Token-based Decision Criteria Are Suboptimal in In-context Learning

Anonymous ACL submission

Abstract

In-**C**ontext **L**earning (ICL) typically utilizes classification criteria from probabilities of manually selected label tokens. However, we argue that such token-based classification criteria lead to suboptimal decision boundaries, despite delicate calibrations through translation and constrained rotation. To address this problem, we propose Hidden Calibration, which renounces token probabilities and uses the nearest centroid classifier on the LM's last hidden states. In detail, we use the nearest centroid classification on the hidden states, assigning the category of the nearest centroid previously observed from a few-shot calibration set to the test sample as the predicted label. Our experiments on 3 models and 10 classification datasets indicate that Hidden Calibration consistently outperforms current token-based calibrations by about 20%. Our further analysis demonstrates that Hidden Calibration finds better classification criteria with less inter-categories overlap, and LMs provide linearly separable intra-category clusters with the help of demonstrations, which supports Hidden Calibration and gives new insights into the conventional ICL.

1 Introduction

In-**c**ontext **L**earning (ICL) (Dong et al., 2022) is a few-shot learning paradigm without any model parameter updates, using **L**anguage **M**odels (LMs). In detail, as shown in Fig. 1-A, B, given a prompt consisting of demonstrations and a query, LMs assign probabilities to the candidate label tokens as the next token of the prompt and choose the label with the highest probability as the prediction. Typically, label tokens are manually selected.

One well-known issue with ICL is that the predicted probabilities are biased (*under-calibrated*), leading to a decrease in prediction performance (Fei et al., 2023; Han et al., 2022; Zhao et al., 2021; Zhou et al., 2023). To solve this issue, previous work calibrates the predicted next token probabilities of the label tokens, performing estimated affine transformations to adjust these probabilities for more faithful predictions.

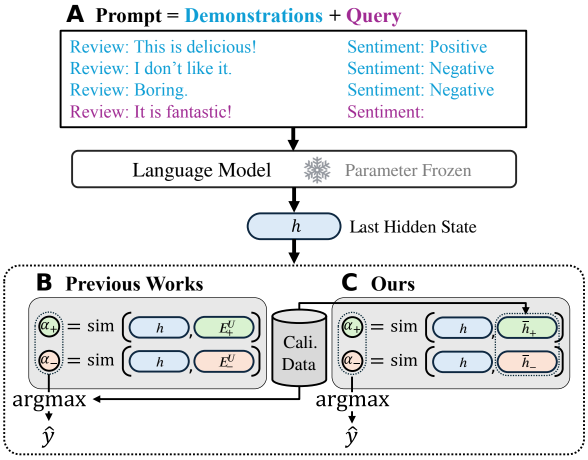


Figure 1: In an ICL diagram, **A.** The prompt of ICL consists of a combination of demonstrations and a query. After encoding the prompt into the last hidden state h, **B.** Previous works use the un-embedding vectors of the label tokens (E^U_+ and E^U_-) to decode the h for prediction, then calibrations are used to adjust the predicted logits. **C.** Our work uses the calibration dataset to calculate centroids ($\bar{h}_+$ and $\bar{h}_-$) to decode the h.

These works are based on a potential assumption: the affine manifolds spanned by the un-embedding vectors[1] of *manually selected* label tokens are good subspaces of the hidden space to distinguish them appropriately, and the output label token probabilities decoded from these subspaces are good classification criteria. However, some practices have pointed out that randomly changing label spaces doesn't critically influence ICL performance (Min et al., 2022c; Wei et al., 2023), which means the selected label subspaces are trivial and unfaithful. Although using the natural label un-embedding seems intuitive, it should be noted that we have no reason to believe that these subspaces have any guarantee

[1] The row vectors in the parameters matrix of LM head.

for good decision boundaries, to allow the coordinate (i.e., label token probabilities) as classification criteria, even if various delicate calibrations are used to move these boundaries *in the subspaces*. This makes us suspect that: **Utilizing manually selected label probabilities as classification criteria may not be good ICL practices**.

Previous work has shown that using the output probabilities of the *full vocabulary* increases ICL performance (Xu et al., 2022; Abbas et al., 2024). This is a good start to avoid the manually selected mapping subspace, but we still think that output probability distributions are inefficient as classification criteria. Therefore, in this paper, we utilize the last hidden states, which are high-ranking and informative precursors of the token probabilities.

In detail, we propose Hidden Calibration, utilizing the spatial pattern of the last hidden states for ICL. **During the training**, we build standard ICL prompts similarly to Fig. 1-A from a calibration set. Inputting them into an LM, we can get the last hidden states of the last tokens. Then, we calculate the centroids of the last hidden states with the same query label, so that we can get a centroid for each label. **During the inference**, we export the last hidden state of the test prompt and find the nearest centroids. The category of the nearest centroid is assigned to the query as the prediction.

Statistically, our method improves the performance of ICL by approximately 20% on 10 text classification datasets and 3 modern LMs. To the best of the author's knowledge, Hidden Calibration consistently outperforms the calibration baselines, achieving a strong state-of-the-art in ICL, with an equal computational cost with previous label probability calibrations.

Our subsequent analysis indicates that Hidden Calibration does find better mapping subspaces that effectively separate data points. In detail, we find that the kernel densities of criteria calculated from Hidden Calibration have less inter-category overlapping than from label probabilities, while such overlapping is proportional to the lower bound of the classification error. This suggests Hidden Calibration finds subspaces with essentially better classification performance.

Moreover, we investigate the principle of Hidden Calibration, that is, the reason why a simple centroid-based linear decision boundary can divide the ICL hidden state properly. We find that LMs provide linearly separable clusters in the hidden states w.r.t. query labels, while demonstration can promote such a process.

Our contributions can be summarized as:

- We analyze the previous calibration practices on ICL, and find their consistent limitations: Using manually selected labels as the projecting subspaces for classification criteria, which are often under-guaranteed.
- We propose Hidden Calibration to address such a problem, eliminating the unreliable human intuition from ICL prediction decoding by the nearest centroid classifier on hidden states instead of human-selected token-based probabilities. Our experiments suggest that Hidden Calibration is a new state-of-the-art.
- Our further analysis indicates that Hidden Calibration does find better classification criteria with less inter-categories overlap, and LMs provide linearly separable intra-category clusters with the help of demonstrations, which supports the Hidden Calibration.

2 Background

This section reviews previous work on ICL and denotes their mathematical descriptions.

2.1 In-context Learning

Prompting. Given a few-shot natural language classification dataset (*demonstration set*) $\mathcal{D} = \left\{\left(x^{(i)}, y^{(i)}\right) \in \mathcal{X} \times \mathcal{Y}\right\}_{i=1}^{n}$, where $x^{(i)}$ and $y^{(i)}$ are the input tokens and label token of i-th data point, and $\mathcal{X}, \mathcal{Y}$ is the input and label space, respectively, we sample a set of k samples $\mathcal{D}^{de} = \left\{x^{(c_i)}, y^{(c_i)}\right\}_{i=1}^{k}$ from $\mathcal{D}$ for a given query x^q. Then, we use a template T to concatenate them in a natural language form into a prompt token sequence: $s = T\left(\mathcal{D}^{de}, x^q\right)$, as shown in Fig. 1-A.

Encoding. A decoder-structured LM receives the prompt token sequence s and encodes it into the *last* (means on the last Transformer layer) hidden state matrix as $H \in \mathbb{R}^{|s| \times d}$ with a length of token $|s|$ and embedding dimension of d. We denote the hidden state of the last token as $h = H_{|s|} \in \mathbb{R}^d$.

Decoding. In a typical ICL setup, one chooses the un-embedding vectors of the label candidates in the output head[2] as the decoding subspace. For

[2] We omit the bias term in the output head (if any) for the sake of simplicity. It can be overridden by a fixed-to-one dimension, or covered by the calibration described below.

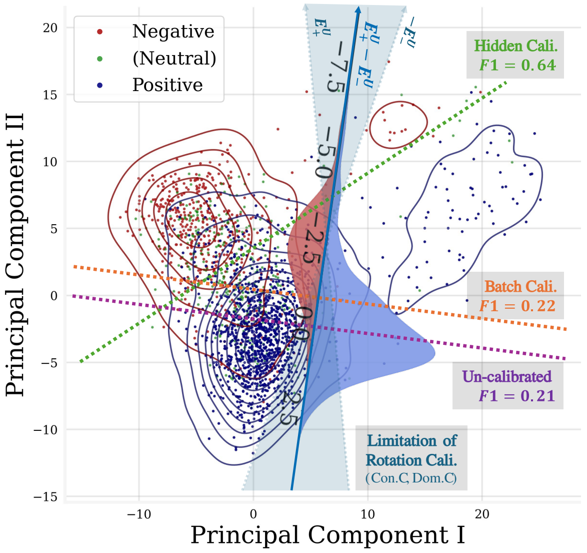


Figure 2: Token probability-based decision boundaries (original & batch calibrated) are suboptimal comparing to hidden state-based boundary. Points and contour lines are ICL's last hidden states and kernel densities mapped by Principal Component Analysis. Oblique coordinate axis is the direction of the un-embedding difference of label tokens $\left(E_+^U - E_-^U\right)$, where the kernel densities of mapped data points are plotted. The rotating calibration by $A \neq I$ (e.g. Contextual Calibration, Domain Calibration) has a limited feasible mapping direction[3].

each label l, the similarity $\alpha_l = \text{sim}(h, E_l^U)$ (usually the dot-product similarity) between h and each un-embedding vector E_l^U is calculated as the output classification criteria α_l (*logits*), as shown in Fig. 1-B for a binary classification example. Then, the category with the highest logits is chosen as the prediction $\hat{y}$, that is: $\hat{y} = \operatorname*{argmax}_{l \in \mathcal{Y}} \text{sim}(h, E_l^U)$.

2.2 Calibration for ICL

However, Zhao et al. (2021) find that simply using the original logits as classification criteria is not a good ICL practice. These logits have considerable prior bias and often tend towards specific labels even if the query is blank or meaningless (Zhao et al., 2021; Fei et al., 2023). Some calibrations have been proposed to mitigate such bias in a linear form: first, the logits are transformed into probabilities as $p = \text{softmax}([\alpha_1, \alpha_2, \ldots, \alpha_{|\mathcal{Y}|}])$, then affine-transformed as calibrated classification criteria $p' = A \odot p + B$, where $A, B \in \mathbb{R}^{|\mathcal{Y}|}$ is the calibration terms estimated from m examples (*calibration set*), and $\odot$ is the Hadamard multiplication. Many practices calculate the background biases by examples with pseudo queries and serving them as the scale terms (Fei et al., 2023; Zhao et al., 2021), while other practices calculate the biases by Gaussian estimation (Han et al., 2022) or the mean value of p during the inference (Zhou et al., 2023).

However, such calibrations are affine transformations on the label token probability, without modifying the E_l^U, allowing only translation of 0-point (by term B) and rotation inside the closure[3] of E_l^U (by term A) to the decoding subspace.

3 Methodology

Based on the above background, in this section, we demonstrate the limitations of the above calibrations, and then propose Hidden Calibration to address such limitations fundamentally.

3.1 Token Probabilities Are Not Good Classification Criteria

To better understand the limitations of the token probability-based paradigm, we show a prototype visualization of the hidden states of ICL prompts. Specifically, we encode ICL prompts (with $k = 8$) for 2,048 instances of SemEval 2014-Task 4 Restaurant (Pontiki et al., 2014) dataset using OPT-2.7B (Zhang et al., 2022) and plot the last hidden states of the last token on a 2D-Principal Component plane in Fig. 2 (See Appendix B.4 for details).

