Doc2LoRA Provides Decodable Representations of Scientific Ideas

Anonymous authors
Paper under double-blind review

Abstract

Representing scientific papers as points in a space lets us search for similar papers and inquire about how fields relate to one another and drive innovation. Beyond search, the vector space of papers invites generation: mixing papers through simple vector operations creates new points, mirroring combinatorial novelty, the recombination of existing ideas into new ones. However, a mixed point often represents an idea no paper has yet realized, with no papers nearby to identify the idea. We propose representing each paper by a LoRA adapter generated by the Doc-to-LoRA hypernetwork. Every point in the space, including mixtures, thus represents a large language model (LLM) open to questions and instructions in natural language. On papers from the American Physical Society (APS), we instruct the LLM at the average of each subfield to name the field in a few words and obtain labels closer to the official names than the labels of five baselines, as judged by word overlap and a panel of five LLM judges. We also ask the LLMs at points between two APS papers to write an abstract and obtain descriptions shifting from one paper to the other in step with the mixing weight. While Doc-to-LoRA is trained for generation, a small invertible transform makes the embeddings competitive for search, on par with SPECTER2 and EmbeddingGemma and close to SBERT. Because the transform is invertible, every point in the transformed space still maps back to an LLM. The embeddings thus serve both search and generation, enabling researchers to question the idea at any point in the space as a starting point for generating new ideas.

1 Introduction

Scientific papers form the foundational record of scientific knowledge and are collected in massive digital libraries (Lo et al., 2020; Priem et al., 2022). Understanding the landscape of science across these libraries is essential for identifying the structural patterns that drive innovation and inform science policy. Embedding models transform each paper into a vector and serve as the central tool for mapping this landscape, enabling literature recommendation, large-scale scientific analysis (Reimers & Gurevych, 2019; Cohan et al., 2020; Su et al., 2023; Beltagy et al., 2019; Ostendorff et al., 2022; Fortunato et al., 2018), and the prediction of future discoveries (Tshitoyan et al., 2019; Peng et al., 2021; Krenn & Zeilinger, 2020; Sourati & Evans, 2023). However, these embeddings serve search rather than generation and are rarely interpretable on their own. Because most embedding models are trained to preserve similarity structures for search, they compress text by discarding internal semantics, obscuring the actual ideas represented by individual papers (Weller et al., 2025; Kuratov et al., 2025).

Since the original text is not recoverable from standard vectors, interpreting the idea of a given point—especially an arithmetic point not paired with an actual paper—relies on heuristics. Labeling nearest documents (Grootendorst, 2023; Pham et al., 2024; Wang et al., 2023) only summarizes surrounding papers instead of describing the point itself. Vector-to-text inversion (Morris et al., 2023; Song & Raghunathan, 2020; Li et al., 2023a) is trained to reconstruct verbatim source text, offering no semantic guarantees for synthetic points constructed in latent space. Both strategies therefore only approximate the underlying idea. Alternative approaches build interpretability into vectors by mapping coordinates to sparse concepts (Koh et al., 2020; Bhalla et al., 2024), fixed answers (Benara et al., 2024), or sparse autoencoder features (O'Neill et al., 2024), but they restrict interpretation to predefined vocabularies.

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.

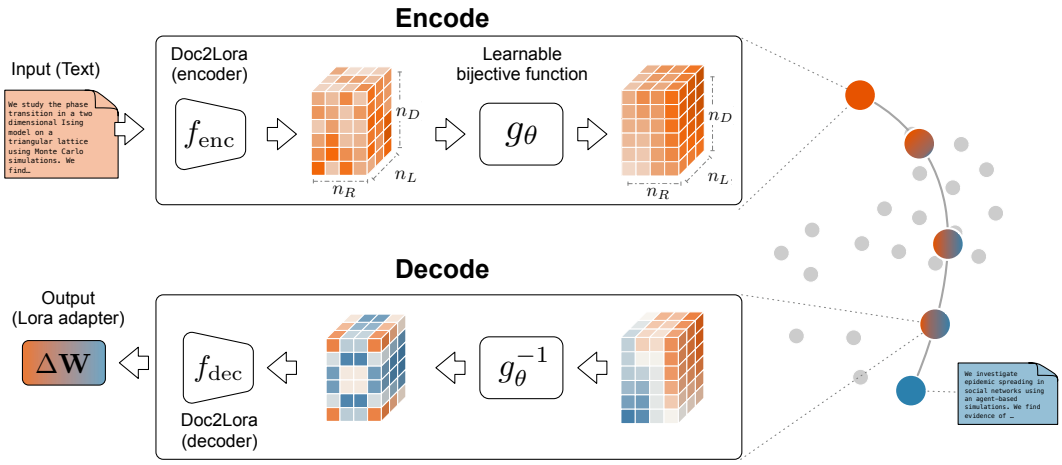

Figure 1: **Doc2LoRA pipeline. Encode:** The frozen D2L hypernetwork f_{enc} compresses a document into an embedding tensor $\mathcal{V}$ of shape (n_L, n_R, n_d), sitting between the MLP and LoRA heads (Section 2.2). A bijective map g_θ (Section 2.4) aligns this space for downstream search. **Decode:** Any point, including interpolations and averages, is mapped back via g_θ^{-1} and expanded by f_{dec} into a LoRA update $\Delta\mathbf{W}$, enabling natural language queries to the base model.

Heuristic labeling and vector decoders offer only passive descriptions of what a point is about. Generating new ideas requires more than description. New ideas often arise from recombining existing ideas, a pattern known as combinatorial novelty (Swanson, 1986; Uzzi et al., 2013; Foster et al., 2015; Youn et al., 2015; Shibayama et al., 2021). Exploring such combinations calls for active interrogation, asking how two ideas can be combined or what specific mechanism carries over from one idea to another. Such interrogation requires embeddings that we can query about ideas through dialogue.

We propose conversational embeddings, where every point in the embedding space represents a large language model (LLM) open to questions in natural language. Because every point in this space is an LLM, performing arithmetic operations—such as averaging a cluster of papers or interpolating between two ideas—yields another LLM that can be queried about the resulting concept.

We enable this paradigm by leveraging Doc-to-LoRA (Charakorn et al., 2026), which extracts a low-dimensional latent representation for each document. While LLMs are typically parameterized by billions of weights, D2L generates a compact adapter that captures only the document-specific changes while shared prior knowledge remains in the frozen base model and is not encoded in the embedding. We use the compact latent tensor produced by its encoder-decoder architecture as our document embedding. This representation isolates what makes that document unique, omitting redundant commonalities present across all documents. Although D2L was originally designed for context distillation (Snell et al., 2022), we show that its latent space has an intuitive geometric structure and is effective as an embedding.

While several works represent texts such as task instructions and documents as LoRA adapters (Chen et al., 2025; Su et al., 2025; Ye & Ren, 2021; Karimi Mahabadi et al., 2021; Phang et al., 2023; Charakorn et al., 2025), we select D2L (Charakorn et al., 2026) as our base hypernetwork because it provides both identifiability and scalability. The same document always maps to the same adapter, and large corpora can be embedded efficiently in a single forward pass. For example, Parametric RAG (Su et al., 2025), which fine-tunes a LoRA adapter per document, lacks identifiability due to random initialization, resulting in different adapters for the same input (Garipov et al., 2018; Frankle et al., 2020; Entezari et al., 2022; Ainsworth et al., 2023). GenerativeAdapter (Chen et al., 2025) achieves identifiability and fast, fixed-size adapter generation, but produces prohibitively large adapters, e.g., 268M parameters (537 MB in `bf16`) per document for Mistral-7B—making corpus-level indexing impractical. In contrast, D2L creates a compact, consistent adapter per document in one pass, compressing each to a 147k-dimensional latent tensor (and poolable to 18k for search; see Section 2.2), achieving both identifiability and scalability.

We test D2L embeddings on both generation and search. On generation, the LLM at the average of a cluster of papers names the field shared by the papers (Section 3.1). The LLMs at points between two papers describe work combining both (Section 3.2). On search, raw D2L embeddings fall behind dedicated text encoders. A small invertible transform trained on citation pairs brings D2L to the level of widely used text encoders while every point still represents an LLM (Section 3.3).

In summary, we turn the embedding space of scientific papers from a space for search into a space for both search and generation. Built on the public SakanaAI D2L checkpoint without retraining, conversational embeddings let researchers question any point in the space, including averages and mixtures of papers, in natural language. A linear map from each embedding to its adapter keeps averaging and mixing well defined (Section 2.2). A small invertible transform aligns the space for search while every point still represents an LLM (Section 2.4). Together, conversational embeddings let researchers move from finding existing papers to asking about ideas no paper has yet realized, a step toward exploring combinatorial novelty across the scientific literature.

2 Methods

2.1 Doc-to-LoRA hypernetwork. D2L is a hypernetwork that converts a document x to an LoRA adapter in one forward pass. Specifically, a document x is first sent to the base LLM, which produces a sequence of hidden states for all tokens in the document. These hidden states are then sent to a perceiver transformer (Jaegle et al., 2021), which compresses into a fixed-size tensor $\mathcal{U} \in \mathbb{R}^{n_L \times n_R \times n_d}$, where n_L is the number of layers in the base LLM, n_R is the number of rank slots, and n_d is the dimension of the hidden states. This fixed-size tensor $\mathcal{U}$ is then sent to an MLP followed by row-wise L_2 normalization, generating another fixed-size tensor $\mathcal{V} \in \mathbb{R}^{n_L \times n_R \times n_d}$. Finally, a linear head $f_{\text{dec},\ell}$ maps $\mathcal{V}_\ell$ to the LoRA adapter matrices, independently for each layer and with no weight sharing.

The resulting LoRA adapter consists of a pair of matrices $\mathbf{A}_\ell \in \mathbb{R}^{m_\ell \times n_R}$ and $\mathbf{B}_\ell \in \mathbb{R}^{n_R \times k_\ell}$ for each layer ℓ, where n_R is the rank of the LoRA adapter. These matrices $\mathbf{A}_\ell$ and $\mathbf{B}_\ell$ are then added to the linear layer $\mathbf{W}_\ell$ of the MLP in the base LLM during inference. Loading the resulting updates $\{\Delta\mathbf{W}_\ell\}$ into the frozen base model yields a document-conditioned model that answers as if x were in its context. We use the public SakanaAI checkpoint on Qwen3-4B (Qwen Team, 2025) in the main text and report other base models in Appendix G. The faithfulness of the decoded output depends on the hypernetwork, as evaluated by Charakorn et al. (2026). We take this as given and focus on the embedding space itself. Paraphrasing the prompt changes the wording but rarely the content of the responses (Appendix H).

2.2 D2L embedding. We leverage the post-MLP tensor $\mathcal{V}$ as the document embedding, as the following linear head $f_{\text{dec},\ell}$ maps it directly to the LoRA adapter matrices, preserving linear equivariance:

$$f_{\text{dec},\ell}\Big(\sum_i w_i \, \mathbf{V}_{i,\ell}\Big) = \sum_i w_i \, f_{\text{dec},\ell}(\mathbf{V}_{i,\ell}). \tag{1}$$

Averaging or interpolating embeddings is therefore equivalent to performing the same operation on the factor matrices $\mathbf{A}_\ell$ and $\mathbf{B}_\ell$. This equivalence does not apply to the adapter updates $\Delta\mathbf{W}_\ell = \mathbf{A}_\ell\mathbf{B}_\ell$ because the product is not linear in the embedding. This linearity is lost if the embedding is taken from the nonlinear pre-MLP matrices. Each $\mathcal{V}$ is a third-order tensor, with ℓ-th slice $\mathbf{V}_\ell \in \mathbb{R}^{n_R \times n_d}$ containing L_2-normalized, n_d=512-dimensional unit vectors per n_R=8 rank (Fig. 1).

When restricted to search, we can downsize the hidden tensor $\mathcal{V}$ while preserving much similarity information by taking the mean over the rank dimension n_R (Spearman ρ of .97–.98 between the full-tensor and pooled cosine similarities for Qwen3-4B, on 500 documents from each evaluation corpus (Appendix A)). In our paper, we use the mean-pooled tensor as embedding for similarity search discussed in Section 3.3, while use the full tensor for generation tasks. Unless otherwise noted, we use the full tensor.

2.3 Decoding documents and interpolants. We decode an embedding by expanding $\mathcal{V}$ into a LoRA adapter and querying the frozen base model with no document in the prompt. We illustrate with two short recipes, an Italian cacio e pepe and a Japanese kake udon, encoded with the Mistral-7B hypernetwork and decoded in full-rank mode (Appendix B). With the Italian adapter loaded, the query "What cheese is used?" returns "The cheese used in this recipe is Pecorino Romano." Next, we

interpolate the two embeddings as $\mathcal{V}(\alpha) = (1-\alpha)\mathcal{V}_A + \alpha\mathcal{V}_B$, where $\mathcal{V}_A$ and $\mathcal{V}_B$ are the embeddings of the two recipes. The endpoints $\alpha = 0$ and $\alpha = 1$ decode to cacio e pepe and kake udon, each with minor paraphrases. At $\alpha = 0.5$ the decode is "Kake Udon with Creamy Parmesan Sauce." The dish coats udon noodles in a melted-cheese sauce and seasons them with mirin and soy. Parmesan replaces Pecorino Romano. The decode also adds heavy cream, which is absent from both sources. We decode greedily, and the examples we report are single runs.

