

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

Hakaze Cho<sup>1,☆</sup>  
Kenshiro Tanaka<sup>1</sup>

Yoshihiro Sakai<sup>1</sup>  
Akira Ishii<sup>1</sup>

Mariko Kato<sup>1</sup>  
Naoya Inoue<sup>1,2</sup>

<sup>1</sup>Japan Advanced Institute of Science and Technology <sup>2</sup>RIKEN

☆Primary Contributor, Correspondence to: yfzhao@jaist.ac.jp

## Abstract

In-Context Learning (ICL) typically utilizes classification criteria from output 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 applied. 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 assign the label of the nearest centroid previously estimated from a calibration set to the test sample as the predicted label. Our experiments on 6 models and 10 classification datasets indicate that Hidden Calibration consistently outperforms current token-based baselines by about 20%~50%, achieving a strong state-of-the-art in ICL. Our further analysis demonstrates that Hidden Calibration finds better classification criteria with less inter-class overlap, and LMs provide linearly separable intra-class clusters with the help of demonstrations, which supports Hidden Calibration and gives new insights into the principle of ICL. Our official code implementation can be found at [https://github.com/hc495/Hidden\\_Calibration](https://github.com/hc495/Hidden_Calibration).

## 1 Introduction

In-context Learning (ICL) (Dong et al., 2022) is a few-shot learning paradigm without model parameter updates on Language Models (LMs). In detail, as shown in Fig. 1-(A, B), given a prompt consisting of demonstrations and a query, LMs conduct causal language modeling operation from the prompt to assign probabilities to the label token candidates designed by hand, and ICL chooses the one with the highest probability as the prediction.

One well-known issue of 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

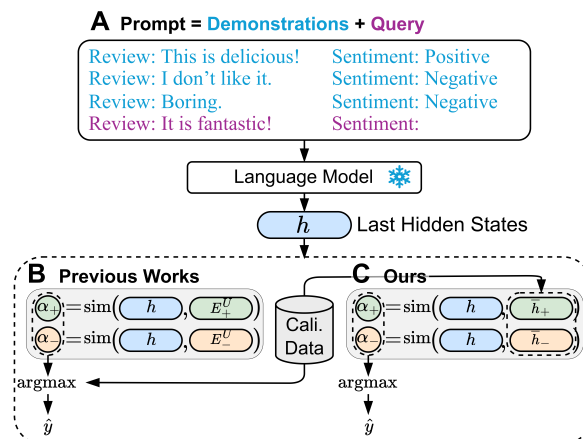


Figure 1: In an ICL diagram, **A**. The prompt of ICL consists of a concatenation of **demonstrations** and a **query**. LMs encode the prompt into the last hidden state  $h$ , then **B**. (previous works) use the un-embedding vectors of the label tokens ( $E_+^U$ ,  $E_-^U$ ) to decode the  $h$  to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

et al., 2021; Zhou et al., 2023). To address this issue, previous work *calibrates* the predicted label token probabilities by performing affine transformations with estimated parameters to adjust these probabilities for more precise predictions.

These previous works and also the vanilla ICL are based on a potential assumption: the affine manifolds spanned by the decoding vectors in the LM head (*un-embedding vectors*) of *manually selected* label tokens are good subspaces of the hidden space to distinguish hidden vectors (i.e., the last hidden states to be fed to the LM head) onto various label appropriately, so that the label token probabilities decoded from these subspaces are accurate classification logits. However, although using the label un-embedding with task-related semantics (e.g. “positive” and “negative”) seems intuitive, it should be noted that we have no reason to believe that these label un-embeddings have any explicit

guarantee for decoding the last hidden states into accurate classification logits (token-based decision criterion), even if various delicate calibrations are used to move these boundaries *inside the subspaces* (see §3.1). Also, some works 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 arbitrary, making a suspicion of: using manually selected label un-embeddings to decode the last hidden states, i.e., **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 classification criteria, but there is still doubt that output probability distributions are not informative enough for classification (see §3.2). Therefore, we utilize the last hidden states instead, which are informative precursors of the token probabilities.

Concretely, we propose Hidden Calibration, training centroid classifiers on the last hidden states of ICL prompts. As shown in Fig. 3, **during the training**, we build standard ICL prompts similarly to Fig. 1-A from a supervised calibration set and input them into the LM to get the last hidden states of the last tokens of ICL prompts. Then, we calculate the centroids of the last hidden states w.r.t. the queries’ label to get a centroid for each label, as an anchor for inference. **During the inference**, we input the test prompt, find the nearest centroid to the last hidden states of the test prompt, and assign the corresponding label of the centroid as the prediction.

