Beyond Plain Demos: A Demo-centric Anchoring Paradigm for In-Context Learning in Alzheimer's Disease Detection

Puzhen Su, Haoran Yin, Yongzhu Miao, Jintao Tang*, Shasha Li*, Ting Wang*

College of Computer Science and Technology, National University of Defense Technology, Changsha, China
{supuzhen, yinhaoran, miaoyz, tangjintao, shashali, tingwang}@nudt.edu.cn

Abstract

Detecting Alzheimer's disease (AD) from narrative transcripts challenges large language models (LLMs): pre-training rarely covers this out-of-distribution task, and all transcript demos describe the same scene, producing highly homogeneous contexts. These factors cripple both the model's built-in task knowledge (**task cognition**) and its ability to surface subtle, class-discriminative cues (**contextual perception**). Because cognition is fixed after pre-training, improving in-context learning (ICL) for AD detection hinges on enriching perception through better demonstration (demo) sets. We demonstrate that standard ICL quickly saturates, its demos lack diversity (context width) and fail to convey fine-grained signals (context depth), and that recent task vector (TV) approaches improve broad task adaptation by injecting TV into the LLMs' hidden states (HSs), they are ill-suited for AD detection due to the mismatch of injection granularity, strength and position. To address these bottlenecks, we introduce **DA4ICL**, a demo-centric anchoring framework that jointly expands context width via ***Diverse and Contrastive Retrieval*** (DCR) and deepens each demo's signal via ***Projected Vector Anchoring*** (PVA) at every Transformer layer. Across three AD benchmarks, DA4ICL achieves large, stable gains over both ICL and TV baselines, charting a new paradigm for fine-grained, OOD and low-resource LLM adaptation.

Code — https://github.com/Eneverg1veup/DA4ICL
Datasets — https://talkbank.org/dementia
Extended version — https://arxiv.org/abs/2511.06826

Introduction

Large language models (LLMs), trained on massive textual corpora, have exhibited impressive adaptability across diverse downstream tasks. Among various adaptation paradigms, in-context learning (ICL) has proven especially effective in low-resource settings (Sun et al. 2023), enabling LLMs to perform new tasks by conditioning on a handful of demonstrations (demos), without any parameter updates (Luo et al. 2024). The effectiveness of ICL fundamentally depends on the *task-related knowledge* acquired by LLMs during pre-training (**cognition**), and the *text-to-label pattern* presented in the demos (**perception**), which together enable the model to infer task objectives and generalize from contextual cues (Zhao et al. 2024). In practice, the model's task cognition is constrained by pre-training exposure (Yu and Ananiadou 2024), making the construction (*retrieval strategies*) and utilization (*inference strategies*) of high-quality demo set critical for effective task perception. In this work, we view the core challenge of ICL as ***how to optimally assemble limited demos to approximate the true task distribution, aggregating complementary task cues to maximize LLMs' perception***.

Despite their remarkable adaptability, current ICL frameworks still encounter fundamental challenges in several practical scenarios, particularly in tasks characterized by low-resource conditions (Srinivasan et al. 2024), out-of-distribution (OOD) shifts (Lin et al. 2020), and vague inter-class difference. A prime and societally significant example of such task is the early detection of Alzheimer's disease (AD) from narrative speech or text (Roark et al. 2011). AD is a devastating neurodegenerative condition that affects millions worldwide with enormous personal and societal costs (Deture and Dickson 2019; Kelley and Petersen 2007). Critically, AD is currently incurable at advanced stages, making early and reliable detection essential for timely intervention and care. AD detection (Roshanzamir, Aghajan, and Soleymani Baghshah 2021) requires classifying the cognitive status (AD or healthy control, HC) of participants (PARs) based on their picture description transcripts. Yet, there are two parallel and intertwined properties that weaken both the cognition and perception of LLMs to achieve effective task adaptation. For **task cognition** (limited), due to privacy concerns and collection costs, AD detection has **extremely scarce and closed-source** datasets, making it a typical OOD task with minimal pre-training exposure. For **task perception** (poor), as all PARs describe the same scene, there exists **pervasive semantic homogeneity** caused by highly similar transcripts even across classes, contributing to weak and ambiguous text-to-label patterns in given demos. Moreover, conventional ICL retrieval strategies are typically limited to a single aspect, most commonly semantic similarity (Liu et al. 2022; Lyu et al. 2022), resulting in demo sets that lack sufficient diversity and fail to provide the discriminative cues required for accurate AD detection. Similarly,

*Corresponding author.

existing inference strategies, like ensemble voting (Su et al. 2025; Hong et al. 2024; Yu et al. 2024), calibration (Abbas et al. 2024), are only effective when ***task-related knowledge is sufficient or the demo set can optimally simulate the underlying task distribution***—a condition rarely satisfied in AD scenario. As a result, robust task adaptation remains out of reach for current ICL methods, motivating the need for more expressive retrieval and demo enrichment mechanisms.

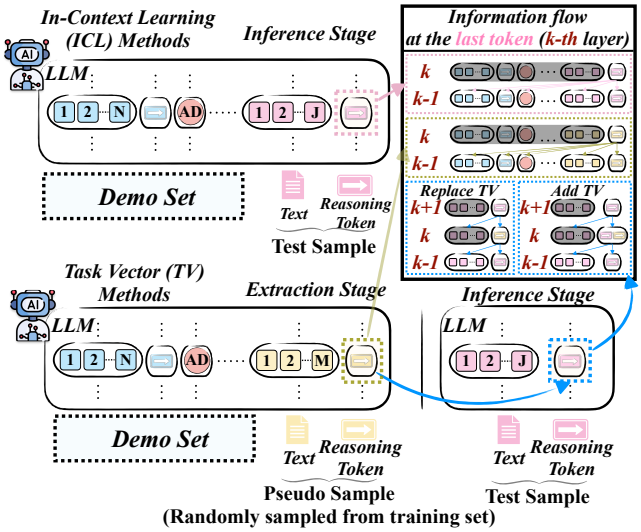

Figure 1: Schematic comparison of information flow and token processing in standard ICL versus TV methods.

