IDEACOMPASS: MAKING EXPLORATION MEASURABLE IN AUTONOMOUS AI RESEARCH AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous machine learning (ML) research agents conduct research by proposing ideas, implementing them, and running experiments. Yet existing autoresearch systems lack an explicit quantitative account of the idea landscape they have explored: they do not directly measure how far apart past ideas are, in which directions the search has moved, or how broadly the search has explored the idea space. General-purpose text embeddings do not fill this gap. We find that they do not reliably capture the distance and relative direction between ML research ideas, placing a change in training duration closer to a learning-rate change than to another duration change. We introduce IDEACOMPASS, a quantitative map of ML ideas: an ML-aware embedding whose vector direction encodes what an idea changes and whose norm encodes how substantial the change is. On this map, an agent's research history becomes a measurable trajectory, from which IDEACOMPASS diagnoses whether stalled search is over-concentrated (collapse) or over-dispersed, and steers subsequent proposals toward diversification or consolidation accordingly. Across eight FML-bench-Lite tasks, IDEACOMPASS improves both Autoresearch and AdaptiveSearch, which become the top two of nine evaluated agents; Autoresearch's pairwise win rate rises from 0.52 to 0.71. The representation achieves 0.740 AUROC on direction discrimination and 0.72 Spearman correlation between norm and modification scale.

1 INTRODUCTION

Autonomous ML research agents conduct research through iterative experimentation: they propose an idea, implement it, evaluate the result, and use the outcome to decide what to try next (Lu et al., 2024; Yamada et al., 2025; Jiang et al., 2025; Toledo et al., 2026; Karpathy, 2026). Many agents retain previous ideas and experimental results as context, giving them access to what has already been tried. Yet these histories provide no quantitative account of the idea space the agent has explored. They do not directly reveal whether exploration has become concentrated or dispersed, or whether recent ideas make small or substantial changes, making these properties difficult to use systematically when deciding what to try next.

A natural way to make these properties measurable is to represent research ideas in a vector space whose geometry reflects ML-relevant relations among ideas. General-purpose text embeddings provide convenient representations for comparing research ideas (Reimers & Gurevych, 2019; Li et al., 2023) and have been used to organize idea exploration (Antoniades et al., 2026). However, their geometry is not explicitly trained around the distinctions that matter for ML research. We distinguish two properties of an ML idea: its *direction*, describing what it changes, and its *magnitude*, describing how substantial the change is. A useful representation should therefore align vector direction with idea direction and vector norm with idea magnitude. Yet general-purpose embeddings do not reliably reflect these ML-specific relations. Table 1 gives a simple example: ideas A and B describe similar changes to training duration in different words, whereas C changes the learning-rate schedule. Despite this, the generic embedding places A closer to C than to B, while the IdeaCompass embedding makes A–B the closest pair.

We introduce IdeaCompass, a framework for making an agent's exploration quantitatively measurable from its research history and using this measurement to guide future search. IdeaCompass maps each executed idea to an ML-aware embedding whose vector direction represents what the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: **Illustrative example of ML idea similarity.** Pairwise cosine distances between research ideas (lower is more similar). **Emb3L** denotes OpenAI's text-embedding-3-large (OpenAI, 2024), and **Ours** denotes the IdeaCompass embedding.

ID	Idea
A	Train for **200 epochs instead of 50**
B	**Quadruple the length** of the training run
C	Decay the **learning rate by 10×** at epochs 50 and 200

Pair	Relation	Emb3L	Ours
A–B	**Similar**	0.525	**0.208**
A–C	Different	0.449	0.526
B–C	Different	0.743	0.436

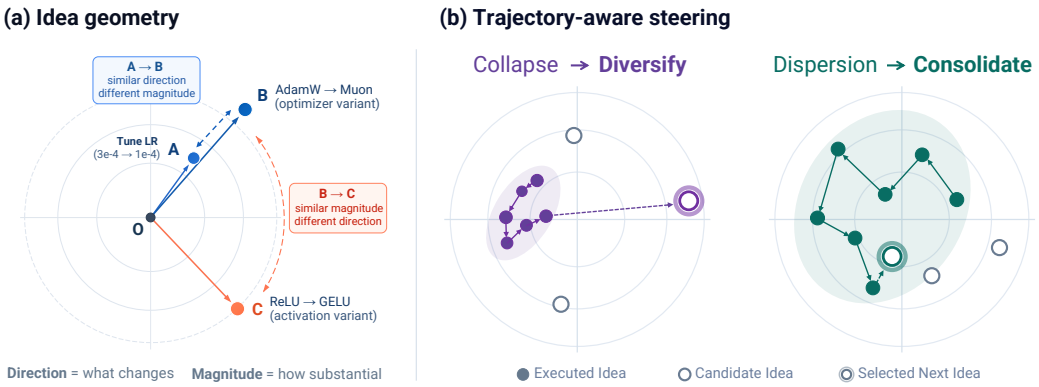


Figure 1: **Overview of IdeaCompass. (a)** An ML-aware idea space where embedding direction captures what an idea changes and norm captures the magnitude of the change. **(b)** Trajectory-aware steering that diversifies under Collapse and consolidates under Dispersion when progress stalls.

idea changes (*direction*) and whose norm represents how substantial that change is (*magnitude*), as illustrated in Figure 1a. We refer to the resulting sequence of embedded ideas as the agent's *research trajectory*. This trajectory makes the structure of past exploration measurable, enabling IdeaCompass to characterize the current search state and guide subsequent exploration.