Focusing on the data points labeled "positive" and "negative", we plot the difference direction $\left(E_+^U - E_-^U\right)$ between the un-embedding vectors of these two label tokens[4]. Then, the coordinates of the mapped hidden states in such a direction are the classification criteria between these two labels[5], serving as the *token-based classification criteria*. In this visualized scenario, the original decision boundary is the orthogonal line at the zero point, the batch calibrated boundary (Zhou et al., 2023) is always parallel to it, and the other calibrations (Contextual Calibration (Zhao et al., 2021), Domain Calibration (Fei et al., 2023)) produces rotated mapping direction $\left(A_+E_+^U - A_-E_-^U\right)$, by

[3] In current practices, the A are calculated from reciprocals of probabilities, which are positive-definite (Note that the calibration is trivial when A is not positively definite: the category with negative A components will never be assigned), and usually do not have significant relative values.

[4] Notice that Principal Component Analysis is an orthogonal transformation, keeping the dot-product and normal line fixed (In fact, beyond orthogonal transformations, they are also centralized. Therefore, the projection axis does not necessarily pass through the coordinate origin). See Appendix B.4.

[5] In most scenarios, we use the dot-product similarity. So, when the classification criteria $\alpha_+ - \alpha_- = h(E_+^U - E_-^U)^T$ is greater than 0, a "positive" label is assigned, and vice versa.

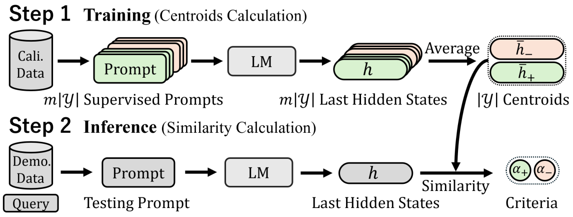

Step 1 Training (Centroids Calculation)
Cali. Data
Prompt
$m|\mathcal{Y}|$ Supervised Prompts
LM
h
$m|\mathcal{Y}|$ Last Hidden States
Average
$\bar{h}_-$
$\bar{h}_+$
$|\mathcal{Y}|$ Centroids
Step 2 Inference (Similarity Calculation)
Demo. Data
Query
Prompt
Testing Prompt
LM
h
Last Hidden States
Similarity
α_+
α_-
Criteria

Figure 3: The diagram of Hidden Calibration. **Step 1:** Calculating the hidden state centroid of each category. **Step 2:** Find the category of the nearest centroid of the text sample to be the prediction.

term A, where $A_+, A_- > 0$ limit[3] the direction inside the closure of E_+^U and $-E_-^U$.

Intuitively, the token-based mapping directions and decision boundaries cannot effectively classify these data points. It is due to the inherent direction of the token un-embedding vectors, regardless of calibration with affine transformation, where the boundary is moved almost parallelly. A straightforward better linear boundary is plotted as the green line, which can be calculated as the equidistant points between both categories' centroids.

3.2 Hidden Calibration

Motivated by the aforementioned limitation, we propose Hidden Calibration, using the centroid similarity as the classification criteria. The process is the same with the nearest centroid classifier, as shown in Fig. 3. First, as training, we calculate the centroid of the last hidden states of data points within each category on a calibration dataset. Then, in the inference, we select the closest centroid of the hidden state of the test sample as the prediction.

In detail, first, we conduct **(1). Training:** Given a calibration set with m supervised prompt-label pair $\left\{\left(s^{(i)}, y^{(i)}\right)\right\}_{i=1}^{m}$, where the $s^{(i)}$s are standard ICL prompts with k demonstrations, and $y^{(i)}$s are the ground-truth labels of corresponding $s^{(i)}$s' query. We use LMs to encode the prompts into last hidden states $h^{(i)}$ with the process mentioned in 2.1 and get a supervised hidden state set $\mathcal{H} = \left\{\left(h^{(i)}, y^{(i)}\right)\right\}_{i=1}^{m}$. Then, we calculate the centroids of category l as: $\bar{h}_l = \mathbb{E}_{\left(h^{(i)}, y^{(i)}\right) \in \mathcal{H}, y^{(i)}=l}\left[h^{(i)}\right]$.

Then, **(2). Inference:** Given a test query, we build an ICL prompt similar to the training step. Encoding it into the last hidden state h, we calculate the similarity[6] between h and every centroid $\bar{h}_l$ as the classification criteria α_l. The category with the highest similarity is assigned as the prediction.

Notice that another intuitive solution to the problem in §3.1 is utilizing the logits or probabilities of the *whole vocabulary* as a classification feature, as shown in previous works (Xu et al., 2022; Abbas et al., 2024). However, the dimensionality of the whole-vocabulary logits is much larger than hidden states, while the difference between them is only an input-irrelevant linear transformation, where no input-relevant information gain is obtained by such transformation. Therefore, we choose the hidden states, a high-rank and informative precursor of token probabilities, as the classification feature.

4 Experiments & Main Results

In this section, we prove the validity of Hidden Calibration by testing its classification performance on 3 models and 10 datasets. Hidden Calibration outperforms all the baselines, and reaches a strong state-of-the-art of ICL.

4.1 Experimental Settings

Models. We use 3 models: OPT-2.7B (Zhang et al., 2022), LLaMa 2 (Touvron et al., 2023), and GPT2-XL (Radford et al., 2019).

Baselines. We use 6 baselines from the previous works, with 4 label token-based methods (None, Con.C, Bat.C, and Dom.C) and 2 whole vocabulary probabilities-based methods (KNN and Cent.C). Details can be found in Appendix B.2.

Datasets. We use 10 commonly used classification datasets with some of the overlength data points excluded. See Appendix B.1 for details.

Other details. All the model checkpoints and datasets are loaded from `HuggingFace`. Macro F1 is used as the classification metric. We use a simple template to generate the prompt, see Appendix B.3. We set $m = 16|\mathcal{Y}|$ (16 examples per category) for calibration, and for the sake of fairness, every baseline method is given equal examples for calibration. All the experiments are repeated 5 times.

4.2 Main Results: Hidden Calibration is A New State-of-the-art of ICL

The classification performance of Hidden Calibration (Hidd.C) and baselines is shown in Fig. 4,

[6] Since it is found that the dot-production similarities in hidden space are unfaithful (Ethayarajh, 2019; Steck et al., 2024), in this practice, the additive inverse of Euclidean distance is used (that is, $\alpha_l = -\left\|h - \bar{h}_l\right\|_2^{\frac{1}{2}}$). However, Appendix D.1 shows that Hidden Calibration acts equally on both similarity measures.

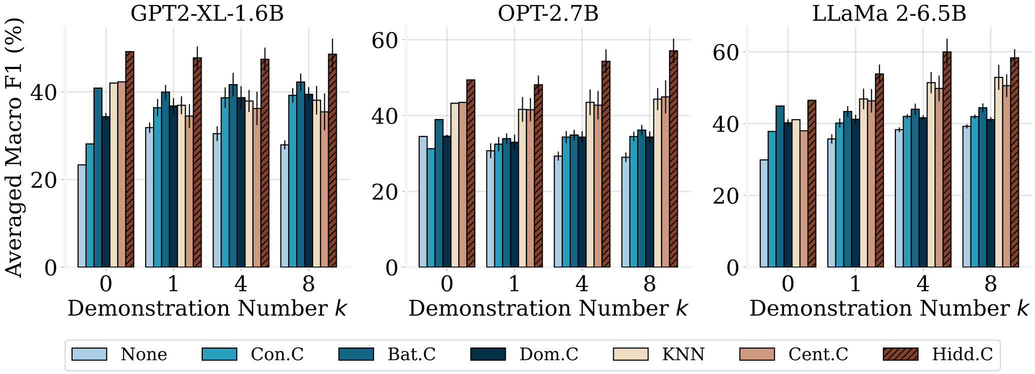


Figure 4: The classification performance (Macro F1(%)) of 3 models averaged on 10 datasets. Hidden Calibration (Hidd.C) is a new state-of-the-art of ICL, where demonstrations consistently improve the performance.

where Hidden Calibration consistently outperforms all the label token-based or vocabulary-based ICL methods. Comparing to the vanilla ICL (None), Hidden Calibration produces a improvement up to around 100%. In general, compared to the highest baseline, Hidden Calibration improves the performance by approximately 20%. Detailed numeric and Accuracy results are in Appendix. C.1.

Especially, compared to the Cent.C baseline proposed by us for a controlled trial, which conducts the same calculation but uses the whole output token probabilities instead of the hidden states, Hidden Calibration outperforms. This confirms our hypothesis that token probability distribution is a less informative classification feature, even if the human-selected label tokens are excluded.

Moreover, our method has little additional computational cost compared to the calibration baselines, making it highly efficient in time and space, as shown in Appendix D.2. Regarding data efficiency of calibration examples, §5.3 proves that even only 1 example per category can help.

Furthermore, in Fig. 4, we find that compared to baselines, Hidden Calibration seems to benefit from demonstrations consistently. In detail, the performances of Hidden Calibration have increasing patterns against the number of demonstrations, while most baselines do not perform similarly. We analyze such a phenomenon in §5.2.

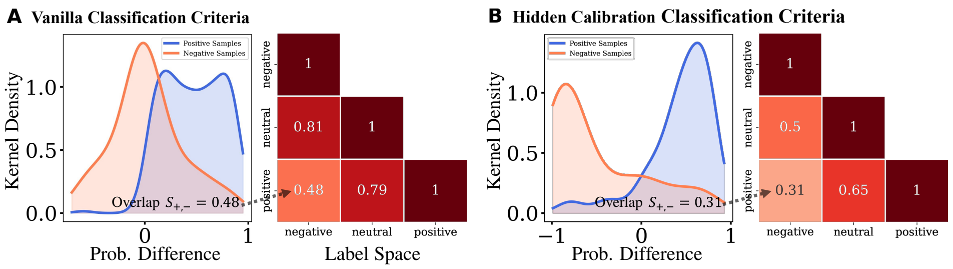
5 Analysis

This section attempts to enhance our understanding of Hidden Calibration through comprehensive analysis. **(1).** We measure the overlapped area of data points mapped into classification criteria and prove that Hidden Calibration finds criteria with smaller overlap, responding to our hypothesis in §3.1. **(2).** We further investigate why using simple linear boundaries can effectively classify ICL data points, as happens in typical ICL and Hidden Calibration. We find that not only do the LMs provide a linearly separable hidden representation for ICL, but the demonstrations also facilitate this process. **(3).** As a guarantee of efficiency, we investigate how are the calibration examples needed in Hidden Calibration, and find that even with one example per category, Hidden Calibration can improve the performance of ICL.

5.1 Effectiveness: Hidden Calibration Finds Criteria with Lower Overlap

In Fig. 2, we mapped all the "positive" and "negative" labeled data points into vanilla classification criteria at the oblique extra coordinate axis, then we find a significant *overlap* in the between the data points in two categories, making it difficult to find suitable classification boundaries in the vertical direction. In this section, we generalize this intuitive observation using the area of such overlap as a metric of classification criteria.

In detail, we first decompose the multi-classification problem into all possible binary classification combinations. Then, for each combination, we sample data points labeled with both ground-truth labels in such a combination. Then, we map them into classification criteria by the method to be evaluted (here, we use the difference of *probabilities*, instead of the difference of *logits* in Fig. 2). To get the continuous representation for area calculation, we run kernel density estimations for both criteria in each ground-truth category, to get two kernel density functions corresponding to queries' label, as shown in Fig. 5. Then, we calculate the overlap area of these two kernel density



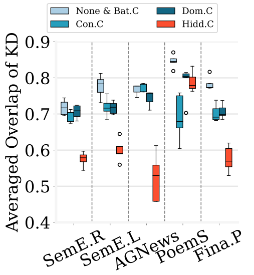


Figure 5: Demonstration of the overlap calculation with GPT2-XL on SemEval 2014-Task 4 Restaurant, $k = 4$. **Curves:** The kernel density of probability difference of l_1 ="positive" and l_2 ="negative". **Heatmaps:** The overlap of various 2-combinations (we plot the combination with the same label where the overlap is 1, but we omit them in averaging).