Empirically, Hidden Calibration improves the ICL performance by approximately more than 20% on 10 text classification datasets and 6 modern LMs (§4.1), with an equal computational cost with previous calibration methods. To the best of the author’s knowledge, Hidden Calibration consistently outperforms the calibration baselines, achieving a strong state-of-the-art in ICL. Additional experiments indicate that Hidden Calibration effectively alleviates the stress of prompt engineering, performing robust accuracy under different prompt designs.

Moreover, our subsequent analysis indicates that Hidden Calibration does find better logits-mapping subspaces that effectively separate data points. In detail, we find that the distribution of classification logits calculated from Hidden Calibration have

less inter-class 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.

Furthermore, 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 more demonstrations 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 predicted probabilities of manually selected label tokens for classification criteria, which is often under-guaranteed.
- We propose Hidden Calibration to address the problem before, eliminating the unreliable decoding on the hand-selected label, and using a centroid classifier on the LM’s last hidden states. Our experiments suggest that Hidden Calibration reaches a strong state-of-the-art.
- Our further analysis indicates that Hidden Calibration does find better classification criteria with less inter-class overlap, and LMs provide linearly separable intra-class clusters with the help of demonstrations, which supports Hidden Calibration to classify samples accurately.

## 2 Background

This section reviews previous work on ICL and denotes their mathematical descriptions as an introduction to the main motivation of this work.

### 2.1 In-context Learning

**Prompting.** Given a few-shot natural language classification dataset (*demonstration set*)  $\mathcal{D} = \{(x^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^n$ , where  $x^{(i)}$  and  $y^{(i)}$  are the input sequence 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  $\mathcal{D}^{de} = \{(x^{(c_i)}, y^{(c_i)})\}_{i=1}^k$  from  $\mathcal{D}$  with an index set  $\{c_i\}_{i=1}^k$  for a given query  $x^q$  as the demonstrations. Then, we use a template  $T$  to concatenate them in a natural language form into a prompt token sequence:  $s = T(\mathcal{D}^{de}, x^q)$ , as shown in Fig. 1-A.

**Encoding.** A decoder-structured LM receives the prompt token sequence  $s$  and encodes it into the

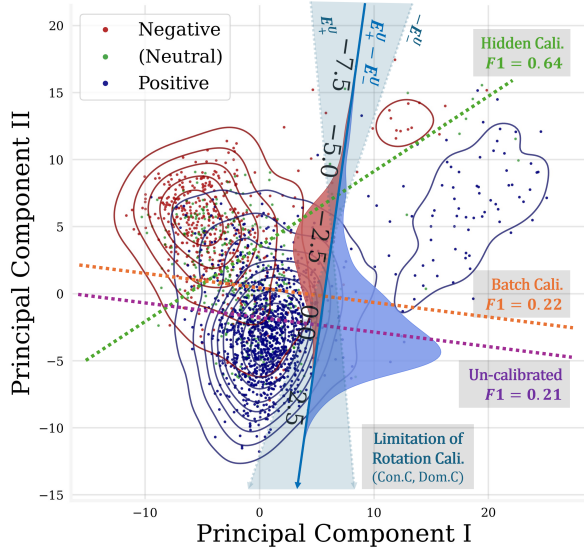


Figure 2: Token probability-based decision boundaries (original & batch calibrated) are suboptimal comparing to centroid-based boundary. Points and contour lines are ICL’s last hidden states and kernel densities mapped by Principal Component Analysis. Oblique coordinate axis is the direction of the un-embedding difference of label tokens ( $E_+^U - E_-^U$ ), where the kernel densities of mapped data points are plotted. The rotating calibration by  $A \neq 1$  (e.g. Contextual Calibration, Domain Calibration) has a limited feasible mapping direction<sup>3</sup>.

last (from 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  $h = H_{|s|} \in \mathbb{R}^d$ .