2.4 Adapting the geometry with a bijective transform. The public hypernetwork is trained for generation. We refine its tensor for search without retraining the base LLM and hypernetwork by adding a linear layer g_θ on top of the embedding, with a bijectivity constraint. More specifically, for each layer ℓ, we transform $\mathbf{V}_\ell \in \mathbb{R}^{n_R \times n_d}$ by

$$\mathbf{V}'_\ell = g_\theta(\mathbf{V}_\ell) = a_\ell \mathbf{V}_\ell \mathbf{C}, \tag{2}$$

where $a_\ell > 0$ controls the scaling of layer ℓ, and $\mathbf{C} \in \mathbb{R}^{n_d \times n_d}$ is an invertible linear matrix. To ensure invertibility, we factor $\mathbf{C}$ as $\mathbf{R}\mathrm{diag}(\sigma)$, where $\mathbf{R} \in \mathbb{R}^{n_d \times n_d}$ is a rotation matrix and $\mathrm{diag}(\sigma) \in \mathbb{R}^{n_d \times n_d}$ is a diagonal matrix with positive entries $\sigma_i > 0$ (Appendix C). This parametrization uses only $n_L + n_d^2$ parameters (262k for Qwen) and brings the embedding on par with dedicated text encoders on the similarity benchmarks we tested. We train g_θ with InfoNCE (van den Oord et al., 2018) on citation pairs (Appendix C) using the OpenAlex citation pairs (negatives in-batch, as in `SPECTER2`), with the base model frozen.

3 Results

3.1 Averaging embeddings names the field a cluster shares. A widespread practice for intepretating clustering results is either manual interpretation of the members, or running a keyword or LLM pipeline that needs careful tuning before it returns a usable name (Grootendorst, 2020; 2022; 2023; Murray et al., 2025). With D2L, labeling becomes as simple as averaging the vectors of a cluster and decoding the mean (Fig. 2c).

We validate cluster labeling on physics papers from the American Physical Society (APS), using the Physics and Astronomy Classification Scheme (PACS) (Appendix A). PACS is a hierarchical classification system for topics in physics, in which each topic is assigned a numeric code. For example, "03.65" is assigned to quantum mechanics. If the last two digits are omitted, it refers to the broader, higher-level topic above it. For example, "03" refers to quantum mechanics and quantum field theory, while "0" refers to general physics. Authors assign codes to their papers to organize them, with the first code indicating the paper's main topic. We group papers based on their first code at each of the three levels of the hierarchy (field, division, subdivision) and calculate the average of the vectors within each group. We decode these averages using the instruction "In 2 to 3 words, name the scientific field that all of these documents belong to" and compare the decoded labels with the official PACS names of the nodes (Appendix D).

We compare D2L to five baselines: `KeyLLM` (Grootendorst, 2023), which prompts an LLM with the original documents to extract keywords; `BERTopic` (Grootendorst, 2022), which represents each given cluster by its most distinctive words (c-TF-IDF); `vec2text` (Morris et al., 2023), which iteratively edits text to match a target embedding; `ICAE` (Ge et al., 2024), which encodes documents into memory slots and decodes them back; and `T2L` (Charakorn et al., 2025), which generates LoRA adapters from a frozen `GTE` encoder (Li et al., 2023c). We also compare against the base model's own hidden states (see Appendix E for details). All methods receive the same input: unmodified paper titles and abstracts.

We quantify the quality of the decoded labels by fuzzy token-set overlap, a score from 0 to 1 with no LLM in the loop (Fig. 2a). D2L leads with $.642 \pm .055$, ahead of `KeyLLM`, `BERTopic`, `vec2text`, `ICAE`, and `T2L`. As an additional validation, we use five LLM judges (`glm-5.3-flash`, `gemini-3.8-flash`, `gpt-5.6-luna`, `muse-spark-1.3`, and `mistral-medium-3.1`) to compare every pair of methods on each node (Fig. 2b). LLM judges consistently prefer D2L's labels over all baselines. A control rules out that any cluster draws a plausible field name: on clusters sampled at random across PACS chapters the judges accept no label at all, against 73.3% for the real nodes (Appendix D.2).

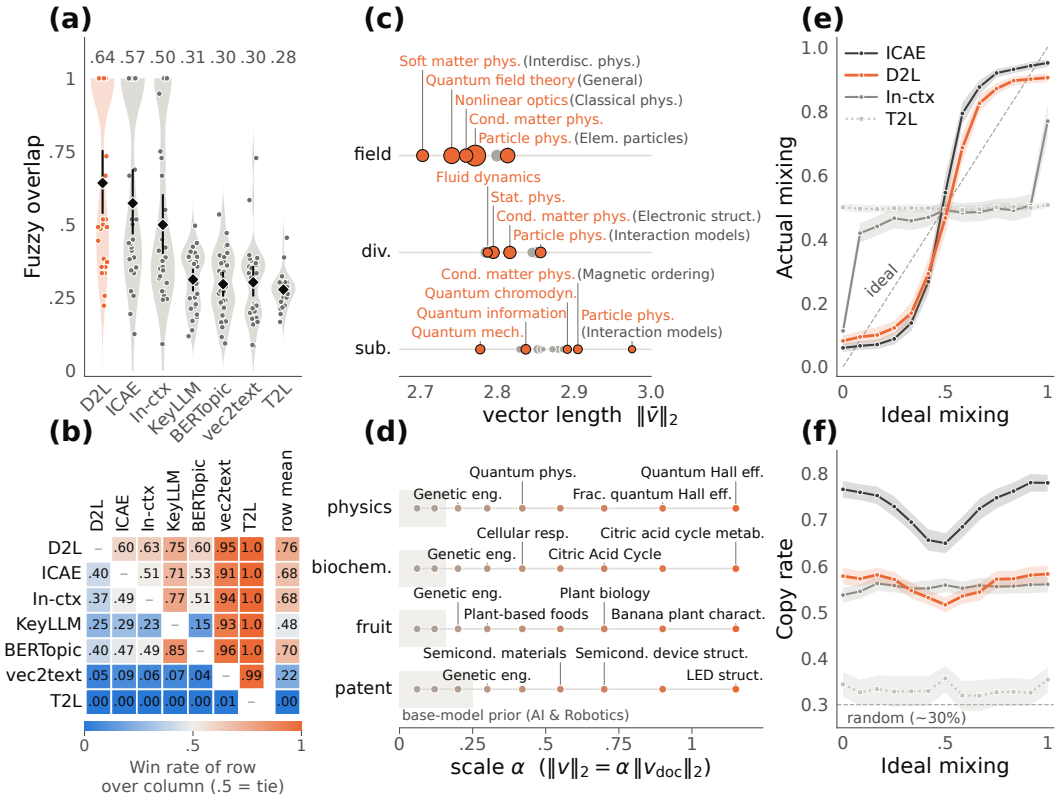

Figure 2: Cluster labeling and mixing with D2L. **(a)** Fuzzy token-set overlap between each decoded label and official 28 PACS labels. Each point is a node, with distribution represented by kernel density estimation (KDE). **(b)** Pairwise judge win rates, showing the fraction of PACS labels where the generated label from the row method is judged closer to the official label than that from the column. Color runs blue (row loses), grey (tie), to orange (row wins), row means at right. **(c)** Cluster centroids, decoded per PACS field/division/subdivision. Horizontal axis is vector length, circle area is the number of papers. Decoded labels (orange) and official PACS names (grey, if different) are shown. **(d)** Specificity as a function of vector length: longer vectors yield specific, shorter more general concepts. For each Wikipedia seed, lines show continuity of label, grey bands mark collapse to base-model prior. **(e)** Decoded abstract position (SBERT space) as a function of B weight, A→B, for 50 near pairs (papers from the same PACS subtopic). D2L and `ICAE` track the target, in-context stays near midpoint, `T2L` flat. **(f)** Verbatim copy rate by method, measured as the fraction of output tokens present in sources. `ICAE` is highest, and D2L and in-context prompting are comparable.

Upon manual inspection, D2L decodes cluster means into formal field names such as "particle physics" and "condensed matter physics" more frequently than all baselines (Fig. 2c). The baselines exhibit four main failure modes. `T2L` produces a handful of unrelated, fixed field names (e.g., "Social psychology") for all nodes; `vec2text` generates indecipherable pseudo-words (e.g., "inferromagnetic spin-diabetic scattering interactions"). `KeyLLM` returns overly specific keyword lists from few medoid papers, and `BERTopic` returns lists of topical words ("quark, mass, model, gauge, higgs, . . .") but never the name of the field. `ICAE` decodes plausible field names labels are too broad ("Materials science") or too specific ("Quantum many-body systems"). The differing performance of D2L and `T2L` shows that just creating a LoRA adapter is not enough. Decodable clustering depends on how the adapter is trained.

D2L avoids generating cluster names that are too broad or too narrow because it interprets the length of the mean vector as an indicator of abstraction level. When document vectors point in the same direction (i.e., are well aligned), their average keeps much of the original length. If they vary widely, averaging pulls the mean closer to zero, shortening the vector. For D2L, we observe that the average vector length grows as we move from general categories (top-level fields) to more detailed ones

Table 1: One near pair (two papers from the same PACS subtopic) decoded along the interpolation path at 25, 50, and 75 % weight on paper B, under the prompt quoted in the text. Paper A is "Statistical mechanics of learning from examples" (PhysRevA 45.6056) and paper B is "Good quantum error-correcting codes exist" (PhysRevA 54.1098). Each cell is the opening of the decoded abstract, cut at 42 words. Terms from paper A are in blue and terms from paper B in orange.

	Ideal mixing weight on paper B		
Method	25 %	50 %	75 %
D2L	This research investigates the learning of real-valued functions in neural networks using a statistical-mechanical framework, focusing on the generalization error of trained networks. The problem involves understanding how well a network can generalize from a finite training set to unseen data, particularly…	This research investigates the existence and properties of quantum error-correcting codes in the context of quantum computing, focusing on the asymptotic behavior of the quantum error-correcting code (QEC) as the number of qubits grows…	This research topic investigates the existence and properties of quantum error-correcting codes, which are essential for protecting quantum information from decoherence. The problem is to determine whether, given a quantum system of qubits, there exists a unitary encoding (quantum map) that transforms…
ICAE	This study investigates the learning of statistical mechanics from feedforward neural networks, focusing on the existence of stochastic, Gibbs-free energy-based models. The study assumes a temperature-dependent distribution of networks and considers both realizable and unrealizable rules…	This research explores the existence of quantum error-correcting codes in the context of quantum mechanics. It discusses the concept of a unitary code, which is a type of error-correcting code that can be defined using a unitary mapping…	This research explores the existence of quantum error-correcting codes (QECCs) in the context of quantum computing. It defines a unitary mapping called a quantum error-correcting code (QECC) that maps qubits to a subspace of the quantum state space…
in-context	This research investigates the statistical-mechanical behavior of quantum neural networks under the influence of both stochastic learning dynamics and quantum decoherence, blending the statistical mechanics of learning from examples (A:75%) with the principles of quantum error correction (B:25%)…	This research investigates the statistical-mechanical behavior of quantum neural networks under the influence of both stochastic learning dynamics and quantum decoherence, blending the statistical mechanics of learning from examples (A) with the principles of quantum error correction (B)…	This research investigates the emergence of quantum spin-glass phases in neural network architectures under the influence of quantum decoherence, blending statistical mechanics of learning (A) with the theory of quantum error correction (B)…

(fine-grained subdivisions; Fig. 2c). While the contraction of cluster means is a general geometric property of averaging diverse vectors, only D2L translates vector length into the appropriate level of abstraction when decoding names. By contrast, sweeping vector length in ICAE leaves decoded labels largely unchanged, and vec2text remains specific until collapsing at low scaling without producing broader field names (Appendix D.3). To test this mechanism, we embed and deliberately rescale Wikipedia introductions (Fig. 2d). With high scaling (α), we get detailed concepts such as "Citric acid cycle metabolism." As we reduce α, the decoded label shifts to something more general, like "Cellular respiration." For Nikola Tesla, shrinking the vector gives "AC power." For a light-emitting diode, it shifts to "Semiconductor device structure" (Appendix D.3). Extremely short vectors produce generic names from the base model, as if the LoRA adapter were absent. If the vector is too long, the output fails to decode because it drifts off the space of meaningful representations.

3.2 Midpoints decode to blends of both sources. D2L can decode text from any point in its embedding space, including points between documents. We test this by interpolating between APS paper pairs with controlled PACS taxonomy distance: 50 "near" pairs from the same subtopic, and 50 "far" pairs from different subtopics, both sampled randomly. For each pair, we decode at 13 points along the interpolation path between documents using D2L, in-context prompting, ICAE, and T2L, all under the prompt *"Write a detailed abstract (four to six sentences) describing this research topic: its problem, methods, and findings."* (Table 1; pair sampling, per-method interpolation, and metrics in Appendix F). In-context prompting has an advantage, as it directly receives both endpoint documents and mixing proportions, while D2L, ICAE, and T2L only receive source content indirectly via parameters, e.g., LoRA weights or memory slots.