Figure 6: The Averaged Overlap of both criteria on GPT2-XL and 5 datasets.

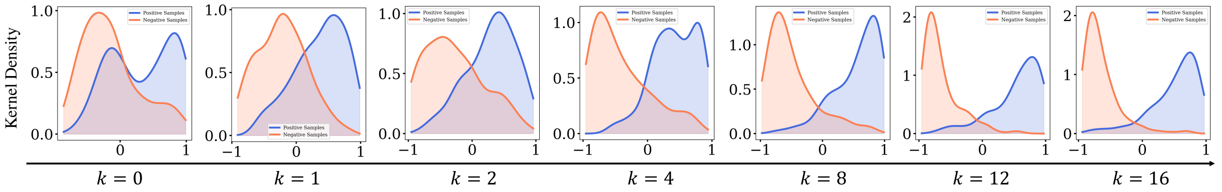


Figure 7: In-context learning dynamics visualized by overlap on OPT-2.7B and SemEval 2014-Task 4 Laptops (Pontiki et al., 2014). The intra-category data points gradually converge to the centroid w.r.t. the demonstrations number.

curves. The final Averaged Overlap is the macro average of overlap area among all possible binary combinations (see Appendix B.5 for details).

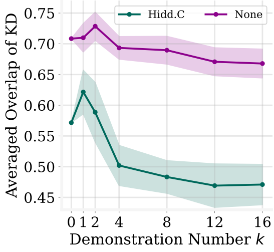
The overlap area of these two curves is double to the *lower bound* of the classifier's error rate among these two labels (see Appendix B.5.3), so Averaged Overlap is an intuitive metric of the classification criteria. The larger the overlap, the more difficult it is for the classifier, even (further) calibrated or ideal, to classify data points correctly, resulting in a potential decrease in classification performance.

We measure the Averaged Overlap of 3 models on 5 datasets (see Appendix B.5.2 for experimental details). The result on GPT2-XL is shown in Fig. 6 (see Appendix C.2 for other models), where the Averaged Overlaps from token-based methods are consistently higher, causing that better classification performance cannot be achieved on such criteria, which confirms our hypothesis in 3.1.

The overlaps from Hidden Calibration is much less than from token-based methods, which means that the Hidden Calibration finds better classification criteria with better possible classification performance than the token-based methods, even delicate calibrations try to transfer or rotate these classification criteria.

5.2 Principle: The Inner Linear-separability

In the practice of Hidden Calibration, we use simple linear boundaries to classify ICL examples, raising doubt on the linear separability of hidden states. In this section, We find that LMs inherently produce linearly separable hidden states corresponding to the ground-truth label, incredibly. Moreover, the demonstrations facilitate this process.



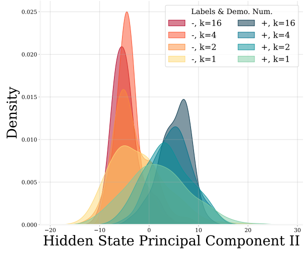
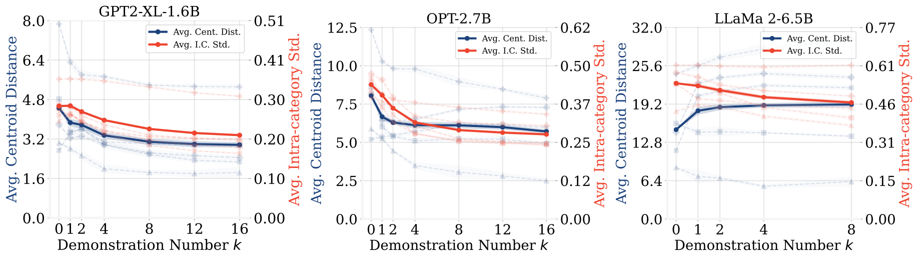


Figure 8: The Averaged Overlap on OPT-2.7B and 5 datasets against the number of demonstrations k.

Figure 9: Hidden state convergence w.r.t. k of Fig. 7 visualized on the direction of principal component II.

As an intuitive visualization, we plot similar curves as the Fig. 5 against the number of demonstrations k to visualize the *in-context learning dynamics* in Fig. 7, where we find that: **(1).** the data points have a little linear separability when $k = 0$, and **(2).** such linear separability is being enhanced following the increment of k, performing intra-category converging dynamics.

We further characterize this process. First, we calculate the Averaged Overlap similar to §5.1 on various k in Fig. 8. We find that the token-based



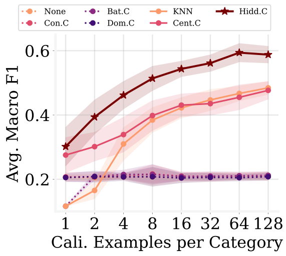


Figure 10: The averaged intra-category standard error of data points and the inter-category averaged centroid distance against k. **Solid lines**: means on 5 datasets; **Dashed lines**: Individual results for each dataset.

Figure 11: Classification performance against calibration examples used.

overlaps remain high and stable w.r.t. k, which indicates that the token-based methods can not benefit much from the demonstrations. However, the overlaps from Hidden Calibration significantly decrease with the increase of k, indicating that Hidden Calibration benefits from the demonstrations, which is consistent with our observations in §4.2.

More generally, we repeat the visualization of Fig. 7 on the **second** principal components of hidden states, instead of classification probability, to get an essential observation in Fig. 9, where as k increases, the hidden state shows clear intra-category cohesive dynamics, enabling linear classifying through the clustering of hidden states.

More directly, we measure the intra-category standard error of data points and the inter-category averaged centroid distance against k (see Appendix B.6 for details), as a joint indicator of intra-category aggregation and inter-category aggregation. The results are shown in Fig. 10, where the two curves are both diminishing, showing an obvious intra- and inter-categories aggregation trend w.r.t. k. However, the inter-category aggregation has weaker and less persistent decreasing trends than the intra-category aggregation only in early demonstrations, or even ascending, which indicates that demonstration enhances intra-category clustering stronger than the inter-category aggregation, which is beneficial to linear classification. Moreover, a model with more parameters shows a stronger difference between these aggregations.

5.3 Efficiency: Even One-shot Centroid Can Help Hidden Calibration

Another concern is how much calibration data is required in Hidden Calibration. We repeat the experiments in §4 with various sizes of the calibration set on OPT-2.7B (see Appendix B.7 for details), from 1 to 128 calibration examples per category. The results are shown in Fig. 11.

Table 1: Transferability of centroid among various datasets with the same label space. Big numbers are the averaged improvement (MF1) compared to vanilla ICL, small numbers are standard error. Statistically significant results ($p < 0.1$) are in **bold**.

Cali. Test	SemE.R	SemE.L	Fina.P	TES
SemE.R	**(+38.75)** ±2.28	**+29.24** ±3.19	+6.32 ±10.55	+7.54 ±8.96
SemE.L	**+20.78** ±7.37	**(+37.33)** ±3.47	-0.40 ±7.37	+8.94 ±8.93
Fina.P	**+7.42** ±4.98	+9.05 ±11.14	**(+37.29)** ±2.30	-4.35 ±6.34
TES	+6.95 ±7.00	**+9.73** ±5.68	-0.51 ±3.83	**(+11.83)** ±3.59

The results indicate that although Hidden Calibration stably benefits from the size of the calibration set, even one sample per category can still make it outperform. This makes our method consistent with the original motivation of in-context learning, that is: making the most efficient use of training samples and calculation resources in low-resource scenarios. While token-based methods can not benefit from scaling the calibration dataset, the KNN method (Xu et al., 2022) underperforms the vanilla ICL with a very small calibration set.

6 Discussion

6.1 Transferability of the Centorid

We have proven that it is not advisable to use the *common* token probability criteria, while, since the centroid criteria are proven to be better than token probability, we are curious: can the centroid calculated in one task be transferred to other tasks with the same label space? Among the datasets sharing the same label space “positive”, “neutral”, and “negative”, we calculate centroids by one dataset and evaluate Hidden Calibration with it on another dataset, on OPT-2.7B, with $k = 4$, $m = 16$. The results are shown in Table 1, where only limited transferability is demonstrated in different domains of the same task (SemE.R and SemE.L), whose behavior is similar to *task vector* (Ilharco et al., 2022; Hendel et al., 2023), while other combination of datasets can not demonstrate considerable transferability. This further exacerbates our doubts about the token-based method: We find that the hidden state distributions have significant differences among various datasets, even if they share a common label space, then utilizing fixed token unembedding vectors to decode these classification criteria is highly unreliable.

Moreover, we repeat this experiment on various k, instead of various datasets, as shown in Table 2. The transferabilities among k are better than on datasets, but still worse than the un-transferred scenario. Notice that $4 \rightarrow 1$ results are much better than $0 \rightarrow 1$, which support our results in §5.2: hidden states with higher k are further converged.

6.2 A Demonstration towards ICL Principles

Our findings may lead to an explanation of the principle of ICL and traditional calibrations. LMs generate distributed representations into separate clusters in the last hidden state. At this point, by dot-product, any non-collinear *arbitrary or plausible* mapping directions should be able to capture and classify these clusters to some extent. Note: The absolute distance in such a direction is not faithful (since the centroids of these hidden states and the coordinate origins in these mapping directions are not necessarily aligned), which leads to the generation of so-called bias, and calibrating these biases can improve the performance to a certain extent. However, in such a paradigm, high-dimensional features are discarded, resulting in overlapping originally linearly separable features in high-dimensional space, leading to a loss of classification accuracy, even if the calibration aligns the coordinate origin.

6.3 Comparison to Previous Works

Comparison to Probe Methods. One concern is that our work can be regarded as a degraded probe (Abbas et al., 2024; Amini and Ciaramita, 2023) of the hidden states. However, we believe our work has more advantages: **In terms of application**, we use fewer samples and require no gradient-based training, which makes our method more user-friendly, efficient, elegant, and interpretable. **In terms of theory**, compared to fitting a universal approximation (Hornik et al., 1989), our method and settings fully utilize the hidden state convergence on decoder LMs, making it a true ICL practice.

Table 2: Transferability of centroid among various k on the same dataset. $k_1 \rightarrow k_2$ is to use centroids estimated by k_1 demonstrations for inference on test examples with k_2 demonstrations. Other annotations are the same as Table. 1

	0→1	**4→1**	**(1→1)**	**1→4**	**(4→4)**
SemE.R	**+9.46** ±1.95	**+22.50** ±14.55	**(+26.14)** ±5.16	**+17.95** ±7.51	**(+38.75)** ±2.28
SemE.L	**+26.80** ±3.20	**+17.18** ±5.61	**(+26.65)** ±2.72	+10.79 ±14.86	**(+37.33)** ±3.47
AGNews	**+42.38** ±2.42	**+40.20** ±1.24	**(+41.02)** ±2.49	**+43.12** ±2.02	**(+46.66)** ±3.77
PoemS	+0.16 ±1.87	+2.12 ±6.18	**(+21.49)** ±2.54	**+8.79** ±1.84	**(+12.96)** ±1.52
Fina.P	-0.13 ±1.88	**+21.40** ±2.90	**(+16.70)** ±3.80	**+10.00** ±13.68	**(+37.30)** ±2.30

Comparison to Supervised Fine-tuning. Some practices (Gu et al., 2023; Min et al., 2022b) build training objectives to fine-tune models for better ICL performance. These efforts are efficient but bulky, while our work avoids such an enormous overhead, making it more usable and elegant.

Comparison to Other Calibrations. Our method is a disruptive innovation for methods based on token probability (even the ones based on the whole vocabulary). Experimental comparisons of these methods have been given in §4.

For more related works, refer to Appendix A.

