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*

[1]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

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, existing inference strategies, like ensemble voting (Hong et al. 2024; Yu et al. 2024), calibration (Abbas et al. 2024),

*Corresponding author.

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.

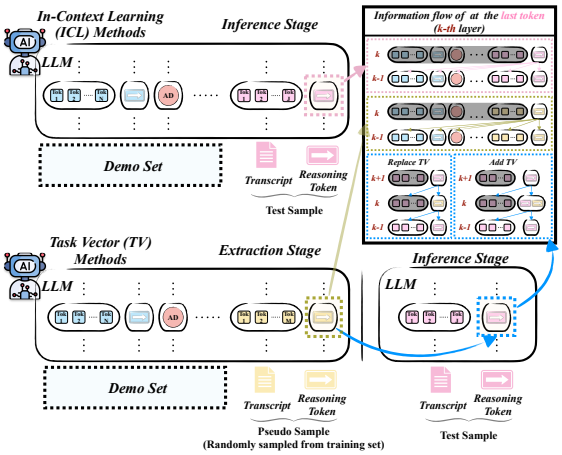

Figure 1: Schematic comparison of information flow and token processing in standard ICL versus TV methods.

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***.

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

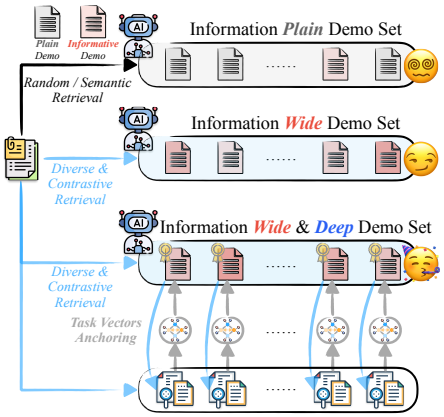

Figure 2: Progressive enrichment of demo sets, from plain to wide and deep, drives more effective and robust in-context reasoning.

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 consistently 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. 2024a; 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.

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

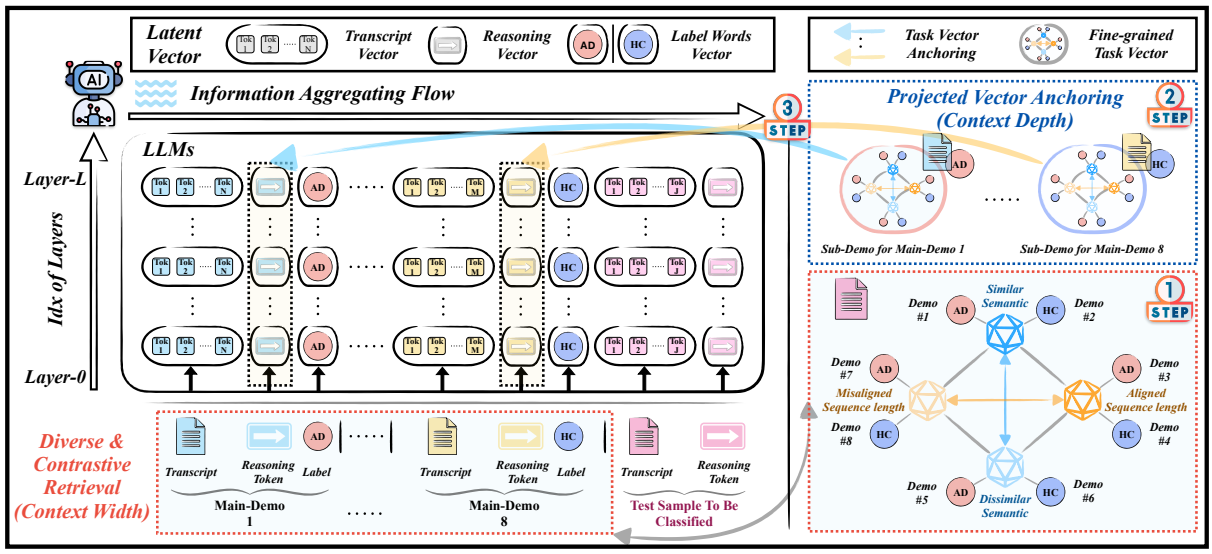

Figure 3: Overview of the DA4ICL framework. Diverse and contrastive demos are selected and enriched via projected vector anchoring across all Transformer layers to provide both wide and deep context for robust AD detection.

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:

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

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 $\to$ 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 $\to$ 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 $\{x_i, \to\}$, where $\{x_i, \to\} \in \text{Seq}_i^{sub}, x_i \in \mathcal{D}_{\text{main}}$.

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:

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

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 details are listed 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, $N = 4$), 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 underperform 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}

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 ($p < 0.01$).

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 sug-

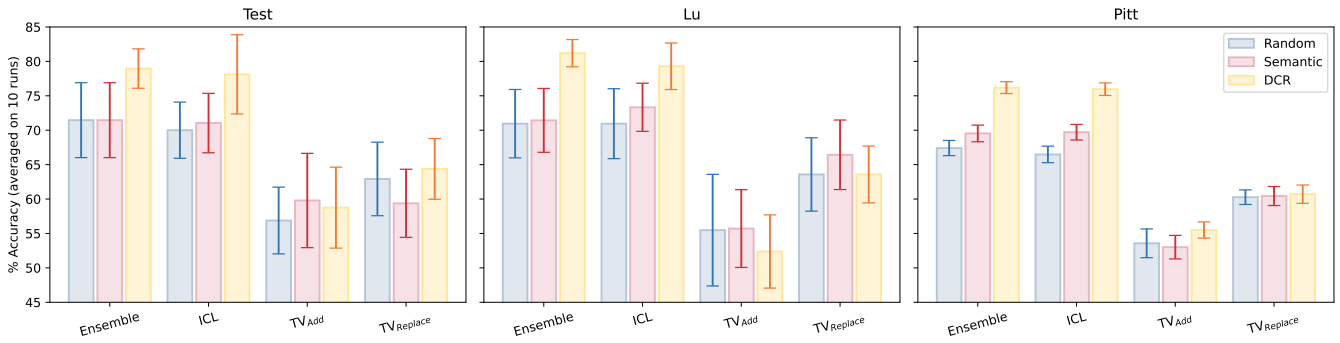

Figure 4: Accuracy comparison after applying DCR to ICL and TV methods (mean ± std over 10 runs). DCR consistently improves ICL performance across all datasets, demonstrating the value of contrastive demo construction. However, DCR fails to enhance conventional TV methods, with performance remaining stable or degrading, highlighting the limitations of TV's injection granularity and alignment. Specific results can be found in Appendix: B.

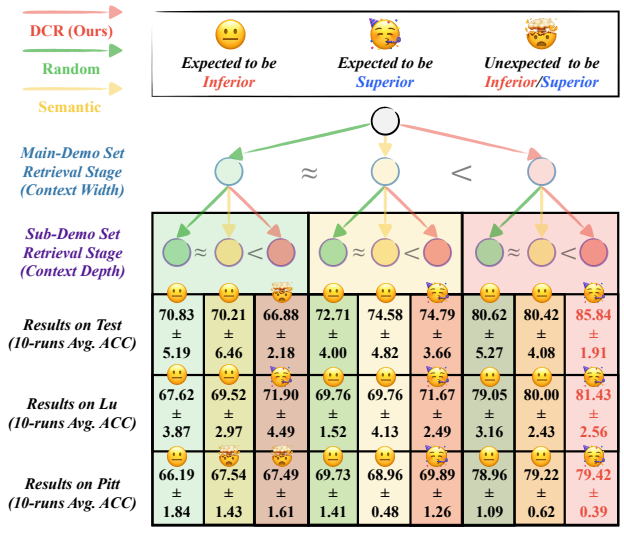

Figure 5: Ablation over two-stage retrieval strategies. Diverse main-demo selection (context width) provides the largest gains, while sub-demo enrichment (context depth) offers further complementary improvements.

gests that existing TV methods cannot effectively absorb the diverse, fine-grained signals DCR provides, due to their injection granularity and alignment limitations.

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 simpler ICL strategies. This validates the need for demo-centric, multi-layer anchoring.

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 (demo-centric) methods. Projection-based PVA achieves the best performance, outperforming addition, replacement, and removal variants across all datasets.

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 set, emphasizing the importance of robust retrieval strategy like DCR.

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 diverse and contrastive retrieval strategy with demo-centric, projection-based TV anchoring, 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.

Balagopalan, A.; Eyre, B.; Robin, J.; Rudzicz, F.; and Novikova, J. 2021. Comparing pre-trained and feature-based models for prediction of Alzheimer's disease based on speech. *Frontiers in aging neuroscience*, 13: 635945.