We evaluate similarity to endpoints using SBERT-based scores. We also calculate the verbatim copy rate (Grusky et al., 2018), which is the fraction of output tokens shared with either source. While D2L and ICAE follow the true mixing weights along the interpolation path, in-context prompting remains near the midpoint, except at the endpoints (Fig. 2e). The copy rate separates ICAE from both alternatives. ICAE reuses source wording at .73 on either stratum, while D2L (.56) and in-context prompting (.52–.55) stay close to each other (Fig. 2f). T2L collapses onto a few input-independent texts. Its decoded position stays within .490–.506 across all 13 points, endpoints included, while

Table 2: Similarity benchmarks, absolute scores. Each row is one task on one field or dataset; *raw* is the D2L embedding and *adapted* applies g_θ, the linear transform trained on OpenAlex. Text encoders are ordered by mean rank. Cells are the bootstrap mean $\pm$ its standard deviation over 1,000 resamples. Bold and shaded indicates the top method per row; the second-best is underlined. For disambiguation, `SPECTER` replaces `SPECTER2`. Full protocol: Appendix G.

		D2L		text encoders				
Task	Field / dataset	raw	adapted	SBERT	GTE	Emb.Gemma	Instructor	SPECTER2
Next-paper (AUC)	Economics	.813±.001	.910±.001	**.937**±.001	.910±.001	.906±.001	.877±.001	.903±.001
	Psychology	.799±.001	.923±.001	**.937**±.001	.912±.001	.910±.001	.888±.001	.908±.001
	Physics	.880±.001	.958±.001	.956±.001	**.962**±.001	.947±.001	.908±.001	.941±.001
Topic (macro-F1)	Economics	.242±.015	.374±.022	**.418**±.022	.398±.023	.362±.021	.379±.021	.344±.021
	Psychology	.217±.010	.340±.020	**.366**±.023	.350±.019	.358±.020	.348±.026	.319±.020
	Physics	.590±.012	.590±.009	.573±.011	.587±.010	.588±.010	.574±.010	**.598**±.007
Collaboration (AUC)	Economics	.596±.011	.629±.012	**.655**±.011	.627±.012	.633±.012	.620±.012	.632±.011
	Psychology	.602±.012	.687±.010	**.691**±.011	.668±.011	.666±.011	.649±.011	.665±.011
	Physics	.792±.004	.795±.004	.832±.003	.839±.003	.828±.003	**.854**±.003	.826±.004
Author disamb. (B^3 F1)	zbMATH	.933±.001	.932±.001	**.944**±.001	.939±.001	.934±.002	.936±.001	.934±.001
	QIAN	.787±.005	.850±.004	.865±.004	**.866**±.004	.832±.004	.832±.004	.832±.004
	ArnetMiner	.678±.004	**.708**±.005	.698±.005	.698±.004	.706±.004	.679±.004	.669±.004
	PubMed	.693±.007	.815±.006	**.832**±.006	.792±.006	.822±.006	.801±.006	.739±.006
	KISTI	.708±.002	.767±.002	**.793**±.002	.785±.002	.771±.002	.754±.002	.746±.002

D2L moves from .081 to .904 (Fig. 2e). The endpoints are therefore not decodable, and interpolation is moot.

The separation between methods is robust to prompt wording. Decoding every interpolation point again under three paraphrases of the instruction leaves the tracking of D2L, `ICAE`, and in-context prompting unchanged (Appendix H).

3.3 Retrieval with D2L embeddings. D2L is designed to induce specific behaviors in LLMs, which may or may not contain topic-level information about each document. If D2L adapters do encode topics within their representations, and if we can extract and align this information geometrically, D2L has practical value as an embedding for retrieval tasks. To test the capacity of D2L embeddings for retrieval, we apply a simple linear transformation to the D2L embedding space, aiming to make topic similarity more directly accessible while preserving decodability, and then evaluate its effectiveness on standard retrieval benchmarks.

We adapt D2L embeddings with a lightweight linear transformation g_θ, containing 262k parameters for Qwen. We train g_θ on 42,332 uniformly-sampled OpenAlex citation pairs, representing less than 0.002% of the 2.56B possible pairs. This supervision rebalances the representation for similarity tasks without sacrificing adapter decodability. We compare D2L embeddings with and without g_θ against commonly used text encoders—`SPECTER2` (Singh et al., 2023), `Instructor` (Su et al., 2023), `EmbeddingGemma` (Gemma Team, 2025), `SBERT` (Reimers & Gurevych, 2019), `GTE` (Li et al., 2023c) on the following tasks (Appendix A for data details and Appendix G for the full benchmark protocol).

With g_θ, D2L sits inside the band of the text encoders on all four retrieval tasks. It performs better than `Instructor` and `SPECTER2`, is comparable to `EmbeddingGemma` and `GTE`, and follows `SBERT`, which leads on 9 of 14 benchmarks. The average ranks are: `SBERT` 2.0, `GTE` 2.9, D2L+g_θ 3.4, `EmbeddingGemma` 3.5, `Instructor` 4.7, `SPECTER2` 5.0, and raw D2L 6.5. All text encoders use default settings without task-specific tuning. We do not claim D2L outperforms current encoders, but it retains similarity information comparable to standard methods. Training the same g_θ in each baseline space improves those encoders marginally or degrades them, with the largest average gain over the 14 benchmarks being +.009 and `SBERT` losing .007. The transform therefore recovers geometry that the D2L space holds without exposing it to cosine similarity, and it gives the baselines no sizeable advantage (Appendix I).

These findings highlight an asymmetry between generation-trained and similarity-trained representations. A strong retrieval encoder does not inherently yield a decodable adapter. `T2L` reads `GTE`, yet it fails to decode documents or support idea arithmetic (Section 3.1, Section 3.2). Conversely,

generation-trained representations can implicitly encode topic geometry even when it is not directly exposed as cosine similarity. Applying g_θ brings this geometry to the surface in D2L, achieving competitive retrieval performance without losing decodability, while offering no meaningful gain to dedicated text encoders.

4 Related work

The landscape of document representation has evolved from static text encoders (Reimers & Gurevych, 2019; Su et al., 2023; Cohan et al., 2020; Singh et al., 2023), which shape papers into geometric spaces where points can be compared, but meaningful text recovery from those points remains limited. Traditional approaches rely on dedicated decoders—either trained a posteriori, as in `vec2text` (Morris et al., 2023), which constrains outputs to short, often incoherent rephrasings for points off the training manifold (Kuratov et al., 2025), or learned jointly via variational autoencoders (Bowman et al., 2016; Li et al., 2020; Gómez-Bombarelli et al., 2018). The latter enable decoding from any point but produce only brief texts and lack the ability to follow instructions, fundamentally constraining interpretability. Our approach with D2L differs: by decoding through a frozen LLM, the embedding gains not only fluency but also instruction-following, making every point in the space *conversationally* interpretable.

Parallel research inserts document content directly into generators. With RAG (Lewis et al., 2020) and compressed prompt engineering (Jiang et al., 2023; Li et al., 2023b; Xu et al., 2024), text remains manipulable only as sequences in the prompt and cannot be averaged or meaningfully composed in vector space. Soft compression—through learned input tokens (Lester et al., 2021; Li & Liang, 2021; Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024; Li et al., 2025; Rau et al., 2025; Cheng et al., 2024) or LoRA adapters (Chen et al., 2025; Su et al., 2025; Charakorn et al., 2026)—enables vector arithmetic, interpolation, and comparison, and the geometry of these learned vectors tracks meaningful task transfer (Vu et al., 2022). Another line of research proposes methods to read the hidden states of a model as text without any learned compression. An example method is Patchscopes (Ghandeharioun et al., 2024), which overwrites a hidden state at a target position and prompts the model to decode it. SelfIE (Chen et al., 2024) interprets individual token embeddings through the same mechanism, and LatentQA (Pan et al., 2026) trains a small encoder to answer natural-language questions about activations. These hidden-state methods require no adapter and no training on the document. Therefore, we consider them as an important reference for whether a learned LoRA representation adds value over what the model already knows (Appendix E). Building on this, our work uses the generated LoRA adapter as the embedding itself, demonstrating that its structure is smooth, expressive, and admits both semantic retrieval and idea synthesis (Section 3.3, Section 3.2).

At a deeper level, collections of model weights—left by fine-tuning tasks—have long served as implicit representations, supporting operations such as averaging for accuracy (Wortsman et al., 2022), constructing skill vectors (Ilharco et al., 2023), and interpolation through training time (Nylund et al., 2024). Fine-tuned models cluster in connected, low-loss regions (Gueta et al., 2023) so that modular adapters can be merged (Chronopoulou et al., 2023; Huang et al., 2024; Yadav et al., 2023). However, these weight-based representations are fundamentally unidentifiable: optimizer randomness erases any canonical correspondence (Garipov et al., 2018; Frankle et al., 2020; Entezari et al., 2022; Ainsworth et al., 2023), and the coordinates are not stable or comparable across models. Hypernetworks (Ha et al., 2017) resolve this by generating adapters in a consistent reference frame. Prior works focused on conditioning these on descriptions or tasks (Ye & Ren, 2021; Karimi Mahabadi et al., 2021; Phang et al., 2023; Charakorn et al., 2025). Among them, `T2L` (Charakorn et al., 2025) is the closest comparison, as it produces a LoRA adapter from text. Unlike D2L, `T2L` does not reconstruct the original document but instead produces task-specific LoRA adapters. It embeds text via a frozen `GTE` encoder (Li et al., 2023c), creating an information bottleneck as the embedding is not optimized for document reconstruction. In contrast, D2L conditions directly on the document for reconstruction, making its adapter an identifiable, decodable representation. We include `T2L` as a baseline (Section 3).

5 Discussion

We showed that the adapter a frozen D2L hypernetwork generates from a paper serves as a decodable, semantically meaningful representation of the idea in it. Using simple arithmetic operations on the embedding, we generated representative labels for clusters and interpolated between ideas. Our core contribution is conversational embeddings that allow researchers to query underlying ideas directly, moving beyond passive inspection of static vectors. This capability enables exploring the semantic structure of research areas without established taxonomies.

D2L embeddings form a smooth semantic space, where averaging and interpolating produce coherent descriptions (Section 3.1, Section 3.2). We hypothesize that this smoothness arises from mapping documents into a shared coordinate system of generative weights, where each point parameterizes a generative model. This explanation remains a conjecture based on empirical observations, and formally evaluating how the resulting distributions blend remains open for future work.

There is a tension between search and generation, as illustrated by `T2L` and D2L. `T2L` is built on `GTE` embeddings for similarity search, which retrieves documents well, but decoding from these embeddings fails to reconstruct the source (Section 3.1). In contrast, D2L adapters are trained for generation and capture full document content, but perform poorly on retrieval by cosine similarity (Section 3.3). We bridge this gap with an invertible map g_θ (Section 2.4), making D2L embeddings usable for search without losing decodability. Trained under the same supervision, this map leaves dedicated text encoders nearly unchanged (Appendix I). The transform therefore surfaces structure already present in the D2L embedding rather than introducing new geometry.

The D2L embedding is a joint function of the document and the base model. The hypernetwork generates an adapter on top of a base model with broad prior knowledge. Content already held by the base model need not be repeated in the adapter. The embedding therefore represents the document relative to this prior. This relative encoding emphasizes what a document adds beyond common knowledge. The prior also fixes the coordinate system. The same document yields different embeddings under different base models, each prior reshaping the geometry. Embeddings from different checkpoints do not share a space and cannot be pooled or compared directly. The choice of base model is part of the representation.

Several limitations remain. First, we test composition only along edges between two ideas. Whether the same behavior holds for mixtures of three or more sources is untested. Second, the embedding depends on its base model as well as the document. Two base models can therefore order the same documents differently. Third, the embedding is large and its encoder is larger. It carries an adapter for every layer, $147{,}456$ coordinates at full rank and $18{,}432$ once pooled, and it is produced by a 4.0B base model against the 110M of `SBERT`. Whether the tensor can be compressed without losing decodability remains untested. Current D2L architectures generate adapters across all layers, though individual layers may not contribute equally to generation. Selectively omitting adapters from certain layers might maintain response quality, or hypernetworks could be trained to target only specific layers to reduce parameter counts. These directions outline architectural refinements for future work.

Conventional embeddings reduce documents to static vectors that cannot render the space between papers as readable text. D2L establishes conversational embeddings where every point expands into an adapter that can be queried in natural language. We expect the ability to query embeddings to be especially valuable for investigating unexplored regions in the landscape of science, including knowledge holes, interdisciplinary gaps, and frontiers of knowledge. By enabling direct dialogue with constructed arithmetic points, conversational embeddings offer a path to gain insights into potential new ideas that lie between existing publications.

AI use statement

We manually wrote the initial drafts of the manuscript, then refined and reviewed the text using generative AI tools. We use AI tools for experiment coding. For literature search, we use Google Scholar and generative AI tools to help search for relevant literature, with all surfaced references checked by the authors. All results and outputs were verified against source data, and we take full responsibility for the content of the paper.