7 Conclusion

In this paper, we analyze the current token-based ICL methods, and point out a common drawback: using token probability as the classification criteria. We propose Hidden Calibration to address such a drawback by discarding the token-based classification criteria. Our experiments show that Hidden Calibration is a new state-of-the-art of ICL. Then, we confirm that Hidden Calibration indeed creates better criteria by reducing the inter-category overlap. Moreover, we find the hidden state convergence promoted by demonstrations, as an explanation of the principle of the performance improvement by a single linear classification boundary in Hidden Calibration.

We hope this work can inspire exploration of the ICL by investigating the hidden state instead of token probabilities, and update the community's understanding of ICL calibration.

8 Limitations

Due to computability limitations, we cannot compare the performance of Hidden Calibration with the baseline based on supervised fine-tuning. However, we believe that Hidden Calibration is not within the same methodology as the fine-tuning method, due to the significant difference in computational cost. So such a lack of comparison will not seriously hurt the soundness of this paper.

We prove that human intuition in the label token choice is not reliable. However, we have not eliminated such human intuition completely from the ICL loop: when we build prompts, we still choose the label token. How to automatically select the optimal label token in the prompt will be an important issue and future research direction for improving the performance of ICL further.

Other probability calibrations can be combined with Hidden Calibration for further performance improvements, since the 0-point is not necessarily an exact classification boundary, as shown in Fig. 7. Also, more complex prompts can be used. However, due to space constraints, we have not attempted this incremental approach, remaining it for future works.

Analysis in §5.2 needs more theoretical and experimental analysis. As we can see, some models (GPT2-XL) do not benefit from demonstrations even through the lens of hidden state aggregation. The differences in such a model behavior need to be explained. An explanation of "how to enhance the intra-category aggregation", and "why such aggregation occurs or not" will be considerably beneficial for understanding the principle of ICL.

Acknowledgments

Invisible during the review.

References

Momin Abbas, Yi Zhou, Parikshit Ram, Nathalie Baracaldo, Horst Samulowitz, Theodoros Salonidis, and Tianyi Chen. 2024. Enhancing in-context learning via linear probe calibration. *arXiv preprint arXiv:2401.12406*.

Afra Amini and Massimiliano Ciaramita. 2023. Probing in context: Toward building robust classifiers via probing large language models. *arXiv preprint arXiv:2305.14171*.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. Improving in-context few-shot learning via self-supervised training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3558–3573.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. 2023. Understanding in-context learning via supportive pretraining data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12660–12673.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2022. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.

Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. 2023. Generative calibration for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2312–2333.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022b. Metaicl: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022c. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system.

Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. 2024. Is cosine-similarity of embeddings really about similarity? In *Companion Proceedings of the ACM on Web Conference 2024*, pages 887–890.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, et al. 2023. Symbol tuning improves in-context learning in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 968–979.

Benfeng Xu, Quan Wang, Zhendong Mao, Yajuan Lyu, Qiaoqiao She, and Yongdong Zhang. 2022. k nn prompting: Beyond-context learning with calibration-free nearest neighbor inference. In *The Eleventh International Conference on Learning Representations*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Yufeng Zhao, Yoshihiro Sakai, and Naoya Inoue. 2024. Noisyicl: A little noise in model parameters calibrates in-context learning. *arXiv preprint arXiv:2402.05515*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. 2023. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. *arXiv preprint arXiv:2309.17249*.

A Related Works

Given the topic of enhancing in-context learning, there are 2 categories of methods focused on such a target.

Model parameter update-based method: Although it is pointed out that the ICL objective is implicitly included in pre-training data (Han et al., 2023), explicitly fixing the gap between the ICL objective and causal language modeling objective can still be beneficial. Such methods are usually based on supervised fine-tuning (Min et al., 2022b; Gu et al., 2023; Wei et al., 2021, 2023; Iyer et al., 2022; Wang et al., 2022), and also self-supervised training (Chen et al., 2022) and non-gradient method (Zhao et al., 2024). Such methods usually require huge amounts of computation and data overhead to update billions of LM parameters.

In contrast, lightweight solutions focus on **(2). Classification criteria-based method (calibration)**: Such methods focus on calculating output category logits, keeping the main feed-forward calculation processes and their parameters unmodified. The original motivation for these works is to eliminate prior bias and unfaithful confidence in ICL, by calibrating the output label probabilities (Fei et al., 2023; Zhao et al., 2021; Han et al., 2022; Zhou et al., 2023; Jiang et al., 2023). While, as described in the main text, some practices without the usage of label probabilities have also been proposed (Xu et al., 2022; Abbas et al., 2024; Min et al., 2022a).

B Experimental Details

B.1 Datasets

In this paper, 10 datasets are used as shown in Table 3. Some datasets do not provide valid splitting, so we randomly split all of them into calibration sets and test sets: For each dataset, we first shuffle it with random seed 42. Then, we choose the 512 data at the tail as the testing data, and the 512 data at the head (all the datasets have more than 1024 examples.) as the calibration data. Each data point in a test set is used once for each experiment trial to build a prompt example and test for performance.

Table 3: Datasets and Abbreviations used in this paper.

Dataset	**Abbr.**
AGNews (Zhang et al., 2015)	AGNews
SemEval 2014-Task 4 Restaurant (Pontiki et al., 2014)	SemE.R
SemEval 2014-Task 4 Laptops (Pontiki et al., 2014)	SemE.L
Poem Sentiment (Sheng and Uthus, 2020)	PoemS
GLUE-RTE (Wang et al., 2019)	RTE
tweet_eval_emotion (Mohammad et al., 2018)	TEE
tweet_eval_hate (Basile et al., 2019)	TEH
tweet_eval_sentiment (Rosenthal et al., 2017)	TES
financial_phrasebank (all agree) (Malo et al., 2014)	FP
rotten_tomatoes (Pang and Lee, 2005)	Rott.T

Table 4: The **additional** (compare to vanilla ICL) time and space on calibration and inference cost of various methods. Hidden Calibration has a similar cost upper bound to other calibrations. $|\mathbb{V}|$ is the vocabulary size.

Method	**Calibration Cost**		**Inference Cost**
	Add. Space	Add. Time	Add. Time
None	0	0	0
Con.C	$O(\|\mathcal{Y}\|)$	$O(m)$	$O(\|\mathcal{Y}\|)$
Bat.C	0	0	$O(m\|\mathcal{Y}\|)$
Dom.C	$O(\|\mathcal{Y}\|)$	$O(m)$	$O(\|\mathcal{Y}\|)$
KNN	$O(m\|\mathbb{V}\|)$	$O(m)$	$O(m\|\mathbb{V}\|)$
Cent.C	$O(\|\mathcal{Y}\|\|\mathbb{V}\|)$	$O(m)$	$O(\|\mathcal{Y}\|\|\mathbb{V}\|)$
Hidd.C	$O(\|\mathcal{Y}\|d)$	$O(m)$	$O(\|\mathcal{Y}\|d)$

AGNews has some over-length examples. So, in the main experiments, we filter out those examples: for LLaMa 2, when $k = 8$, we filter out all the examples with a string length greater than 512. And in the experiments in §5.2, for all the models, we filter out all the examples with a string length greater than 256 for all the k.

B.2 Baselines

6 baselines (1 vanilla and 5 calibrations) are used in this paper. Here we introduce the 5 calibration baseline.

Contextual Calibration (Con.C). Proposed by Zhao et al. (2021), Con.C uses empty queries with normal demonstrations as calibration samples. We input $m|\mathcal{Y}|$ samples with empty queries into the model and get the averaged normalized label

Table 5: Prompt templates used in this paper.

Dataset	Prompt Template	Label Space
AGNews	Input: <x>, Label: <y>	world, sport, business, science
SemE.R	Input: <x>, Aspect: <a>, Label: <y>	positive, neutral, negative
SemE.L	Input: <x>, Aspect: <a>, Label: <y>	positive, neutral, negative
PoemS	Input: <x>, Label: <y>	positive, neutral, negative, mix
RTE	Input: <x>, Text 2: <a>, Label: <y>	include, neutral
TEE	Input: <x>, Label: <y>	anger, joy, positive, sad
TEH	Input: <x>, Label: <y>	normal, hate
TES	Input: <x>, Label: <y>	positive, neutral, negative
FP	Input: <x>, Label: <y>	positive, neutral, negative
Rott.T	Input: <x>, Label: <y>	positive, negative

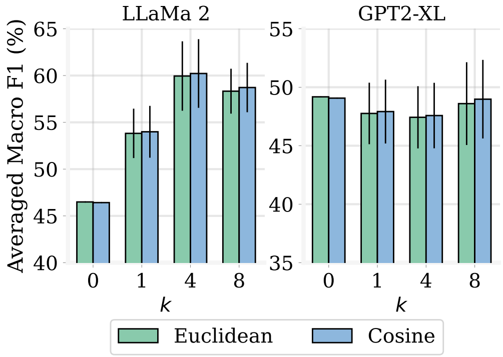


Figure 12: The classification performance (Macro F1(%)) of Hidden Calibration with difference similarity measure.

probabilities $\bar{p'}$ among m samples. We take the reciprocal of the probabilities as calibration term $A = \bar{p'}^{-1}$, while the $B = \mathbf{0}$.

Batch Calibration (Bat.C). Proposed by Zhou et al. (2023), Bat.C is an inference-time calibration, using the negative averaged normalized label probabilities $-\bar{p}$ of $m|\mathcal{Y}|$ samples in inference time as the calibration term $B = -\bar{p}$, while the $A = \mathbf{1}$, where $\mathbf{1}$ is the all-one vector.

Domain Calibration (Dom.C). Proposed by Fei et al. (2023), Dom.C acts similarly to the Con.C. The difference is that it uses a random sequence sampled on the random tokens from the calibration dataset as queries instead of empty ones. We fix the sampled length to 32.

KNN Prompt (KNN). Proposed by Xu et al. (2022), KNN uses the whole output vocabulary probability distribution as the classification feature, instead of the label tokens. First, features of calibration examples are calculated as k-NN anchors. Then, during the inference, a k-NN classifier is used to classify the feature from the test samples. We use m examples per category to calculate the anchors for k-NN, and the nearest neighbor number is set to 3.

Central Calibration (Cent.C). This is the control method proposed by us. The calculation process is completely consistent with the Hidden Calibration, except that the hidden state is not used, and the whole output vocabulary probability distribution consistent with KNN is used. This method compares with Hidden Calibration to prove that the output probability distribution is not a good classification feature for ICL.

Notice that: these label-probability-based methods (Con.C, Bat.C, Dom.C) use A or B *along*, which may be another major drawback of these calibration methods: According to Fig. 2, if a calibration rotates the mapping direction suitably, and transfer the 0-point properly, a decision boundary close to the Hidden Calibration can be found. This also leads to a new research direction for calibration: the simultaneous usage of translation and rotation methods.

B.3 Prompts

In this paper, we use a minimum prompt template shown in Table 5.

To facilitate the replication of label probability-based methods, we limit the label space to one token by synonymous conversion. Note that Hidden Calibration does not need to meet such a one-token requirement. That is, Hidden Calibration can be applied to classification datasets of any length on the label.

B.4 Details of Visualization in §3.1

Principle Component Analysis (PCA). Given a hidden state set $\mathcal{H} = \left\{h^{(i)}\right\}_{i=1}^{n}$, we span all the hidden state vector into a matrix $H \in \mathbb{R}^{n\times d}$. The covariance matrix is $\operatorname{cov}(H) = \left(H - \bar{H}\right)^T \left(H - \bar{H}\right)$, where the $\bar{H}$ is the matrix spanned by the element-wise average vectors $\bar{h}$ of hidden state set $\mathcal{H}$. We conduct Eigenvalue Decomposition on $\operatorname{cov}(H)$ and adjust the dimensions to arrange the eigenvalues Λ in a descending order along the row:

$$\operatorname{cov}(H) = P\Lambda P^T, \tag{1}$$

where the $P \in \mathbb{R}^{d\times d}$ is an orthogonal matrix. Taking the top-$\tilde{d}$ lines of P and span them into $\tilde{P} \in \mathbb{R}^{d\times \tilde{d}}$, we get the principle component mapping:

$$\mathrm{PCA}_{\mathcal{H}}(h) = \left(h - \bar{h}\right)\tilde{P} = h\tilde{P} - \bar{h}\tilde{P}. \tag{2}$$

Notice that $\tilde{P}\tilde{P}^T = I$, where I is the identity matrix.