To make this geometry reflect ML-relevant properties, we collect a dataset of ML research ideas and annotate each idea with its direction and magnitude. We train the encoder so that ideas pursuing similar ML modifications have aligned vector directions despite differences in wording, while ideas representing larger modifications have larger vector norms. The resulting embedding space gives vector direction and norm explicit ML-specific interpretations, enabling ideas to be compared along these two dimensions and the evolution of exploration to be characterized over time.

With exploration represented as a measurable trajectory, we use this quantitative account to distinguish two states of stalled exploration. Search collapse occurs when progress has stalled while recent exploration remains overly concentrated relative to the run's search history, whereas search dispersion occurs when progress has stalled while recent exploration remains broadly dispersed relative to that history. IdeaCompass responds differently to the two states: it promotes diversification under collapse and consolidation under dispersion (Figure 1b), using the learned geometry to guide subsequent exploration. Our analysis of search episodes shows that IdeaCompass changes subsequent trajectory dynamics, with a more consistent response under collapse and more host-dependent behavior under dispersion (Section 4.5).

On the representation test set, the learned geometry achieves an AUROC of 0.740 for direction discrimination and a Spearman correlation of 0.72 between vector norm and modification scope (Section 4.3). We evaluate IdeaCompass on eight FML-bench-Lite tasks with Autoresearch and AdaptiveSearch as host agents (Zou et al., 2026; Karpathy, 2026). IdeaCompass improves both hosts: Autoresearch's mean-of-three pairwise win rate increases from 0.516 to 0.714, while AdaptiveSearch's increases from 0.568 to 0.651. The two IdeaCompass-equipped agents rank in the top two under all three benchmark-level statistics in both repeat views. Ablations further show lower

performance when replacing our ML-aware encoder with a general-purpose embedding model, selecting candidates randomly, or using only a single candidate.

Our contributions are:

- **Quantitative geometry of ML research ideas that captures distances and relative directions among ideas.** We learn an embedding whose direction captures what an idea changes and whose norm reflects how substantial the change is, making an agent's research trajectory measurable.
- **Measurable trajectories for diagnosis and guidance.** We use the resulting research trajectory to characterize stalled exploration as collapse or dispersion and guide subsequent proposals toward diversification or consolidation accordingly.
- **Empirical validation of the learned geometry and its search utility.** We validate the learned geometry through direction and magnitude evaluations and demonstrate improved performance on eight FML-bench-Lite tasks with two hosts, supported by search-behavior analysis and ablations.

2 Related Work

Autonomous ML research agents iteratively propose, implement, and evaluate candidate improvements using different search strategies. The AI Scientist series automates a broad workflow spanning ideation, experimentation, and paper generation (Lu et al., 2024; Yamada et al., 2025). AIDE organizes iterative ML engineering as tree search (Jiang et al., 2025), AIRA studies greedy, MCTS, and evolutionary search policies (Toledo et al., 2026), OpenEvolve uses LLM-guided evolutionary search over populations of candidate programs (Sharma, 2025), and Autoresearch follows a lightweight sequential loop that repeatedly modifies a codebase and retains successful changes (Karpathy, 2026). Recent systems have also explored richer mechanisms for preserving experience over longer research processes. MARS introduces reflective memory across search branches (Chen et al., 2026), while AutoScientists, Iris, and ScienceFlow maintain shared experimental context, evolving task knowledge, or recoverable execution histories to support continued exploration and revision (Gao et al., 2026; Fu et al., 2026; Zhao et al., 2026).

Recent work has more directly addressed exploration in autonomous research. FML-bench studies research agents from the perspective of search dynamics, and AdaptiveSearch responds to stagnation by moving from focused single-path search to broader multi-branch exploration (Zou et al., 2026). This adapts the breadth of search when progress stalls. IdeaCompass takes a complementary approach: it explicitly represents the trajectory of executed research ideas and adapts exploration according to how that trajectory is distributed, allowing stalled search to call for either diversification or consolidation. Heuresis studies search strategies across quality, diversity, and novelty, including strategies that organize exploration using general-purpose embeddings of research ideas (Antoniades et al., 2026). Such embeddings provide semantic similarity, but do not explicitly distinguish what an idea changes from how substantial the change is. IdeaCompass explicitly supervises these properties, with direction capturing what an idea changes and magnitude capturing how substantial the change is. Broader work on scientific ideation has likewise considered diversity and novelty, including quality-diversity objectives and embedding-based novelty assessment (Gumma et al., 2026; Keya et al., 2026). Complementary empirical analyses report that, in the studied settings, AI-generated research ideas are more concentrated than corresponding human-authored research (Tang & Yang, 2026).

General-purpose text embeddings, including Sentence-BERT, GTE, OpenAI's text-embedding-3-large, and Gemini Embedding, provide broadly useful representations for semantic similarity and retrieval (Reimers & Gurevych, 2019; Li et al., 2023; OpenAI, 2024; Lee et al., 2025). However, their geometry is not designed to distinguish the direction and magnitude of ML research modifications. Scientific embeddings such as SPECTER and SciNCL show that representations can instead be specialized to task-relevant relations (Cohan et al., 2020; Ostendorff et al., 2022). IdeaCompass specializes this geometry for ML research actions, explicitly representing modification direction and magnitude for measuring and guiding research exploration.

Table 2: Direction categories and magnitude levels used to annotate ML research ideas.

Direction				Magnitude	
1	Data preprocessing	6	Optimizer / LR schedule	1	Hyperparameter tuning
2	Data augmentation	7	Regularization	2	Variant substitution
3	Feature engineering	8	Training procedure	3	Mechanism addition
4	Model architecture	9	Inference / post-processing	4	Component replacement
5	Loss function	10	Other	5	Paradigm shift

3 IDEACOMPASS

IdeaCompass augments a search-based ML research agent with an idea embedding model and a trajectory-aware controller. The embedding model represents each executed idea by its direction and magnitude; the sequence of embeddings forms a *research trajectory*. The controller combines this trajectory with validation progress to diagnose the search state and guide idea generation when progress stalls.

