From Predictions to Coefficients

Scores such as r_i , sensitivities, costs, and similarities are usually estimated. Optimization then treats them as data, so a perfect solver can still make a poor decision when coefficients are biased or shifted. Calibration, uncertainty sets, causal evaluation, and post-decision monitoring are therefore part of the optimization pipeline. The relationship label is equally important: a natural MP reformulation can be valuable for counterfactual analysis even if a latency-critical implementation uses sorting or greedy search.

13.2 Relaxations, Certificates, and Approximation

An LP relaxation of a BIP supplies a bound, not necessarily a deployable decision. Total unimodularity in (4) is exceptional because it closes the integrality gap under stated conditions. SDP

Table 3. Cross-paradigm comparison. Complexity statements are typical worst-case characterizations, not instance-level runtime promises.

Paradigm	Variables	Convexity	Typical complexity	Common solution methods	Scalability bottleneck	Representative ML/AI role
LP	Continuous; sometimes integral by structure	Convex	Polynomial-time algorithms; often very fast in practice	Simplex, interior point, network algorithms, decomposition	Rows/columns and communication	Sparse recovery; global association
QP	Continuous	Convex for PSD Hessian; otherwise nonconvex	Polynomial for convex QP; hard in general	KKT solves, active set, first order	Dense Hessians and conditioning	Local reconstruction weights
BIP/MIP	Binary plus optional continuous	Nonconvex feasible set	NP-hard in general	Branch-and-cut, relaxation, decomposition, heuristics	Search tree and weak bounds	Selection, assignment, routing, structured models
Conic	Continuous matrices/vectors	Convex for standard cones	Polynomial-time theory	Interior point, first-order splitting, spectral methods	SDP matrix storage/factorization	Group norms; lifted sparse components
Bilevel	Outer and inner, continuous or discrete	Usually nonconvex	Hard even for simple subclasses	Unrolling, implicit differentiation, nested search	Repeated inner solves and hyper-gradient memory	Model/configuration design
Multi-objective	Usually continuous parameters	Inherits objective geometry	Pareto set may be large	Scalarization, common-descent QP, continuation	Many tasks/objectives and gradient storage	Conflicting task losses
Inverse	Objective parameters plus forward variables	Convex only for selected models	Depends on forward problem and identifiability	KKT/Bellman constraints, constraint generation	Repeated forward-oracle calls	Recover latent preferences/rewards
DRO	Model plus adversarial distribution	Convex in selected loss/ambiguity pairs; neural outer nonconvex	Inner problem can be infinite-dimensional before duality	Dual reformulation, reweighting, stochastic descent-ascent	Adversarial sampling and ambiguity calibration	Robust representations/alignment
Submodular	Set-valued/discrete	Discrete diminishing returns	NP-hard maximization; approximable under structure	Greedy, lazy/stochastic greedy, local search	Marginal-gain evaluation and ground-set size	Coverage/diversity under budgets
Min-max	Two or more players	Favorable if convex-concave; neural games are not	Global neural equilibria generally intractable	Alternating gradients, extragradient, regularization	Instability and unequal player progress	Adversarial generation

relaxation in (22) similarly gives a bound, but recovering a rank-one loading can lose optimality. Submodular greedy guarantees compare against the optimum of the specified set function; they do not validate that function as a proxy for human utility. For nonconvex bilevel and min-max learning, a stationary point is weaker than a global certificate and should be described accordingly.

13.3 Decomposition and Scale

Many large models are separable except for a few budgets: notification decisions couple through quotas, quantization choices through hardware limits, and compute levels through a batch budget. Lagrangian multipliers turn shared constraints into prices and expose parallel subproblems. Column generation is useful when feasible actions are numerous but only a few enter the master problem.

Candidate generation before constrained reranking is indispensable in retrieval, but its recall ceiling must be distinguished from the optimizer's gap on the reduced candidate set.

13.4 Choosing a Formulation

Modeling should begin with the decision and guarantee required. Use network LP when incidence structure is genuine; do not add binaries merely because the underlying objects are discrete. Use MIP when logical feasibility and certificates matter at manageable scale. Use SOCP/SDP when norm or matrix geometry provides a useful convex representation or bound. Use bilevel models when one learned response is conditional on another decision, DRO when the uncertainty is distributional, submodularity when coverage exhibits diminishing returns, and min-max models when there are genuinely opposed players. Hybrid algorithms are normal, but the reported guarantee must correspond to the part actually solved.

14 Research Challenges and Opportunities

End-to-end uncertainty. Most formulations separate prediction from optimization. A central challenge is to propagate epistemic and distributional uncertainty in relevance, cost, and response models into decisions without making robust models unusably conservative.

Differentiation through discrete and approximate solvers. Learning coefficients end to end requires gradients through LP/MIP layers, greedy subset selection, or truncated inner training. Useful estimators must reconcile discontinuity, bias, memory, and solver tolerances while preserving feasibility.

Scalable certificates. Information-access instances can contain billions of candidate actions. Better decompositions, learned primal heuristics, safe screening, and distributed bounds could make certificates available beyond small offline models. A learned heuristic should accelerate search without being allowed to invalidate exact pruning.