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.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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. 2024a. 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, F.; Huang, C.; Su, P.; and et al. 2024b. SPZ: A Semantic Perturbation-based Data Augmentation Method with Zonal-Mixing for Alzheimer's Disease Detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15429–15439.

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.

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*.

Appendix

Appendix Overview. This appendix complements the technical content of our paper with task specifications, implementation details, mechanistic analyses, and additional experiments, following the structure below:

Appendix A. *Task and Dataset Details.* defines the task (*A.1 Task Definition: Alzheimer's Disease Detection from Narrative Speech*), summarizes datasets, demographics and licenses (*A.2 Datasets, Demographics and Licenses*), details preprocessing (*A.3 Preprocessing*), and highlights key challenges (*A.4 Task Challenges*). Correlated figures and tables are Fig. 6, Tab. 3.

Appendix B. *Supplementary Implementation Details.* documents the backbone and hardware (*B.1 Details of Backbone Model*: *Model and Hardware*, *Model Scale Considerations*); demo construction (*B.2 Details of Demo Construction*: *Prompt Construction*, *Demo Retrieval and Construction*, *Hyperparameters and Inference*); and experimental settings (*B.3 Details of Experimental Settings*: *Statistical Analysis*, *Evaluation Metrics*). Correlated figures and tables are Fig. 7, Tab. 4.

Appendix C. *Supplementary Mechanistic Analysis.* formalizes ICL in the residual-stream view (*C.1 In-Context Learning (ICL): Residual Stream Formulation*: *Residual Stream in Transformers*, *Generation Flow in ICL*, *Final Token as Contextual Anchor*, *Layer-wise Information Refinement*); specifies TV extraction/injection (*C.2 Task Vector (TV) Method: Extraction and Injection Mechanism*: *Task Vector Extraction*, *Single-point Task Vector Injection*, *Vertical Propagation and Horizontal Limitation*); visualizes design limits (*C.3 Visualizing Limitations of TV Injection*); and details our diverse/contrastive retrieval (*C.4 Demo Construction: Diverse and Contrastive Retrieval*: *Stage 1: Main-Demo Set Construction*, *Remarks*). Correlated figures and tables are Fig. 8, Fig. 9, Tab. 5.

Appendix D. ***Supplementary Experiments.*** reports ablations for PVA (*D.1 Ablation Results for PVA Module*), full metrics across datasets and shots (*D.2 Full Evaluation Metrics*), supervised baselines and LoRA configurations (*D.3 Comparison of Supervised Baselines*: *Fine-tuning Methods*, *Detailed Information of LoRA*, *Hyperparameters*, *Classification Task Setup*, *Generation Task Setup*), efficiency/resource measurements on 48 GB/80 GB GPUs (*D.4 Efficiency and Resource Analysis*), and a quantitative–qualitative error analysis of Top-2 runs (*D.5 Error Analysis*: *Overall Trends*, *Error Direction Asymmetry*, *Qualitative findings*). Correlated figures and tables are Fig. 10, Tab. 6, Tab. 7, Tab. 8, Tab. 9, Tab. 10, Tab. 11, Tab. 12, Tab. 13, Tab. 14.

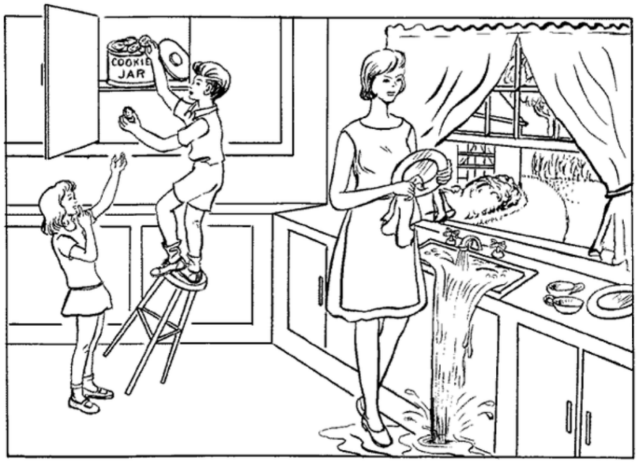

Figure 6: Image for the Cookie Theft picture description task. All participants are asked to describe the details and actions present in this scene.

A. Task and Dataset Details

A.1 Task Definition: Alzheimer's Disease Detection from Narrative Speech The primary task evaluated in this study is Alzheimer's disease detection from narrative speech transcripts, specifically through the Cookie Theft picture description protocol. In this task, participants are shown a standard image (see Fig. 6) depicting a kitchen scene with several salient elements. Each participant is instructed to describe, in as much detail as possible, everything occurring in the image. Their spoken descriptions are transcribed and subsequently used as input for automatic AD detection.

The task is to classify each transcript as originating from an individual with Alzheimer's disease (AD) or from a healthy control (HC). This presents a challenging scenario due to the high semantic similarity of transcripts (as all subjects describe the same scene) and the subtle linguistic differences associated with early-stage AD.

A.2 Datasets, Demographics and Licenses Experiments are conducted on three benchmark corpora derived from DementiaBank: ADReSS (Train/Test), Lu, and Pitt. Tab. 3 summarizes the demographic information for each split.

ADReSS (Train/Test): Age- and gender-balanced splits, ensuring comparability between AD and HC groups. **Lu:** Contains older participants with a modest gender imbalance, serving as an out-of-distribution evaluation scenario. **Pitt:** The largest corpus, with greater demographic skew and the most challenging OOD setting.

All datasets are sourced from DementiaBank (Train, Test, Lu, Pitt). The DementiaBank is a shared database of multimedia interactions for the study of communication in dementia, a sub database of TalkBank. Train/test splits are strictly separated, and no overlap exists. The use of TalkBank data is governed by the Creative Commons CC BY-NC-SA 3.0 copyright license. and subject to additional requirements:

- Researchers must abide by the TalkBank Code of Ethics and maintain confidentiality NIH Certificate of Confidentiality.
- Access to password-protected corpora is limited to authorized academic members, and students must be supervised by faculty.
- Raw data cannot be redistributed or uploaded to external platforms.

Note: All processed results in this paper are based on data access compliant with DementiaBank policies. Data cannot be made public due to license restrictions but can be obtained by qualified researchers through DementiaBank as above.

A.3 Preprocessing All audio recordings are transcribed to text. Preprocessing includes removal of non-speech markers, normalization of punctuation, and lowercasing. Only transcripts corresponding to the Cookie Theft description are retained. Speaker and interviewer markers are excluded.

A.4 Task Challenges Challenges include limited data size and pre-training exposure (due to privacy and recruitment constraints) and semantic homogeneity (since all participants describe the same image). These characteristics make the AD detection an out-of-distribution (OOD) task, and hinder its adaptation and generalization ability.

B. Supplementary Implementation Details

B.1 Details of Backbone Model

Model and Hardware. All experiments are conducted with a frozen **Llama-3.1-8B-Instruct** language model (Huggingface Transformers), without any parameter updates or further pre-training. All inference was performed on a single NVIDIA Quadro RTX 8000 (48GB GPU), with the efficiency and resource analysis experiments in **Appendix D.4** additionally introducing an NVIDIA A100 (80GB GPU) for comparison.

Model Scale Considerations. We used a single backbone because, under AD data scarcity, the pre-training exposure is limited across vary scales. Nevertheless, model capacity is expected to modulate both absolute performance and the *marginal* benefit of our components. We argue the following hypothesis:

Dataset	Alzheimer's Disease		Healthy Control	
	Age (mean±std)	Gender (M/F)	Age (mean±std)	Gender (M/F)
Train	66.4 ± 6.4	24 / 30	66.8 ± 6.6	24 / 30
Test	66.1 ± 6.9	11 / 13	66.1 ± 7.3	11 / 13
Lu	73.9 ± 8.6	7 / 8	79.2 ± 8.9	12 / 15
Pitt	64.9 ± 7.7	89 / 154	71.4 ± 8.5	117 / 189

Table 3: Demographic statistics of AD and HC groups in the ADReSS (Train/Test), Lu, and Pitt corpora.