3.1 IDEA EMBEDDING MODEL

We distinguish an idea's *direction*, describing what it changes, from its *magnitude*, describing how substantial the change is.

Direction and magnitude supervision. Each idea receives one or more direction labels and a magnitude annotation following Table 2. Additional fine-grained annotations used during training are detailed in Appendix C. Direction supervision is represented by a multi-hot vector w, while magnitude supervision is represented by an ordinal label ℓ. We write $\ell_i \prec \ell_j$ when idea j represents a more substantial change than idea i. Training uses only this ordering, not the numerical values assigned to the ordinal levels.

Encoder and embedding geometry. The idea encoder E_θ mean-pools the token states of a pre-trained text encoder and applies a linear projection to obtain $z = E_\theta(x) \in \mathbb{R}^D$ for idea x. We train the encoder so that angular similarity reflects direction-label similarity and the Euclidean norm $\|z\|$ reflects magnitude. Embeddings are left unnormalized.

Training objective. For a batch of N annotated ideas, let $z_i = E_\theta(x_i)$. Direction is supervised by a soft-label contrastive loss, with weights $s_{ij} = w_i^\top w_j / (\|w_i\| \, \|w_j\|) \in [0, 1]$ measuring direction-label overlap. With temperature τ, the loss is

$$\mathcal{L}_{\text{dir}} = -\frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j \neq i} s_{ij} \log \frac{\exp(\cos(z_i, z_j)/\tau)}{\sum_{r \neq i} \exp(\cos(z_i, z_r)/\tau)}. \tag{1}$$

For magnitude, let $\Omega = \{(i, j) : \ell_i \prec \ell_j\}$ denote the ordered pairs in the batch. We use the ranking loss

$$\mathcal{L}_{\text{mag}} = \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \text{softplus}(\|z_i\| - \|z_j\| + \delta), \tag{2}$$

where $\delta \geq 0$ is a margin. The loss specifies neither target norms nor equal spacing between magnitude categories.

To provide direction predictions for the controller's trajectory summary (Section 3.2.2), we use a linear multi-label direction head with output $\hat{w} = \text{sigmoid}(A_h z + b_h)$, where A_h and b_h are learned parameters. It is supervised by $\mathcal{L}_{\text{dec}} = \frac{1}{N} \sum_{i=1}^{N} \text{BCE}(\hat{w}_i, w_i)$, where binary cross-entropy (BCE) is averaged over the direction outputs. The encoder and direction head are trained end to end with

$$\mathcal{L} = \alpha \, \mathcal{L}_{\text{dir}} + \beta \, \mathcal{L}_{\text{mag}} + \mu \, \mathcal{L}_{\text{dec}}, \tag{3}$$

where α, β, μ weight the three losses.

3.2 IDEACOMPASS CONTROLLER

The controller uses the learned idea space to guide an existing search-based ML research agent, the *host*. We index the search by the number of completed research steps $t = 0, 1, 2, \ldots$. For $t \geq 1$, x_t denotes the idea executed at step t; after t completed steps, the next proposal is x_{t+1}.

The controller diagnoses the search state from validation progress and the recent research trajectory, first at step t_0 and then every m completed steps. The verdict remains active until the next diagnosis.

3.2.1 DIAGNOSIS

At a diagnosis following step t, the controller assesses the window $\mathcal{W}_t = \{t - W + 1, \ldots, t\}$ of the W most recent executed ideas. Let I_t denote the best validation improvement over the task baseline achieved by step t, with larger values indicating better performance. Progress is stalled when the gain in the running best over the last W steps, $G_t = I_t - I_{t-W}$, satisfies $G_t \leq \varepsilon$, where ε is a tolerance for validation noise.

We measure the spread of the recent embeddings $z_{t-W+1}, \ldots, z_t$ as

$$S_t = \frac{1}{W} \sum_{i \in \mathcal{W}_t} \|z_i - \bar{z}_t\|^2. \tag{4}$$

Here, $\bar{z}_t = \frac{1}{W} \sum_{i \in \mathcal{W}_t} z_i$ is their centroid. Because embeddings are unnormalized, variation in both direction and magnitude contributes to S_t. To assess whether exploration is unusually concentrated or broad within the run, we use the empirical quantile rank $q_t \in [0, 1]$ of S_t among the window spreads available up to step t.

For quantile thresholds $\rho_{\text{low}} < \rho_{\text{high}}$, the controller assigns

$$v_t = \begin{cases} \text{Healthy}, & G_t > \varepsilon, \\ \text{Collapse}, & G_t \leq \varepsilon \ \wedge \ q_t \leq \rho_{\text{low}}, \\ \text{Dispersion}, & G_t \leq \varepsilon \ \wedge \ q_t \geq \rho_{\text{high}}, \\ \text{Uncertain}, & \text{otherwise}. \end{cases} \tag{5}$$

Collapse and Dispersion denote stalled progress with unusually concentrated and broad recent exploration, respectively. Healthy and Uncertain trigger no intervention.

3.2.2 INTERVENTION

Let v denote the active verdict after t completed steps. If $v \in \{\text{Collapse}, \text{Dispersion}\}$, IdeaCompass augments the host prompt with v, a histogram of predicted directions, and the mean and range of embedding norms.