Recent works (Hendel, Geva, and Globerson 2023) introduced Task Vector (TV) as a rapid task adaptation approach, which injects latent TV into the hidden states (HSs) of the test sample (at reasoning token, i.e., $\rightarrow$). TV methods typically split ICL into two stages: **1) extraction**, where a demo set is concatenated with a randomly selected pseudo sample and the last $\rightarrow$ token's HS is extracted as the TV, and **2) inference**, where the label for a test sample is predicted by injecting this vector into its $\rightarrow$ token. While effective for generic tasks(Yang et al. 2025), TV methods present two fundamental limitations for fine-grained tasks such as AD detection. First, in ICL, each demo serves as an anchor (Wang et al. 2023; Pang et al. 2024), aggregating semantic information and guiding the final prediction. However, existing TV paradigms are fundamentally **test-sample-centric** (see Fig. 1), they inject adaptation signals at the final $\rightarrow$ token (i.e., in test sample), discarding the distributed information encoded in preceding demo anchors and instead relying on a single, pseudo-sample-derived TV. Such injection paradigm neglects context diversity and fine-grained cues which are crucial in tasks with subtle inter-class variation. Second, most TV methods inject TVs via **addition or replacement**, directly altering both the ***direction*** and ***magnitude*** of the HS, which may risk distorting the original semantic information and introduce instability or bias. Consequently, ***existing TV approaches often fail to deliver reliable performance on OOD, low-resource, and fine-grained tasks like AD detection***.

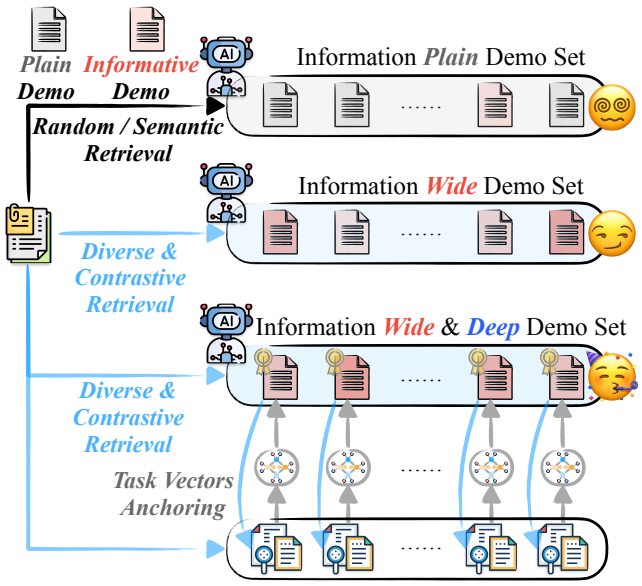

Figure 2: Progressive enrichment of demo sets, from plain to wide and deep, drives more effective and robust in-context reasoning.

To overcome these limitations of both ICL and TV paradigms, we propose *Demo-centric Anchoring for In-Context Learning* (DA4ICL), a paradigm shift that rethinks how demo sets are constructed and how task information is integrated. Our approach is driven by two core motivations: (1) ***maximizing demo context width by diversifying and contrasting the information available to LLM***, and (2) ***enhancing context depth by reinforcing each demo with fine-grained and demo-centric signals***. As shown in Fig. 2, DA4ICL first employs a *Diverse and Contrastive Retrieval* (DCR) strategy to construct a context-wide *main-demo* set, selecting demos from two complementary perspectives (i.e., semantic, structural), and broadening the contextual cues available for adaptation. Next, for each *main-demo*, a second-stage retrieval process identifies a set of *sub-demos*, enabling us to extract detailed, demo-specific TVs. We then introduce *Projected Vector Anchoring* (PVA) mechanism, which projects these fine-grained TVs into corresponding $\rightarrow$ tokens of each *main-demo* across all Transformer layers, treating each demo as an anchor for subsequent reasoning. Unlike previous test-centric TV approaches, DA4ICL leaves the test sample token unmodified. During inference, the LLM naturally aggregates information from all enriched demo anchors through masked self-attention, thereby enhancing both stability and precision. By shifting from a single, test-centric injection to a distributed, demo-centric anchoring paradigm, DA4ICL substantially increases the diversity and discriminative value of context available to the LLM. This enables LLM to leverage more nuanced cues and improves task perception by addressing both context width and depth bottlenecks. Experimental results across three AD detection datasets confirm that our method consis-

tently and significantly outperforms both conventional ICL and TV baselines, establishing a new paradigm for effective adaptation in challenging NLP tasks. Our main contributions are:

- We propose a demo-centric anchoring paradigm that shifts task information integration from the test token to each demo anchor, enhancing fine-grained in-context reasoning.
- We introduce a diverse and contrastive retrieval strategy (DCR) to maximize context width, capturing multi-dimensional and complementary contextual cues for robust adaptation.
- We design a projection-based, layer-wise anchoring mechanism (PVA) that deepens context integration by injecting fine-grained task vectors into demo anchors without distorting original semantics.

Related Works

In-Context Learning Methods. Recent studies (Zhao et al. 2024; Yu and Ananiadou 2024; Kossen, Gal, and Rainforth 2024) have highlighted two aspects determining ICL effectiveness in LLMs: the recognition of task objectives (**task cognition**) and leveraging relevant contextual cues (**task perception**). Task cognition, defined as the latent task-specific knowledge acquired during pre-training, fundamentally constrains ICL adaptation, when tasks lack sufficient pre-training exposure, models often default to superficial label copying from semantically similar demos (Ali, Wolf, and Titov 2024). Furthermore, in view of information flow, Transformer models heavily rely on aggregating semantic signals at the final token's HS for inference (Wang et al. 2023; Pang et al. 2024), making the diversity and informativeness of demos critical for effective adaptation. To address these limitations, previous works primarily explored retrieval strategies based on semantic similarity (Liu et al. 2022; Lyu et al. 2022) and ensemble-based inference (Khalifa et al. 2023; Mojarradi et al. 2024). However, these approaches remain fundamentally constrained in OOD and fine-grained tasks like AD detection, where semantically homogeneous inputs and subtle class distinctions exacerbate the bottlenecks of existing ICL paradigms on AD detection (Balamurali and Chen 2024; Li et al. 2025a).