Reproducibility Statement

All experiments can be fully reproduced using the code, configurations, and Snakemake workflow included with this submission. The workflow covers every result and figure in the manuscript. The main results and manuscript figures are generated from the released `SakanaAI/doc-to-lora` checkpoints. Cluster-label comparisons and incoherent-cluster controls are judged by a panel of five language models from different providers, ensuring that no single model or naming convention dominates the scoring. The models are accessed via established APIs such as Google Vertex AI and OpenRouter. Importantly, no model judges its own output, and no label is rewritten by another model, so the scoring process is independent and unbiased. While the external judge models are versioned proprietary endpoints and may change over time, scoring within a fixed panel is deterministic and reproducible, i.e., inputs are presented in a stable order, and responses are cached, ensuring that repeated runs produce identical outcomes. Any label overlap analysis is model-free and fully reproducible from the released outputs.

References

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *Proceedings of ICLR*, 2023.

Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

Vinamra Benara, Chandan Singh, John X. Morris, Richard Antonello, Ion Stoica, Alexander G. Huth, and Jianfeng Gao. Crafting interpretable embeddings by asking LLMs questions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P. Calmon, and Himabindu Lakkaraju. Interpreting CLIP with sparse linear concept embeddings (SpLiCE). In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 2016.

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. Text-to-LoRA: Instant transformer adaption. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Rujikorn Charakorn, Edoardo Cetin, Shinnosuke Uesaka, and Robert Tjarko Lange. Doc-to-LoRA: Learning to instantly internalize contexts, 2026.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. SelfIE: Self-interpretation of large language model embeddings. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Tong Chen, Hao Fang, Patrick Xia, Xiaodong Liu, Benjamin Van Durme, Luke Zettlemoyer, Jianfeng Gao, and Hao Cheng. Generative adapter: Contextualizing language models in parameters with a single forward pass. In *Proceedings of ICLR*, 2025. URL https://openreview.net/forum?id=bc3sUsS6ck.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xRAG: Extreme context compression for retrieval-augmented generation with one token. In *Advances in Neural Information Processing Systems*, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Alexandra Chronopoulou, Matthew E. Peters, Alexander Fraser, and Jesse Dodge. AdapterSoup: Weight averaging to improve generalization of pretrained language models. In *Findings of EACL*, 2023.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of ACL*, 2020.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *Proceedings of ICLR*, 2022.

Santo Fortunato, Carl T. Bergstrom, Katy Börner, James A. Evans, Dirk Helbing, Staša Milojević, Alexander M. Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, Alessandro Vespignani, Ludo Waltman, Dashun Wang, and Albert-László Barabási. Science of science. *Science*, 359 (6379):eaao0185, 2018.

Jacob G. Foster, Andrey Rzhetsky, and James A. Evans. Tradition and innovation in scientists' research strategies. *American Sociological Review*, 80(5):875–908, 2015.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems*, 2018.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *Proceedings of ICLR*, 2024.

Gemma Team. Embeddinggemma: Powerful and lightweight text representations, 2025.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Maarten Grootendorst. KeyBERT: Minimal keyword extraction with BERT. https://github.com/MaartenGr/KeyBERT, 2020.

Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure, 2022.

Maarten Grootendorst. KeyLLM: Keyword extraction with large language models. https://maartengr.github.io/KeyBERT/guides/keyllm.html, 2023.

Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of NAACL*, pp. 708–719, 2018.

Almog Gueta, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. Knowledge is a region in weight space for fine-tuned language models. In *Findings of EMNLP*, 2023.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *Proceedings of ICLR*, 2017.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. In *Conference on Language Modeling (COLM)*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *Proceedings of ICLR*, 2023.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In *Proceedings of ICML*, 2021.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of ICML*, 2020.

Mario Krenn and Anton Zeilinger. Predicting research trends with semantic and neural networks with an application in quantum physics. *Proceedings of the National Academy of Sciences*, 117 (4):1910–1916, 2020.

Yuri Kuratov et al. Cramming 1568 tokens into a single vector and back again: Exploring the limits of embedding space capacity. In *Proceedings of ACL*, 2025.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of EMNLP*, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mario Lezcano-Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 2019.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of EMNLP*, 2020.

Haoran Li, Mingshi Xu, and Yangqiu Song. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In *Findings of the Association for Computational Linguistics: ACL*, pp. 14022–14040, 2023a.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of ACL*, 2021.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b.

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*, 2023c.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xCompressor: Generalized prompt compression for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25081–25091. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1219. URL https://aclanthology.org/2025.acl-long.1219/.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. Text embeddings reveal (almost) as much as text. In *Proceedings of EMNLP*, 2023.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Dakota Murray, Chaoqun Ni, Weiye Gu, and Trevor Hubbard. Using language models to label clusters of scientific documents. *Scientometrics*, 2025. doi: 10.1007/s11192-025-05445-5.

Kai Nylund, Suchin Gururangan, and Noah A. Smith. Time is encoded in the weights of finetuned language models. In *Proceedings of ACL*, 2024. URL https://aclanthology.org/2024.acl-long.141.

Charles O'Neill, Christine Ye, Kartheik Iyer, and John F. Wu. Disentangling dense embeddings with sparse autoencoders, 2024.

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*, 2022.

Alexander Pan, Lijie Chen, and Jacob Steinhardt. LatentQA: Teaching LLMs to decode activations into natural language. In *International Conference on Learning Representations*, 2026.

Hao Peng, Qing Ke, Ceren Budak, Daniel M. Romero, and Yong-Yeol Ahn. Neural embeddings of scholarly periodicals reveal complex disciplinary organizations. *Science Advances*, 7(17): eabb9004, 2021. doi: 10.1126/sciadv.abb9004.

Chau Minh Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. TopicGPT: A prompt-based topic modeling framework. In *Proceedings of NAACL*, 2024.

Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. HyperTuning: Toward adapting large language models without back-propagation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Jason Priem, Heather Piwowar, and Richard Orr. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, 2022.

Qwen Team. Qwen3 technical report, 2025.

David Rau, Shuai Wang, Hervé Déjean, Stéphane Clinchant, and Jaap Kamps. Context embeddings for efficient answer generation in retrieval-augmented generation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, pp. 493–502. ACM, 2025. doi: 10.1145/3701551.3703527. URL https://doi.org/10.1145/3701551.3703527.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP*, 2019.

Sotaro Shibayama, Deyun Yin, and Kuniko Matsumoto. Measuring novelty in science with word embedding. *PLOS ONE*, 16(7):e0254034, 2021.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. SciRepEval: A multi-format benchmark for scientific document representations. In *Proceedings of EMNLP*, 2023.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, 2022.

Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 377–390, 2020.

Jamshid Sourati and James A. Evans. Accelerating science with human-aware artificial intelligence. *Nature Human Behaviour*, 7:1682–1696, 2023.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of ACL*, 2023.

Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. Parametric retrieval augmented generation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1240–1250, 2025.

Shivashankar Subramanian, Daniel King, Doug Downey, and Sergey Feldman. S2AND: A benchmark and evaluation system for author name disambiguation. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2021.

Don R. Swanson. Undiscovered public knowledge. *The Library Quarterly*, 56(2):103–118, 1986.

Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571:95–98, 2019.

Brian Uzzi, Satyam Mukherjee, Michael Stringer, and Ben Jones. Atypical combinations and scientific impact. *Science*, 342(6157):468–472, 2013.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of ACL*, 2022.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. Goal-driven explainable clustering via language descriptions. In *Proceedings of EMNLP*, 2023.

Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval, 2025.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of ICML*, 2022.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *International Conference on Learning Representations*, 2024.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

Qinyuan Ye and Xiang Ren. Learning to generate task-specific adapters from task description. In *Proceedings of ACL*, pp. 646–653, 2021.

Hyejin Youn, Deborah Strumsky, Luís M. A. Bettencourt, and José Lobo. Invention as a combinatorial process: evidence from US patents. *Journal of the Royal Society Interface*, 12(106):20150272, 2015.

A Datasets

We evaluate on three academic domain corpora and the five S2AND author name disambiguation datasets. Physics comes from the American Physical Society (APS) corpus. Economics and Psychology are subsets cut out of OpenAlex (a bulk snapshot from March 2026) based on journals. We convert every paper into a single string `Title: … Abstract: …` and embed it once.

Physics (APS). The APS corpus contains 644,022 papers from 18 *Physical Review* journals (1893–2020), 8.3M citation edges, and author-specified PACS topic codes. Although the public release omits abstracts, we recover them from OpenAlex via DOI matching. 99.3% of the APS papers match OpenAlex works, and 97.3% of those contain abstracts. We embed all 644,022 papers regardless of whether we find an abstract. Every encoder receives the same string. Missing abstracts therefore introduce no bias into the comparison. To label clusters, we use PACS nodes from all three levels of the hierarchy (Section 3.1). Condensed matter accounts for 42.5% of the corpus.

Economics and Psychology (OpenAlex). For each discipline, we extract Q1/Q2 journals from the Scimago Journal Rank (SJR) in the target subject areas and match them by ISSN to OpenAlex venues. Economics uses "Economics, Econometrics and Finance" (1,035 journals), and Psychology uses "Psychology" (1,166 journals). A paper is included if it appears in a matched venue, has a non-empty scientific-work type and title, and has at least one author with ≥ 2 papers within the slice. This final condition excludes one-off author–venue combinations and preserves the connectivity of the authorship network. For text-based tasks, we also require abstracts of ≥ 300 characters and primary Scopus/ASJC topic labels. This filter leaves 565,475 Economics papers and 986,800 Psychology papers. Economics spans 26 major fields and 244 subfields. Psychology spans 26 major fields and 245 subfields. The Scopus/ASJC labels come from OpenAlex and are independent of the SJR venue rules.

Train-evaluation overlap. Both the transformation training pool and the evaluation corpus were obtained from OpenAlex. We validate the overlap by checking that evaluation papers account for less than 1% of the training set, and training citation edges falling within a single evaluation field constitute no more than 1.24%. Since we do not use any author labels for training g_θ, the author-merging supervision is completely separated. To verify the absence of information leakage, we retrained g_θ using only citations from 2018 or earlier; the change in next-paper AUC was at most .005, and the ranking of our method remained unaffected. Similarly, even when topic classification is split by time (training ≤ 2016, testing > 2016), the encoder's rankings are preserved. For Physics, since the labels are journal codes and the benchmark ended in 2015, we did not perform a temporal split.

Author-name disambiguation (S2AND). The five subsets of S2AND (Subramanian et al., 2021) are zbMATH (15,181 papers), QIAN (6,542), ArnetMiner (7,144), PubMed (2,871), and KISTI (40,383). Each provides ground-truth author clusters for ambiguous name blocks. We use title and abstract texts where available, leaving name blocks and metadata untouched.

B Recipe decoding and fusion example

We illustrate recipe fusion using two short recipe documents—Italian (cacio e pepe) and Japanese (kake udon), each encoded independently by the Mistral-7B D2L hypernetwork. We decoded the recipes at both endpoints and their interpolation (full-rank mode, Section 2.2), by prompting: "Write out this recipe: give the dish a name, list the ingredients, and describe the preparation steps." For example, with only the Italian adapter active, the model answers "The cheese used in this recipe is Pecorino Romano." when asked about the cheese, even without seeing the recipe text. Mistral-7B was used (rather than Qwen3-4B) for its more natural recipe outputs, but this does not affect the results. All outputs below are shown verbatim.