Prompt Component	**Content Example**
System Prompt	`You are an experienced clinician specializing in dementia care. You will analyze descriptions provided by subjects who are describing a scene from the "Cookie Theft" picture, used in the Boston Diagnostic Aphasia Examination. In this task, subjects are asked to describe everything they see in the picture, which depicts a chaotic kitchen scene ... Your task is to analyze the subject's description to determine whether it indicates [HEALTHY CONTROL] mental condition or [Alzheimer's Disease] mental condition. Give your answer (Start with [HEALTHY CONTROL] or Start with [Alzheimer's Disease]) first.`
Demo Example 1	`Here is the example: ##Text:<The mother is washing the dishes ...> ##Answer: The condition of this description is -> HEALTHY CONTROL`
Demo Example 2	`Here is the example: ##Text:<The boy is standing on a stool to reach ...> ##Answer: The condition of this description is -> HEALTHY CONTROL`
Test Query	`Now you should analyze the testing subject's description: ##Text:<Subject's description here> ##Answer: I think The condition of this description will be ->`

Table 4: Prompt construction template for ICL and TV methods. The system prompt provides clinical context. Each demo example presents a subject's Cookie Theft description and its label. The test query asks the model to analyze a new description and output the predicted condition. All prompts are in English.

- Larger LLMs, endowed with stronger task perception and richer pre-trained priors, should deliver higher baselines; their *absolute* gains from DCR and PVA will likely shrink due to diminishing returns, yet *relative* gains can persist because both modules operate on high-quality hidden states (HS) and mainly improve decision calibration and context routing rather than raw capability.
- Smaller LLMs, with weaker HS quality, are expected to see larger *absolute* gains, DCR widens salient cues in the prompt space and PVA deepens integration at separator tokens, while their upper bounds remain constrained by HS fidelity.
- Across scales, we anticipate a monotone trade-off between compute/latency and robustness: deeper/multi-layer anchoring yields consistent improvements, but its cost is amortized more effectively on mid-to-large models that already maintain stable HS geometry.

A full multi-scale analysis (e.g., 3B/8B/14B/70B) is left as future work with both absolute and relative gains reported.

B.2 Details of Demo Construction

Prompt Construction. All prompts are in English, following the standard in-context learning format (Details in Tab. 4) with demonstration pairs. Each prompt is constructed as a sequence of k demonstration pairs, each formatted as "Input $\rightarrow$ Output", followed by the test input and separator:

$$[x_1 \rightarrow y_1;\ x_2 \rightarrow y_2;\ \ldots;\ x_k \rightarrow y_k;\ x_{\text{test}} \rightarrow]$$

A concrete illustration is shown below.

Demo Retrieval and Construction. Semantic similarity is computed using the cosine similarity between the test sample and candidate demos, based on the last hidden state

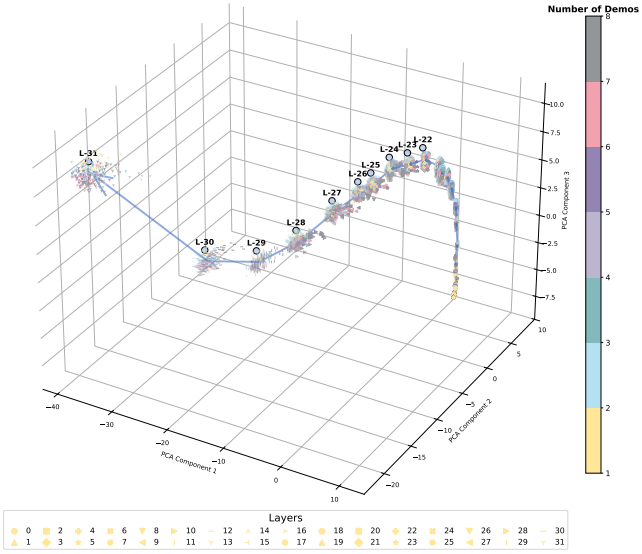

Figure 7: Layer-wise PCA visualization of demo separator token representations from shallow (right) to deep (left) layers. Clustering by class emerges in higher layers, illustrating progressive semantic refinement.

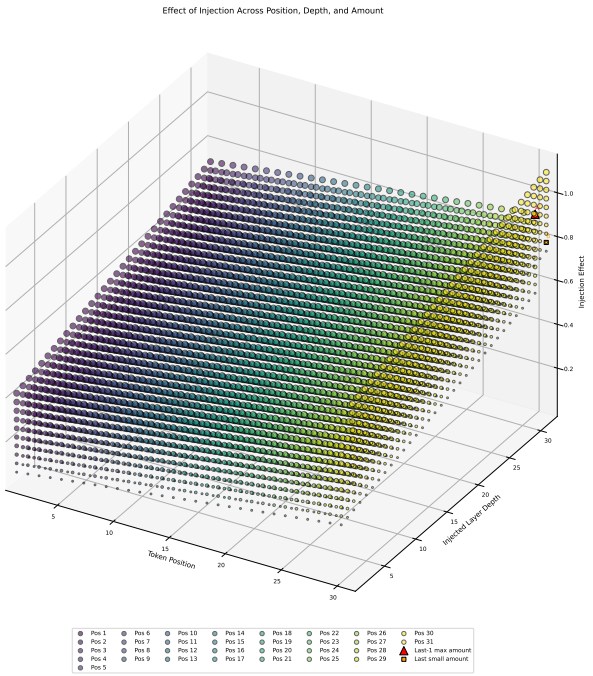

Figure 8: Conceptual illustration of Task Vector (TV) injection effects. The x-axis denotes the token position, the y-axis indicates injected layer depth, and the z-axis (color) represents the relative injection effect. This figure serves as a schematic visualization to clarify the propagation pattern rather than an empirical result. Injections applied to the final (anchor) token accumulate and dominate downstream influence, while earlier token injections have weaker or dispersed effects.

of the model at the final token. Length similarity is measured by token count. Eight main demonstrations (4 AD / 4 Control, see **Appendix C**) are selected for each test sample via four orthogonal criteria (semantic/length, similarity/dissimilarity), with each main demo paired to a sub-demo for task vector extraction. Label balance is strictly enforced; no overlap occurs between main and sub-demo sets.

Hyperparameters and Inference. As illustrated in the PCA analysis (Fig. 7), middle layers undergo a rapid shift along PC-3 that then stabilizes near deep layers. Strong anchoring here can perturb downstream feature composition. Thus, the layer-wise injection strength $\gamma^{(l)}$ is set to 1.0 for the first and last 8 layers, and 0.2 for the middle 16 layers. Each experiment is repeated for 10 runs with different random seeds. All other hyper-parameters are listed in Tab. 5.

B.3 Details of Experimental settings

Statistical Analysis. Results are reported as mean $\pm$ standard deviation over 10 independent runs. Paired t-tests and Wilcoxon signed-rank tests are computed via `scipy.stats`. No multiple-comparison correction is applied, since only one-to-one comparisons are made per setting.

Evaluation Metrics. All results are reported on metrics (Accuracy, Precision, Recall, F1). For each run, metrics are computed on the test set and then averaged.

C. Supplementary Mechanistic Analysis

To rigorously complement the theoretical foundation discussed in the technical content of our paper, this section provides a detailed and formalized analysis of the underlying mechanisms of In-Context Learning (ICL) and Task Vector (TV) methods from the perspective of the Transformer's residual stream formulation, alongside an extended discussion on TV injection effects.

C.1 In-Context Learning (ICL): Residual Stream Formulation

Residual Stream in Transformers. Following previous formulations, we represent the hidden state at token t and layer l as part of a recursive residual stream:

$$h_t^{(l)} = h_t^{(l-1)} + a_t^{(l)} + m_t^{(l)}, \tag{10}$$

where $a_t^{(l)}$ and $m_t^{(l)}$ denote multi-head attention (MHA) and multi-layer perceptron (MLP) modules, respectively. The attention mechanism aggregates information across tokens:

$$\begin{aligned} a_t^{(l)} &= \mathrm{MHA}(h_t^{(l-1)}) \\ &= \sum_{i=1}^{T} \mathrm{softmax}\left(\frac{(W_Q h_t^{(l-1)})^\top (W_K h_i^{(l-1)})}{\sqrt{d}}\right) W_V h_i^{(l-1)}, \end{aligned} \tag{11}$$

where T is the sequence length and W_Q, W_K, W_V are learnable projection matrices. The MLP module introduces nonlinear transformations:

$$m_t^{(l)} = W_2\left(\sigma(W_1(h_t^{(l-1)} + a_t^{(l)}))\right), \tag{12}$$

Parameter	Value
Backbone Model	Llama-3.1-8B-Instruct (frozen)
Max sequence length	512
Demo count per test	8 (4 pairs)
Batch size	1
Layer-wise γ	1.0 (0-7)/ 0.2 (8-23)/ 1.0 (24-31)
Runs per experiment	10
GPU	Quadro RTX 8000, 48GB
Metrics	Precision, Recall, F1, Accuracy

Table 5: Main hyperparameters and computational resources.

where σ is a nonlinear activation function. Collectively, these mechanisms progressively refine hidden representations layer-by-layer.