The host's idea generator then produces K candidates $x^{(1)}, \ldots, x^{(K)}$ from the augmented prompt, with embeddings $z^{(k)} = E_\theta(x^{(k)})$. For each candidate, let $d_k = \min_{i \in \mathcal{W}_t} \{1 - \cos(z^{(k)}, z_i)\}$ be its directional distance to the nearest recent idea.

Collapse favors a direction less similar to those recently explored; Dispersion favors a direction more similar to at least one recent idea. For the active verdict v, the directional selection rule is

$$k^\star = \begin{cases} \arg\max\limits_{1 \leq k \leq K} d_k, & v = \text{Collapse}, \\ \arg\min\limits_{1 \leq k \leq K} d_k, & v = \text{Dispersion}. \end{cases} \tag{6}$$

Under Dispersion, if all candidates share the same nearest recent idea under directional distance, with embedding z°, we override directional selection and take $k^\star = \arg\max_{1 \leq k \leq K} \big| \|z^{(k)}\| - \|z^\circ\| \big|$. Here, norm difference serves as a heuristic for varying modification size relative to the shared neighbor.

The selected idea $x^{(k^\star)}$ is implemented and evaluated by the host. Algorithm 1 in Appendix A summarizes the full procedure.

4 EXPERIMENTS

4.1 SETUP

We evaluate on FML-bench-Lite (Zou et al., 2026), which spans eight tasks across five ML domains: Continual Learning (PyCIL), Data Efficiency (USB), Generalization (DomainBed-CM and DomainBed-OH), Robustness (ART and OpenOOD), and Privacy (Opacus and PrivacyMeter). We evaluate IdeaCompass with Autoresearch and AdaptiveSearch (Karpathy, 2026; Zou et al., 2026), comparing each equipped agent with its original host. We also include The AI Scientist v1 and v2, AIDE, AIRA, and OpenEvolve for broader comparison (Lu et al., 2024; Yamada et al., 2025; Jiang et al., 2025; Toledo et al., 2026; Sharma, 2025). All agent-level experiments use the FML-bench task environment with 100 validation steps per run and three repeats per condition. The main comparison uses GPT-5.4 (OpenAI, 2026a), and IdeaCompass does not add validation executions. Ablation and LLM-backbone studies use DomainBed-CM; the threshold and component ablations use GPT-5.4-mini (OpenAI, 2026b), while the backbone study compares GPT-5.4, Qwen3.8-Flash (Alibaba Cloud, 2026), and Gemini-3.5-Flash-Lite (Google DeepMind, 2026).

We report normalized improvement (NI) over the task baseline following FML-bench (Zou et al., 2026). Agents select code by validation performance during search, and final NI is computed on the held-out test evaluation of the selected code. We report mean-of-3 and best-of-3 results across the three repeats, with benchmark-level NI averaged over the eight tasks; standard deviations reported with mean-of-3 NI are computed across the three repeats. Pairwise win rate averages an agent's strict wins against the other agents on each task's native test metric, with ties ignored. For mean-of-3, win rates are computed within each repeat and then averaged; for best-of-3, each task-agent pair uses its best repeat before comparison. Mean rank averages each agent's within-task rank by NI, with lower values indicating better performance.

4.2 IMPLEMENTATION DETAILS

We collect 33,536 annotated research-idea examples from agent trajectories across 18 FML-bench tasks, split by run into 26,672/3,414/3,450 training, validation, and test examples. Claude Sonnet 5 (Anthropic, 2026) annotates each example with one or more of the ten direction categories and one of the five magnitude scopes in Table 2. Additional data and annotation details are provided in Appendix C. Hyperparameter tuning changes parameter settings only; variant substitution switches between standard alternatives such as SGD and Adam; mechanism addition introduces a known technique such as dropout or mixup; component replacement changes an entire component's method, such as its head structure or loss family; and paradigm shift changes the overall learning paradigm, such as supervised to self-supervised learning. We assign the highest scope realized in the code diff.

We fine-tune gte-modernbert-base (Tongyi Lab, Alibaba Group, 2025) end to end with mean pooling over token states and $D = 256$. We train for 20 epochs with AdamW using $\alpha = 0.5$, $\beta = 0.3$, $\mu = 0.3$, $\tau = 0.07$, and $\delta = 0.01$ times the batch-mean norm. For the controller, we use $W = 10$, $m = 5$, $t_0 = 20$, $K = 2$, $\varepsilon = 5 \times 10^{-4}$, $\rho_{\text{low}} = 0.20$, and $\rho_{\text{high}} = 0.80$.

4.3 IDEA REPRESENTATION EVALUATION

We evaluate whether the learned geometry reflects the direction and magnitude of ML modifications on the full representation test set. We compare IdeaCompass with the general-purpose text-embedding-3-large model (OpenAI, 2024) and the scientific-document representation model SPECTER2 (Singh et al., 2023), using the same idea texts for all three models.

Direction. We assess direction discrimination using pairs of ideas from the same task. Positive pairs share at least one direction category in Table 2; negative pairs have disjoint direction categories. This yields 146,032 positive and 89,583 negative pairs, scored by cosine similarity. Figure 2(a) shows that IdeaCompass separates the two groups more effectively, achieving an AUROC of 0.740, compared with 0.609 for text-embedding-3-large and 0.570 for SPECTER2. These results support alignment with the annotated direction categories.