Task Vector Methods. TV methods (Merullo, Eickhoff, and Pavlick 2024) have recently emerged as an alternative to ICL, by injecting a task-specific vector directly into LLM's HSs. In these approaches (Yang et al. 2025), an LLM's few-shot demos are first compressed into a single TV, typically by extracting the HS at the reasoning token (i.e. the final separator token $\rightarrow$ in a prompt that maps inputs to outputs). This TV is then injected at the corresponding position during test sample's forward pass. While effective for broad tasks (Li et al. 2024; Todd et al. 2024), TV methods face fundamental limitations in low-resource, OOD, and fine-grained settings like AD detection. First, existing TV paradigms are based on single-layer, last-token injection (Dong et al. 2025), assuming the extracted TV captures the full task context, which is rarely satisfied for tasks that require subtle or distributed cues. Pseudo-sample overfitting (Hendel, Geva, and Globerson 2023) is another key issue, where TV injection can overfit to the idiosyncrasies of the pseudo-sample, neglecting important distinctions. This is particularly problematic for AD detection, where nuanced inter-class variations are crucial. Second, TV injection typically involves addition (Liu et al. 2024a) or replacement (Liu et al. 2024b), modifying both the ***direction*** and ***magnitude*** of the HS. This often leads to semantic misalignment, distorting the original meaning and introducing instability or bias, especially when the injected TV is poorly aligned with the context. These challenges underscore the need for fine-grained, demo-aware, and projecting-based injection strategies.

Methodology

Preliminary: Why Test-centric TV Injection Fails. In decoder-only LLMs, each token's HS is updated layer by layer via residual connections (Li et al. 2025b; Saglam et al. 2024):

$$h_t^{(\ell)} = h_t^{(\ell-1)} + \mathrm{mha}(h_t^{(\ell-1)}) + \mathrm{mlp}(h_t^{(\ell-1)} + \mathrm{mha}(h_t^{(\ell-1)})), \tag{1}$$

where mha and mlp denote multi-head attention and feed-forward modules. This recursive structure enables each layer to integrate contextual signals from preceding tokens and layers. At inference, the LLM predicts the next token x_{t+1} via:

$$P(x_{t+1} \mid x_{\leq t}) = \mathrm{softmax}(\mathbf{W}_{\mathrm{LM}} \cdot h_t^{(L)}), \tag{2}$$

where $\mathbf{W}_{\mathrm{LM}}$ is the unembedding matrix that maps the last token's final-layer HS $h_t^{(L)}$ to vocabulary logits. This encourages existing TV methods to inject TVs into the test $\rightarrow$ token at a single layer. However, this *test-centric* injection is coarse-grained and prone to overfitting in fine-grained tasks like AD detection. Since the $\rightarrow$ tokens in demos act as local anchors encoding class-specific cues, injecting signals into them can further enhance demo representation, but shallow-layer and signals often dissipate through the residual stream. To preserve their influence, TVs must be anchored across all layers. These insights motivate our **demo-centric, full-layer anchoring** design.

DA4ICL Framework. To address the intrinsic information bottleneck and granularity misalignment inherent in standard ICL and existing TV methods, we propose **DA4ICL**, an enhanced ICL framework combining a novel retrieval strategy and a refined TV anchoring mechanism. Our DA4ICL (see Fig. 3) introduces two modules: (1) ***Diverse and Contrastive Retrieval*** (DCR) to enrich demo sets along multiple complementary dimensions, and (2) ***Projected Vector Anchoring*** (PVA) to anchor fine-grained, layer-wise TVs at demo-level $\rightarrow$ tokens. The DCR module mitigates the insufficient context width caused by conventional retrieval, while the PVA module realigns the injection granularity of TVs from test-centric to demo-centric anchoring, ensuring robust and context-deep adaptation for the AD detection task.

Diverse and Contrastive Retrieval (DCR)

The DCR module aims to construct informative and contextually diverse demo sets in two sequential stages.

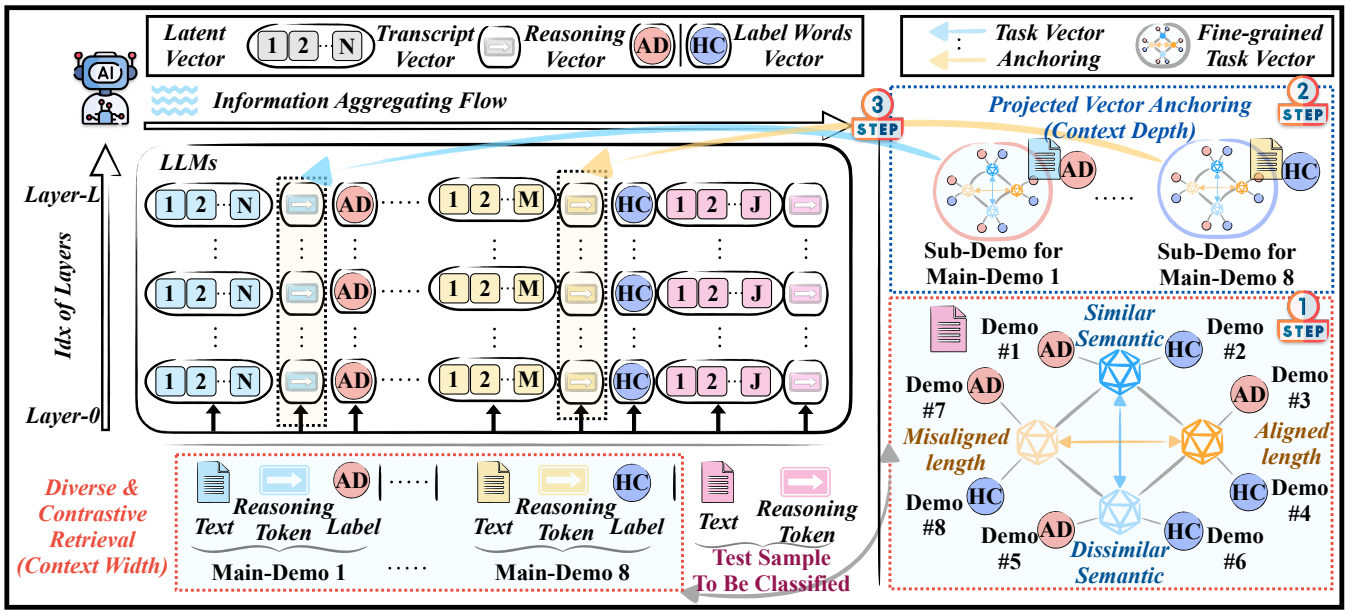

Figure 3: Overview of the DA4ICL framework. Diverse and contrastive demos are selected and enriched via projected vector anchoring across all Transformer layers to provide both wide and deep context for robust AD detection.