Source document A (Italian).

```
Cacio e Pepe (Roman pasta with cheese and black pepper)

A classic recipe of the Roman tradition with only three ingredients.
↪  Serves 2.

Ingredients:
```

```
- 200 g spaghetti (or tonnarelli)
- 100 g Pecorino Romano cheese, finely grated
- 2 teaspoons whole black peppercorns
- coarse salt for the pasta water

Preparation:
1. Toast the black peppercorns in a dry skillet over medium heat until
   fragrant, then crush them coarsely in a mortar.
2. Bring a pot of lightly salted water to a boil and cook the spaghetti
↪  until
   one minute short of al dente. Use less water than usual so the cooking
   water becomes rich in starch.
3. While the pasta cooks, mix the grated Pecorino Romano with a ladle of
↪  the
   hot, starchy pasta water, stirring vigorously until it forms a thick,
   smooth cream with no lumps.
4. Transfer the spaghetti to the skillet with the crushed pepper, add a
↪  splash
   of pasta water, and finish cooking for one minute, tossing constantly.
5. Off the heat, add the Pecorino cream and toss vigorously, loosening
↪  with
   more pasta water as needed, until every strand is coated in a glossy
   ↪  sauce.
6. Serve immediately with extra grated Pecorino and a final grind of
↪  pepper.
```

Source document B (Japanese).

```
Kake Udon (Japanese udon noodles in hot dashi broth)

A simple and warming Japanese noodle soup built on dashi stock. Serves 2.

Ingredients:
- 2 portions fresh or frozen udon noodles
- 600 ml dashi stock (kombu and katsuobushi)
- 3 tablespoons soy sauce
- 2 tablespoons mirin
- 1 teaspoon sugar
- 2 spring onions, thinly sliced
- shichimi togarashi (Japanese seven-spice), to taste
- optional toppings: kamaboko fish cake, tempura flakes (tenkasu), a soft
↪  egg

Preparation:
1. Make the dashi: soak a piece of kombu in cold water for 30 minutes,
↪  bring
   it almost to a boil and remove the kombu, then add a handful of
   katsuobushi (dried bonito flakes), turn off the heat, let it steep for
   2 minutes, and strain.
2. Season the dashi with the soy sauce, mirin, and sugar, and keep the
↪  broth
   hot over low heat without boiling.
3. Cook the udon noodles in a separate pot of unsalted boiling water
↪  according
   to the package directions, then drain and rinse briefly to remove
   ↪  surface
   starch.
4. Divide the noodles between two deep bowls and ladle the hot seasoned
↪  broth
   over them.
5. Top with sliced spring onions and any optional toppings, and finish
↪  with a
   pinch of shichimi togarashi. Serve immediately, very hot.
```

Decoded midpoint, $\alpha = 0.5$.

```
Recipe Name: Kake Udon with Creamy Parmesan Sauce

Ingredients:
- 200 g fresh udon noodles
- 2 tablespoons unsalted butter
- 2 tablespoons mirin
- 2 tablespoons soy sauce
- 1 teaspoon wasabi paste
- 1/2 cup grated Parmesan cheese
- 1/2 cup heavy cream
- 1 spring onion, thinly sliced
- Optional: a pinch of freshly ground black pepper

Preparation Steps:
1. Cook the fresh udon noodles according to the package instructions.
↪  Drain
   and rinse with cold water.
2. In a deep skillet or pan, melt the unsalted butter over medium heat.
3. Add the mirin and cook fsr 1 minute.
4. Stir in the soy sauce and wasabi paste.
5. Gradually add the grated Parmesan cheese, stirring constantly to
↪  prevent
   lumps from forming.
6. Pour in the heavy cream and bring to a simmer.
7. Cook for 2 minutes, stirring occasionally, until the sauce has
↪  thickened.
8. Add the cooked udon noodles to the skillet and toss to coat them
↪  evenly
   with the sauce.
9. Remove from heat and top with the sliced spring onion.
10. Optional: finish with a pinch of freshly ground black pepper.
11. Serve immediately.
```

The midpoint describes a dish that appears in neither source. Udon noodles coated in a melted-cheese cream sauce, the cacio e pepe technique, seasoned with the mirin and soy of the Japanese broth and finished with the spring onion of one source and the black pepper of the other. The endpoints are faithful. $\alpha = 0$ decodes to cacio e pepe and $\alpha = 1$ to kake udon, each with minor paraphrases. Off-midpoint interpolants snap to the nearer source ($\alpha = 0.25$ decodes to cacio e pepe; $\alpha = 0.75$ to kake udon with butter entering the broth). We note that the blend is imperfect in an instructive way. Namely, the midpoint substitutes Parmesan for Pecorino Romano and introduces heavy cream found in neither source, so the fusion is a plausible blend of the two cuisines rather than a constraint-satisfying merge of the two ingredient lists.

C Bijective transforms

The transform g_θ must be invertible so that we can always recover the adapted representation needed for decoding (Section 2.4). We guarantee this invertibility by structuring the channel transformation matrix $\mathbf{C}$. Specifically, we write $\mathbf{C}$ as

$$\mathbf{C} = \mathbf{R}\,\mathrm{diag}(\sigma_1, \ldots, \sigma_{n_d}), \qquad \sigma_1, \ldots, \sigma_{n_d} > 0, \tag{3}$$

where $\mathbf{R}$ is a rotation matrix (orthogonal) and $\mathrm{diag}(\sigma_1, \ldots, \sigma_{n_d})$ is a diagonal matrix with all positive entries. Both of these components are invertible, i.e., a diagonal matrix with positive values is inverted simply by taking the reciprocal of each entry, and a rotation is inverted by transposing it. Therefore, their product $\mathbf{C}$ is also invertible.

To make sure that $\mathbf{R}$ stays a true rotation during training, we construct it by exponentiating a skew-symmetric matrix:

$$\mathbf{R} = \exp(\mathbf{P} - \mathbf{P}^\top) \tag{4}$$

where $\mathbf{P}$ is any square matrix of size $n_d \times n_d$ (with no constraints). The difference $\mathbf{P} - \mathbf{P}^\top$ is always skew-symmetric by construction, and exponentiating any skew-symmetric matrix produces

an orthogonal (rotation) matrix (Lezcano-Casado & Martínez-Rubio, 2019), which lets us freely optimize $\mathbf{P}$ as unconstrained parameters.

The full inverse of g_θ is given by

$$g_\theta^{-1}(\mathbf{V}') = a_\ell^{-1}\mathbf{V}'\mathbf{C}^{-1}, \tag{5}$$

where $a_\ell > 0$ is a scalar learned separately for each layer (Section 2.4). Since g_θ is linear, operates along the channel axis, and uses a simple per-layer scaling, it commutes with mean-pooling over the rank axis. In other words, it produces the same result if we pool the rank dimension before or after applying g_θ. Therefore, we can use g_θ, trained on full-rank tensors, on mean-pooled vectors as well.

We initialize g_θ to the identity (i.e., no transformation initially) and keep all parameters of the base language model frozen. Training for g_θ uses 42,332 examples from OpenAlex (see Appendix A) paired up as anchor, positive, easy negative, and hard negative. The easy and hard negatives follow the sampling protocol of `SPECTER`. An easy negative is a paper drawn uniformly at random from the corpus, excluding the anchor and the papers it cites. A hard negative is a paper cited by the positive but not by the anchor, i.e., at distance two from the anchor. The optimization uses an InfoNCE contrastive loss with temperature 0.05, in which each anchor is scored against its own hard and easy negative together with the other 255 positives in the batch. We train for 8,000 Adam steps with batch size 256 at a learning rate of 10^{-3} and no added regularization.

D Cluster-label evaluation protocol

We evaluate labeling over a 28-node PACS hierarchy slice (four fields, each with two categories and two subcategories). All methods are given the same cluster members and asked to assign scientific field labels. Compared methods include D2L, `ICAE`, `KeyLLM`, `BERTopic`, `vec2text`, `T2L`, and a baseline (`in-context`) using only the base Qwen3-4B LLM prompted with abstracts from the 20 members nearest each centroid. `ActPatch` (hidden-state baseline) is evaluated separately in Appendix E.

D2L receives its instruction with the cluster documents internalized in the adapter and no document text in the prompt:

> *In 2 to 3 words, name the scientific field that all of these documents belong to. Reply with only the field name.*

`ICAE` receives the same prompt verbatim and decodes against its averaged memory slots. `T2L` receives the same prompt and decodes using its training prompt format. `T2L` turns a document into an adapter in three stages (a `GTE` vector, a 64-dimensional TaskEncoder output, and the LoRA weights), so a cluster could be averaged at any of them. We average the TaskEncoder outputs, as in the mixing comparison (Appendix F); averaging at the other two stages yields similar unrelated field names ("Social Sciences", "Information science"). `KeyLLM`, `BERTopic`, and `vec2text` cannot take this instruction. `KeyLLM` is a keyword extractor returning a ranked list of keywords. `BERTopic` represents a cluster by its ten most distinctive words (Appendix D.1). `vec2text` is an iterative inverter trained to reconstruct text from an embedding and has no prompt interface. We therefore score every method on its native output without rewriting any label through a second language model. Passing outputs through an auxiliary namer would test the namer. The comparison asks whether the representation itself supports the abstraction.

This design categorizes the methods based on output length. Compared to the official PACS labels of 4.0 words, both D2L and `ICAE` respond to prompts with 2.4 words each. Therefore, their scores are directly comparable. `KeyLLM` averages 8.8 words, `BERTopic` 10 words, and `vec2text` 15.5 words. The inability to adhere to a concise naming format is an inherent characteristic of the methods, and we do not apply length corrections. `T2L` outputs average 2.0 words. Across all methods, D2L, `ICAE`, `T2L`, and `vec2text` operate exclusively on embeddings, whereas `KeyLLM`, `BERTopic`, and the `in-context` read the member text. Among the embedding methods, D2L differs in that it reads the abstraction level from the length of the mean adapter vector (Appendix D.3).

Fuzzy overlap compares native outputs directly against official PACS labels using a deterministic token-set ratio in `rapidfuzz`. The score ranges from 0 to 1, involves no language model, and reproduces deterministically from the released strings. Because fuzzy overlap measures surface tokens,

Table 3: Cluster-labeling protocols comparing embedding-arithmetic (top) and text-based (bottom) methods without downstream language model post-processing: "Operates on" denotes the representation averaged and labeled, "#docs" indicates consumed member documents per node, and "Native output" defines directly evaluated raw outputs.

Method	Operates on	#docs	Native output (scored as-is)
D2L	embedding (mean adapter)	up to 2,000	field name (2–3 words)
ICAE	embedding (mean memory slots)	8	field name (2–3 words)
T2L	embedding (mean TaskEncoder output)	up to 2,000	field name (2–3 words)
vec2text	embedding (mean GTR vector, inverted)	20	reconstruction (no prompt)
KeyLLM	text (medoid document)	1	keyword list
BERTopic	text (members, c-TF-IDF)	up to 2,000	keyword list
in-context	text (centroid-nearest abstracts)	20	field name (2–3 words)

verbose outputs receive lower scores against four-word reference labels, and the metric cannot credit valid paraphrases.

The **pairwise win rate** evalutes each method using five language-model judges (Reproducibility Statement, p. 10). For each cluster, judges compare the official PACS label and two candidate method labels, picking the closer one in meaning or declaring a tie; all pairs are shown in both label orders to remove position bias, i.e., a win counts only if both orders agree, otherwise scored as a tie (0.5). The seven methods of Table 4 enter the round robin, each compared with every other on every node. A method's win rate is the fraction of its comparisons that it wins, averaged over nodes. Fig. 2b shows the rate of each row method against each column method, and its row mean is the win rate. Judges rate D2L labels as tied with official PACS names on 32% of judge-node decisions. By comparison, this rate is 25% for ICAE, 22% for the in-context, and 0% for KeyLLM, BERTopic, and vec2text.

Table 4: Cluster-label quality over 28 PACS nodes. Each method's native output is scored against the true node label. **Fuzzy overlap:** deterministic token-set ratio with the reference. **Pairwise:** win rate from a five-judge round robin over the seven methods below; ties score 0.5 and require both label orders to agree. $\pm$ is the bootstrap s.d. over 1,000 node resamples. **Words:** mean label length (*PACS:* 4.0). The ground-truth control has no win rate. Best (excluding control) in bold. Full pairwise matrix: Fig. 2b.

Method	Words	Fuzzy overlap	Pairwise
Ground truth (control)	4.0	1.000	—
D2L	2.4	**.642 ± .055**	**.755 ± .032**
ICAE	2.4	.573 ± .058	.678 ± .039
KeyLLM	8.8	.311 ± .019	.474 ± .034
vec2text	15.5	.302 ± .026	.215 ± .021
BERTopic (top 10)	10.0	.296 ± .020	.696 ± .033
T2L	2.0	.277 ± .010	.002 ± .001
in-context	2.4	.499 ± .051	.678 ± .031

D2L achieved the highest scores on both metrics, with a fuzzy overlap of $.642 \pm .055$ and a pairwise win rate of $.755 \pm .032$ (Table 4). Two of the five judges rank D2L first by win rate and three rank BERTopic first, because BERTopic beats the weakest methods by wider margins (e.g., .850 against KeyLLM, where D2L has .750). Head to head, the judges still prefer D2L over BERTopic (.604; Appendix D.1). D2L 's direct baseline is ICAE, which processes identical instructions and generates outputs of the same average length. D2L outperforms ICAE in pairwise win rates (.755 vs. .678). Since their marginal intervals overlap in terms of fuzzy overlap ($.642 \pm .055$ vs. $.573 \pm .058$), we examine the paired node-level differences. The mean paired difference is $+.069$, the bootstrap standard deviation is .034, and the 95% confidence interval is $[.016, .145]$. Out of a total of 28 nodes, D2L achieved a higher overlap at 12 nodes, ICAE achieved it at 1 node, and 15 nodes were tied. In direct head-to-head comparisons, D2L and ICAE tied in 103 out of 140 decisions; D2L won approximately two-thirds of the remaining decisions, resulting in a head-to-head win rate of .596. Compared to in-context (pairwise .678, fuzzy .499), D2L recorded a head-to-head win rate of

.629, with 78 ties. Therefore, decoding an averaged adapter recovers the field at least as well as reading the member abstracts.

D2L is the top performer on both metrics and wins every head-to-head comparison. Compared to its closest baseline, ICAE, which operates under the same conditions, D2L leads in win rate, and their other scores are similar enough to warrant direct comparison. Analysis at the node level shows D2L typically outperforms or ties with ICAE. In head-to-head matchups, D2L has a clear edge, and it also surpasses the in-context approach. Overall, decoding an averaged adapter is at least as effective as reading member abstracts.

T2L performs worst on both metrics, outputting only two field names ("Social Sciences" and "Social psychology") across all 28 nodes, with a fuzzy overlap floor of .277. It loses almost every pairwise comparison, including against vec2text. The method's mapping does not yield decodable representations. The synthesized adapters expanded from the frozen GTE encoder cannot generate meaningful labels. We note that cluster-labeling is not the intended use case and goal for T2L.