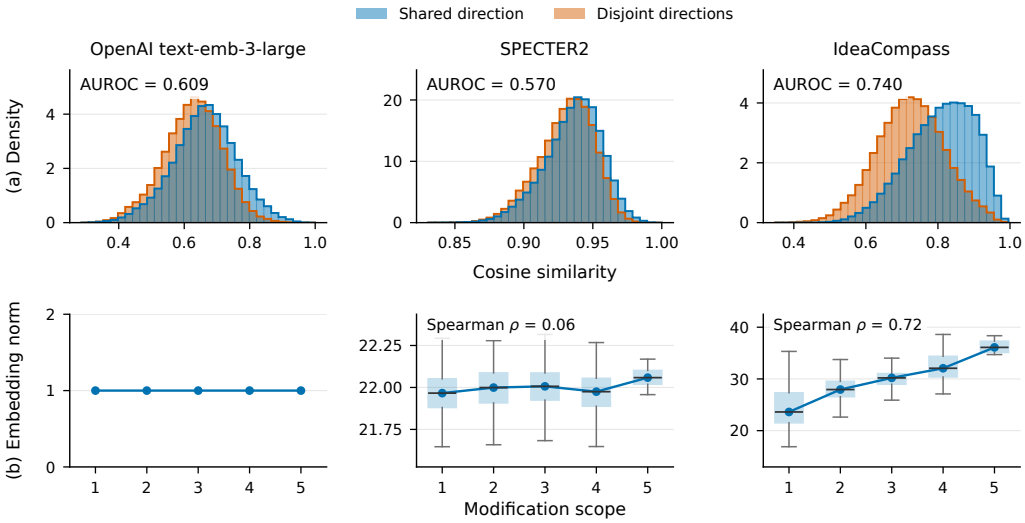


Figure 2: **Direction and magnitude evaluation.** **(a)** Cosine similarity for idea pairs with shared versus disjoint direction categories. **(b)** Embedding norm versus modification scope; Spearman ρ cannot be computed for OpenAI's text-embedding-3-large because all embeddings have unit norm.

Magnitude. We evaluate the association between embedding norms and the five-level modification scope on the same test-set ideas. Figure 2(b) shows that IdeaCompass assigns larger norms to broader modifications, with Spearman $\rho = 0.72$ across individual ideas. SPECTER2's norms exhibit almost no association with scope ($\rho = 0.06$), while text-embedding-3-large returns unit-norm vectors. Together, the two evaluations support the intended roles of vector direction and norm: distinguishing modification directions and representing the scale of the modification.

4.4 Evaluation on Research Agents

Table 3 shows that IdeaCompass improves both hosts in NI, pairwise win rate, and mean rank under both repeat views. In the mean-of-three view, Autoresearch's NI rises from 0.1598 to 0.1936, its pairwise win rate from 0.516 to 0.714, and its mean rank improves from 5.00 to 2.88. AdaptiveSearch's NI rises from 0.1811 to 0.1972, with corresponding improvements in win rate and rank. The two equipped hosts occupy the top two positions in every benchmark-level column, with IdeaCompass-equipped AdaptiveSearch achieving the highest mean-of-three NI and IdeaCompass-equipped Autoresearch the highest win rate and best mean rank.

The task-level results in Table 6 (Appendix B) show that these gains are distributed unevenly. Mean NI improves on seven of eight tasks for Autoresearch and six of eight for AdaptiveSearch, with the largest absolute gains on DomainBed-CM. Improvements also appear on tasks such as PyCIL, USB, and OpenOOD. However, Autoresearch's mean NI decreases on Opacus, while AdaptiveSearch's decreases on PrivacyMeter and ART. Thus, trajectory-aware guidance improves benchmark-level performance without uniformly improving every host–task combination.

4.5 Search Behavior after Collapse and Dispersion

We examine how search evolves after collapse and dispersion across the eight tasks and three repeats for each host, with and without IdeaCompass. To compare the two settings in the same geometry, we embed the histories from runs without IdeaCompass using the deployed IdeaCompass encoder and apply the same diagnostic rule. An episode consists of consecutive diagnoses of the same abnormal state. Table 4 reports episode-level means of duration in validation steps, spread change $\log(S_{t+10}/S_t)$, and the increase in validation best-so-far over ten steps from onset t. Positive spread change indicates expansion, while negative change indicates contraction.

Under Collapse, IdeaCompass shortens abnormal episodes, increases trajectory expansion, and improves subsequent gain for both hosts. For example, with Autoresearch, episode duration decreases

Table 3: **Results on FML-bench-Lite.** Green deltas below each IdeaCompass-equipped row show the change relative to the corresponding host.

Agent	Mean-of-3			Best-of-3		
	NI	Win Rate	Mean Rank	NI	Win Rate	Mean Rank
TASv1	0.0765±.0039	0.266	7.12	0.0945	0.250	7.12
TASv2	0.1602±.0288	0.495	5.00	0.1924	0.391	5.69
AIDE	0.1653±.0172	0.469	5.12	0.2047	0.469	5.06
AIRA	0.1143±.0633	0.234	7.38	0.1841	0.203	7.25
OpenEvolve	0.1426±.0076	0.516	4.88	0.1913	0.469	5.06
AdaptiveSearch	0.1811±.0222	0.568	4.12	0.2118	0.547	4.62
w/ IdeaCompass (Ours)	**0.1972**±.0390 (+0.0161)	0.651 (+0.083)	3.50 (-0.62)	0.2405 (+0.0287)	0.734 (+0.187)	2.94 (-1.68)
Autoresearch	0.1598±.0090	0.516	5.00	0.1835	0.578	4.38
w/ IdeaCompass (Ours)	0.1936±.0347 (+0.0338)	**0.714** (+0.198)	**2.88** (-2.12)	**0.2463** (+0.0628)	**0.766** (+0.188)	**2.88** (-1.50)

Table 4: Search behavior following Collapse and Dispersion, with and without IdeaCompass.