Stage 1: Main-Demo Set Construction (Width Enrichment). For each test sample d_{test}, we construct a *main-demo* set by retrieving a pair of AD/HC demos under each of four complementary criteria: (i) semantic similarity, (ii) semantic dissimilarity, (iii) length similarity, and (iv) length dissimilarity. Let $\phi(d)$ represent the final-layer HS at the last $\rightarrow$ token of demo d, and $\ell(d)$ denote its sequence length. Formally, for each criterion $c \in \{sim\phi, dis\phi, sim\ell, dis\ell\}$, we select:

$$d_+^c = \underset{(x,y)\in\mathcal{S},\, y=AD}{\text{argext}} \; f_c(d_{\text{test}}, (x,y)),$$
$$d_-^c = \underset{(x,y)\in\mathcal{S},\, y=HC}{\text{argext}} \; f_c(d_{\text{test}}, (x,y)), \tag{3}$$

where ext is max or min according to the criterion c, and f_c denotes cosine similarity or length difference accordingly. The resulting *main-demo* set $\mathcal{D}_{\text{main}}$ comprises eight demos:

$$\mathcal{D}_{\text{main}} = \{d_+^c, d_-^c \mid c \in \{sim\phi, dis\phi, sim\ell, dis\ell\}\}. \tag{4}$$

Stage 2: Sub-Demo Set Construction (Depth Enrichment). To deepen and enrich the context around each *main-demo* $d_i = (x_i, y_i)$ ($d_i \in \mathcal{D}_{\text{main}}$), we further retrieve a set of 8 *sub-demos* using the same four criteria but centered on d_i itself. The resulting augmented sequence Seq_i for each main demo d_i is:

$$\text{Seq}_i^{sub} = \{x_{i,1} \rightarrow y_{i,1}, \ldots, x_{i,8} \rightarrow y_{i,8}, x_i \rightarrow\}, \tag{5}$$

where $\{(x_{i,j}, y_{i,j})\}_{j=1}^{8}$ are the *sub-demo* set (pre-computed) providing rich contrast and contextual detail. Crucially, each *main-demo* serves as its own pseudo-sample for subsequent TV extraction, circumventing external pseudo-sample selection and enhancing representation consistency.

Projected Vector Anchoring (PVA)

Existing TV injection methods suffer from two key limitations: (1) *they inject at only the test sample's $\rightarrow$ token (test-centric), neglecting distributed demo cues*, and (2) *they use addition or replacement, which distorts both the direction and magnitude of HSs, risking semantic misalignment*. To address these issues, we propose Projected Vector Anchoring (PVA), a demo-centric, layer-wise, and projection-based mechanism that ensures every demo anchor mostly contributes to the LLM's reasoning at all depths.

Task Vector Extraction. In decoder-only LLMs, the HS at the last token and last layer overwhelmingly determines next-token prediction, due to the LLM's architecture. Signals injected at earlier positions can easily be washed out during forward propagation.

Therefore, to ensure that every demo anchor robustly contributes to the test prediction, we anchor TVs at every layer, making their influence persistent and cumulative throughout the residual stream. For each main demo d_i, we expand it with *sub-demo* retrieval (see Eq. 5), and extract the HS at its $\rightarrow$ token t_i of all layers:

$$\mathbf{v}i^{(\ell)} = \mathbf{h}_{i,t_i}^{(\ell)}, \quad \ell = 1, \ldots, L, \tag{6}$$

where tokens in the $t_i - 1$ and t_i-th position of Seq_i^{sub} are $\{x_i, \rightarrow\}$, where $\{x_i, \rightarrow\} \in \text{Seq}_i^{sub}, x_i \in \mathcal{D}_{\text{main}}$.

Projected and Layer-wise Anchoring. At inference, for each *main-demo* d_i and each layer ℓ, we refine its HS by projecting the extracted TV onto the original representation, rather than naive addition or replacement, to ensure semantic alignment and robust adaptation.

First, we normalize the extracted TV to match the scale of the layer's typical HSs:

$$\bar{v}_i^{(\ell)} = \mu^{(\ell)} \cdot \frac{v_i^{(\ell)}}{|v_i^{(\ell)}|_2} \tag{7}$$

where $\mu^{(\ell)}$ is the average ℓ_2-norm of HSs at layer ℓ. This step ensures the injected signal is calibrated to the expected scale of LLM HSs. Next, we compute the projection of the normalized TV onto the original HS direction:

$$p_i^{(\ell)} = \frac{\langle \bar{v}_i^{(\ell)}, h_{i,t_i}^{(\ell)} \rangle}{|h_{i,t_i}^{(\ell)}|_2^2 + \epsilon} \cdot h_{i,t_i}^{(\ell)} \tag{8}$$

This operation preserves the original semantic direction of the *main-demo* anchor, modulating only its magnitude, and thereby avoids distorting semantic. Finally, we update the HS with a layer-specific scaling factor:

$$h_{i,t_i}^{(\ell)\prime} = h_{i,t_i}^{(\ell)} + \gamma^{(\ell)} \cdot p_i^{(\ell)}. \tag{9}$$

This flexible, projection-based anchoring ensures that demo-specific task information is injected in a manner that is both fine-grained and semantically consistent, leading to more stable and interpretable adaptation. More detailed mechanisms are listed in **Appendix: C**.

Experiments

Experimental Setup

Our experiments are structured to progressively analyze how DA4ICL overcomes the key challenges in AD detection, which are, limited task perception, insufficient demo diversity, and mismatched injection granularity.

We begin by benchmarking DA4ICL against ICL and TV baselines under varied retrieval and inference settings, revealing the performance saturation of ICL and the ineffectiveness of test-centric TV injection. We then isolate the role of demo diversity by applying our retrieval strategy (DCR) to both DA4ICL and existing ICL/TV pipelines, confirming its universal benefit for ICL but limited utility for conventional TV methods. Finally, we investigate why TV fails to leverage DCR, demonstrating that our projection-based, multi-layer anchoring (PVA) is essential to effectively inject fine-grained task signals. ***Together, these experiments explain both the failure modes of previous methods and the mechanisms behind DA4ICL's improvements.***

Datasets. We perform experiments on three widely-adopted AD corpora: the *ADReSS Challenge dataset* (Luz et al. 2020), the *Lu corpus* (Lanzi et al. 2023), and the *Pitt corpus* (Becker et al. 1994). The *ADReSS Train* split contains 54 AD and 54 HC (*Test* split: 24 vs. 24), providing a balanced in-distribution benchmark. In contrast, the *Lu corpus* comprises 15 AD vs. 27 HC, and the *Pitt corpus* comprises 243 AD vs. 306 HC, introducing pronounced class imbalance. For our experiments, every demo is drawn exclusively from ADReSS Train split, with the rest corpora strictly held out for evaluation. All transcripts follow the standard CHAT protocol (MacWhinney 2000), ensuring consistent preprocessing and enabling direct cross-corpus comparison. More details can be found in **Appendix: A**.