Generation Flow in ICL. In ICL, given a demonstration set $D = \{(x_i, y_i)\}_{i=1}^{k}$ concatenated with a test query x_{test}:

$$\text{Seq}(D) = [x_1 \to y_1; \ldots; x_k \to y_k; x_{\text{test}} \to], \tag{13}$$

the model predicts labels sequentially, with the next-token probability:

$$p(x_t \mid x_{<t}) = \text{softmax}(W_{LM} h_{t-1}^{(L)}), \tag{14}$$

where W_{LM} is the final projection matrix mapping hidden states to token logits.

Final Token as Contextual Anchor. Although each separator token ($\to$) locally integrates contextual information within individual demos, the final token at position T in the last layer (L) serves as the **global anchor**:

$$h_{\to,T}^{(L)} = \text{TransformerLayer}^{(L)}([h_1^{(L-1)}, \ldots, h_T^{(L-1)}]). \tag{15}$$

This global anchor token integrates and consolidates all previously aggregated context across demonstrations and the test input, thus critically influencing model predictions.

Layer-wise Information Refinement. The Transformer progressively transforms representations from general and shallow features in early layers to increasingly specialized and discriminative task-related representations in deeper layers. This hierarchical refinement ensures the final-layer representations are rich in task-specific semantics.

C.2 Task Vector (TV) Method: Extraction and Injection Mechanism

Task Vector Extraction. Standard TV methods extract task-specific latent vectors at a chosen intermediate layer ℓ by concatenating a demo set with a randomly chosen pseudo-query x_{pseudo}:

$$\text{Seq}(D) = [x_1 \to y_1; \ldots; x_k \to y_k; x_{\text{pseudo}} \to]. \tag{16}$$

The extracted task vector $v^{(\ell)}$ is then:

$$v^{(\ell)} = h_{\to,\text{pseudo}}^{(\ell)}, \tag{17}$$

where $h_{\to,\text{pseudo}}^{(\ell)}$ represents the hidden state at the pseudo-query separator token.

Single-point Task Vector Injection. During inference, the task vector is injected into the separator token of the test input at layer ℓ:

$$h_{\to,\text{test}}^{(\ell)\prime} = f(h_{\to,\text{test}}^{(\ell)}, v^{(\ell)}), \tag{18}$$

where $f(\cdot)$ can be soft (additive, $0 \leq \gamma < 1$) or hard (replacement, $\gamma = 1$):

$$f(h, v) = (1 - \gamma) \cdot h + \gamma \cdot v. \tag{19}$$

Vertical Propagation and Horizontal Limitation. The injected vector modifies hidden states and propagates vertically across subsequent layers:

$$h_{\to,\text{test}}^{(\ell+1)} = \text{TransformerLayer}^{(\ell+1)}([h_1^{(\ell)}, \ldots, h_{\to,\text{test}}^{(\ell)\prime}]). \tag{20}$$

However, horizontally, the injected vector only influences subsequent layers at the test position, significantly limiting contextual integration capabilities and potentially causing pseudo-sample overfitting, especially when using single or shallow-layer injection.

C.3 Visualizing Limitations of TV Injection To clarify the impact of injection design, we visually illustrate how TV effectiveness is influenced by three key design choices: injection position, injection depth, and the number of injection layers (Fig. 8). In this illustration:

- **Position (x-axis):** Indicates the relative token position of injection (left: early, right: late).
- **Layer Depth (y-axis):** Denotes injection at shallow (front) versus deep (behind) layers.
- **Injection Breadth (Sphere size):** Larger spheres represent injection spanning more layers; smaller spheres reflect fewer layers (or single-layer injection).

This visualization clearly demonstrates that:

- Injecting at very shallow layers has limited vertical propagation potential, even high quality of TV, offering insufficient task specialization.
- A bad Injection at deep layers may disrupt context-sensitive representations, causing semantic distortion or overfitting.
- Single-layer or limited-layer injection results in restricted horizontal propagation, failing to enrich broader contextual understanding adequately.

Thus, the TV method's effectiveness critically depends on balanced choices regarding these dimensions. The limitations inherent in conventional single-layer, test-centric injection motivate our proposed demo-centric, multi-layer projection anchoring strategy, which distributes and harmonizes task adaptation signals more effectively across both horizontal and vertical dimensions.

This rigorous preliminary formalization and expanded visualization collectively provide clear theoretical grounding for our proposed methodological improvements and elucidate the necessity of adopting demo-centric, multi-layer injection approaches for robust adaptation in nuanced tasks such as Alzheimer's disease detection.

Methods	Sig.	Test		Lu		Pitt	
		F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)
ICL_{Van}	–	$72.13_{4.15}$	$70.00_{4.08}$	$79.53_{3.58}$	$70.95_{5.08}$	$70.24_{1.12}$	$66.48_{1.20}$
+Van PVA	✗	$73.75_{4.77}$	$70.83_{5.19}$	$78.01_{2.44}$	$67.62_{3.87}$	$70.32_{1.43}$	$66.19_{1.84}$
+Sem PVA	✗	$72.72_{6.43}$	$70.21_{6.46}$	$79.09_{2.53}$	$69.52_{2.97}$	$71.68_{1.26}$	$67.54_{1.43}$
+DCR PVA	✗	$70.17_{2.94}$	$66.88_{2.18}$	$80.54_{3.05}$	$71.90_{4.49}$	$71.54_{1.40}$	$67.49_{1.61}$
ICL_{Sem}	–	$74.20_{3.69}$	$71.04_{4.32}$	$82.02_{2.11}$	$73.33_{3.50}$	$73.79_{1.21}$	$69.71_{1.13}$
+Van PVA	✗	$75.76_{3.61}$	$72.71_{4.00}$	$80.06_{0.96}$	$69.76_{1.52}$	$73.91_{1.08}$	$69.73_{1.41}$
+Sem PVA	✗	$77.20_{3.91}$	$74.58_{4.82}$	$79.81_{2.75}$	$69.76_{4.13}$	$73.03_{0.46}$	$68.96_{0.48}$
+DCR PVA	✗	$77.20_{3.30}$	$74.79_{3.66}$	$81.16_{1.40}$	$71.67_{2.49}$	$73.90_{0.99}$	$69.89_{1.26}$
ICL_{DCR}	–	$76.75_{4.53}$	$77.29_{4.41}$	$83.71_{2.11}$	$77.86_{2.83}$	$76.62_{1.41}$	$75.94_{1.21}$
+Van PVA	✓	$80.78_{5.05}$	$80.62_{5.27}$	$84.82_{2.34}$	$79.05_{3.16}$	$79.86_{1.13}$	$78.96_{1.09}$
+Sem PVA	✓	$80.56_{4.11}$	$80.42_{4.08}$	$85.41_{1.78}$	$80.00_{2.43}$	$80.16_{0.59}$	$79.22_{0.62}$
+DCR PVA (Ours)	✓	$86.11_{1.92}$	$85.83_{1.91}$	$86.51_{1.83}$	$81.43_{2.56}$	$80.23_{0.42}$	$79.42_{0.39}$

Table 6: Ablation results for the PVA module under different demo retrieval settings (4 demo pairs) on Test, Lu, and Pitt datasets. Each cell reports mean_{std} over 10 runs. "Sig" indicates whether the PVA variant significantly outperforms its base ICL (✓) or not (✗).

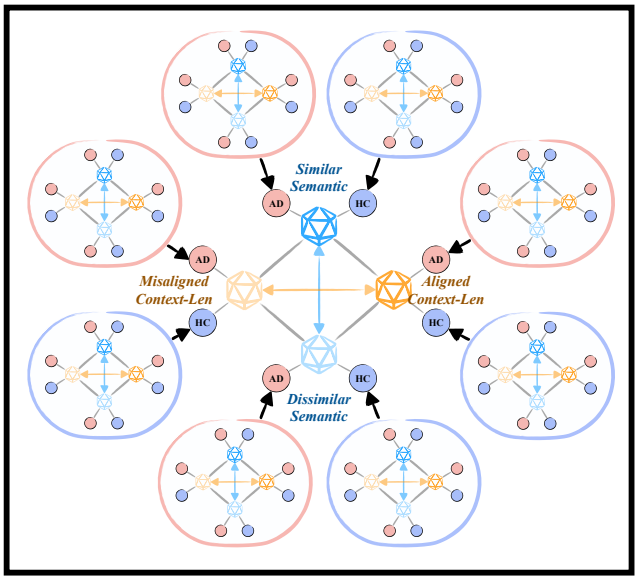

Figure 9: Demo-centric, projection-based full-layer anchoring: Each main demo separator is injected with its specific task vector at all layers, enabling distributed and context-rich adaptation.