Native outputs for every PACS node appear in Table 12 in Appendix J. D2L outputs match official designations, producing "Quantum chromodynamics" for code 12.38 and "Fluid dynamics" for code 47. KeyLLM reflects single-document specificity, returning "square-lattice Heisenberg antiferromagnet, S=1/2, nearest-neighbor exchange" for magnetic ordering (75.10). ICAE identifies "Condensed matter physics" for code 71 and "Nonlinear optics" for code 42.65, but occasionally shifts hierarchy levels, returning "Materials science" for bulk band structure (71.20) and "Quantum many-body systems" for general physics (0). vec2text reconstructions are garbled, yielding "in-distance interactions between electron-bonding spectral properties" for code 3 and "inferromagnetic spin-diabetic scattering interactions" for code 75.

D.1 BERTopic baseline. The standard BERTopic pipeline finds clusters (UMAP + HDBSCAN) and then uses c-TF-IDF to extract words that are frequent within a cluster and rare elsewhere; as our evaluation assumes clusters are fixed, we run BERTopic in its manual topic modeling mode (bypassing clustering via empty BaseDimensionalityReduction and BaseCluster, with PACS nodes as clusters) to test only the labeling step. We match D2L's setup by providing identical titles, abstracts, a maximum of 2,000 papers per node (seed 0), and precomputed SBERT embeddings. Since c-TF-IDF compares each cluster to all others, output depends on the full set of clusters. Therefore, we fit a model for each PACS hierarchy level, including all groups at that level with at least 100 papers: 9 fields (18k papers), 65 divisions (99k), and 393 subdivisions (305k), not just the nodes we score.

We evaluate BERTopic using its default top-10 keyword list as the label, and also try a shorter top-3 list to control for label length, since longer labels are penalized by token-set metrics. While BERTopic's fuzzy overlap score is .296 for the top-10 (or .346 for the top-3), similar to KeyLLM (.311) and vec2text (.302), the judges tend to group BERTopic with methods that successfully name the field. In direct comparisons, BERTopic performs worse than D2L (winning only .396 of matchups), about the same as in-context (.493), close to ICAE (.471), and better than KeyLLM (.850) and vec2text (.964). This suggests that fuzzy overlap in keyword-based labels is influenced more by length and wording than by content. Although using the top-3 list improves fuzzy overlap by .05, judges only prefer it over the top-10 list in 17.5% of cases.

BERTopic never outputs the field name itself; instead, it produces lists of common subtopics found within each field, like "quark, mass, model, gauge, higgs, theory, qcd, decays, decay, production" for Elementary particles (node 6), or "quantum, time, state, systems, states, model, entanglement, black, field, phase" for General (node 0). Because few words are unique to a field, these lists cannot represent the field name clearly. Judges never considered BERTopic's label equivalent to the correct field name (0% of cases, both for the regular and top-3 lists), whereas methods like D2L, ICAE, and in-context tied with the correct name 32%, 25%, and 22% of the time, respectively. The top-3 word list was evaluated separately from the main results, always using the same judges, protocol, and random seed. Finally, we did not include results from the standard BERTopic pipeline, as it produces its own clusters instead of labeling the original PACS nodes.

D.2 Validity of cluster labels. We test whether decoded cluster labels require a coherent member set (Table 5). For each of 30 real PACS nodes, we construct two control clusters of identical size by

sampling papers uniformly across PACS chapters outside the target node. We decode one control at its native centroid length of 11.15 and a second control scaled to the real-node mean of 11.49.

A five-judge panel sees each label alongside twelve sampled member titles and is not told which method or cluster type a row comes from. The panel accepts the real-node label on 73.3% of clusters and rejects 3.3% (one of 30). The remainder are judged partial. Judges accept zero control labels, rejecting 100% of the native control and 96.7% of the norm-matched control. Control decodings collapse almost uniformly to "Quantum field theory". The norm-matched control earns the same verdict as the native control.

Table 5: Scores from the blinded five-judge panel on the incoherent-cluster control experiment. *PACS node (real)* denotes actual clusters, while the control rows consist of clusters of the same size sampled across different PACS chapters and decoded either at their native centroid norm or rescaled to the real nodes' norm. *belongs=yes* and *rejected* indicate the proportion of clusters whose label the panel accepted as valid for the members or rejected entirely. The rest were judged partial.

cluster set	n	belongs=yes	rejected
PACS node (real)	30	73.3%	3.3%
cross-chapter control	60	0%	100%
cross-chapter, norm-matched	60	0%	96.7%

D.3 Adapter length sets label granularity. Vector length governs the level of abstraction in decoded D2L embeddings. Section 3.1 shows that scaling an embedding by α shifts the decoded label from specific topics to general fields (for instance, moving from citric acid cycle to cellular respiration) until near the origin the label reverts to the prior distribution of the prompt. We evaluate this magnitude sweep on `ICAE` and `vec2text` across five seed documents covering a physics concept, a biochemistry process, a fruit, a historical person, and a patent. Fig. 2d shows the D2L sweep on four of these seeds. The person seed appears only in the tables of Appendix J.

Neither baseline exhibits upward semantic abstraction under vector scaling. `ICAE` remains semantically unchanged across lengths, returning the specific concept label at every value of α with variations limited to capitalization and surface wording. `vec2text` maintains surface-level specificity while the reconstruction remains coherent, then decomposes into unrelated text fragments at low α without naming broader fields (Table 11). Among the three embedding methods on this sweep, the specific-to-general ladder is specific to D2L.

E Activation patching does not recover the source document

LatentQA (Pan et al., 2026) could not be reproduced from the released code and checkpoint. Intended as a trained activation decoder baseline, it yields no reported metrics. The claim we could have drawn is that a trained activation decoder also fails, and a broken baseline produces exactly the appearance our result wants. We instead report a training-free readout of Qwen3-4B-Instruct hidden states via activation patching (Ghandeharioun et al., 2024; Chen et al., 2024).

For each document, we mean-pool a hidden state h from a chosen layer, insert h into a fixed slot of the completion prompt, and generate tokens. We sweep layers, slot counts, and decoding style. The question prompt

> *Document: ?*
>
> *In one sentence, what is the specific topic of this document?*

reports that the document is empty. The continuation prompt

> *The following is a scientific abstract.*

carries topic information. The sweep selects layer 15 with eight slots. This baseline is `ActPatch`. Both D2L and `ActPatch` inject a vector into the same model and prompt, differing only in vector origin. Generated outputs are compared to those from the unpatched prompt to control for prompt prior.

We first test whether the resulting embeddings can identify source documents. Tested on 500 abstracts per method retrieving each among 99 same-subfield distractors, D2L retrieved its source $91.8\% \pm 1.2$ (MRR 0.946 ± 0.008), while `ActPatch` managed just $6.0\% \pm 1.1$ (MRR 0.132 ± 0.011), which is near chance (1.0%). Decodes from activations retain weak document signal. Their SBERT similarity to the source is only 0.211 above baseline (0.794 for D2L). `ActPatch` decodes also often collapse into repetition (100 of 500, vs 1 for D2L).

We interpolate the same 100 document pairs as Section 3.2 by decoding three points along each interpolant ($\alpha = 0, 0.5, 1$), using a linear mix of full D2L embeddings or mean-pooled hidden states for `ActPatch`. Each decode D of papers A and B is scored by how much its SBERT similarity (all-mpnet-base-v2, see Appendix F) to A and B exceeds the baseline similarity between A and B.

$$\gamma(D; A, B) = \min\big(\text{sim}(D, A),\ \text{sim}(D, B)\big) - \text{sim}(A, B). \tag{6}$$

A verbatim copy of A or B scores $\gamma = 0$. $\gamma > 0$ indicates D is closer to both sources than they are to each other. $\gamma < 0$ means at least one similarity falls below the baseline.

The D2L midpoint achieves $\gamma = .171 \pm .011$, positive for 99 of 100 pairs, with endpoints at -0.020 and -0.008. The `ActPatch` midpoint yields $\gamma = -0.039 \pm .015$. Out of the 50 closest pairs, where the sources are most similar and the required threshold for positivity is highest, D2L remains positive for all, while `ActPatch` does so for 8.

Across 28 PACS node means, D2L reaches fuzzy overlap 0.575 ± 0.043 while `ActPatch` gets 0.476 ± 0.038. Judge agreement is 0.650 ± 0.068 for D2L and 0.350 ± 0.065 for `ActPatch`. The D2L label is closer on 15 nodes, the activation label wins on 5, and there are 8 ties. We note that our results do not eliminate the possibility of hidden state-based decoding strategies. In fact, we note that the `ActPatch` is train-free and also small in size compared to D2L. We instead clarify that D2L provides values beyond bare hidden-state-based decoding.

F Fusion protocol and metrics

We sample paper pairs from the APS corpus, grouped by PACS taxonomy distance. The near group (L1) has pairs within the same subdivision, the far group (L5) has pairs from different subtopics. For each group, we draw 50 pairs and label each paper A or B. Fig. 2e and Fig. 2f show the near group. For each pair, we evaluate all methods on 13 interpolation weights from 0 to 1 in steps of 1/12, representing how much paper B contributes. Each method generates outputs at these 13 points using its representation space and the prompt in Section 3.2. D2L interpolates the full-rank embeddings of A and B, decoding with Qwen3-4B. `ICAE` combines memory slots with matching weights then decodes with its native decoder. `T2L` interpolates TaskEncoder outputs (64 dimensions per document) then runs the hypernetwork to make a LoRA adapter. If we interpolate instead at the frozen `GTE` input or at the final LoRA factors, the curve stays flat. For the in-context baseline, Qwen3-4B receives both document texts and the mixing percentages directly. These explicit percentages give the in-context baseline mixing weights that are not available to the other methods.

```
You are given two short documents:

[A] <corner A>

[B] <corner B>

Imagine ONE research topic that blends these two in the
↪  proportions
A:<pa>%, B:<pb>%. Write a detailed abstract (four to six
↪  sentences)
describing this research topic: its problem, methods, and findings.
```

We measure where decoded abstracts land on the axis between the two source papers in SBERT space using all-mpnet-base-v2. Let $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{d}$ denote the embeddings of paper A, paper B, and the decoded abstract. The position t is the least-squares projection $t = (\mathbf{d} - \mathbf{a}) \cdot (\mathbf{b} - \mathbf{a}) / \|\mathbf{b} - \mathbf{a}\|^2$. Under this metric, $t = 0$ corresponds to paper A, and $t = 1$ corresponds to paper B. Values of t can fall outside the unit interval $[0, 1]$ and are not clipped. Fig. 2e plots the mean position over pairs at each grid point with its 95% bootstrap interval.

We measure verbatim copying as the fraction of word tokens in the decoded abstract that appear identically in either source abstract. Word tokens are lowercase alphabetic character sequences of length three or greater. Abstracts randomly sampled from the corpus share roughly 30% of their tokens by chance. We display this floor as a dashed line in Fig. 2f.

G Similarity-benchmark protocol

The field-level benchmarks cover Economics, Psychology, and Physics, with data described in Appendix A. For D2L, we evaluate embeddings produced by the Qwen3-4B checkpoint across the primary benchmark, including all three field tasks and the five S2AND datasets. For retrieval, we mean-pool the adapter tensors across the rank dimension as defined in Section 2.2. Because the transformation g_θ is linear, pooling before or after applying g_θ yields identical vectors. Text baselines comprise `SPECTER2` (Singh et al., 2023), `SBERT` (all-mpnet-base-v2) (Reimers & Gurevych, 2019), `Instructor` (instructor-large) (Su et al., 2023), `EmbeddingGemma` (`embeddinggemma-300m`) (Gemma Team, 2025), and `GTE` (`gte-large-en-v1.5`) (Li et al., 2023c) used by `T2L`. All text encoders run with default settings without task-specific tuning. Table 2 lists the absolute scores, and the mean ranks cited in Section 3.3 are computed from that table.

For the S2AND datasets, we evaluate against the earlier `SPECTER` model because the S2AND system is designed around those representations. The benchmark does not provide `SPECTER2` vectors. Generating vectors locally would require changing both the text preprocessing and the encoder, which causes the evaluation to diverge from the published benchmark. Because official vectors do not exist for the other text baselines, we compute their embeddings ourselves from the published titles and abstracts.

For next-paper prediction, papers are split per author over time. Each query paper is paired with the next paper written by the same author, which forms a positive pair. We also pair the query paper with a randomly sampled paper that the author did not write, which forms a negative pair. For topic classification, we assign OpenAlex papers to one of 26 Scopus or ASJC classes, APS papers to one of 18 Physical Review codes, and sample 60000 papers seen by all encoders, splitting into 80 percent training and 20 percent test, not based on year. Collaboration prediction uses anchor years 2000 and 2008 for APS and 2008, 2012, and 2016 for Economics and Psychology. At each anchor, an author is represented by averaging their papers from the previous four years. Candidate pairs are author pairs at network distance two who share at least one coauthor and have no joint publication. A positive example is a candidate pair who coauthor a paper together in the next three years after the anchor, while a negative is a pair who do not coauthor in that period. Because positive and negative pairs have identical local coauthorship structure, only content similarity can distinguish them. We pool the candidate pairs of all anchor windows and compute a single AUC-ROC over the pooled pairs. The AUC-ROC of each window is reported in Table 10. For author-name disambiguation, we use the five S2AND subsets from Appendix A; each signature is represented by the embedding of its paper. Signatures are clustered by agglomerative average linkage with a cosine-distance threshold tuned on training data and performance is reported using B3 F1.