Baselines. We compare our method comprehensively against two main families of strong LLM-based baselines: **ICL** methods and **TV** methods. Within the ICL family, we evaluate: (1) *Vanilla ICL*, which randomly selects demonstrations from the support set without considering relevance, (2) *Semantic ICL* (Liu et al. 2022), which retrieves demos semantically closest to the test sample, measured via cosine similarity in latent space, and (3) *Ensemble ICL* (Hong et al. 2024), which aggregates predictions from multiple independently sampled or retrieved demonstration sets through majority voting, aiming to enhance prediction robustness. Collectively, these ICL variants examine both the construction of demonstration contexts and the inference mechanisms within conventional ICL paradigms. For the TV family, we evaluate methods that directly manipulate HSs to inject latent task representations. Specifically, we test two core injection approaches: (4) *Replace TV* (Hendel, Geva, and Globerson 2023), which replaces the HS at final $\rightarrow$ position directly with a TV derived from demos, and (5) *Add TV* (Liu et al. 2024a), which injects the TV into the original HS via addition. Both methods consider two variants in demos retrieval for extraction: random retrieval and semantic similarity-based retrieval. Thus, these TV baselines thoroughly explore the mechanisms (replacement vs. addition) and the content (random vs. semantic) of latent task signal injection, allowing precise comparisons to our proposed demo-centric anchoring strategy.

Implementation Details. We implement all methods using the `Llama3.1-8B-instruct` model as the backbone to ensure consistent and fair comparisons, and conduct experiments on a single NVIDIA Quadro RTX 8000 (48GB GPU). To stabilize outputs, we set temperature of 0.1 and top-k of 50 across all experiments. For demo retrieval, we uniformly sample demos in pairs (AD/HC per pair) and evaluate performance under varying demo pair counts. For TV baselines, TVs are consistently extracted and injected at the final HS layer at the last token ($\rightarrow$) of the input sequence. The injection strength parameter is set as 0.5 for Add TV. For DA4ICL, the $\gamma(\ell)$ is set as 1 for $\ell \in [0, 7] \cup [24, 31]$, and 0.2 for $\ell \in [8, 23]$. More implementation details are listed in **Appendix: B**, and we conduct several supplementary experiments, which can be found in **Appendix: D**.

Evaluation Metrics. We evaluate performance comprehensively using accuracy and F1-score. All experimental results are averaged over 10 independent runs under identical conditions to ensure statistical robustness and reliability.

Main Results

Tab. 1 presents the comparative results of DA4ICL and a comprehensive set of baselines, including vanilla, semantic, ensemble ICL, and two representative TV injection methods on three AD detection datasets.

LLM's task cognition is insufficient for AD detection. The zero-shot results (ICL_{Van}, N=0) are consistently poor across all datasets (e.g., Acc 57.71% on Test/Lu and 55.72% on Pitt), directly confirming that LLMs lack the necessary task cognition for AD detection in the absence of demos.

Limited gains from current ICL strategies. Increasing the N or switching from random to semantic retrieval yields limited and fluctuated improvements. For instance, on Test, accuracy rises only from 67.50% ($N = 1$) to 70.00% ($N = 4$), from 70.00% (vanilla, $N = 4$) to 71.04% (semantic,

Dataset	Metric (%)	N	ICL_{Van}	ICL_{Sem}	ICL_{Ens}	$\text{TV}^{\text{Add}}_{\text{Van}}$	$\text{TV}^{\text{Add}}_{\text{Sem}}$	$\text{TV}^{\text{Rep}}_{\text{Van}}$	$\text{TV}^{\text{Rep}}_{\text{Sem}}$	Ours
Test	F1	0	51.50±7.34	-	54.71±7.20	-	-	-	-	-
		1	69.33±4.22	68.93±3.88	64.43±4.62	51.23±6.62	52.28±8.33	67.11±3.12	65.00±4.94	-
		2	68.29±3.71	68.70±5.14	71.42±3.71	54.60±5.77	56.02±8.84	68.40±4.27	64.46±2.86	-
		3	70.66±1.83	73.71±4.73	70.20±5.24	51.04±6.17	51.44±7.95	66.94±4.72	69.68±5.40	-
		4	72.13±4.15	74.20±3.69	<u>75.32±4.08</u>	54.47±5.21	56.31±7.99	66.97±3.96	64.15±4.65	**86.11**† ± 1.92
	Acc.	0	57.71±6.46	-	59.58±5.76	-	-	-	-	-
		1	67.50±5.12	66.46±4.00	62.50±5.19	54.58±5.88	57.08±5.12	62.92±3.46	61.46±5.76	-
		2	66.67±3.36	66.88±5.39	70.21±3.73	58.13±5.22	61.04±6.72	65.21±4.66	59.17±2.83	-
		3	68.33±2.43	71.25±5.65	68.96±5.31	55.62±6.79	57.08±4.86	62.08±4.82	65.42±5.45	-
		4	70.00±4.08	71.04±4.32	<u>71.46±5.44</u>	56.88±4.85	59.79±6.85	62.92±5.34	59.38±4.95	**85.83**† ± 1.91
Lu	F1	0	63.41±7.24	-	63.59±5.59	-	-	-	-	-
		1	76.45±3.03	77.89±2.69	77.71±3.99	61.10±7.41	64.79±6.62	75.31±4.16	74.10±3.48	-
		2	79.31±2.86	78.01±2.65	78.91±3.12	62.12±6.34	64.54±4.53	76.06±1.71	74.51±3.19	-
		3	79.83±4.10	78.81±2.54	80.44±3.24	58.42±6.72	61.55±5.31	76.75±3.43	73.77±3.54	-
		4	79.53±3.58	<u>82.02±2.11</u>	79.88±3.28	62.06±7.81	61.91±5.65	74.46±4.28	76.54±3.46	**86.12**† ± 1.73
	Acc.	0	55.71±8.05	-	56.19±5.65	-	-	-	-	-
		1	66.90±4.05	69.52±3.33	68.57±5.19	54.76±6.82	59.29±6.43	65.00±5.43	63.57±3.99	-
		2	70.95±4.10	69.05±3.53	70.00±4.29	55.48±6.03	57.38±4.32	66.43±2.49	63.57±4.27	-
		3	71.19±6.25	68.81±3.44	72.14±4.27	53.10±5.84	54.52±4.70	67.14±4.86	62.86±4.54	-
		4	70.95±5.08	<u>73.33±3.50</u>	70.95±4.97	55.48±8.11	55.71±5.65	63.57±5.33	66.43±5.05	**80.95**† ± 2.51
Pitt	F1	0	55.23±2.00	-	56.30±2.67	-	-	-	-	-
		1	67.18±1.11	67.59±1.21	67.41±1.89	55.33±2.24	54.71±1.72	66.92±1.21	67.00±0.72	-
		2	67.74±1.35	68.18±1.83	69.11±1.40	53.86±2.35	54.31±2.38	66.31±1.15	67.63±1.63	-
		3	68.81±1.40	71.70±0.83	69.44±0.93	55.10±2.33	54.22±2.65	67.10±1.44	66.99±1.23	-
		4	70.24±1.12	<u>73.79±1.21</u>	70.66±1.07	53.45±2.44	52.60±1.80	67.45±1.02	67.64±1.39	**80.23**† ± 0.42
	Acc.	0	55.72±1.96	-	56.92±2.21	-	-	-	-	-
		1	63.33±1.12	64.34±1.21	63.42±1.83	55.06±1.62	54.39±1.62	59.85±1.43	59.64±0.91	-
		2	64.63±1.15	65.39±1.79	66.01±1.59	53.92±2.11	53.83±2.44	59.03±1.27	60.55±1.82	-
		3	65.43±1.55	68.12±0.93	66.25±0.95	54.35±1.37	53.77±2.31	59.67±1.71	59.82±1.55	-
		4	66.48±1.20	<u>69.71±1.13</u>	67.41±1.10	53.57±2.09	53.01±1.71	60.27±1.06	60.44±1.38	**79.42**† ± 0.39