Host	State	W/ IdeaCompass?	Duration	Spread Change	Subsequent Gain ↑
AdaptiveSearch	Collapse	× *(original host)*	8.3	+0.17	0.0060
		✓	5.2	+0.36	**0.0161**
	Dispersion	× *(original host)*	6.7	-0.27	0.0041
		✓	7.3	-0.26	**0.0114**
Autoresearch	Collapse	× *(original host)*	6.7	+0.26	0.0020
		✓	5.2	+0.47	**0.0070**
	Dispersion	× *(original host)*	7.8	-0.11	0.0051
		✓	6.8	-0.40	**0.0069**

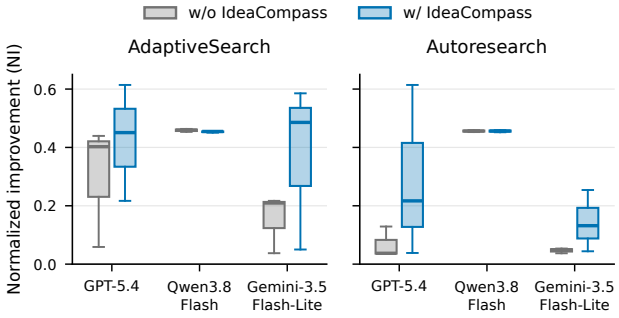
from 6.7 to 5.2 steps, spread change increases from $+0.26$ to $+0.47$, and subsequent gain rises from 0.0020 to 0.0070. Under Dispersion, the response is more host-dependent: AdaptiveSearch improves subsequent gain despite a slightly longer episode duration (6.7 to 7.3 steps), while its spread dynamics remain nearly unchanged (-0.27 vs. -0.26), whereas Autoresearch shows greater contraction (-0.11 to -0.40) together with shorter episodes. Overall, IdeaCompass produces a more consistent trajectory response under Collapse, with more host-dependent effects under Dispersion.

4.6 Ablation Studies

Quantile thresholds. On DomainBed-CM, we vary the spread thresholds while keeping the encoder, $K = 2$, and candidate-selection rule fixed; $(0.20, 0.80)$ is the default setting. We use three Autoresearch repeats with GPT-5.4-mini.

The default thresholds yield the highest mean NI (0.4983), compared with 0.1832 for the unmodified host. The $(0.10, 0.90)$ setting attains the best individual run (0.6141), but a lower mean and larger variation. Both $(0.05, 0.95)$ and $(0.30, 0.70)$ fall below the host's mean NI. This comparison shows that performance depends on the diagnostic thresholds; neither restricting intervention to the most extreme spread values nor extending it to a broader range consistently improves the outcome.

Embedding model, number of candidates, and selection rule. With thresholds fixed at the default $(0.20, 0.80)$, we ablate the embedding model, number of candidates, and candidate-selection rule. The default combination achieves a mean NI of 0.4983. Replacing the learned encoder with text-embedding-3-large reduces it to 0.0668, while random candidate selection and a single candidate yield 0.1752 and 0.1501, respectively. These results are consistent with a benefit from the learned representation and IdeaCompass candidate selection in this setting. Generating additional candidates alone does not reproduce the benefit of selecting them according to the diagnosed search state.



(a) LLM backbones

ρ_{low}	ρ_{high}	**Mean ± Std**	**Best**
–	–	0.1832 ± 0.0519	0.2167
0.05	0.95	0.0933 ± 0.1074	0.2170
0.10	0.90	0.2894 ± 0.2952	**0.6141**
0.20	0.80	**0.4983** ± 0.1588	0.5974
0.30	0.70	0.1021 ± 0.0990	0.2164

(b) Quantile thresholds

Figure 3: **Sensitivity analyses. (a)** Performance across LLM backbones with and without IdeaCompass. **(b)** Ablation of the spread quantile thresholds used to identify Collapse and Dispersion.

Table 5: **Component ablations.** We vary the embedding model, number of candidates K, and selection rule. *IdeaCompass* denotes our candidate-selection rule.

Embedding model	K	**Selection**	**Mean ± Std**	**Best**
Ours	2	IdeaCompass	**0.4983** ± 0.1588	**0.5974**
	2	random	0.1752 ± 0.0411	0.2168
	1	none	0.1501 ± 0.1153	0.2168
text-embedding-3-large	2	IdeaCompass	0.0668 ± 0.0387	0.0965

Sensitivity to LLM backbones. We compare GPT-5.4, Qwen3.8-Flash, and Gemini-3.5-Flash-Lite on DomainBed-CM with both hosts. IdeaCompass improves both hosts with GPT-5.4 and Gemini-3.5-Flash-Lite; for example, with GPT-5.4, mean NI increases from 0.067 to 0.290 for Autoresearch and from 0.300 to 0.427 for AdaptiveSearch. The effect is not uniform across backbones: with Qwen3.8-Flash, Autoresearch is unchanged at 0.456, while AdaptiveSearch decreases slightly from 0.459 to 0.453. Thus, the effect is observed with more than one backbone but depends on the host and model.

4.7 Token Usage and API Cost

IdeaCompass modestly increases token usage relative to its hosts. At 100 validation steps, estimated GPT-5.4 API cost (OpenAI, 2026a) is approximately $1.06\times$ that of Autoresearch and $1.08\times$ that of AdaptiveSearch under matched pricing and cache-rate assumptions. Mean input/output token counts are 2.15M/0.31M versus 1.89M/0.31M for Autoresearch, and 2.12M/0.31M versus 1.90M/0.29M for AdaptiveSearch, excluding encoder training and local embedding computation.