Dot-product after PCA. Suppose we have dot-product with vector[7] h and E in the original space $\mathbb{R}^d$, producing the dot-product similarity classification criterion α:

$$\alpha = h\left(E^T - \mathbf{0}^T\right). \tag{3}$$

When we conduct a same PCA on both h and E^T to get dot-product similarity in a dimensionality-reduced space similar to Fig. 2:

$$\tilde{\alpha} = \mathrm{PCA}_{\mathcal{H}}(h)\underbrace{\left(\mathrm{PCA}_{\mathcal{H}}(E)^T - \mathrm{PCA}_{\mathcal{H}}(\mathbf{0})^T\right)}_{\text{Mapping direction selected after PCA}} \tag{4}$$

$$= \left(h\tilde{P} - \bar{h}\tilde{P}\right)\left(E\tilde{P}\right)^T \tag{5}$$

$$= h\tilde{P}\tilde{P}^T E^T - \bar{h}\tilde{P}\tilde{P}^T E^T \tag{6}$$

$$= \alpha - \bar{h}E^T. \tag{7}$$

Notice that we use the mapping direction $\left(\mathrm{PCA}_{\mathcal{H}}(E)^T - \mathrm{PCA}_{\mathcal{H}}(\mathbf{0})^T\right)$ after the PCA, instead of $\left(\mathrm{PCA}_{\mathcal{H}}(E)^T - \mathbf{0}^T\right)$, and this is the reason why the oblique axis in Fig. 2 does not necessarily pass through the coordinate origin. In such a scenario, the dot productions after PCA only differ by a fixed constant bias $-\bar{h}E^T$ from the ones before PCA. This is the reason why the normal line of oblique axis on the 0-point doesn't pass the coordinate origin of the 2D-plane in Fig. 2.

Decision Boundary after PCA. Notice that the decision boundary of two categories l_1 and l_2 in an non-rotated ICL scenario is:

$$\mathcal{B} = \left\{h \middle| hE_{l_1}^T - hE_{l_2}^T = C\right\}. \tag{8}$$

Where the C is the calibration term without rotation. Notice that it is a hyperplane in $\mathbb{R}^d$ with normal vector $(E_{l_1} - E_{l_2})^T$. Also, the normal plane which pass the 0-point of direction $(E_{l_1} - E_{l_2})^T$ in $\mathbb{R}^{\tilde{d}}$ after PCA is:

$$\begin{aligned}\tilde{\mathcal{B}} = \{&\mathrm{PCA}_{\mathcal{H}}(h) | \mathrm{PCA}_{\mathcal{H}}(h) \\ &(\mathrm{PCA}_{\mathcal{H}}(E_{l_1} - E_{l_2}) - \mathrm{PCA}_{\mathcal{H}}(\mathbf{0}))^T = 0\}.\end{aligned} \tag{9}$$

By the aforementioned transformation, we have:

$$\tilde{\mathcal{B}} = \left\{\mathrm{PCA}_{\mathcal{H}}(h) | hE_{l_1}^T - hE_{l_2}^T = \bar{h}\left(E_{l_1}^T - E_{l_2}^T\right)\right\}. \tag{10}$$

That is, the dimensionality-reduced decision boundary $\tilde{\mathcal{B}}$ is perpendicular to the mapped direction $(\mathrm{PCA}_{\mathcal{H}}(E_{l_1} - E_{l_2}) - \mathrm{PCA}_{\mathcal{H}}(\mathbf{0}))$, and biased only by a constant $\left(\bar{h}\left(E_{l_1}^T - E_{l_2}^T\right) - C\right)$ on the classification criteria comparing to the original space. Specifically, in the two-dimensional case, it is a straight line that may not necessarily pass through the coordinate origin, as shown in Fig. 2.

B.5 Details of Experiment in §5.1

B.5.1 Calculation Details of Averaged Overlap

First, we divide the $|\mathcal{Y}|$-way classification task into $\mathbb{C}(|\mathcal{Y}|, 2)$ 2-way classification task[8], to allow us to use a scalar to characterize the classification criteria for each 2-combination (similar to what we do to the "positive" and "negative" examples in Fig. 2). Then, for each chosen 2-combination, w.l.o.g, given labels denoted as l_1 and l_2, we build prompt-label sets[9] as:

$$\mathcal{S}_{l_j} = \left\{T\left(\mathcal{D}^{de,(i)}, x^{(c_i)}\right) \middle| y^{(c_i)} = l_j\right\}_{l_j \in \{l_1, l_2\}}^{n_{l_j}}, \tag{11}$$

where c_i is the sampled query index. That is, we sample queries annotated with these two labels and build prompt sets, then collect the prompts with the same query label l_j into $\mathcal{S}_{l_j}$, with a size n_{l_j}.

Then, for each prompt $s^{(i)} = T\left(\mathcal{D}^{de,(i)}, x^{(i)}\right) \in \mathcal{S}_{l_j}$, we run decoders (vanilla, Con.C, Dom.C and Hidden Calibration) with probability normlization $f_{l_1}(\cdot)$ and $f_{l_2}(\cdot)$ to get the classification probabilities of assigning label l_1 and l_2 as $\alpha_1^{(i)} = f_{l_1}\left(s^{(i)}\right)$ and $\alpha_2^{(i)} = f_{l_2}\left(s^{(i)}\right)$. We calculate the difference between $\alpha_1^{(i)}$ and $\alpha_2^{(i)}$ and collect them into a set:

$$\mathcal{A}_{l_j} = \left\{\alpha_1^{(i)} - \alpha_2^{(i)} \middle| s^{(i)} \in \mathcal{S}_{l_j}\right\}_{i=1}^{n_{l_j}}. \tag{12}$$

Now, for the 2-combination of labels (l_1, l_2), we get $\mathcal{A}_{l_1}$ and $\mathcal{A}_{l_2}$, whose elements are the probabilities difference between assigning l_1 and assigning l_2 to example $s^{(i)}$. The difference between $\mathcal{A}_{l_1}$ and $\mathcal{A}_{l_2}$ is: the elements in $\mathcal{A}_{l_1}$ are from $s^{(i)}$s with queries labeled by ground-truth l_1, and vice versa. We obtain continuous probability density functions of $\mathcal{A}_{l_1}$ and $\mathcal{A}_{l_2}$ as $p_{l_1}(\cdot)$ and $p_{l_2}(\cdot)$ by kernel density estimation, as the curves in Fig. 5.

Then, we calculate the overlap area of these curves:

$$S_{l_1, l_2} = \int_{-1}^{1} \min\left[p_{l_1}(x), p_{l_2}(x)\right] dx. \tag{13}$$

[7] Due to excessive superscripts, in this section, we omit the superscripts U in the notation of un-embedding E_l^U.

[8] The $\mathbb{C}(m, n)$ is the n-combination number from m elements.

[9] Notice that the T is the prompting function.

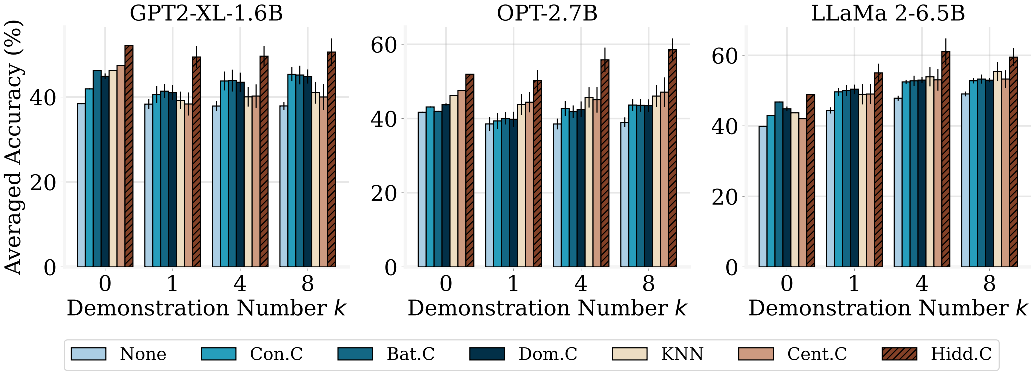


Figure 13: The classification performance (Accuracy(%)) of 3 models averaged on 10 datasets.

For each combination[10] in the $\mathbb{C}(|\mathcal{Y}|, 2)$ 2-combinations, we repeat to calculate the $S_{\cdot,\cdot}$, and average them as the **Averaged Overlap** $\bar{S}$.

$$\bar{S} = \frac{1}{\mathbb{C}(|\mathcal{Y}|, 2)} \sum_{i=1}^{|\mathcal{Y}|} \sum_{j=i+1}^{|\mathcal{Y}|} S_{l_i,l_j}. \tag{14}$$

B.5.2 Experimental Details

We conduct experiments resulting Fig. 6 on 3 models with SemEval 2014-Task 4 Restaurant, SemEval 2014-Task 4 Laptops, AGNews, Poem Sentiment, and fiancial_phrasebank, given the demonstration number $k = 4$ and calibration example numbers $m = 16$. We use the whole 512 examples on the test split for each dataset and repeat 5 times.

B.5.3 Proof: *the Overlap Area is Double to the Error's Lower Bound*

Suppose a label combination l_1 and l_2, w.l.o.g., we have a ground truth probability density function $p_{l_1}(x)$ and $p_{l_2}(x)$ on a criterion $x \in \mathbb{X}$, same as the curves in Fig. 5. Given a specific value of criterion x, the upper-bound classification performance is determined by majority vote, which is the most accurate method on such a point, resulting in a density of error classification:

$$e(x)_{l_1,l_2} \leqslant \min\left[p_{l_1}(x), p_{l_2}(x)\right]. \tag{15}$$

So, the integral error rate:

$$\mathcal{E}_{l_1,l_2} \leqslant \frac{\int_{x\in\mathbb{X}} \min\left[p_{l_1}(x), p_{l_2}(x)\right] dx}{\int_{x\in\mathbb{X}} p_{l_1}(x)dx + \int_{x\in\mathbb{X}} p_{l_2}(x)dx} \tag{16}$$

$$= \frac{1}{2}\int_{x\in\mathbb{X}} \min\left[p_{l_1}(x), p_{l_2}(x)\right] dx \tag{17}$$

$$= \frac{1}{2} S_{l_1,l_2}. \tag{18}$$

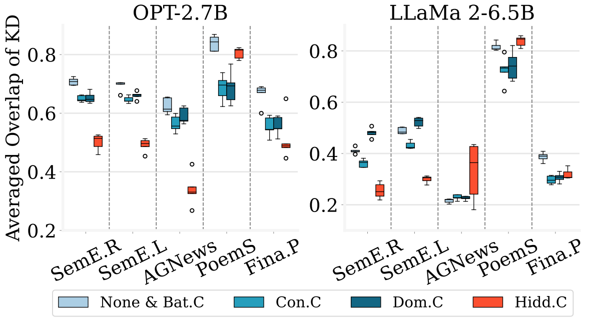


Figure 14: The augmented results on 2 models of Fig. 6.