Table 1: Mean ± std of F1-score and Accuracy over 10 runs for ICL (Van, Sem, Ens), TV (Van^{Add}, Sem^{Add}, Van^{Rep}, Sem^{Rep}), and DA4ICL (Ours) on three AD detection datasets (Test, Lu, Pitt) with varying demo counts (N). Bold entries denote the best result, and underlined entries denote the second-best. Significance is shown with † for DA4ICL compared to second-best baselines. Statistical significance was measured with a paired t-test ($p < 0.005$) and a Wilcoxon signed-rank test ($p < 0.01$).

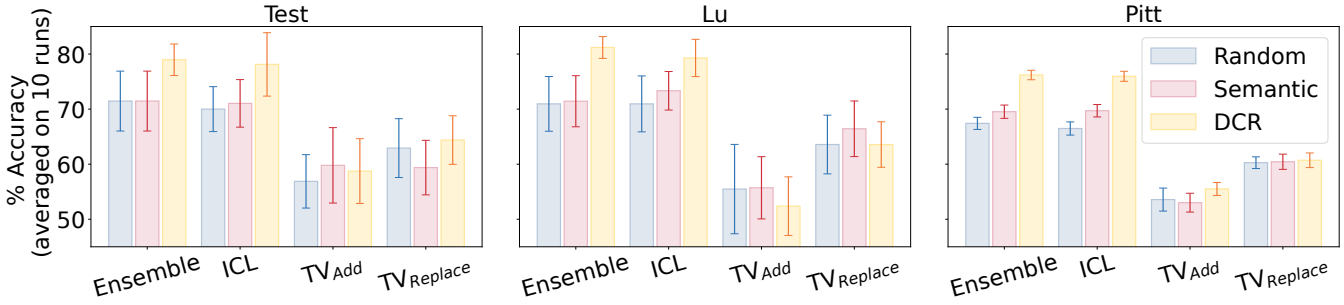

Figure 4: Accuracy comparison after applying DCR to ICL and TV methods (10 runs). DCR consistently improves ICL performance across all datasets, showing the value of DCR demo construction. However, DCR fails to enhance TV methods, with performance remaining stable or degrading, highlighting the limitations of TV's injection granularity and alignment.

$N = 4$), and F1 scores follow a similar trend. Notably, on the Lu dataset, both F1 scores and accuracy decrease when moving from ICL_{Van} to ICL_{Sem} ($N = 2, 3$), illustrating instability and lack of robustness. ICL_{Ens} offers minor gains (Test F1/Acc: 75.32/71.46%), further confirming the limited effect.

Conventional TV injection fails for AD adaptation. Both addition and replacement TV methods consistently un-

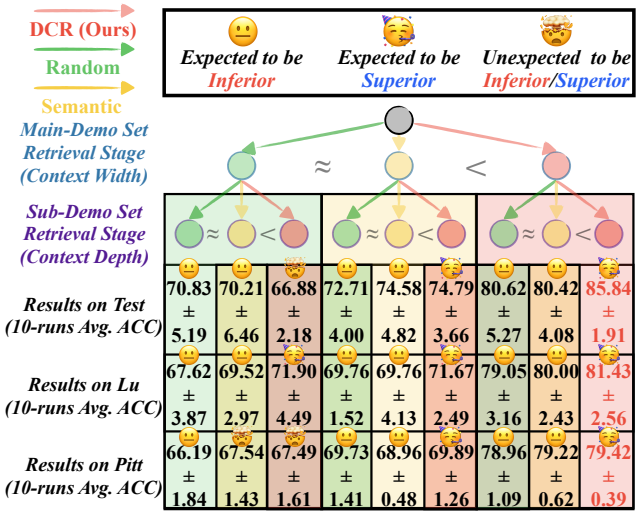

Figure 5: Ablation over two-stage retrieval strategies. Diverse main-demo selection (context width) provides the largest gains, while sub-demo enrichment (context depth) offers further complementary improvements.

derperform ICL baselines, regardless of retrieval or injection strategy across dataset. For example, TV_{Van}^{Add} and TV_{Sem}^{Add} achieve only 56.31/54.47% F1 and TV_{Van}^{Rep} and TV_{Sem}^{Rep} reaches 66.97/64.15% F1 on Test, well below ICL, and with accuracy following the same trend. Importantly, semantic retrieval provides no consistent benefit. These findings confirm that single-layer, last-token TV injection fundamentally fails to support robust task adaptation in AD detection.