Uncertainties are computed via 1,000 bootstrap resamples of per-unit evaluation scores; the $\pm$ in Table 2 denotes the standard deviation across a method's own resamples, with each method resampled independently (so intervals are conservative for comparisons). Mean ranks in Section 3.3 average over all fourteen benchmarks (not raw scores) to account for differing metrics and scales. Resampling units: next-paper prediction resamples clustered queries; collaboration prediction, candidate author pairs; topic classification, test papers; disambiguation, signatures within name blocks, with B^3 precision and recall drawn together to compute F1.

Relative model rankings in Table 2 remain stable across all three base models (Table 9 in Appendix J). The smallest model, Gemma-2-2b, trails Qwen3-4B by up to $\sim$.05 on the field tasks but exceeds it on Physics collaboration (.823 against .795). Mistral-7B performs comparably to Qwen3-4B and outperforms it on Physics next-paper prediction (.965). Each cell in Table 9 reports raw and $+g_\theta$ scores using the evaluation pools, metrics, and bootstrap procedures from Table 2, omitting standard deviations for readability. Because text baseline scores are independent of the base model, we report them only in Table 2.

H PROMPT SENSITIVITY

Paraphrasing the decode prompt changes the wording but rarely the content (Table 6). For PACS cluster labels and two-paper midpoint descriptions, we decoded each point under eight semantically equivalent prompts, with generation settings held fixed. We score the eight outputs of a point by SBERT cosine. Two different PACS nodes decoded under one prompt have mean cosine .463, and two different midpoints have mean cosine .244.

The eight PACS-label prompts follow (1 is the wording of Section 3.1).

1. In 2 to 3 words, name the scientific field that all of these documents belong to. Reply with only the field name.
2. Using 2 or 3 words, state the scientific field shared by all of these documents. Answer with the field name only.
3. What scientific field do all of these documents belong to? Answer in 2-3 words, nothing else.
4. Name, in at most three words, the research field common to these documents. Output only the name.
5. Give the shared scientific field of these documents as a 2-3 word phrase and nothing more.
6. Identify the field of science that covers all of these documents. Reply with a two or three word field name only.
7. In a phrase of two to three words, what is the common scientific field of these documents? Reply with the phrase alone.
8. Summarise the scientific field of these documents in 2-3 words. Do not add any explanation.

The eight midpoint-description prompts follow (1 is the 2–3 sentence combined-idea prompt).

1. You have read two research papers and hold a single combined idea. In 2 to 3 sentences, describe the combined research idea. Reply with the description only.
2. Describe, in two or three sentences, the single research idea that combines the two papers you have read. Output only the description.
3. What research idea sits between the two papers you have read? Answer in 2-3 sentences.
4. In at most three sentences, state the merged research idea formed from the two papers.
5. Summarise the blended research idea of the two internalised papers in two to three sentences.
6. Give a 2-3 sentence description of the research idea that fuses the two papers you hold.
7. Express, in two or three sentences, the combination of the two papers as one research idea.
8. Write two or three sentences describing the single idea that unites the two papers you read.

PACS labels mostly stay the same string. Eight prompts produce 3.2 distinct labels on average. The most frequent label appears in 72% of the outputs, and 51% of prompt pairs match exactly.

Midpoint descriptions produce a new string under every prompt. Mean pairwise cosine is .705. The lowest pair is .507, still well above the .244 baseline between different points. Two descriptions of the same midpoint therefore remain closer than descriptions of two different midpoints, even when the prompt changes.

The wording in the paper is typical of this paraphrase set. Similarity to that wording equals or exceeds the mean pairwise similarity (.841 for labels, .705 for midpoints).

Table 6: Prompt sensitivity. K is the number of semantically equivalent prompts per point, the first being the one the paper reports. *mean pairwise sim.* averages SBERT cosine over prompt pairs for the same point; *sim. to the reported prompt* compares the other seven against it. *identical strings* is the fraction of those prompt pairs that emit the same text after lowercasing. Two different PACS nodes decoded under one prompt have mean cosine .463. Two different midpoints decoded under one prompt have mean cosine .244.

decode family	units	K	mean pairwise sim.	sim. to the reported prompt	identical strings
PACS node labels (2–3 words)	30	8	0.787	0.841	51%
two-paper midpoint descriptions	40	8	0.705	0.705	0%

We paraphrased the four-to-six sentence abstract prompt from Section 3.2 three times. The wordings are listed as follows:

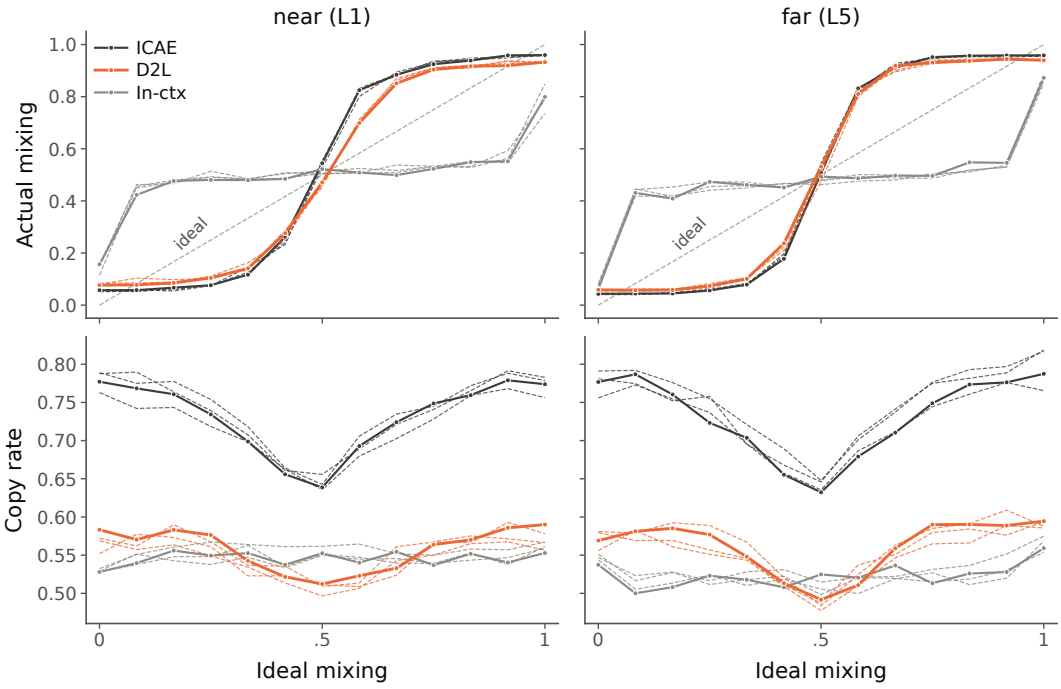

Figure 3: Decoded abstracts along the A–B interpolation under paraphrase, near pairs (left) and far pairs (right). Solid lines are the wording reported in Section 3.2 and dashed lines its three paraphrases. *Top*, where the decoded abstract lands on the A→B axis in SBERT space against the requested mix weight, with the diagonal marking proportional tracking. D2L and ICAE sit near a source paper at the endpoints and blend through the middle, while in-context prompting stays near the midpoint across the whole interior. *Bottom*, verbatim copy rate. ICAE sits above the other two everywhere, and D2L and in-context prompting overlap. Paraphrase moves every curve by less than the gap between methods.

1. Write a detailed abstract (four to six sentences) describing this research topic: its problem, methods, and findings.
2. In four to six sentences, write a detailed abstract for this research topic, covering the problem it addresses, the methods it uses, and its findings.
3. Produce a detailed abstract of this research topic – its problem, its methods and its findings – in four to six sentences.
4. Describe this research topic as a detailed abstract of four to six sentences, stating the problem addressed, the methods used and the results obtained.

For each wording, we decoded 20 near pairs and 20 far pairs at 13 mix weights for each method (D2L, in-context prompting, ICAE), totaling 6240 decodes. The in-context baseline used the same two-document template from Appendix F, changing only the instruction.

Decoded abstracts closely tracked the requested mix for every wording (see Fig. 3). For D2L, slope of decoded position versus mix weight ranged from 1.11 to 1.20. For ICAE, 1.19 to 1.24. In-context prompting ranged from 0.29 to 0.41. No wording changed the slope by more than 0.04. Copy rate stayed consistent. ICAE copied the most, averaging .72 to .75, and never matched the other methods. D2L and in-context prompting were both near .55.

Changing the mix point rewrote abstracts much more than changing the prompt. Abstracts at the same point under different wordings had SBERT cosine similarity of 0.886 for D2L, 0.924 for in-context prompting, and 0.923 for ICAE. Abstracts at different points under one prompt had much lower cosine (0.146 for D2L, 0.256 for in-context prompting, 0.125 for ICAE).

I BENCHMARK PERFORMANCE OF THE BASELINE ENCODERS WITH g_θ

We used the same type of transformation (the same g_θ transform) in each baseline embedding space and repeated all 14 benchmarks, following the same training steps as for D2L. However, the number of parameters in the transform varies between models: the D2L transform has 262,692 parameters, while the transforms for the baseline encoders are larger—590,593 for text embeddings with 768 dimensions, and 1,049,601 for those with 1,024 dimensions.

D2L gains +.061 on average, improving on 11 of the 14 benchmarks (Fig. 4 and Table 7). In contrast, baseline text encoders gain at most +.009 on average, and SBERT, the strongest baseline, loses .007 and degrades on 10 of the 14 benchmarks. In summary, for text encoders trained for similarity search, post adaptation via g_θ offers no recoverable structure and disrupts existing geometry. In D2L, it recovers latent citation-aligned geometry unexposed to cosine similarity.

Table 7: Change in benchmark score when the same transform g_θ is trained in each embedding space. Each row summarizes one space over the 14 benchmarks of Table 2. *mean* Δ: score with g_θ minus score without, averaged over the benchmarks. *range*: smallest and largest change. *better* and *worse*: number of benchmarks with a significant gain or loss, i.e., a 95% bootstrap interval of the change entirely above or below zero. Each bootstrap sample scores the space with and without g_θ on the same resampled evaluation units. This pairing yields a narrower interval for the change than the separate intervals of Table 2.

space	mean Δ	range	better	worse
`ICAE`	+.178	[-.004, +.549]	13/14	1/14
D2L (Qwen3-4B)	+.061	[-.009, +.133]	11/14	1/14
`Instructor`	+.009	[-.020, +.040]	7/14	2/14
`GTE`	+.007	[-.022, +.029]	8/14	2/14
`EmbeddingGemma`	+.001	[-.044, +.025]	6/14	3/14
`SPECTER2` / `SPECTER`	+.001	[-.034, +.023]	5/14	4/14
`SBERT`	-.007	[-.017, +.005]	1/14	10/14

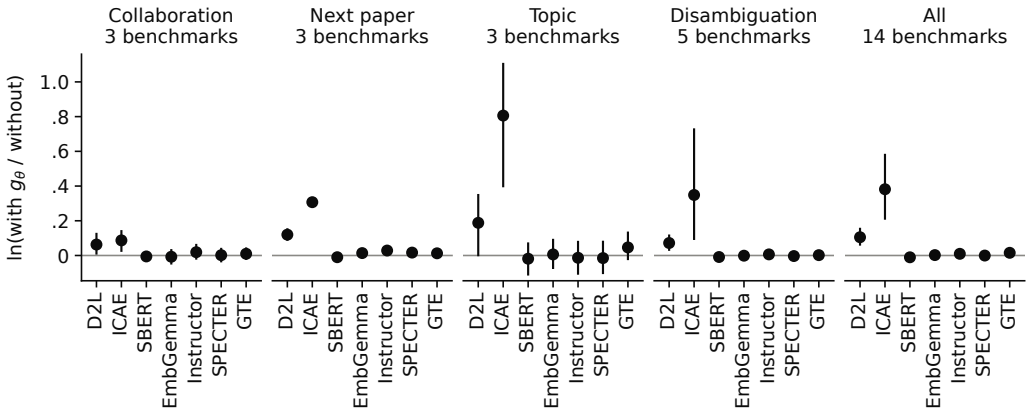

Figure 4: Gain from the shared transform g_θ in every space on the 14 benchmarks of Table 2. For each benchmark and space we take the natural logarithm of the score with g_θ divided by the score without it. Each circle averages this log ratio over the benchmarks of a panel. The first four panels are collaboration prediction, next-paper prediction and topic classification on Physics, Economics and Psychology, and author-name disambiguation (B^3 F1) on the five S2AND datasets. The last panel averages all 14. Whiskers are 95% intervals of a two-level bootstrap that resamples the panel's benchmarks and redraws each score with and without g_θ from its own bootstrap distribution. EmbGemma is `EmbeddingGemma`, and SPECTER is `SPECTER2` on the field benchmarks and `SPECTER` on S2AND, each with its own transform.