B.6 Details of Experiment in §5.2

B.6.1 Calculation of the Distance and Standard Error

Averaged Centroid Distance. Given a $|\mathcal{Y}|$-way classification task, for each label l we build its corresponding prompt set $\mathcal{S}_l = \left\{s^{(c_i)} | y^{(c_i)} = l\right\}_{i=1}^{n_l}$, where $s^{(c_i)}$ is the prompt with query labeled by l, and c_i is the sampled query index. We encode it into a hidden state set $\mathcal{H}_l = \left\{h^{(i)}\right\}_{i=1}^{n_l}$, and calculate its centroid $\bar{h}_l$, as what we do in Hidden Calibration:

$$\bar{h}_l = \frac{1}{n_l} \sum_{h^{(i)}\in\mathcal{H}_l} h^{(i)}. \tag{19}$$

For every 2-combination of labels l and l', we calculate the distance of their centroid, and the average among all the 2-combination is used as the Averaged Centroid Distance:

$$\text{ACD} = \frac{1}{\mathbb{C}(|\mathcal{Y}|, 2)} \sum_{i=1}^{|\mathcal{Y}|} \sum_{j=i+1}^{|\mathcal{Y}|} \left\|\bar{h}_i - \bar{h}_j\right\|_2. \tag{20}$$

Averaged Intra-category Standard Error. Given the hidden state set $\mathcal{H}_l = \left\{h^{(i)}\right\}_{i=1}^{n_l}$ w.r.t.

[10] Notice that on $S_{\cdot,\cdot}$, the labels are rotational symmetry.

the label l, we span all the hidden state vectors into a matrix $H_l \in \mathbb{R}^{n_l \times d}$. The covariance matrix is $\left(H_l - \bar{H}_l\right)^T \left(H_l - \bar{H}_l\right)$, where the $\bar{H}_l$ is the matrix spanned by the element-wise average vectors of hidden state set $\mathcal{H}_l$. Notice that the ACD is a first-order moment, for a proper comparison, we use the average on the diagonal elements of the element-wise square root of the covariance matrix as the intra-category standard error metric for category l. We average all the standard errors from all the categories as the Averaged Intra-category Standard Error:

$$\text{AIS} = \frac{1}{|\mathcal{Y}|d} \sum_{i=1}^{|\mathcal{Y}|} \text{tr} \left[\sqrt{\left(H_i - \bar{H}_i\right)^T \left(H_i - \bar{H}_i\right)} \right]. \tag{21}$$

B.6.2 Experimental Details

We conduct experiments resulting Fig. 10 on 3 models with SemEval 2014-Task 4 Restaurant, SemEval 2014-Task 4 Laptops, AGNews, Poem Sentiment, and fiancial_phrasebank, given the calibration example numbers $m = 16$. We use the whole 512 examples on the test split for each dataset and repeat 5 times.

B.7 Experimental Details in §5.3

We conduct experiments resulting Fig. 11 on OPT-2.7B with 4 datasets: SemEval 2014-Task 4 Restaurant, SemEval 2014-Task 4 Laptops, AGNews, and Poem Sentiment, given the demonstration numbers $k = 4$ and repeat 5 times.

C Detailed Results

C.1 Details of Main Results

Numerical details of Fig. 4 are shown in Table 7, 8, and 6. Accuracy results is shown in Fig. 13.

C.2 Details of Averaged Overlaps Results

The augmented results on the other 2 models of Fig. 6 are shown in Fig. 14.

D Additional Analysis

D.1 The Similarity Measures Used in Hidden Calibration

In §3.2, we use the Euclidean distance as the similarity measure. But this is not the only option. Intuitively, we can choose other similarity measures as alternatives. Since we get inspired by observation with dot-production similarity, we have an obligation to check the performance on such a measure instead of the Euclidean distance. This section uses cosine similarity as an example to illustrate that there is no significant performance difference between these measures. We use cosine similarity to repeat the results in §4.2 on LLaMa 2 and GPT2-XL.

The results are shown in Fig. 12, where the performance based on these two measures is close, without statistical difference. This indicates that the hidden space has good properties of both metric and vector space, and Hidden Calibration acts equally on these measures.

D.2 Time and Space Cost of Hidden Calibration and baselines

We analyze the upper bound of the additional space-time cost of the baseline method and Hidden Calibration, as shown in Table 4. Here, we are most concerned about the inference time cost, and Hidden Calibration is the fastest among all the non-label-based methods.

Since the product $|\mathcal{Y}|d$ is usually not very large, Hidden Calibration does not add considerable inference overhead. In contrast, KNN may be incredibly slow as the calibration dataset scales.

E Statements

E.1 License for Artifacts

Models. GPT2-XL and OPT-2.7B is under the MIT license, LLaMa 2 is under its specific license.

Datasets. We list the open-source license for the datasets used in this paper as follows:

- CC-by-4.0: Poem Sentiment, SemEval 2014-Task 4 Restaurant, SemEval 2014-Task 4 Laptops, tweet_eval_emotion, tweet_eval_hate, tweet_eval_hate

- CC-by-SA-3.0: financial_phrasebank, GLUE-RTE

- Unknown: AGNews, rotten_tomatoes

Consistency of Usage. Models and data are used with their original usage.

E.2 AI Agent Usage

AI Agents are only used for writing improving and grammar checking in this paper.

Table 6: Classification performance (Macro F1(%)) on GPT2-XL. $\mathrm{mean}_{\mathrm{std}}$, top-2 results are in **bold**.

	GPT-2 XL	**AGNews**	**SemE.R**	**SemE.L**	**PoemS**	**RTE**	**TEE**	**TEH**	**TES**	**Fina.P**	**Rott.T**	**Average**
k=0	None	16.53$_{0.00}$	9.87$_{0.00}$	12.31$_{0.00}$	8.75$_{0.00}$	48.31$_{0.00}$	19.40$_{0.00}$	37.56$_{0.00}$	21.14$_{0.00}$	25.36$_{0.00}$	34.16$_{0.00}$	23.34
	Con.C	30.04$_{0.00}$	9.87$_{0.00}$	12.31$_{0.00}$	8.11$_{0.00}$	47.46$_{0.00}$	9.68$_{0.00}$	42.97$_{0.00}$	27.01$_{0.00}$	33.14$_{0.00}$	60.75$_{0.00}$	28.13
	Bat.C	42.56$_{0.00}$	37.29$_{0.00}$	**46.21**$_{0.00}$	18.92$_{0.00}$	**51.29**$_{0.00}$	26.39$_{0.00}$	**44.45**$_{0.00}$	**42.52**$_{0.00}$	32.00$_{0.00}$	**66.96**$_{0.00}$	40.86
	Dom.C	38.00$_{0.60}$	24.76$_{0.04}$	26.58$_{1.06}$	20.52$_{0.99}$	38.00$_{0.60}$	14.34$_{0.00}$	37.56$_{0.00}$	**42.70**$_{1.00}$	30.14$_{1.06}$	**67.42**$_{2.19}$	34.33
	KNN	50.63$_{0.00}$	**39.62**$_{0.00}$	42.31$_{0.00}$	26.69$_{0.00}$	**49.57**$_{0.00}$	30.79$_{0.00}$	**44.68**$_{0.00}$	34.12$_{0.00}$	39.30$_{0.00}$	62.37$_{0.00}$	42.01
	Cent.C	**52.54**$_{0.00}$	39.53$_{0.00}$	45.37$_{0.00}$	**27.23**$_{0.00}$	44.92$_{0.00}$	**32.71**$_{0.00}$	41.93$_{0.00}$	32.15$_{0.00}$	**43.15**$_{0.00}$	63.32$_{0.00}$	**42.29**
	Hidd.C	**82.02**$_{0.00}$	**44.73**$_{0.00}$	**54.45**$_{0.00}$	**32.81**$_{0.00}$	47.36$_{0.00}$	**43.45**$_{0.00}$	42.95$_{0.00}$	36.78$_{0.00}$	**47.38**$_{0.00}$	59.83$_{0.00}$	**49.18**
k=1	None	20.95$_{1.29}$	36.72$_{1.19}$	31.60$_{1.33}$	21.21$_{1.72}$	49.47$_{2.22}$	22.81$_{0.94}$	37.56$_{0.00}$	31.74$_{1.43}$	30.15$_{1.97}$	36.20$_{0.56}$	31.84
	Con.C	24.15$_{1.13}$	41.88$_{1.73}$	38.92$_{3.32}$	24.82$_{2.55}$	47.64$_{3.38}$	20.70$_{1.20}$	37.56$_{0.00}$	33.06$_{1.29}$	34.93$_{1.31}$	60.21$_{3.98}$	36.39
	Bat.C	30.02$_{1.49}$	**45.04**$_{0.77}$	**41.10**$_{4.41}$	25.20$_{0.48}$	**49.58**$_{1.88}$	25.85$_{2.51}$	**48.02**$_{0.94}$	**34.92**$_{0.91}$	**35.02**$_{1.14}$	**64.72**$_{1.92}$	**39.95**
	Dom.C	22.17$_{1.15}$	44.86$_{1.16}$	41.02$_{4.91}$	**25.61**$_{1.27}$	46.81$_{2.09}$	18.92$_{1.58}$	37.56$_{0.00}$	**33.96**$_{0.90}$	34.72$_{1.56}$	62.27$_{3.33}$	36.79
	KNN	**32.14**$_{0.85}$	36.88$_{2.76}$	37.29$_{3.20}$	21.44$_{1.57}$	48.18$_{2.09}$	**27.44**$_{1.75}$	43.52$_{3.73}$	33.85$_{1.08}$	**36.23**$_{2.07}$	52.41$_{1.44}$	36.94
	Cent.C	26.74$_{2.18}$	33.00$_{1.48}$	32.02$_{3.63}$	18.75$_{2.78}$	47.82$_{3.27}$	23.82$_{3.95}$	44.07$_{1.52}$	30.68$_{2.12}$	29.58$_{1.98}$	58.41$_{4.30}$	34.49
	Hidd.C	**65.15**$_{1.77}$	**49.16**$_{3.43}$	**51.56**$_{2.83}$	**32.83**$_{2.32}$	**50.47**$_{1.41}$	**36.17**$_{2.74}$	**49.16**$_{1.47}$	33.55$_{4.45}$	44.02$_{3.36}$	**65.55**$_{2.47}$	**47.76**
k=4	None	21.87$_{4.32}$	33.14$_{1.46}$	41.03$_{2.14}$	20.11$_{1.47}$	40.48$_{1.15}$	17.98$_{0.33}$	38.19$_{1.41}$	29.06$_{1.70}$	28.86$_{2.07}$	33.81$_{0.72}$	30.45
	Con.C	24.22$_{10.00}$	44.76$_{0.98}$	**48.90**$_{2.71}$	21.95$_{1.28}$	36.33$_{1.11}$	24.40$_{0.82}$	37.51$_{0.11}$	**37.30**$_{2.49}$	41.65$_{2.96}$	69.57$_{1.11}$	38.66
	Bat.C	26.97$_{9.22}$	44.48$_{1.67}$	46.94$_{1.72}$	21.93$_{1.24}$	47.17$_{1.96}$	**28.86**$_{1.88}$	46.94$_{2.76}$	**36.49**$_{2.64}$	**44.82**$_{2.54}$	**71.94**$_{1.24}$	**41.65**
	Dom.C	25.30$_{10.23}$	**45.44**$_{1.54}$	47.01$_{2.20}$	**23.05**$_{1.05}$	36.79$_{1.68}$	27.11$_{1.72}$	37.79$_{0.51}$	34.67$_{0.81}$	42.54$_{2.53}$	67.00$_{3.89}$	38.67
	KNN	**33.93**$_{2.04}$	37.57$_{3.54}$	38.35$_{2.35}$	21.72$_{3.05}$	**48.55**$_{2.97}$	26.20$_{1.96}$	**48.71**$_{2.74}$	30.10$_{2.12}$	36.57$_{2.56}$	57.36$_{1.72}$	37.91
	Cent.C	32.98$_{2.56}$	37.24$_{5.80}$	32.71$_{4.50}$	18.47$_{1.88}$	45.78$_{2.93}$	24.21$_{4.17}$	**48.83**$_{3.57}$	29.75$_{2.50}$	33.30$_{7.56}$	58.81$_{2.86}$	36.21
	Hidd.C	**49.55**$_{3.29}$	**50.81**$_{2.16}$	**54.16**$_{3.62}$	**24.96**$_{2.56}$	**49.28**$_{2.42}$	**39.13**$_{2.80}$	48.48$_{1.62}$	34.43$_{2.46}$	**50.80**$_{3.51}$	**72.70**$_{2.19}$	**47.43**
k=8	None	19.23$_{0.78}$	32.79$_{2.13}$	33.36$_{1.02}$	17.93$_{1.08}$	37.37$_{2.07}$	15.75$_{0.75}$	37.56$_{0.00}$	25.15$_{0.99}$	26.45$_{1.13}$	33.55$_{0.45}$	27.91
	Con.C	18.38$_{0.28}$	**47.06**$_{3.77}$	**52.40**$_{0.96}$	19.70$_{2.05}$	35.98$_{0.00}$	26.80$_{2.00}$	37.56$_{0.00}$	**38.12**$_{3.15}$	41.05$_{2.25}$	**75.05**$_{2.12}$	39.21
	Bat.C	22.32$_{1.22}$	45.96$_{2.08}$	48.25$_{0.79}$	**21.20**$_{1.55}$	**45.82**$_{4.68}$	**29.23**$_{1.22}$	47.84$_{1.36}$	**38.30**$_{3.09}$	**48.78**$_{2.06}$	74.81$_{1.58}$	**42.25**
	Dom.C	21.85$_{0.95}$	45.78$_{2.48}$	49.91$_{0.74}$	20.80$_{1.97}$	35.98$_{0.00}$	28.50$_{1.30}$	37.56$_{0.60}$	36.20$_{2.44}$	47.99$_{1.84}$	69.87$_{4.70}$	39.44
	KNN	**39.81**$_{1.81}$	38.83$_{4.38}$	34.38$_{3.72}$	20.92$_{2.98}$	43.03$_{5.31}$	26.49$_{3.84}$	**49.00**$_{1.23}$	29.33$_{3.74}$	39.00$_{1.91}$	60.08$_{3.64}$	38.09
	Cent.C	41.52$_{4.18}$	37.73$_{3.15}$	32.21$_{2.80}$	14.66$_{4.00}$	45.11$_{6.67}$	21.83$_{6.60}$	48.97$_{3.71}$	25.81$_{2.72}$	31.32$_{3.89}$	55.23$_{4.35}$	35.44
	Hidd.C	**57.10**$_{5.54}$	**49.86**$_{2.55}$	**58.26**$_{3.15}$	**24.42**$_{1.28}$	**48.48**$_{4.48}$	**35.48**$_{5.04}$	**51.50**$_{2.18}$	27.46$_{6.30}$	**56.99**$_{1.72}$	**76.50**$_{3.14}$	**48.60**