DA4ICL achieves substantial and stable improvements. In contrast, DA4ICL delivers the highest F1 and accuracy across all datasets (Test: 86.11/85.83%, Lu: 86.12/80.95%, Pitt: 80.23/79.42%), with the lowest variance among all methods. These results confirm that enriching demos through diverse, contrastive retrieval and demo-wise vector anchoring decisively overcomes the adaptation bottlenecks and granularity mismatches of prior ICL and TV paradigms, enabling reliable and robust AD detection.

Dissecting Two-Stage Retrieval and the Generality of DCR

We first examine the effectiveness of DCR from two perspectives: within our full DA4ICL framework and when integrated into existing ICL and TV baselines.

Context width is essential and depth provides complementary gains. Ablation of retrieval strategies (Fig. 5) reveals that applying DCR to the *main-demo* stage consistently yields the highest accuracy and lowest variance, regardless of the *sub-demo* strategy. Removing DCR from the *main-demo* stage, replacing it with random or semantic retrieval, leads to notable drops in performance, highlighting that context width (diverse and contrastive *main-demos*) is a prerequisite for effective adaptation, while context depth (*sub-demos*) provides complementary gains.

DCR generalizes to ICL, but fails to activate standard TV methods. We further apply DCR to ICL_{Van}, ICL_{Ens}, TV_{Add}, and TV_{Rep} (Fig. 4). Results show that DCR significantly boosts both standard and ensemble ICL across all datasets, confirming that contrastive demo construction improves robustness and generalization. However, such improvement fails to transfer to conventional TV methods, performance remains unchanged or even degraded. This suggests that existing TV methods cannot effectively absorb the diverse, fine-grained signals DCR provides, due to their injection granularity and alignment limitations.

Dataset	Acc (%)			
	Addition	Replacement	w/o injection	PVA
Test	77.50±4.04	69.38±2.95	77.29±4.41	**85.83**±1.91
Lu	77.62±3.23	75.71±1.78	77.86±2.83	**81.43**±2.56
Pitt	76.41±0.91	72.82±1.00	75.94±1.21	**79.42**±0.39

Table 2: Ablation on anchoring methods. Projection-based PVA achieves the best performance, outperforming addition, replacement, and removal variants across all datasets.

Why Direct TV Injection Fails and PVA Matters

To understand why standard TV methods struggle, we further dissect their injection strategy and contrast them with our proposed PVA module.

Test-centric injection with coarse granularity undermines adaptation. Standard TV methods inject TVs only at test sample's $\rightarrow$ token and single layer. Such test-centric, single-token and single-layer injection discards distributed demo cues and fails to propagate fine-grained distinctions. As shown in Fig. 4, even when empowered with DCR, these methods do not improve and often performing worse than ICL methods.

PVA preserves semantic alignment and achieves fine-grained control. Ablation results in Tab. 2 confirm that PVA's projection-based, layer-wise anchoring substantially outperforms naive alternatives. These results show that direct addition or replacement disrupts the original semantic direction of the HSs, while PVA preserves alignment and allows stable, context-sensitive adaptation. Notably, PVA can amplify both helpful and harmful cues of the demo set, emphasizing the importance of DCR strategy.

Conclusion

We revisit the core limitations of ICL and TV methods for AD detection, showing that both demo context narrowness and misaligned vector injection hinder task perception. DA4ICL addresses these challenges by integrating DCR strategy with demo-centric PVA mechanism, jointly expanding context width and depth while preserving semantic alignment. Experiments on three AD datasets demonstrate substantial and stable improvements over previous methods. We believe this demo-centric anchoring paradigm offers a promising foundation for fine-grained, low-resource and OOD task adaptation with LLMs.

Acknowledgments

This work was supported by the Key Research and Development Project of Hunan Province (No. 2025JK2119), Foundation of NUDT (HQKYZH2025KD004), the Leading Science and Technology Innovation Talents Program of Furong Project (2025RC1048).

References

Abbas, M.; Zhou, Y.; Ram, P.; Baracaldo, N.; Samulowitz, H.; Salonidis, T.; and Chen, T. 2024. Enhancing In-context Learning via Linear Probe Calibration. In Dasgupta, S.; Mandt, S.; and Li, Y., eds., *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, 307–315. PMLR.

Ali, A.; Wolf, L.; and Titov, I. 2024. Mitigating Copy Bias in In-Context Learning through Neuron Pruning. *ArXiv preprint*, abs/2410.01288.

Balamurali, B. T.; and Chen, J. 2024. Performance Assessment of ChatGPT versus Bard in Detecting Alzheimer's Dementia. *Diagnostics*, 14(8): 817.

Becker, J. T.; Boiler, F.; Lopez, O. L.; Saxton, J.; and McGonigle, K. L. 1994. The natural history of Alzheimer's disease: description of study cohort and accuracy of diagnosis. *Archives of Neurology*, 51(6): 585–594.

Deture, M. A.; and Dickson, D. W. 2019. The neuropathological diagnosis of Alzheimer's disease. *Molecular Neurodegeneration*, 14(1).

Dong, Y.; Jiang, J.; Zhu, Z.; and Ning, X. 2025. Understanding Task Vectors in In-Context Learning: Emergence, Functionality, and Limitations. *arXiv preprint arXiv:2506.09048*.

Hendel, R.; Geva, M.; and Globerson, A. 2023. In-Context Learning Creates Task Vectors. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 9318–9333. Singapore: Association for Computational Linguistics.

Hong, G.; van Krieken, E.; Ponti, E.; Malkin, N.; and Minervini, P. 2024. Mixtures of In-Context Learners. *ArXiv preprint*, abs/2411.02830.

Kelley, B. J.; and Petersen, R. C. 2007. Alzheimer's disease and mild cognitive impairment. *Neurologic Clinics*, 25(3): 577–609.

Khalifa, M.; Logeswaran, L.; Lee, M.; Lee, H.; Wang, L.; and Pavlick, E. 2023. Exploring Demonstration Ensembling for In-Context Learning. *ArXiv preprint*, abs/2308.08780.