`ICAE` is a method that reduces the amount of text fed into the context by compressing the input document into a fixed number of tokens called "memory slots." It was not originally trained for

Table 8: Scores of every space without g_θ, averaged over the benchmarks of each task (three fields for collaboration, next-paper and topic, five S2AND datasets for disambiguation). Per-benchmark scores with intervals are in `figs/groupc_similarity_symmetric.tex`.

space	Collab. (AUC)	Next paper (AUC)	Topic (F1)	Disamb. (B^3 F1)
D2L (Qwen3-4B)	.663	.823	.358	.760
`ICAE` (Mistral-7B)	.660	.677	.227	.617
`SBERT`	.726	.941	.443	.827
`EmbeddingGemma`	.712	.919	.433	.813
`Instructor`	.708	.887	.425	.800
`SPECTER2` / `SPECTER`	.708	.915	.417	.784
`GTE`	.711	.925	.434	.816

representation learning, and our benchmarks confirm that it is the space with the lowest search performance (Table 8). However, as with D2L, the space is simply not structured to be used directly with cosine similarity. The information useful for search is still contained within the space. In fact, applying post-adaptation, as with D2L, improves performance by an average of $+.178$ and results in performance on par with D2L. Across the 14 benchmarks, the pairwise difference between the two adapted spaces averages $-.007$, with the maximum difference in either direction being .056 (Physics collaboration prediction, `ICAE` ahead). Therefore, once both compression spaces have g_θ, the two spaces exhibit virtually equivalent performance in terms of retrieval scores. What distinguishes the two is their decoding behavior: `ICAE` reproduces a higher proportion of the source text (Fig. 2f), and its cluster labels underperform D2L with a score of .60 in the blind evaluation (Section 3.1).

J Full result tables

This appendix collects the tables that the preceding appendices read against. Every table is generated by the workflow from the same caches as the figures in the main text.

Table 9 repeats the field tasks for the three base models of Appendix G. Table 10 splits the collaboration AUC of Table 2 into its anchor windows. Table 11 is the `vec2text` magnitude sweep of Appendix D.3. Table 12 closes the appendix with the native output of every embedding method for every node of the PACS slice scored in Appendix D.

Table 9: Per-encoder field-task matrix. Each cell reports the two D2L variants as raw / +g_θ (the frozen embedding and the single general citation transform), with metrics, evaluation pools, and bootstrap protocol as in Table 2 (standard deviations omitted). Text-baseline scores are encoder-independent and given in Table 2.

Encoder	Field	Next-paper (AUC)	Topic (F1)	Collaboration (AUC)
Gemma-2-2b	Economics	.792 / .888	.217 / .317	.592 / .617
	Psychology	.786 / .910	.180 / .319	.593 / .673
	Physics	.864 / .946	.569 / .582	.827 / .823
Qwen3-4B	Economics	.813 / .910	.242 / .374	.596 / .629
	Psychology	.799 / .923	.217 / .340	.602 / .687
	Physics	.880 / .958	.590 / .590	.792 / .795
Mistral-7B	Economics	.825 / .913	.234 / .368	.599 / .639
	Psychology	.815 / .926	.217 / .348	.601 / .681
	Physics	.876 / .965	.591 / .595	.820 / .781

Table 10: AUC-ROC of collaboration prediction for each anchor year. At an anchor year, each author is represented by the mean embedding of their papers from the previous four years. Candidate pairs are authors two steps apart in the coauthorship network who have not published together. A candidate pair is positive if the two authors coauthor a paper within the three years after the anchor year. Each column scores the candidate pairs of one anchor year. The *pooled* column scores the candidate pairs of all anchor years together and is the number reported in Table 2. D2L is the Qwen3-4B embedding with and without g_θ (Appendix G).

method	Physics			Economics				Psychology			
	2000	2008	pooled	2008	2012	2016	pooled	2008	2012	2016	pooled
D2L (raw)	.786	.848	.792	.646	.556	.579	.596	.581	.610	.627	.602
$+g_\theta$	.784	.849	.795	.680	.594	.603	.628	.672	.721	.678	.688
SPECTER2	.788	.857	.826	.672	.593	.621	.632	.651	.679	.676	.665
SBERT	.826	.887	.832	.696	.626	.637	.655	.691	.718	.667	.691
Instructor	.831	.843	.854	.676	.577	.597	.620	.645	.657	.648	.649
EmbeddingGemma	.798	.908	.829	.684	.596	.612	.633	.663	.688	.652	.667
GTE	.814	.825	.838	.670	.593	.610	.627	.665	.682	.659	.668

Table 12: Cluster labels for every node of a four-branch PACS slice. Each cell is the method's native output, exactly the string scored in Table 4, truncated at a word boundary for the page but scored in full. D2L and ICAE answer the field prompt; KeyLLM returns a keyword list and vec2text a reconstruction, which is why only the first two columns read as field names. The first column gives the PACS code and a short name, indented by tree depth, with field rows in bold.

	Native output (as scored)			
PACS node	**D2L**	**KeyLLM**	**ICAE**	**vec2text**
0 General	**Quantum field theory**	partial decoherence, thermalization, time-domain …	Quantum many-body systems	coupled-dissonance correlations in one-way systems. The …
03 Quantum mechanics, QFT	**Quantum optics**	quantum state transfer, entanglement, four qubits …	Quantum optics	coupled entanglement of one-state quantums, where the …
03.65 Quantum mechanics	**Quantum mechanics**	quantum decoherence, single qubit, random matrix theory …	Quantum mechanics	as an empirical measure of the interpolation of one-state …
03.67 Quantum information	**Quantum information**	Measurement-based quantum computing, AKLT state, spin-1 …	Quantum information	entanglement of single-state quantum computations, so …
05 Statistical physics	**Statistical physics**	Disturbance spreading, Incommensurate systems …	Statistical mechanics	diffuse-phase dynamics in univariate quantization. This …
05.40 Fluctuations, noise	**Statistical physics**	bistable systems, colored noise, stochastic relaxation …	Statistical physics	diffusion-display coefficients in one-variable …
05.45 Nonlinear dynamics, chaos	**Nonlinear dynamics**	separatrix map, resonant dynamics, global chaos onset …	Nonlinear dynamics	coupled chaotic quantization in classical dynamics. This …

	Native output (as scored)			
PACS node	**D2L**	**KeyLLM**	**ICAE**	**vec2text**
2 Classical physics	**Nonlinear optics**	Nonlinear light pulse propagation, Polarized light …	Quantum optics	multi-mode scattering of neutrons via classical optical …
42 Optics	**Quantum optics**	Photon antibunching, Photonic molecule, Coupled cavities …	Quantum optics	coupled photon-diametric scattering of neutrons in …
42.50 Quantum optics	**Quantum optics**	mesoscopic states, cavity-field state, atom-field …	Quantum optics	two-level physics involving scattering of coherent atoms …
42.65 Nonlinear optics	**Nonlinear optics**	solitons, spontaneous symmetry breaking, photonic …	Nonlinear optics	nonsimultaneous low-wave propagation in coherent optical …
47 Fluid dynamics	**Fluid dynamics**	Two-dimensional turbulence, freely decaying turbulence …	Fluid dynamics	fluid-displacement dynamics in experimentally correlated …
47.20 Flow instabilities	**Fluid dynamics**	Bifurcation diversity, Thermocapillary liquid layers …	Fluid dynamics	in-convection instability in linear-wavelength fluid …
47.27 Turbulent flows	**Fluid dynamics**	Mean-field approximation, turbulence theory …	Turbulence theory	in turbulence-displacement dynamics, the initial …
3 Condensed matter	**Condensed matter physics**	Quasi-one-dimensional electron systems, t-J model …	Condensed matter physics	in-distance interactions between electron-bonding …
71 Electronic structure	**Condensed matter physics**	semiclassical action, dynamical mean-field theory, local …	Condensed matter physics	phase-dimersion configuration of classical electron-fluid …
71.10 Many-electron systems	**Condensed matter physics**	Mott physics, Mott transition, spin fluctuations …	Condensed Matter Physics	one-dimensional quasi-dotonal model where the Hubbard …
71.20 Bulk band structure	**Condensed matter physics**	Electronic structure, Pyrochlore metals, Cd2Os2O7 …	Materials science	(dBa)-structure calculations of the respective metallic …
75 Magnetic materials	**Condensed matter physics**	Quantum Monte Carlo, sign problem, itinerant …	Condensed Matter Physics	inferromagnetic spin-diabetic scattering interactions …
75.10 Magnetic ordering	**Condensed matter physics**	square-lattice Heisenberg antiferromagnet, S=1/2 …	Condensed Matter Physics	in anastomosing-phase-spatial-inverse quantum state, with …
75.30 Ordered magnets	**Condensed matter physics**	metal-insulator transition, La1-xCaxMnO3, manganites …	Magnetism	inferomorphic magnetic spin-phases at 0–0,0–0,0–0,0–0,0–0

	Native output (as scored)			
PACS node	**D2L**	**KeyLLM**	**ICAE**	**vec2text**
6 Elementary particles	**Particle physics**	Left-right symmetry, LHC, SU(2)_L x SU(2)_R x U(1)_B-L …	High Energy Physics	scattering of resonant singularity quarks on the …
11 Fields and particles	**Quantum field theory**	light-front Hamiltonian, spontaneous chiral symmetry …	Quantum Field Theory	with the symmetries of fermion quantum diffusion. This …
11.10 Field theory	**Quantum field theory**	sine-Gordon field theory, massive, (1+1)-dimensional …	Quantum field theory	symmetries of the classical quantization-dynamic theory …
11.15 Gauge field theories	**Quantum field theory**	SU(2) gauge theory, adjoint Dirac flavor, infrared …	Quantum field theory	scattering of d-quantity quantums. This d-quantity …
12 Particle systematics	**Particle physics**	Left-Right symmetry, LHC, SU(2)_L x SU(2)_R x U(1)_B-L …	High Energy Physics	scattering of low-mass quark quantities. This approach …
12.38 Quantum chromodynamics	**Quantum chromodynamics**	Nucleon strange quark content, Lattice QCD, Two-flavor …	Nuclear physics	with d-quantity scattering of d-quantity laton particles …
12.60 Beyond standard model	**Particle physics**	SO(10), supersymmetry breaking, gauge mediation, grand …	Particle physics	high scattering-symmetry sbN, the experimental model will …

Table 11: `vec2text` magnitude sweep on the same five seed documents. `vec2text` is an inverter and takes no prompt, so we read back its reconstruction directly. The reconstruction stays on topic at the same surface specificity for larger α and degrades into off-topic fragments at small α, never abstracting toward the broader field. Entries truncated to leading terms.

α	**Physics**	**Biochemistry**	**Fruit**	**Person**	**Patent (LED)**
1.15	is the fractional quantum effect of accumulating electrons in certain atoms. …	(abbreviated CY), commonly ATP, is an enzyme cycle for the production …	a banana is an edible plantation fruit, traditionally specializing in long, …	Nikola Tesla, an American engineer from Serbia, was an 1860s American …	comprised an ion-doped light-emitting semiconductor forming a nitride structure, nitrid
0.9	quantum fraction is a function of the Hall effect in physics: …	enzyme cycle is the metabolic use of ATP for generating cryptic …	fruit is a elongated banana. This is due to the variety …	Nikola Tesla was a Serbian engineer and American engineer, known for …	structure comprised a light-emitting compound of a double-type nitride semiconductor (e.g.
0.7	quantum mechanics is the fraction of the Hall effect which is …	acids used for the production of glycogen. The Krebs cycle is …	plant species is a banana. The name derives from a linguistic …	engineering genius Nikola Tesla is a Serbian engineer who contributed to …	structure of a double-layered nickel semiconductor (indicated in the introduction of …
0.55	equation for the physics of fractional heating is a contribution to …	the CYCY process of carbohydrates is a way of generating ATP, …	botanical designation of a plant. A banana is a combination of …	American mechanical engineer. Nikola Tesla is credited for the production of …	layer of a g-type nickel plating (indicated in the following table: …
0.42	equation for quantum mechanics is a result of the publication of …	the metabolic cycle of enzymes is a contribution to the aforementioned …	botanical name for a banana. This is a variant of the …	American electrical engineer. Nikola Tesla is credited for the contribution of …	a GLTI semiconductor pairing of the following structures: (a) (initiated) (a) …
0.3	a number of awards in the field of physics	biology is a key component of the Crypt of the Chemical …	biology of plants is a classic in the discipline of banana …	American electrical engineer who was a major contributor to the creation …	a layered semiconductor for the neo-Neo-Geo-Neo-Neo-Dai
0.2	a prestigious award in the field of physics for the 1860 …	biology is a classic in the vein of the MIT CYPATHE, …	classic in Asian agriculture. The title of a chapter is a …	American mechanical engineer. Tesla contributed to a number of awards including …	a DP-type semiconductor engineering is
0.12	a prestigious journal in the area of physics entitled	a.k.a. The Chemical Cycle of the As - a -does it …	classic in the history of the Venus Project	a history of Tesla engineering. Contributor to the 97th edition of …	a fusion of the following disciplines
0.06	classic in the history of physics. 109.	a.k.a. 179. 2004: The Academy of Sciences published a review of …	classic in the history of engineering. William Rice Publishers	1925. The publication of a book on the history of electricity …	a.k.a. DP Engineering