5 Conclusion

Autonomous ML research agents record what they have tried, but these histories alone provide little quantitative structure for understanding how exploration evolves. We introduced IdeaCompass, which maps ML research ideas into a geometry where vector direction captures what an idea changes and vector norm captures how substantial the change is, turning research histories into measurable trajectories. Direct evaluation shows that this geometry captures the intended direction and magnitude structure, while our agent-level experiments show that these measurements are actionable: when progress stalls, IdeaCompass distinguishes overly concentrated from broadly dispersed exploration and uses this diagnosis to guide subsequent search. Across eight FML-bench-Lite tasks, equipping Autoresearch and AdaptiveSearch with IdeaCompass improves their benchmark-level performance, with the two equipped agents ranking in the top two under all three benchmark-level statistics in both repeat views. More broadly, our results suggest that autonomous research need not rely on unstructured histories alone: explicitly measuring how ideas relate and how exploration evolves can provide a useful basis for deciding where to search next.

AI USE STATEMENT

In this work, we used generative AI tools to annotate research ideas for training the IdeaCompass embedding model. Additionally, we used generative AI tools for manuscript editing and literature search. AI-assisted manuscript content was reviewed by the authors; references identified with AI assistance were verified against the original sources, and reported results were checked against the underlying experimental records. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide detailed descriptions of the IdeaCompass representation and search procedure, benchmark and evaluation protocol, data construction and annotation process, model training configuration, and controller settings.

REFERENCES

Alibaba Cloud. Qwen3.8-Flash, 2026. https://www.alibabacloud.com/help/en/model-studio/qwen3-8-flash.

Anthropic. Introducing Claude Sonnet 5. https://www.anthropic.com/news/claude-sonnet-5, June 2026.

Antonis Antoniades, Deepak Nathani, Ritam Saha, Alfonso Amayuelas, Ivan Bercovich, Zhaotian Weng, Vignesh Baskaran, Kunal Bhatia, and William Yang Wang. Heuresis: Search strategies for autonomous ai research agents across quality, diversity and novelty. *arXiv preprint arXiv:2606.25198*, 2026.

Jiefeng Chen, Bhavana Dalvi Mishra, Jaehyun Nam, Rui Meng, Tomas Pfister, and Jinsung Yoon. Mars: Modular agent with reflective search for automated ai research. *arXiv preprint arXiv:2602.02660*, 2026.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. Specter: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2270–2282, 2020.

Shaokang Fu, Yulong Tao, Linbo Jin, Jiarong Zhao, Qiming Shi, Tianjun Pan, Haonan Li, Chengyu Wang, Jia Wu, and Chengfu Huo. Beyond solution-centric search: Adaptive inquiry and knowledge revision for autonomous ml engineering. *arXiv preprint arXiv:2608.02143*, 2026.

Shanghua Gao, Ada Fang, and Marinka Zitnik. Autoscientists: Self-organizing agent teams for long-running scientific experimentation. *arXiv preprint arXiv:2605.28655*, 2026.

Google DeepMind. Gemini 3.5 flash-lite, July 2026. https://deepmind.google/models/model-cards/gemini-3-5-flash-lite/.

Varun Gumma, Navonil Majumder, Soumitra Sinhahajari, and Soujanya Poria. Ideagent: Agentic quality-diversity search for research idea generation. *arXiv preprint arXiv:2607.22375*, 2026.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. Aide: Ai-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*, 2025.

Andrej Karpathy. autoresearch: Ai agents running research on single-gpu nanochat training automatically, 2026. URL https://github.com/karpathy/autoresearch.

Farhana Keya, Gollam Rabby, Sören Auer, Sahar Vahdati, Prasenjit Mitra, and Mohamad Yaser Jaradeh. Sci-idea: Context-aware scientific ideation using token and sentence embeddings. *Machine Learning*, 115(5):103, 2026.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, et al. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*, 2025.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

OpenAI. New embedding models and API updates. OpenAI, January 2024. URL https://openai.com/index/new-embedding-models-and-api-updates/. Introduces text-embedding-3-large.

OpenAI. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/, March 2026a.

OpenAI. Introducing GPT-5.4 mini and nano. https://openai.com/index/introducing-gpt-5-4-mini-and-nano/, March 2026b.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 11670–11688, 2022.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Asankhaya Sharma. Openevolve: an open-source evolutionary coding agent, 2025. URL https://github.com/algorithmicsuperintelligence/openevolve.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. Scirepeval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 5548–5566, 2023.

Yixuan Tang and Yi Yang. Ai research agents narrow scientific exploration. *arXiv preprint arXiv:2605.27905*, 2026.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Lupidi, et al. Ai research agents for machine learning: Search, exploration, and generalization in mle-bench. *Advances in Neural Information Processing Systems*, 38:35309–35348, 2026.

Tongyi Lab, Alibaba Group. gte-modernbert-base. https://huggingface.co/Alibaba-NLP/gte-modernbert-base, 2025.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Mingming Zhao, Jiqian Dong, Kangping Xu, Zadid Hasan, Chengrui Fan, Shan Jiang, Shuai Mao, Ting Lingya, Linyi Zou, Tailin Zhou, et al. Scienceflow: A long-horizon agent for ml research, scientific discovery and beyond. *arXiv preprint arXiv:2608.14354*, 2026.

Qiran Zou, Hou Hei Lam, Wenhao Zhao, Tingting Chen, Yiming Tang, Samson Yu, Yingtao Zhu, Srinivas Anumasa, Zufeng Zhang, Tianyi Zhang, et al. Fml-bench: A controlled study of ai research agent strategies from the perspective of search dynamics. *arXiv preprint arXiv:2605.17373*, 2026.