Dynamic objectives and constraints. LLM inference, notification systems, and autonomous discovery operate in nonstationary environments. Static optimality is inadequate when actions alter future data, preferences, or scientific goals. Online and dynamic formulations need regret or stability statements aligned with the actual feedback loop.

Semantics of diversity and fairness. Pairwise similarity, category quotas, and coverage features are convenient proxies. Research should connect them to user-level outcomes, intersectional constraints, and causal exposure effects while retaining algorithms that scale to retrieval and recommendation workloads.

Auditability of agentic optimization. When agents revise objectives or compose tools, reproducibility requires versioned objectives, constraint provenance, solver status, and traces of approximations. This is a modeling opportunity: provenance and safety conditions can become first-class constraints rather than post hoc documentation.

Benchmarking formulation fidelity. Comparisons should separate coefficient error, candidate-generation loss, relaxation gap, optimization gap, and execution drift. Reporting only final task quality obscures whether the mathematical model, its learned inputs, or its solver is responsible for failure.

References

- [1] Kolos Cs. Ágoston and Marianna Eisenberg-Nagy. 2024. Mixed Integer Linear Programming Formulation for K-means Clustering Problem. *Central European Journal of Operations Research* 32, 1 (2024), 11–27. doi:10.1007/s10100-023-00881-1

- [2] Dimitris Bertsimas and Jack Dunn. 2017. Optimal Classification Trees. *Machine Learning* 106, 7 (2017), 1039–1082. doi:10.1007/s10994-017-5633-9
- [3] Dimitris Bertsimas, Angela King, and Rahul Mazumder. 2016. Best Subset Selection via a Modern Optimization Lens. *The Annals of Statistics* 44, 2 (2016), 813–852. doi:10.1214/15-AOS1388
- [4] Jan Pablo Burgard, Carina Moreira Costa, Christopher Hojny, Thomas Kleinert, and Martin Schmidt. 2023. Mixed-Integer Programming Techniques for the Minimum Sum-of-Squares Clustering Problem. *Journal of Global Optimization* 87, 1 (2023), 133–189. doi:10.1007/s10898-022-01267-4
- [5] Alexandre d’Aspremont, Laurent El Ghaoui, Michael I. Jordan, and Gert R. G. Lanckriet. 2004. A Direct Formulation for Sparse PCA Using Semidefinite Programming. In *Advances in Neural Information Processing Systems*, Vol. 17. 41–48. <https://proceedings.neurips.cc/paper/2628-a-direct-formulation-for-sparse-pca-using-semidefinite-programming>
- [6] Chaosheng Dong, Yiran Chen, and Bo Zeng. 2018. Generalized inverse optimization through online learning. *Advances in Neural Information Processing Systems* 31 (2018).
- [7] Chaosheng Dong and Yijia Wang. 2024. Towards generalized inverse reinforcement learning. *arXiv preprint arXiv:2402.07246* (2024).
- [8] David L. Donoho. 2006. Compressed Sensing. *IEEE Transactions on Information Theory* 52, 4 (2006), 1289–1306. doi:10.1109/TIT.2006.871582
- [9] Yuanqi Du, Botao Yu, Tianyu Liu, Tony Shen, Junwu Chen, Jan G. Rittig, Kunyang Sun, Yikun Zhang, Aarti Krishnan, Yu Zhang, Daniel Rosen, Rosali Pirone, Zhangde Song, Bo Zhou, Cassandra Masschelein, Yingze Wang, Haorui Wang, Haojun Jia, Chao Zhang, Hongyu Zhao, Martin Ester, Nir Hacohen, Teresa Head-Gordon, Carla P. Gomes, Huan Sun, Chenru Duan, Philippe Schwaller, and Wengong Jin. 2025. Accelerating Scientific Discovery with Autonomous Goal-evolving Agents. *arXiv preprint arXiv:2512.21782* (2025). arXiv:2512.21782 doi:10.48550/arXiv.2512.21782
- [10] Bo Fang, Yuxin Song, Haoyuan Sun, Qiangqiang Wu, Wenhao Wu, and Antoni B. Chan. 2026. Threading Keyframe with Narratives: MLLMs as Strong Long Video Comprehenders. In *The Fourteenth International Conference on Learning Representations*. https://proceedings.iclr.cc/paper_files/paper/2026/hash/675a31be2d051be53d57bcd52c6631f-Abstract-Conference.html
- [11] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazi, and Massimiliano Pontil. 2018. Bilevel Programming for Hyperparameter Optimization and Meta-Learning. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 1568–1577. <https://proceedings.mlr.press/v80/franceschi18a.html>
- [12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Vol. 27. 2672–2680. https://proceedings.neurips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html
- [13] Kohei Hirata, Daichi Amagata, Sumio Fujita, and Takahiro Hara. 2023. Categorical Diversity-Aware Inner Product Search. *IEEE Access* 11 (2023), 2586–2596. doi:10.1109/ACCESS.2023.3234072
- [14] Hui Lin and Jeff Bilmes. 2011. A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 510–520. <https://aclanthology.org/P11-1052/>
- [15] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. DARTS: Differentiable Architecture Search. In *The Seventh International Conference on Learning Representations*. <https://openreview.net/forum?id=S1eYHoC5FX>
- [16] Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B. Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. 2024. LLM and Simulation as Bilevel Optimizers: A New Paradigm to Advance Physical Scientific Discovery. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*. PMLR, 33940–33962. <https://proceedings.mlr.press/v235/ma24m.html>
- [17] Michinari Momma, Chaosheng Dong, and Jia Liu. 2022. A multi-objective/multi-task learning framework induced by pareto stationarity. In *International Conference on Machine Learning*. PMLR, 15895–15907.
- [18] Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann, 663–670. <https://ai.stanford.edu/~ang/papers/icml00-irl.pdf>
- [19] Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290, 5500 (2000), 2323–2326. doi:10.1126/science.290.5500.2323
- [20] Ozan Sener and Vladlen Koltun. 2018. Multi-Task Learning as Multi-Objective Optimization. In *Advances in Neural Information Processing Systems*, Vol. 31. <https://proceedings.neurips.cc/paper/2018/hash/432aca3a1e345e339f35a30c8f65edce-Abstract.html>
- [21] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Model Parameters. In *The Thirteenth International Conference on Learning Representations*. <https://openreview.net/forum?id=4FWaWZtd2n>