C.4 Demo Construction: Diverse and Contrastive Retrieval A core component of our approach is the construction of a high-diversity, contrastively balanced demo set for each test query via a two-stage retrieval mechanism.

Stage 1: Main-Demo Set Construction. For every test sample d_{test}, we retrieve 8 main demos, sampling one contrastive pair (AD/HC) for each of four orthogonal criteria:

- **Semantic Similarity:** Pair with the highest cosine similarity in ϕ-space (where ϕ is an LLM-derived embedding).

$$\begin{aligned} d_+^{sim} &= \underset{(x,y)\in\mathcal{S},y=AD}{\arg\max} \cos\big(\phi(d_{\text{test}}),\, \phi((x,y))\big), \\ d_-^{sim} &= \underset{(x,y)\in\mathcal{S},y=HC}{\arg\max} \cos\big(\phi(d_{\text{test}}),\, \phi((x,y))\big) \end{aligned} \tag{21}$$

- **Semantic Dissimilarity:** Pair with the lowest cosine similarity.

$$\begin{aligned} d_+^{dis} &= \underset{(x,y)\in\mathcal{S},y=AD}{\arg\min} \cos\big(\phi(d_{\text{test}}),\, \phi((x,y))\big), \\ d_-^{dis} &= \underset{(x,y)\in\mathcal{S},y=HC}{\arg\min} \cos\big(\phi(d_{\text{test}}),\, \phi((x,y))\big) \end{aligned} \tag{22}$$

- **Length Similarity:** Pair whose text length is closest to d_{test}.

$$\begin{aligned} d_+^{sim\ell} &= \underset{(x,y)\in\mathcal{S},y=AD}{\arg\min} \big|\ell(d_{\text{test}}) - \ell((x,y))\big|, \\ d_-^{sim\ell} &= \underset{(x,y)\in\mathcal{S},y=HC}{\arg\min} \big|\ell(d_{\text{test}}) - \ell((x,y))\big| \end{aligned} \tag{23}$$

- **Length Dissimilarity:** Pair whose length is most distant.

$$\begin{aligned} d_+^{dis\ell} &= \underset{(x,y)\in\mathcal{S},y=AD}{\arg\max} \big|\ell(d_{\text{test}}) - \ell((x,y))\big|, \\ d_-^{dis\ell} &= \underset{(x,y)\in\mathcal{S},y=HC}{\arg\max} \big|\ell(d_{\text{test}}) - \ell((x,y))\big| \end{aligned} \tag{24}$$

Here, $\ell(d)$ denotes the sequence length of demo d. Together, the selected demos form the main demo set:

$$\mathcal{D}_{\text{main}}(d_{test}) = \{d_+^{sim}, d_-^{sim}, d_+^{dis}, d_-^{dis}, d_+^{sim\ell}, d_-^{sim\ell}, d_+^{dis\ell}, d_-^{dis\ell}\}. \tag{25}$$

Stage 2: Sub-Demo Set Construction. For each main demo $d_i \in \mathcal{D}_{\text{main}}$ from Stage 1, we construct a *sub-demo set* to extract a demo-centric TV. The retrieval criterion is

same with Stage 1. Concretely, for a main demo d_{main}, we select:

$$\mathcal{D}_{\text{sub}}(d_i) = \{d_+^{sim}, d_-^{sim}, d_+^{dis}, d_-^{dis}, d_+^{sim\ell}, d_-^{sim\ell}, d_+^{dis\ell}, d_-^{dis\ell}\}. \tag{26}$$

As Fig. 9 shows, repeated for each dimension and balanced across AD/HC labels, yields a diverse set of 8 demos per test sample, providing a rich, contrastive context by covering the semantic and structural extremes of the demo space. All similarities are computed directly using the final-layer hidden representations of the LLM, ensuring alignment with the model's own reasoning space and avoiding external embedding mismatch. This design enables the prompt to cover both "close" and "distant" neighbors in both semantic and structural spaces, ensuring that the few-shot context exposes the model to a spectrum of possible features, mitigating overfitting and improving generalization.

Hyperparameters	Value
Maximum Input Length	1024 tokens
Maximum New Tokens	105 tokens
Temperature	0.95
Top-p Sampling	0.7
Batch Size	2

Table 7: Hyperparameters for LoRA fine-tuning and evaluation.

D. Supplementary Experiments

D.1 Ablation Results for PVA Module Tab. 6 presents the ablation study for the Projected Vector Anchoring (PVA) module under various demo retrieval settings (4 demo pairs) on Test, Lu, and Pitt datasets. Statistical significance is indicated for each variant.

D.2 Full Evaluation Metrics Tab. 14 reports the complete evaluation results (Precision, Recall, F1-score, and Accuracy) for all methods across the Test, Lu, and Pitt datasets, with varying numbers of demonstrations (N). Each entry shows the mean and standard deviation over multiple runs.

D.3 Comparison of Supervised Baselines As an additional comparison, Tab. 8 reports representative supervised baselines alongside the DA4ICL result. All reported results are based on the best-performing run.

Fine-tuning Methods 1) BERT (coarse-grained) (Balagopalan et al. 2021): Utilizes the CLS token for classification; 2) BERT (fine-grained) (Balagopalan et al. 2021): Extracts representations via GlobalMaxPooling; 3) SPZ (Li et al. 2024b): Employs semantic perturbations and zonal-mixing to enhance robustness; 4) LoRA (CLS) (Hu et al. 2022): Applies low-rank adaptation for parameter-efficient fine-tuning on LLM under classification task; 5) LoRA (GEN) (Hu et al. 2022): Applies low-rank adaptation for parameter-efficient fine-tuning on LLM under text generation task.

Detailed Information of LoRA This section provides details of LoRA method used in our experiments. LoRA is evaluated under two supervised paradigms: **classification tasks** and **generation tasks**, to assess its adaptability in AD detection scenarios. Hyperparameters and task-specific configurations are detailed below.

Hyperparameters. For hyperparameters like *lora-rank (γ)*, *lora-alpha (α)* and *target modules*[1], we use default settings to ensure consistency and robustness across experiments. Tab. 7 summarizes the hyperparameters used for fine-tuning and evaluating LoRA in both task paradigms.

LoRA Classification Task Setup. In the classification task, LoRA is trained to differentiate between AD and normal cognitive states based on textual descriptions of the "Cookie Theft" picture. Each data instance comprises an instruction, an input description, and an output label. The details of prompt construction in classification task are shown in Tab. 12.

LoRA Generation Task Setup. In the generation task, LoRA is supervised to produce textual descriptions based on specified mental states (e.g., alzheimer's disease or normal cognitive conditions). This paradigm evaluates the model's capacity to simulate descriptive outputs conditioned on cognitive states. The details of prompt construction in classification task are shown in Tab. 13.

The classification-based LoRA evaluates the LLMs' ability to differentiate between AD and control transcripts by assigning discrete labels, while the generation-based LoRA assesses its capacity to simulate descriptive patterns under specified cognitive conditions. These complementary configurations ensure a robust evaluation of LoRA's adaptability and effectiveness across diverse tasks relevant to AD detection.

Analysis of Results Comparison with supervised SOTA. While our scope targets LLM adaptation under limited task cognition and weak contextual perception (hence the emphasis on ICL/TV families), we additionally benchmark against representative supervised baselines (Tab. 8) to address practical utility. On *Test* dataset, DA4ICL attains second-best Accuracy/F1, trailing only SPZ, and on *Lu* corpus, it matches the best Accuracy and remains near-best F1. On the large OOD *Pitt* corpus, it achieves the top Accuracy and F1. These results indicate that, despite using a frozen backbone with only 8-shot prompting (no gradient updates), DA4ICL is competitive with fine-tuned encoders (BERT variants) and strong data augmentation methods (SPZ), and surpasses LoRA fine-tuning settings under both classification and generation paradigms. In short, DA4ICL narrows the gap to supervised SOTA on ID-style splits and provides stronger transfer on OOD corpora, a property that is critical for real clinical deployment where domain shift is the norm.

Why this matters for medical use. Supervised pipelines typically require labeled training, multiple training runs, and careful regularization to avoid overfitting to specific corpus. DA4ICL instead operates in a label-efficient, training-free

[1]Here we set $\gamma = 16$, $\alpha = 8$ and set all attention modules (Q,K,V,O) as target modules.