Kossen, J.; Gal, Y.; and Rainforth, T. 2024. In-Context Learning Learns Label Relationships but Is Not Conventional Learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Lanzi, A. M.; Saylor, A. K.; Fromm, D.; Liu, H.; MacWhinney, B.; and Cohen, M. 2023. DementiaBank: Theoretical rationale, protocol, and illustrative analyses. *American Journal of Speech-Language Pathology*.

Li, C.; Li, R.; Field, T. S.; and Carenini, G. 2025a. DeltaKNN: Improving Demonstration Selection in In-Context Learning for Alzheimer's Disease Detection. *ArXiv preprint*, abs/2506.03476.

Li, D.; Liu, Z.; Hu, X.; Sun, Z.; Hu, B.; and Zhang, M. 2024. In-Context Learning State Vector with Inner and Momentum Optimization. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024), Poster Track*.

Li, Z.; Xu, Z.; Han, L.; Gao, Y.; Wen, S.; Liu, D.; Wang, H.; and Metaxas, D. N. 2025b. Implicit In-Context Learning. In *Proceedings of the International Conference on Learning Representations*.

Lin, B. Y.; Lee, S.; Khanna, R.; and Ren, X. 2020. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6862–6868. Online: Association for Computational Linguistics.

Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; and Chen, W. 2022. What Makes Good In-Context Examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 100–114. Dublin, Ireland and Online: Association for Computational Linguistics.

Liu, S.; Ye, H.; Xing, L.; and Zou, J. Y. 2024a. In-Context Vectors: Making In Context Learning More Effective and Controllable Through Latent Space Steering. In *Proceedings of the 41st International Conference on Machine Learning*.

Liu, Z.; Li, D.; Hu, X.; Zhao, X.; Chen, Y.; Hu, B.; and Zhang, M. 2024b. Take Off the Training Wheels! Progressive In-Context Learning for Effective Alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2743–2757. Miami, Florida, USA: Association for Computational Linguistics.

Luo, M.; Xu, X.; Liu, Y.; Pasupat, P.; and Kazemi, M. 2024. In-Context Learning with Retrieved Demonstrations for Language Models: A Survey. *ArXiv preprint*, abs/2401.11624.

Luz, S.; Haider, F.; de la Fuente, S.; Fromm, D.; and MacWhinney, B. 2020. Alzheimer's Dementia Recognition Through Spontaneous Speech: The ADReSS Challenge. In Meng, H.; Xu, B.; and Zheng, T. F., eds., *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, 2172–2176. ISCA.

Lyu, X.; Min, S.; Beltagy, I.; Zettlemoyer, L.; and Hajishirzi, H. 2022. Z-ICL: Zero-shot in-context learning with pseudo-demonstrations. *ArXiv preprint*, abs/2212.09865.

MacWhinney, B. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Merullo, J.; Eickhoff, C.; and Pavlick, E. 2024. Language Models Implement Simple Word2Vec-style Vector Arithmetic. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 5030–5047. Mexico City, Mexico: Association for Computational Linguistics.

Mojarradi, M. M.; Yang, L.; McCraith, R.; and Mahdi, A. 2024. Improving In-Context Learning with Small Language Model Ensembles. *ArXiv preprint*, abs/2410.21868.

Pang, J.; Ye, F.; Wong, D.; He, X.; Chen, W.; and Wang, L. 2024. Anchor-Based Large Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 4958–4976. Bangkok, Thailand: Association for Computational Linguistics.

Roark, B.; Mitchell, M.; Hosom, J.-P.; Hollingshead, K.; and Kaye, J. 2011. Spoken language derived measures for detecting mild cognitive impairment. *IEEE transactions on audio, speech, and language processing*, 19(7): 2081–2090.

Roshanzamir, A.; Aghajan, H.; and Soleymani Baghshah, M. 2021. Transformer-based deep neural network language models for Alzheimer's disease risk assessment from targeted speech. *BMC Medical Informatics and Decision Making*, 21: 1–14.

Saglam, B.; Yang, Z.; Kalogerias, D.; and Karbasi, A. 2024. Learning Task Representations from In-Context Learning. In *Proceedings of the ICML 2024 Workshop on In-Context Learning (Poster)*.

Srinivasan, K. P. V.; Gumpena, P.; Yattapu, M.; and Brahmbhatt, V. H. 2024. Comparative Analysis of Different Efficient Fine Tuning Methods of Large Language Models (LLMs) in Low-Resource Setting. *ArXiv preprint*, abs/2405.13181.

Su, P.; Miao, Y.; Guo, C.; Tang, J.; Li, S.; and Wang, T. 2025. Explicit Knowledge-Guided In-Context Learning for Early Detection of Alzheimer's Disease. arXiv:2511.06215.

Sun, X.; Li, X.; Li, J.; Wu, F.; Guo, S.; Zhang, T.; and Wang, G. 2023. Text Classification via Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 8990–9005. Association for Computational Linguistics.

Todd, E.; Li, M. L.; Sen Sharma, A.; Mueller, A.; Wallace, B. C.; and Bau, D. 2024. Function Vectors in Large Language Models. In *Proceedings of the 12th International Conference on Learning Representations*.

Wang, L.; Li, L.; Dai, D.; Chen, D.; Zhou, H.; Meng, F.; Zhou, J.; and Sun, X. 2023. Label Words are Anchors: An Information Flow Perspective for Understanding In-Context Learning. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 9840–9855. Singapore: Association for Computational Linguistics.

Yang, L.; Lin, Z.; Lee, K.; Papailiopoulos, D.; and Nowak, R. 2025. Task Vectors in In-Context Learning: Emergence, Formation, and Benefit. *ArXiv preprint*, abs/2501.09240.

Yu, Y.; Shen, J.; Liu, T.; Qin, Z.; Yan, J. N.; Liu, J.; Zhang, C.; and Bendersky, M. 2024. Explanation-Aware Soft Ensemble Empowers Large Language Model In-Context Learning. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14002–24. Bangkok, Thailand: Association for Computational Linguistics.

Yu, Z.; and Ananiadou, S. 2024. How Do Large Language Models Learn In-Context? Query and Key Matrices of In-Context Heads Are Two Towers for Metric Learning. *arXiv*.

Zhao, A.; Ye, F.; Fu, J.; and Shen, X. 2024. Unveiling In-Context Learning: A Coordinate System to Understand Its Working Mechanism. *arXiv*.