- [22] Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in Data Subset Selection and Active Learning. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*. PMLR, 1954–1963. <https://proceedings.mlr.press/v37/wei15.html>
- [23] Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Xiang Wang, and Xiangnan He. 2023. Understanding Contrastive Learning via Distributionally Robust Optimization. In *Advances in Neural Information Processing Systems*, Vol. 36. doi:10.52202/075280-1010
- [24] Quan Xiao, Hui Yuan, A F M Saif, Gaowen Liu, Ramana Rao Kompella, Mengdi Wang, and Tianyi Chen. 2025. A First-order Generative Bilevel Optimization Framework for Diffusion Models. In *Proceedings of the 42nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 267)*. PMLR, 68535–68558. <https://proceedings.mlr.press/v267/xiao25i.html>
- [25] Shijia Xu, Zhou Wu, Xiaolong Jia, Yu Wang, Kai Liu, and April Xiaowen Dong. 2026. Self-Correcting RAG: Enhancing Faithfulness via MMKP Context Selection and NLI-Guided MCTS. *arXiv preprint arXiv:2604.10734* (2026). arXiv:2604.10734 doi:10.48550/arXiv.2604.10734
- [26] Zaiyan Xu, Sushil Vemuri, Kishan Panaganti, Dileep Kalathil, Rahul Jain, and Deepak Ramachandran. 2025. Robust LLM Alignment via Distributionally Robust Direct Preference Optimization. In *Advances in Neural Information Processing Systems*, Vol. 38. doi:10.52202/085713-0912
- [27] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael W. Mahoney, and Kurt Keutzer. 2021. HAWQ-V3: Dyadic Neural Network Quantization. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 11875–11886. <https://proceedings.mlr.press/v139/yao21a.html>
- [28] Ming Yuan and Yi Lin. 2006. Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68, 1 (2006), 49–67. doi:10.1111/j.1467-9868.2005.00532.x
- [29] Michelle Yuan, Khushbu Pahwa, Shuaichen Chang, Mustafa Kaba, Jiarong Jiang, Xiaofei Ma, Yi Zhang, and Monica Sunkara. 2025. Automated Composition of Agents: A Knapsack Approach for Agentic Component Selection. In *Advances in Neural Information Processing Systems*, Vol. 38. https://proceedings.neurips.cc/paper_files/paper/2025/hash/39c51d0fe64c273413a4d358361b3daa-Abstract-Conference.html
- [30] Li Zhang, Yuan Li, and Ramakant Nevatia. 2008. Global Data Association for Multi-Object Tracking Using Network Flows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–8. doi:10.1109/CVPR.2008.4587584
- [31] Mi Zhang and Neil Hurley. 2008. Avoiding Monotony: Improving the Diversity of Recommendation Lists. In *Proceedings of the Second ACM Conference on Recommender Systems*. ACM, 123–130. doi:10.1145/1454008.1454030
- [32] Xingwen Zhang, Feng Qi, Zhigang Hua, and Shuang Yang. 2020. Solving Billion-Scale Knapsack Problems. In *Proceedings of The Web Conference 2020*. ACM, 3105–3111. doi:10.1145/3366423.3380084
- [33] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 36. doi:10.52202/075280-1506
- [34] Bo Zhao, Koichiro Narita, Burkay Orten, and John Egan. 2018. Notification Volume Control and Optimization System at Pinterest. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1012–1020. doi:10.1145/3219819.3219906
- [35] Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, Zonghao Chen, and Tong Zhang. 2022. Probabilistic Bilevel Coreset Selection. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 27287–27302. <https://proceedings.mlr.press/v162/zhou22h.html>
- [36] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. 2022. Mixture-of-Experts with Expert Choice Routing. In *Advances in Neural Information Processing Systems*, Vol. 35. doi:10.52202/068431-0515