Method	Shot	Test (%)				Lu (%)				Pitt (%)			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
BERT$_c$	N/A	81.25	94.11	66.67	78.05	**83.33**	**85.71**	88.89	87.27	70.12	82.87	58.49	68.58
BERT$_f$	N/A	81.25	82.61	79.17	80.85	80.95	80.64	92.59	86.21	77.05	85.16	71.24	77.58
SPZ	N/A	**91.66**	**95.45**	87.50	**91.30**	**83.33**	83.33	92.59	**87.72**	77.96	**88.07**	69.93	77.96
LoRA$_{cls}$	N/A	60.40	55.81	**100.00**	71.64	64.29	65.00	**96.29**	77.61	57.92	57.19	**97.39**	72.07
LoRA$_{gen}$	N/A	41.67	43.33	54.17	48.15	47.62	60.87	51.85	56.00	51.18	57.19	49.35	52.98
DA4ICL (Ours, best)	8	89.58	88.00	91.67	89.80	**83.33**	83.33	92.59	87.22	**80.33**	86.67	76.47	**81.25**

Table 8: Representative supervised baselines on Test / Lu / Pitt, and DA4ICL result. Bold entries denote the best result, and underlined entries denote the second-best.

Setting	Variant	Elapsed (s)	Peak Alloc (MB)	Peak Reserved (MB)
48GB GPU				
Vanilla ICL	–	15.65 ± 0.27	16160.05 ± 0.54	16364.80 ± 146.40
Semantic ICL	–	15.72 ± 0.18	16160.05 ± 0.54	16364.80 ± 146.40
DCR ICL	–	44.99 ± 1.95	16677.95 ± 0.17	16904.60 ± 16.20
Vanilla TV	Add	60.47 ± 1.24	18618.89 ± 225.60	21126.00 ± 997.70
	Replace	63.17 ± 5.18	18567.99 ± 134.72	21090.80 ± 1385.58
Semantic TV	Add	79.13 ± 1.75	18630.14 ± 141.89	22666.00 ± 942.08
	Replace	71.29 ± 4.92	18594.08 ± 120.34	22926.80 ± 1648.29
DCR TV	Add	80.24 ± 2.17	19370.89 ± 118.89	21107.60 ± 640.76
	Replace	70.08 ± 7.29	19304.88 ± 67.69	21458.20 ± 971.24
DA4ICL	–	521.63 ± 42.68	27865.15 ± 0.00	31651.80 ± 17.40

Table 9: Runtime and peak GPU memory on a 48GB GPU. Mean $\pm$ SD over 10 runs.

Setting	Variant	Elapsed (s)	Peak Alloc (MB)	Peak Reserved (MB)
80GB GPU				
Vanilla ICL	–	12.64 ± 2.17	16501.42 ± 29.30	17053.00 ± 226.22
Semantic ICL	–	20.36 ± 0.74	16545.22 ± 16.65	17033.80 ± 135.46
DCR ICL	–	26.35 ± 0.35	16678.37 ± 1.16	16971.60 ± 10.80
Vanilla TV	Add	35.64 ± 0.94	17477.90 ± 105.37	19094.40 ± 556.18
	Replace	35.40 ± 0.58	17428.73 ± 94.81	19142.20 ± 645.57
Semantic TV	Add	36.65 ± 0.96	17442.27 ± 38.88	19734.00 ± 492.66
	Replace	37.14 ± 1.10	17463.38 ± 61.94	19531.60 ± 532.08
DCR TV	Add	41.20 ± 0.61	17897.43 ± 105.93	19091.20 ± 490.37
	Replace	41.33 ± 0.83	17866.03 ± 52.96	18852.60 ± 329.37
DA4ICL	–	283.72 ± 2.18	26299.99 ± 2.81	27813.00 ± 279.34

Table 10: Runtime and peak GPU memory on an 80GB GPU. Mean $\pm$ SD over 10 runs.

alternative, where DCR widens salient cues in the prompt space while PVA deepens multi-layer context integration at anchor positions, yielding better cross-corpus calibration without additional training data or parameters updates. Practically, this reduces development friction (e.g., no re-training when moving across clinics/corpora) and lowers the risk of distributional mismatch.

Complementary strengths. Encoder-based supervised baselines (e.g., BERT$_c$/BERT$_f$) excel when train/test share distribution, leveraging discriminative representations learned from labels; DA4ICL's gains stem from improved *context routing* inside a frozen LLM, which is inherently robust to label scarcity and lexical drift. The patterns in Tab. 8 reflect this complementarity: SPZ leads on smaller, cleaner splits (*Test*), while DA4ICL dominates on *Pitt* where stylistic and demographic variability amplify OOD shift.

Limitations and scope. Our supervised comparisons focus on widely used text-based baselines (BERT family,

Dataset	Method	Run	N	Errors	FN (1→0)	FP (0→1)	Errors (%)	FN (%)	FP (%)
Test	ICL Van	1	48	13	4	9	27.08	8.33	18.75
Test	ICL Van	2	48	13	0	13	27.08	0.00	27.08
Test	DA4ICL	1	48	5	2	3	10.42	4.17	6.25
Test	DA4ICL	2	48	7	3	4	14.58	6.25	8.33
Lu	ICL Van	1	32	9	2	7	28.12	6.25	21.88
Lu	ICL Van	2	32	12	5	7	37.50	15.62	21.88
Lu	DA4ICL	1	32	7	2	5	21.88	6.25	15.62
Lu	DA4ICL	2	32	7	2	5	21.88	6.25	15.62
Pitt	ICL Van	1	549	152	66	86	27.69	12.02	15.67
Pitt	ICL Van	2	549	162	78	84	29.60	14.21	15.30
Pitt	DA4ICL	1	549	108	72	36	19.67	13.12	6.56
Pitt	DA4ICL	2	549	114	80	34	20.76	14.57	6.19

Table 11: Top-2 runs error details per dataset and method. Percentages are relative to N.

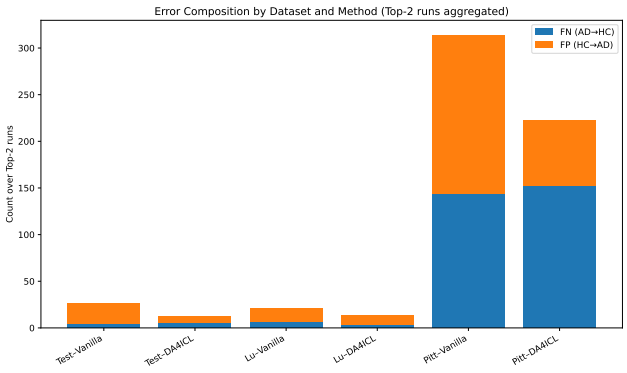

Figure 10: Error composition across datasets for the Top-2 performing runs of Vanilla ICL and DA4ICL. Each bar represents the sum of false negatives (FN, AD→HC) and false positives (FP, HC→AD) across Top-2 performing runs. DA4ICL exhibits a clear reduction in total errors and a more symmetric FN–FP distribution across all datasets, indicating improved contextual calibration and robustness, particularly under out-of-distribution conditions (Lu and Pitt).

SPZ) and parameter-efficient LLM adaptation (LoRA), yet broader clinical SOTA also includes multi-modal systems and task-specific Transformers. A full audit of such systems, especially those using audio cues, falls outside our text-only scope but is a valuable direction. Integrating paralinguistics and multi-modal descriptors into the DCR&PVA framework and expand cross-corpus validation to additional clinical cohorts to further probe generalization under real-world shift is promising in future works.

D.4 Efficiency and Resource Analysis To assess computational overhead and scalability, we measured wall-clock runtime and peak GPU memory during inference for all baselines and our proposed DA4ICL under identical hardware conditions (48GB and 80GB NVIDIA GPUs). All measurements were obtained using `torch.cuda.max_memory_allocated()` and `max_memory_reserved()` after resetting CUDA memory statistics at the start of each run. As summarized in Tab. 9 and Tab. 10, standard ICL variants (Vanilla, Semantic retrieval) exhibit lightweight computational overhead, requiring approximately 16–17 GB of allocated memory and 15–26 s of runtime. DCR retrieval introduces additional retrieval and prompt-assembly cost, slightly increasing inference time but maintaining comparable memory usage. Task Vector (TV) methods incur higher resource demand due to per-layer injection and fusion operations, leading to 18–19 GB allocation and roughly double runtime relative to pure ICL. In contrast, the full DA4ICL pipeline, combining multi-layer projection, demo-centric anchoring, and retrieval, represents the most resource-intensive configuration (up to 27.9 GB on 48GB GPU and 26.3 GB on 80GB GPU), yet remains stable on a single GPU. These results confirm that DA4ICL achieves substantial performance gains with moderate and manageable computational overhead, preserving practical deployability in real-world low-resource inference settings.