A IDEACOMPASS SEARCH PROCEDURE

Algorithm 1 summarizes how IdeaCompass is integrated into the host agent's search loop. The controller periodically diagnoses the search state from validation progress and the recent research trajectory. When the verdict is Collapse or Dispersion, IdeaCompass augments idea generation and selects among the resulting candidates; otherwise, the host proceeds with its original proposal. The selected idea is then implemented and evaluated through the host's standard workflow, without adding validation executions.

Algorithm 1 IdeaCompass inside the host's search loop

Require: Host agent; trained E_θ and direction head; step budget T;
$W, m, t_0, K, \varepsilon, \rho_{\text{low}}, \rho_{\text{high}}$.

$v \leftarrow$ none
for $t = 0, 1, \ldots, T-1$ **do**
  if $t \in \{t_0,\ t_0 + m,\ t_0 + 2m, \ldots\}$ **then** ▷ *Diagnosis*
    $v \leftarrow v_t$, computed from G_t and q_t via equation 5
  end if
  if $v \in \{\text{Collapse}, \text{Dispersion}\}$ **then** ▷ *Intervention*
    Augment the host prompt with v, the recent direction histogram, and norm statistics
    Generate candidates $\{x^{(k)}\}_{k=1}^{K}$; embed $z^{(k)} \leftarrow E_\theta(x^{(k)})$
    $d_k \leftarrow \min_{i \in \mathcal{W}_t} \left[1 - \cos(z^{(k)}, z_i)\right], \quad k = 1, \ldots, K$
    Select $k^\star$ via equation 6, with the norm-based override under Dispersion
    $x_{t+1} \leftarrow x^{(k^\star)}$
  else
    $x_{t+1} \leftarrow$ the host's own proposal
  end if
  The host implements and evaluates x_{t+1}; $z_{t+1} \leftarrow E_\theta(x_{t+1})$
end for

B TASK-LEVEL RESULTS

Table 6 provides the task-level results underlying the benchmark-level aggregates in Table 3. For each task, Mean and Best correspond to the mean-of-three and best-of-three views defined in Section 4.1; the final two rows report the corresponding averages across tasks.

Table 6: Task-level NI on FML-bench-Lite. AR, AS, and IC denote Autoresearch, AdaptiveSearch, and IdeaCompass, respectively.

Task	Statistic	TASv1	TASv2	AIDE	AR	AIRA	OpenEvolve	AS	AS+IC	AR+IC
PyCIL	Mean	0.0302	**0.0639**	0.0564	0.0261	0.0201	0.0370	0.0263	0.0416	0.0506
	Best	0.0597	0.0676	0.0617	0.0342	0.0275	0.0609	0.0402	0.0502	**0.0685**
USB	Mean	0.0398	0.0386	0.0293	0.0478	0.0188	**0.0671**	0.0443	0.0508	0.0557
	Best	0.0474	0.0479	0.0379	**0.0807**	0.0239	0.0794	0.0484	0.0543	0.0737
DomainBed-CM	Mean	0.0713	0.2448	0.2306	0.0672	0.1650	0.2425	0.3003	**0.4272**	0.2897
	Best	0.1007	0.3924	0.4379	0.1289	0.4080	0.4544	0.4395	**0.6141**	**0.6141**
DomainBed-OH	Mean	0.0065	**0.0137**	0.0108	0.0047	0.0116	0.0126	0.0084	0.0102	0.0065
	Best	0.0094	**0.0153**	**0.0153**	0.0082	0.0135	**0.0153**	0.0141	**0.0153**	0.0106
OpenOOD	Mean	0.0230	0.0392	0.0119	0.0571	0.0213	0.0064	0.0488	0.0532	**0.0595**
	Best	0.0610	0.0562	0.0347	**0.0698**	0.0236	0.0191	0.0548	0.0614	0.0646
Opacus	Mean	0.0000	0.0000	0.0059	0.0540	0.0000	0.0031	0.0461	**0.0546**	0.0382
	Best	0.0000	0.0000	0.0177	0.0704	0.0000	0.0051	0.0636	**0.0730**	0.0590
PrivacyMeter	Mean	0.0123	0.4871	0.5311	0.5734	0.3749	0.3130	0.5130	0.5015	**0.5822**
	Best	0.0239	0.5439	0.5556	0.5977	0.5745	0.4306	0.5492	0.5733	**0.5984**
ART	Mean	0.4289	0.3945	0.4465	0.4481	0.3028	0.4590	0.4614	0.4381	**0.4665**
	Best	0.4539	0.4160	0.4765	0.4783	0.4019	0.4657	**0.4846**	0.4825	0.4814
Average Mean		0.0765	0.1602	0.1653	0.1598	0.1143	0.1426	0.1811	**0.1972**	0.1936
Average Best		0.0945	0.1924	0.2047	0.1835	0.1841	0.1913	0.2118	0.2405	**0.2463**

C Additional Representation Data and Annotation Details

The representation dataset is collected from 18 FML-bench tasks, including the eight FML-bench-Lite tasks used for downstream evaluation; the representation split is therefore run-disjoint rather than task-disjoint. Representation training uses idea annotations only and does not use associated experimental outcomes, including validation/test performance, reward, or measured improvement.

Beyond the direction and magnitude annotations in Table 2, we use finer-grained auxiliary annotations. Modification type describes how a modification is performed and takes one of six values: add, remove, replace, combine, tune, or restructure. Novelty is labeled as either known or novel. Known refers to an established method, including one that is new only to the current repository, while novel refers to a non-standard mechanism proposed by the agent. Magnitude scope s and novelty are combined into a ten-level ordinal label. Known modifications use $2(s-1)+1$, and novel modifications use $2(s-1)+2$. The magnitude loss uses only the ordering of these levels.