Table 7: Classification performance (Macro F1(%)) on OPT-2.7B. $\mathrm{mean}_{\mathrm{std}}$, top-2 results are in **bold**.

OPT 2.7B		AGNews	SemE.R	SemE.L	PoemS	RTE	TEE	TEH	TES	Fina.P	Rott.T	Average
k=0	None	$27.67_{0.00}$	$24.62_{0.00}$	$31.07_{0.00}$	$25.00_{0.00}$	$51.19_{0.00}$	$24.43_{0.00}$	$39.31_{0.00}$	$34.08_{0.00}$	$\mathbf{47.23}_{0.00}$	$40.33_{0.00}$	34.49
	Con.C	$20.72_{0.00}$	$20.77_{0.00}$	$29.33_{0.00}$	$16.51_{0.00}$	$34.63_{0.00}$	$16.25_{0.00}$	$31.37_{0.00}$	$30.65_{0.00}$	$44.88_{0.00}$	$\mathbf{67.48}_{0.00}$	31.26
	Bat.C	$28.58_{0.00}$	$31.32_{0.00}$	$34.31_{0.00}$	$25.44_{0.00}$	$\mathbf{53.23}_{0.00}$	$26.81_{0.00}$	$43.60_{0.00}$	$34.34_{0.00}$	$45.25_{0.00}$	$\mathbf{66.51}_{0.00}$	38.94
	Dom.C	$27.55_{0.06}$	$20.53_{0.05}$	$30.90_{0.84}$	$16.72_{0.40}$	$45.07_{0.12}$	$18.01_{0.18}$	$43.30_{0.12}$	$31.11_{0.24}$	$\mathbf{46.16}_{0.47}$	$66.23_{1.28}$	34.56
	KNN	$52.50_{0.00}$	$31.39_{0.00}$	$\mathbf{44.16}_{0.00}$	$25.72_{0.00}$	$\mathbf{52.08}_{0.00}$	$36.15_{0.00}$	$51.60_{0.00}$	$\mathbf{39.45}_{0.00}$	$42.26_{0.00}$	$57.16_{0.00}$	43.25
	Cent.C	$\mathbf{55.97}_{0.00}$	$\mathbf{31.65}_{0.00}$	$43.23_{0.00}$	$\mathbf{35.86}_{0.00}$	$41.98_{0.00}$	$\mathbf{41.27}_{0.00}$	$\mathbf{53.44}_{0.00}$	$35.25_{0.00}$	$37.54_{0.00}$	$58.60_{0.00}$	**43.48**
	Hidd.C	$\mathbf{75.01}_{0.00}$	$\mathbf{41.94}_{0.00}$	$\mathbf{52.14}_{0.00}$	$\mathbf{39.92}_{0.00}$	$45.64_{0.00}$	$\mathbf{45.90}_{0.00}$	$\mathbf{52.93}_{0.00}$	$\mathbf{35.67}_{0.00}$	$43.24_{0.00}$	$61.71_{0.00}$	**49.41**
k=1	None	$24.16_{0.71}$	$18.76_{0.50}$	$24.98_{1.74}$	$12.23_{1.19}$	$\mathbf{50.75}_{3.01}$	$20.72_{3.33}$	$\mathbf{51.55}_{4.28}$	$23.64_{1.19}$	$31.40_{1.83}$	$48.94_{1.90}$	30.71
	Con.C	$22.57_{1.38}$	$20.15_{1.78}$	$24.48_{1.88}$	$13.17_{0.63}$	$\mathbf{50.75}_{3.01}$	$24.30_{1.69}$	$\mathbf{51.69}_{4.03}$	$23.02_{0.00}$	$28.88_{1.95}$	$65.63_{1.72}$	32.46
	Bat.C	$27.43_{1.13}$	$20.48_{0.90}$	$26.78_{1.17}$	$14.21_{1.05}$	$50.25_{3.25}$	$26.44_{1.41}$	$50.45_{1.76}$	$23.34_{2.30}$	$28.81_{0.24}$	$\mathbf{70.88}_{0.65}$	33.91
	Dom.C	$24.06_{0.94}$	$19.68_{1.26}$	$24.24_{1.73}$	$13.78_{2.27}$	$\mathbf{50.75}_{3.01}$	$23.58_{2.54}$	$51.54_{3.86}$	$23.14_{0.83}$	$28.81_{1.72}$	$\mathbf{69.98}_{1.81}$	32.96
	KNN	$48.15_{2.50}$	$\mathbf{42.35}_{4.06}$	$39.01_{5.24}$	$25.52_{1.94}$	$\mathbf{53.07}_{3.06}$	$32.62_{3.02}$	$49.78_{0.58}$	$\mathbf{31.90}_{3.64}$	$36.59_{3.67}$	$57.31_{4.89}$	**41.63**
	Cent.C	$\mathbf{49.21}_{2.81}$	$39.46_{4.26}$	$\mathbf{42.30}_{4.65}$	$\mathbf{29.60}_{5.30}$	$48.48_{3.85}$	$\mathbf{34.32}_{2.51}$	$50.69_{1.21}$	$31.00_{1.34}$	$\mathbf{36.69}_{2.28}$	$53.70_{2.51}$	41.54
	Hidd.C	$\mathbf{65.18}_{2.39}$	$\mathbf{44.91}_{5.14}$	$\mathbf{51.62}_{2.09}$	$\mathbf{33.72}_{2.24}$	$50.40_{1.80}$	$\mathbf{45.00}_{2.79}$	$49.53_{2.92}$	$\mathbf{35.02}_{0.47}$	$\mathbf{48.10}_{3.33}$	$57.72_{1.42}$	**48.12**
k=4	None	$22.91_{1.05}$	$20.84_{1.16}$	$25.44_{1.60}$	$12.46_{1.29}$	$49.70_{3.22}$	$14.83_{0.16}$	$40.68_{0.44}$	$23.78_{0.99}$	$28.62_{1.13}$	$53.73_{0.95}$	29.30
	Con.C	$22.81_{1.04}$	$20.70_{1.02}$	$27.74_{1.79}$	$12.62_{1.32}$	$\mathbf{50.06}_{3.86}$	$21.92_{1.03}$	$41.21_{0.70}$	$25.86_{1.65}$	$36.74_{1.68}$	$\mathbf{83.50}_{1.82}$	34.32
	Bat.C	$25.40_{1.04}$	$20.07_{1.40}$	$26.49_{0.86}$	$11.46_{1.24}$	$47.61_{2.78}$	$23.92_{0.96}$	$45.71_{1.33}$	$26.91_{1.24}$	$38.88_{1.70}$	$82.22_{1.13}$	34.87
	Dom.C	$22.32_{1.17}$	$20.71_{1.39}$	$25.96_{1.97}$	$12.64_{1.24}$	$\mathbf{50.37}_{3.22}$	$21.74_{1.30}$	$41.30_{0.54}$	$26.15_{1.41}$	$38.01_{1.69}$	$\mathbf{83.99}_{1.45}$	34.32
	KNN	$49.36_{2.28}$	$49.71_{5.26}$	$47.66_{3.87}$	$22.18_{1.17}$	$48.74_{3.76}$	$\mathbf{32.05}_{4.50}$	$\mathbf{47.35}_{3.18}$	$\mathbf{30.76}_{3.92}$	$\mathbf{40.78}_{1.86}$	$66.33_{4.19}$	**43.49**
	Cent.C	$\mathbf{49.91}_{4.46}$	$\mathbf{51.72}_{2.80}$	$\mathbf{48.96}_{3.00}$	$\mathbf{21.63}_{3.55}$	$47.02_{2.37}$	$29.84_{3.43}$	$46.99_{2.05}$	$25.39_{3.05}$	$40.23_{8.32}$	$65.80_{4.54}$	42.75
	Hidd.C	$\mathbf{69.56}_{3.62}$	$\mathbf{59.59}_{1.97}$	$\mathbf{62.77}_{3.08}$	$\mathbf{25.42}_{0.79}$	$47.94_{3.16}$	$\mathbf{47.47}_{5.86}$	$\mathbf{49.46}_{1.80}$	$\mathbf{35.61}_{3.45}$	$\mathbf{65.91}_{2.01}$	$79.47_{5.88}$	**54.32**
k=8	None	$21.52_{1.03}$	$21.24_{1.48}$	$24.11_{1.51}$	$12.84_{0.73}$	$48.05_{3.65}$	$14.54_{0.00}$	$37.54_{0.03}$	$21.85_{1.07}$	$28.47_{1.48}$	$59.86_{1.63}$	29.00
	Con.C	$22.26_{0.33}$	$23.84_{1.26}$	$30.27_{1.27}$	$13.02_{0.89}$	$47.58_{3.27}$	$23.06_{1.10}$	$37.80_{0.48}$	$25.91_{0.79}$	$37.67_{1.44}$	$83.42_{1.74}$	34.48
	Bat.C	$24.38_{1.49}$	$22.28_{1.04}$	$28.05_{1.65}$	$13.31_{0.50}$	$47.50_{2.86}$	$24.46_{0.50}$	$49.30_{1.63}$	$27.96_{1.51}$	$40.39_{1.53}$	$84.03_{1.31}$	36.16
	Dom.C	$22.38_{0.70}$	$21.33_{1.57}$	$26.63_{1.36}$	$13.18_{1.08}$	$47.48_{3.87}$	$23.63_{1.08}$	$37.59_{0.25}$	$25.25_{1.11}$	$39.27_{2.20}$	$\mathbf{86.67}_{1.55}$	34.34
	KNN	$47.45_{1.64}$	$55.08_{1.89}$	$48.16_{3.38}$	$22.61_{2.20}$	$\mathbf{49.56}_{2.68}$	$\mathbf{34.80}_{3.18}$	$49.60_{2.26}$	$\mathbf{30.77}_{2.79}$	$38.73_{3.52}$	$66.65_{5.67}$	44.34
	Cent.C	$\mathbf{47.50}_{4.46}$	$\mathbf{55.83}_{4.46}$	$\mathbf{51.06}_{8.42}$	$\mathbf{22.95}_{2.22}$	$48.22_{2.35}$	$33.57_{4.12}$	$\mathbf{49.66}_{2.73}$	$29.18_{4.37}$	$\mathbf{43.87}_{2.57}$	$67.49_{8.07}$	**44.93**
	Hidd.C	$\mathbf{65.27}_{1.64}$	$\mathbf{63.95}_{2.51}$	$\mathbf{66.48}_{2.53}$	$\mathbf{23.70}_{1.76}$	$\mathbf{49.86}_{2.86}$	$\mathbf{52.24}_{2.39}$	$\mathbf{50.57}_{3.28}$	$\mathbf{41.94}_{6.42}$	$\mathbf{71.16}_{4.13}$	$\mathbf{85.78}_{4.39}$	**57.09**