D.5 Error Analysis To further examine robustness and failure patterns, we analyzed the **Top-2 performing runs** of both Vanilla ICL and DA4ICL on the *Test*, *Lu*, and *Pitt* datasets (See Fig. 10 and Tab. 11). Each dataset was evaluated by counting (i) total misclassifications and (ii) the direction of error: false negatives (AD→HC) and false positives (HC→AD).

Overall Trends. DA4ICL consistently reduced total error counts across all datasets and runs. On the *Test* set (48 samples), total errors dropped from 13/13 under Vanilla ICL to 5/7 under DA4ICL; on *Lu* (32 samples), from 9/12 to 7/7; and on the large-scale *Pitt* corpus (549 samples), from 152/162 to 108/114. This reduction demonstrates improved stability and generalization, especially under out-of-distribution (OOD) shifts.

Error Direction Asymmetry. Vanilla ICL exhibited a strong asymmetry in error direction, disproportionately over-predicting the positive (AD) class. For instance, in the *Test* runs, false positives (HC→AD) occurred 9/13 times versus only 4/0 false negatives, suggesting an over-reliance on pathological lexical cues. Similarly, on *Pitt*, false pos-

itives (86/84) exceeded false negatives (66/78). In contrast, DA4ICL largely balanced this distribution: e.g., on *Test*, errors were nearly symmetric (2–3 AD→HC vs. 3–4 HC→AD), and on *Lu*, the ratios matched exactly (2–2 vs. 5–5). This balance indicates that the proposed democentric anchoring improves contextual calibration rather than merely shifting the decision boundary.

Qualitative findings. Beyond aggregate counts, the Top-2 misclassified transcripts reveal three recurring failure patterns that persist even under DA4ICL.

- ***Lexical stereotype bias***: utterances richly mentioning canonical cues (overflowing sink, tipping stool, cookie jar) with otherwise fluent, orderly narration are sometimes predicted as HC even when their microstructure shows AD-like markers (frequent fillers/repairs "uh/hm...", local incoherence, object substitutions such as *ladder* for stool). This drives residual AD→HC errors: when surface content units align strongly with the picture's **gold schema**, the model's decision can be dominated by content coverage rather than form.
- ***Noisy idiosyncrasy as pathology***: rare insertions or off-picture intrusions (e.g., joking or fantastical clauses like a "race horse jumping through the window"), plus pet phrases ("what the devil is xxx?"), inflate HC→AD false positives by mimicking semantic drift and pragmatic breakdown. DA4ICL reduces but does not eliminate this tendency.
- ***Information density vs. sequencing***: very concise, list-like descriptions that enumerate entities without linking actions/relations (weak event chaining, minimal causal connectives) remain ambiguous. When the final anchor token aggregates a short, weakly structured context, both ICL and DA4ICL can be brittle to small lexical fluctuations, yielding paired FP/FN flips across runs.

Prompt Component	Details
System Prompt	You are an experienced clinician specializing in dementia care. You will analyze descriptions provided by subjects who are describing a scene from the "Cookie Theft" picture, used in the Boston Diagnostic Aphasia Examination. Your task is to analyze the subject's description to determine whether it indicates Alzheimer or Normal. Provide your answer in the format: **[A. Alzheimer Disease]** or **[B.Control]**.
Input	Here is the example: [I see uh two kids up at the cookie jar, one on a stool the other standing on the floor. cupboard door is opened. mother's washing the dishes. the water is running overflowing the sink. and uh there's two cups and a plate on the counter. and she's washing holding a plate in her hand. curtains at the windows. the cookie jar has the lid off. hm hm that's about it. cupboards underneath the sink. cupboards underneath the other cupboards. uh kid falling off the stool. the girl laughing at him. cookies in the cookie jar with the lid off. he has a cookie in his hand. and that's it.]
Output	The description is identified as **[B.Control]**.

Table 12: LoRA setup for classification tasks.

Prompt Component	Content Example
System Prompt	You are a volunteer for a picture description task. Below is the given comprehensive description of the task: "The picture depicts a chaotic kitchen scene. On the left, a young girl is pointing upwards towards a boy who is standing on a stool and reaching for a cookie jar labeled 'COOKIE JAR' in an open cupboard. The boy is losing his balance and appears to be falling. In the center, a woman is calmly washing dishes at a sink that is overflowing with water, which is spilling onto the floor. To the right, there is a counter with plates and a bowl. Outside the window above the sink, there are trees and possibly a house in the distance." You are required to describe the picture based on your specified mental condition.
Input	**AD Instruction** ['You are in dementia, and you are required to describe the picture.', 'You are an Alzheimerś patient, and you are required to describe the picture.', 'You are a patient with Alzheimerś disease, and you are required to describe the picture.', 'You are a patient with dementia, and you are required to describe the picture.', 'You are a patient with Alzheimerś, and you are required to describe the picture.', 'You are a patient with dementia, and you are required to describe the picture.', 'You are a patient with Alzheimerś disease, and you are required to describe the picture.', 'You are a patient with Alzheimerś, and you are required to describe the picture.', 'You are a patient with dementia, and you are required to describe the picture.', 'You are a patient with Alzheimerś disease, and you are required to describe the picture.'] **Control Instruction** *['You are mentally healthy, and you are required to describe the picture.', 'You are a normal person, and you are required to describe the picture.', 'You are a person with normal mental health, and you are required to describe the picture.']* You can start your description now.
Output	#Certain Transcripts in Training Dataset#

Table 13: LoRA setup for generation tasks.

Methods	N	Test				Lu				Pitt			
		Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc
ICL$_{Van}$	0	$60.73_{9.36}$	$45.00_{7.17}$	$51.50_{7.34}$	$57.71_{6.46}$	$67.45_{7.08}$	$60.00_{8.08}$	$63.41_{7.24}$	$55.71_{8.05}$	$63.28_{2.34}$	$49.02_{1.97}$	$55.23_{2.00}$	$55.72_{1.96}$
	1	$66.07_{5.57}$	$73.33_{5.65}$	$69.33_{4.22}$	$67.50_{5.12}$	$70.46_{2.82}$	$83.70_{5.02}$	$76.45_{3.03}$	$66.90_{4.05}$	$67.03_{1.03}$	$67.35_{1.58}$	$67.18_{1.11}$	$63.33_{1.12}$
	2	$65.15_{3.34}$	$72.08_{6.19}$	$68.29_{3.71}$	$66.67_{3.36}$	$73.28_{3.55}$	$86.67_{4.74}$	$79.31_{2.86}$	$70.95_{4.10}$	$68.87_{0.91}$	$66.67_{2.13}$	$67.74_{1.35}$	$64.63_{1.15}$
	3	$66.01_{2.86}$	$76.25_{3.75}$	$70.66_{1.83}$	$68.33_{2.43}$	$72.82_{4.46}$	$88.52_{5.09}$	$79.83_{4.10}$	$71.19_{6.25}$	$69.21_{1.46}$	$68.43_{1.61}$	$68.81_{1.40}$	$65.43_{1.55}$
	4	$67.31_{3.48}$	$77.92_{6.47}$	$72.13_{4.15}$	$70.00_{4.08}$	$72.80_{3.74}$	$87.78_{4.70}$	$79.53_{3.58}$	$70.95_{5.08}$	$69.54_{1.14}$	$70.98_{1.56}$	$70.24_{1.12}$	$66.48_{1.20}$
ICL$_{Sem}$	1	$64.23_{3.46}$	$74.58_{5.73}$	$68.93_{3.88}$	$66.46_{4.00}$	$72.99_{2.50}$	$83.70_{5.02}$	$77.89_{2.69}$	$69.52_{3.33}$	$68.49_{1.16}$	$66.73_{1.73}$	$67.59_{1.21}$	$64.34_{1.21}$
	2	$65.36_{5.38}$	$72.92_{7.74}$	$68.70_{5.14}$	$66.88_{5.39}$	$71.84_{2.77}$	$85.56_{4.81}$	$78.01_{2.65}$	$69.05_{3.53}$	$69.93_{1.73}$	$66.57_{2.68}$	$68.18_{1.83}$	$65.39_{1.79}$
	3	$68.37_{5.90}$	$80.42_{6.19}$	$73.71_{4.73}$	$71.25_{5.65}$	$69.92_{1.98}$	$90.37_{4.12}$	$78.81_{2.54}$	$68.81_{3.44}$	$70.98_{1.06}$	$72.45_{1.36}$	$71.70_{0.83}$	$68.12_{0.93}$
	4	$67.21_{4.82}$	$83.33_{6.45}$	$74.20_{3.69}$	$71.04_{4.32}$	$72.54_{2.79}$	$94.44_{2.48}$	$82.02_{2.11}$	$73.33_{3.50}$	$71.25_{0.99}$	$76.57_{2.33}$	$73.79_{1.21}$	$69.71_{1.13}$
ICL$_{Ens}$	0	$62.57_{7.56}$	$49.58_{10.28}$	$54.71_{7.20}$	$59.58_{5.76}$	$68.03_{4.26}$	$60.00_{7.73}$	$63.59_{5.59}$	$56.19_{5.65}$	$64.73_{2.34}$	$49.84_{2.95}$	$56.30_{2.67}$	$56.92_{2.21}$
	1	$61.63_{5.76}$	$67.92_{6.19}$	$64.43_{4.62}$	$62.50_{5.19}$	$71.33_{3.34}$	$85.56_{6.30}$	$77.71_{3.99}$	$68.57_{5.19}$	$66.93_{1.41}$	$67.91_{2.49}$	$67.41_{1.89}$	$63.42_{1.83}$
	2	$68.70_{3.72}$	$74.58_{5.42}$	$71.42_{3.71}$	$70.21_{3.73}$	$72.02_{3.06}$	$87.41_{4.74}$	$78.91_{3.12}$	$70.00_{4.29}$	$70.04_{1.57}$	$68.20_{1.39}$	$69.11_{1.40}$	$66.01_{1.59}$
	3	$67.47_{4.84}$	$73.33_{6.77}$	$70.20_{5.24}$	$68.96_{5.31}$	$73.23_{2.52}$	$89.26_{4.52}$	$80.44_{3.24}$	$72.14_{4.27}$	$70.10_{1.06}$	$68.82_{1.57}$	$69.44_{0.93}$	$66.25_{0.95}$
	4	$66.75_{5.05}$	$86.67_{4.08}$	$75.32_{4.08}$	$71.46_{5.44}$	$72.16_{3.73}$	$89.63_{4.32}$	$79.88_{3.28}$	$70.95_{4.97}$	$70.93_{1.16}$	$70.42_{1.69}$	$70.66_{1.07}$	$67.41_{1.10}$
TV$^{Add}_{Van}$	1	$55.29_{6.44}$	$47.92_{7.28}$	$51.23_{6.62}$	$54.58_{5.88}$	$67.78_{6.05}$	$55.93_{8.83}$	$61.10_{7.41}$	$54.76_{6.82}$	$62.01_{1.64}$	$50.00_{3.01}$	$55.33_{2.24}$	$55.06_{1.62}$
	2	$59.64_{6.13}$	$50.42_{5.73}$	$54.60_{5.77}$	$58.13_{5.22}$	$68.22_{4.91}$	$57.41_{8.49}$	$62.12_{6.34}$	$55.48_{6.03}$	$60.93_{2.32}$	$48.30_{2.66}$	$53.86_{2.35}$	$53.92_{2.11}$
	3	$57.79_{8.36}$	$46.25_{6.57}$	$51.04_{6.17}$	$55.62_{6.79}$	$67.37_{4.76}$	$51.85_{8.28}$	$58.42_{6.72}$	$53.10_{5.84}$	$60.95_{1.22}$	$50.36_{3.41}$	$55.10_{2.33}$	$54.35_{1.37}$
	4	$57.89_{6.25}$	$51.67_{5.65}$	$54.47_{5.21}$	$56.88_{4.85}$	$68.39_{7.75}$	$57.04_{8.31}$	$62.06_{7.81}$	$55.48_{8.11}$	$60.56_{2.35}$	$47.88_{2.72}$	$53.45_{2.44}$	$53.57_{2.09}$
TV$^{Add}_{Sem}$	1	$58.82_{7.06}$	$48.33_{2.53}$	$52.28_{8.33}$	$57.08_{5.12}$	$72.66_{5.87}$	$58.89_{8.52}$	$64.79_{6.62}$	$59.29_{6.43}$	$61.26_{1.78}$	$49.44_{1.83}$	$54.71_{1.72}$	$54.39_{1.62}$
	2	$64.11_{7.93}$	$50.42_{10.45}$	$56.02_{8.84}$	$61.04_{6.72}$	$69.37_{4.21}$	$60.74_{6.67}$	$64.54_{4.53}$	$57.38_{4.32}$	$60.57_{2.77}$	$49.25_{2.37}$	$54.31_{2.38}$	$53.83_{2.44}$
	3	$58.64_{5.98}$	$46.25_{9.58}$	$51.44_{7.95}$	$57.08_{4.86}$	$67.11_{3.80}$	$57.04_{6.67}$	$61.55_{5.31}$	$54.52_{4.70}$	$60.50_{2.52}$	$49.18_{3.07}$	$54.22_{2.65}$	$53.77_{2.31}$
	4	$61.47_{8.01}$	$52.08_{8.39}$	$56.31_{7.99}$	$59.79_{6.85}$	$68.96_{4.64}$	$56.30_{6.58}$	$61.91_{5.65}$	$55.71_{5.65}$	$60.10_{2.11}$	$46.80_{2.08}$	$52.60_{1.80}$	$53.01_{1.71}$
TV$^{Rep}_{Van}$	1	$60.34_{2.72}$	$75.83_{5.83}$	$67.11_{3.12}$	$62.92_{3.46}$	$68.80_{3.24}$	$83.33_{6.47}$	$75.31_{4.16}$	$65.00_{5.43}$	$61.89_{1.21}$	$72.88_{1.83}$	$66.92_{1.21}$	$59.85_{1.43}$
	2	$62.72_{4.08}$	$75.42_{6.02}$	$68.40_{4.27}$	$65.21_{4.66}$	$70.32_{2.4}$	$82.96_{3.39}$	$76.06_{1.71}$	$66.43_{2.49}$	$61.21_{0.96}$	$72.35_{1.74}$	$66.31_{1.15}$	$59.03_{1.27}$
	3	$59.25_{3.55}$	$77.08_{7.28}$	$66.94_{4.72}$	$62.08_{4.82}$	$70.46_{3.35}$	$84.44_{5.19}$	$76.75_{3.43}$	$67.14_{4.86}$	$61.53_{1.30}$	$73.79_{2.03}$	$67.10_{1.44}$	$59.67_{1.71}$
	4	$60.71_{4.91}$	$75.00_{4.93}$	$66.97_{3.96}$	$62.92_{5.34}$	$67.64_{3.25}$	$82.96_{6.46}$	$74.46_{4.28}$	$63.57_{5.33}$	$62.07_{0.72}$	$73.86_{1.57}$	$67.45_{1.02}$	$60.27_{1.06}$
TV$^{Rep}_{Sem}$	1	$59.85_{5.53}$	$71.68_{7.64}$	$65.00_{4.94}$	$61.46_{5.76}$	$68.09_{2.09}$	$81.48_{6.42}$	$74.10_{3.48}$	$63.57_{3.99}$	$61.55_{0.75}$	$73.53_{1.00}$	$67.00_{0.72}$	$59.64_{0.91}$
	2	$57.04_{2.16}$	$74.17_{4.49}$	$64.46_{2.86}$	$59.17_{2.83}$	$67.68_{2.69}$	$82.96_{4.74}$	$74.51_{3.19}$	$63.57_{4.27}$	$62.31_{1.40}$	$73.99_{2.58}$	$67.63_{1.63}$	$60.55_{1.82}$
	3	$61.95_{4.12}$	$80.00_{8.90}$	$69.68_{5.40}$	$65.42_{5.45}$	$67.50_{2.85}$	$81.48_{5.74}$	$73.77_{3.54}$	$62.86_{4.54}$	$61.79_{1.25}$	$73.17_{1.64}$	$66.99_{1.23}$	$59.82_{1.55}$
	4	$57.47_{4.10}$	$72.92_{7.03}$	$64.15_{4.65}$	$59.38_{4.95}$	$69.56_{3.55}$	$85.19_{4.38}$	$76.54_{3.46}$	$66.43_{5.05}$	$62.14_{0.90}$	$74.22_{2.17}$	$67.64_{1.39}$	$60.44_{1.38}$
Ours	4	$\mathbf{84.55}_{3.02}$	$\mathbf{87.92}_{4.14}$	$\mathbf{86.11}_{1.92}$	$\mathbf{85.83}_{1.91}$	$\mathbf{81.08}_{2.22}$	$\mathbf{91.85}_{1.56}$	$\mathbf{86.12}_{1.73}$	$\mathbf{80.95}_{2.51}$	$\mathbf{86.34}_{0.50}$	$\mathbf{74.93}_{0.71}$	$\mathbf{80.23}_{0.42}$	$\mathbf{79.42}_{0.39}$

Table 14: Main experimental results on the Test, Lu, and Pitt datasets. Each entry reports mean_{std} over ten runs.