Table 8: Classification performance (Macro F1(%)) on LLaMa 2. $\mathrm{mean}_{\mathrm{std}}$, top-2 results are in **bold**.

LLaMa 2		**AGNews**	**SemE.R**	**SemE.L**	**PoemS**	**RTE**	**TEE**	**TEH**	**TES**	**Fina.P**	**Rott.T**	**Average**
k=0	None	23.74$_{0.00}$	44.36$_{0.00}$	32.48$_{0.00}$	16.23$_{0.00}$	35.88$_{0.00}$	17.01$_{0.00}$	46.06$_{0.00}$	30.59$_{0.00}$	19.64$_{0.00}$	32.90$_{0.00}$	29.89
	Con.C	22.76$_{0.00}$	44.50$_{0.00}$	42.83$_{0.00}$	10.53$_{0.00}$	43.88$_{0.00}$	24.06$_{0.00}$	42.42$_{0.00}$	33.32$_{0.00}$	54.92$_{0.00}$	**58.97**$_{0.00}$	37.82
	Bat.C	37.26$_{0.00}$	49.03$_{0.00}$	**54.33**$_{0.00}$	19.70$_{0.00}$	**47.34**$_{0.00}$	**31.64**$_{0.00}$	**51.06**$_{0.00}$	**39.82**$_{0.00}$	**55.76**$_{0.00}$	**63.07**$_{0.00}$	**44.90**
	Dom.C	28.11$_{1.43}$	47.88$_{0.45}$	49.74$_{1.40}$	**26.58**$_{1.27}$	39.99$_{0.13}$	23.00$_{1.11}$	37.48$_{0.00}$	33.33$_{0.21}$	**61.03**$_{0.76}$	54.96$_{2.78}$	40.21
	KNN	**40.24**$_{0.00}$	47.65$_{0.00}$	49.15$_{0.00}$	20.56$_{0.00}$	**50.93**$_{0.00}$	26.65$_{0.00}$	42.35$_{0.00}$	33.63$_{0.00}$	46.44$_{0.00}$	53.26$_{0.00}$	41.09
	Cent.C	34.36$_{0.00}$	**50.87**$_{0.00}$	46.36$_{0.00}$	17.57$_{0.00}$	42.15$_{0.00}$	25.09$_{0.00}$	42.31$_{0.00}$	28.21$_{0.00}$	43.81$_{0.00}$	49.38$_{0.00}$	38.01
	Hidd.C	**62.46**$_{0.00}$	**50.90**$_{0.00}$	**55.77**$_{0.00}$	**22.49**$_{0.00}$	46.94$_{0.00}$	**34.06**$_{0.00}$	**47.37**$_{0.00}$	**34.29**$_{0.00}$	54.84$_{0.00}$	56.37$_{0.00}$	**46.49**
k=1	None	13.97$_{1.15}$	51.57$_{0.30}$	51.50$_{0.47}$	22.28$_{0.98}$	36.26$_{0.80}$	27.80$_{1.58}$	40.16$_{0.83}$	30.78$_{2.77}$	27.89$_{2.18}$	55.03$_{1.91}$	35.72
	Con.C	13.29$_{1.62}$	52.36$_{0.58}$	54.12$_{0.50}$	24.16$_{1.29}$	38.53$_{0.92}$	27.77$_{1.16}$	40.63$_{1.52}$	37.39$_{1.86}$	37.23$_{0.68}$	**75.91**$_{2.44}$	40.14
	Bat.C	22.05$_{1.07}$	52.24$_{0.44}$	53.77$_{1.27}$	**24.75**$_{1.51}$	**49.77**$_{3.25}$	28.94$_{0.90}$	47.72$_{1.19}$	**40.25**$_{2.30}$	38.43$_{1.91}$	75.77$_{1.37}$	43.37
	Dom.C	11.89$_{1.01}$	51.80$_{0.38}$	53.98$_{0.45}$	24.23$_{1.13}$	**49.33**$_{3.82}$	29.17$_{1.44}$	38.23$_{0.85}$	39.02$_{1.72}$	34.95$_{1.35}$	**79.44**$_{0.58}$	41.20
	KNN	**52.50**$_{3.58}$	56.19$_{5.59}$	61.24$_{2.42}$	21.91$_{0.91}$	48.82$_{1.90}$	**30.66**$_{3.56}$	**49.65**$_{3.15}$	34.86$_{1.11}$	**40.26**$_{3.40}$	75.26$_{3.34}$	**46.87**
	Cent.C	46.17$_{0.93}$	**62.09**$_{3.04}$	**64.50**$_{1.76}$	23.29$_{2.57}$	46.66$_{3.16}$	27.80$_{2.36}$	45.63$_{2.33}$	33.10$_{2.42}$	38.99$_{3.56}$	75.16$_{2.33}$	46.34
	Hidd.C	**61.88**$_{0.68}$	**64.83**$_{1.85}$	**69.05**$_{2.29}$	**24.26**$_{1.37}$	48.32$_{2.35}$	**42.77**$_{2.38}$	**51.55**$_{1.27}$	**40.72**$_{1.91}$	**63.79**$_{0.93}$	71.00$_{5.23}$	**53.82**
k=4	None	9.57$_{0.26}$	53.45$_{0.34}$	53.76$_{0.62}$	19.82$_{0.51}$	35.90$_{0.18}$	29.30$_{1.47}$	37.56$_{0.00}$	36.55$_{0.97}$	37.86$_{0.78}$	69.31$_{1.43}$	38.31
	Con.C	9.35$_{0.23}$	53.49$_{0.33}$	54.21$_{0.30}$	**27.01**$_{1.15}$	37.65$_{1.74}$	**31.37**$_{1.07}$	37.56$_{0.00}$	39.34$_{0.47}$	39.28$_{0.55}$	90.86$_{0.35}$	42.01
	Bat.C	19.33$_{1.51}$	52.58$_{0.64}$	54.40$_{0.69}$	24.62$_{0.99}$	46.81$_{5.22}$	31.26$_{1.58}$	48.81$_{2.93}$	38.47$_{0.33}$	37.05$_{0.60}$	86.62$_{1.15}$	44.00
	Dom.C	9.49$_{0.32}$	52.12$_{0.79}$	54.14$_{0.28}$	**26.47**$_{0.87}$	39.59$_{2.03}$	30.75$_{1.06}$	37.56$_{1.43}$	**39.52**$_{0.22}$	34.71$_{0.58}$	**91.84**$_{0.88}$	41.62
	KNN	**66.47**$_{6.82}$	60.00$_{0.92}$	62.87$_{2.54}$	21.76$_{3.20}$	54.41$_{4.95}$	29.96$_{2.19}$	49.46$_{2.22}$	31.90$_{3.97}$	**49.09**$_{5.20}$	88.18$_{1.17}$	**51.41**
	Cent.C	60.62$_{4.62}$	**63.61**$_{2.06}$	**64.55**$_{3.77}$	21.42$_{4.73}$	**55.50**$_{5.61}$	28.41$_{2.51}$	**50.72**$_{3.29}$	21.12$_{5.70}$	42.48$_{2.81}$	89.55$_{0.74}$	49.80
	Hidd.C	**68.22**$_{6.82}$	**65.64**$_{1.38}$	**71.38**$_{2.00}$	25.47$_{2.10}$	**58.00**$_{4.38}$	**48.48**$_{4.32}$	**51.83**$_{3.03}$	**49.44**$_{3.20}$	**68.79**$_{9.71}$	**92.26**$_{0.18}$	**59.95**
k=8	None	8.03$_{0.00}$	53.90$_{0.31}$	54.12$_{0.38}$	20.47$_{0.49}$	35.98$_{0.00}$	30.77$_{1.50}$	37.56$_{0.00}$	38.59$_{1.26}$	39.25$_{1.04}$	73.77$_{0.60}$	39.24
	Con.C	8.15$_{0.26}$	53.72$_{0.30}$	54.26$_{0.37}$	**28.72**$_{1.91}$	35.98$_{0.00}$	31.40$_{1.94}$	37.56$_{0.00}$	39.33$_{0.53}$	38.80$_{0.46}$	91.52$_{0.38}$	41.94
	Bat.C	17.58$_{1.19}$	53.06$_{0.52}$	54.15$_{0.28}$	25.62$_{0.85}$	51.85$_{3.70}$	30.66$_{1.48}$	48.45$_{2.63}$	38.44$_{0.46}$	36.28$_{0.37}$	88.11$_{0.62}$	44.42
	Dom.C	8.03$_{0.00}$	52.26$_{0.43}$	54.26$_{0.47}$	26.50$_{0.99}$	36.44$_{1.05}$	29.28$_{2.06}$	37.56$_{0.00}$	**39.72**$_{0.41}$	35.21$_{0.86}$	**91.91**$_{0.49}$	41.12
	KNN	**69.41**$_{4.38}$	59.12$_{4.16}$	62.69$_{3.74}$	23.15$_{2.09}$	**54.86**$_{4.00}$	**34.89**$_{2.06}$	48.44$_{3.62}$	36.27$_{4.36}$	**50.27**$_{5.38}$	89.60$_{1.32}$	**52.87**
	Cent.C	**60.04**$_{8.07}$	**61.80**$_{1.44}$	**66.16**$_{2.05}$	23.17$_{1.31}$	53.64$_{3.23}$	32.16$_{2.62}$	**51.74**$_{1.94}$	30.29$_{3.91}$	36.55$_{6.71}$	90.12$_{0.52}$	50.57
	Hidd.C	59.17$_{4.08}$	**65.88**$_{0.84}$	**70.07**$_{2.32}$	**27.85**$_{2.56}$	**55.79**$_{3.05}$	**46.70**$_{5.37}$	**53.95**$_{2.36}$	**43.90**$_{0.48}$	**67.69**$_{2.29}$	**92.26**$_{0.63}$	**58.33**