**Constrained Decoding.** In a typical ICL setup, one chooses the un-embedding vectors of the label candidates in the output head<sup>1</sup> to decode  $h$  as the prediction. That is, for each label  $l$ , the similarity  $\alpha_l = \text{sim}(h, E_l^U)$  (usually the dot-product similarity) between  $h$  and the un-embedding vector  $E_l^U$  is calculated as the output logits  $\alpha_l$ , as shown in Fig. 1-B for a binary classification example. Then, the label with the highest logits is chosen as the prediction  $\hat{y}$ , that is:  $\hat{y} = \underset{l \in \mathcal{Y}}{\text{argmax}} \text{sim}(h, E_l^U)$ .

## 2.2 Token-probability Calibration for ICL

However, Zhao et al. (2021) find that simply using the original logits for classification can not lead to a good ICL practice, since 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

<sup>1</sup>We omit the bias term in the output head (if any) for the sake of simplicity, which can be overridden by a fixed-to-one dimension, or covered by the calibration described below.

form: first, the logits are transformed into probabilities as  $p = \text{softmax}([\alpha_1, \alpha_2, \dots, \alpha_{|\mathcal{Y}|}])$ , 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  $m$  training examples from a calibration set, and  $\odot$  is the Hadamard multiplication. Various estimations for  $A$  and  $B$  are used: some practices use examples with pseudo queries terms (Fei et al., 2023; Zhao et al., 2021), while other practices use Gaussian estimation on real prompts (Han et al., 2022) or the mean value of  $p$  during the inference (Zhou et al., 2023).

However, as to be discussed in §3.1 current calibrations are affine transformations on label token probability, without modifying the  $E_l^U$ , causing only limited improvement to ICL performance.

## 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 label token probability-based ICL, we show a prototypical visualization of the hidden states of ICL prompts (aforementioned  $h$ ). Specifically, we input 2,048 ICL prompts (with  $k = 8$ ) built from of SemEval 2014-Task 4 Restaurant (Pontiki et al., 2014) into OPT-2.7B (Zhang et al., 2022) and plot the  $h$  on a 2D-Principal Component plane in Fig. 2 (detailed in Appendix A.4).

As a simple 2-way case, focusing on the data points labeled “positive” and “negative”, we plot the difference direction ( $E_+^U - E_-^U$ ) between the un-embedding vectors of these two label tokens<sup>2</sup>. Then, the coordinates of the projected hidden states in this direction are the difference of predicted logits between these two labels, serving as a *token-based classification criteria*, i.e., when the coordinate is positive, a “positive” label will be assigned, and vice versa. Therefore, in this visualized scenario, the orthogonal line at the zero point is the *original decision boundary*, points below this boundary are classified as “positive”, and

<sup>2</sup>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 A.4.

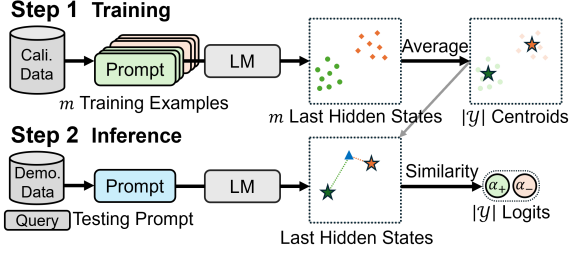


Figure 3: The diagram of Hidden Calibration. **Step 1:** Calculating the hidden state centroid of each label. **Step 2:** Find the label of the nearest centroid of the text sample to be the prediction.

vice versa. The **batch calibrated boundary** (Zhou et al., 2023) is always parallel to the original one, and the other calibrations (Contextual Calibration (Zhao et al., 2021), Domain Calibration (Fei et al., 2023)) produce **rotated mapping directions** ( $A_+E_+^U - A_-E_-^U$ ), by positive-definite term  $A$  and thus rotated decision boundaries, with limited direction<sup>3</sup> between  $E_+^U$  and  $-E_-^U$ .

Intuitively, as shown in Fig. 2, the token-based decision boundaries cannot effectively classify these data points, which is due to the inherent direction of the token un-embedding vectors, regardless of limited affine transformation by calibration. A straightforward better linear boundary is the **equidistant points** between both classes’ centroids, we try to find it as follows.

### 3.2 Hidden Calibration

Motivated by the visualization, we propose Hidden Calibration, using the centroid similarity as the classification logits. In practice, we use a 2-step paradigm as shown in Fig. 3: first, as training, we calculate the centroid of the last hidden states of data points within each class on some (of amount  $m$ ) prompt-label training examples. Then, in the inference, we select the closest centroid of the test prompt’s hidden state as the prediction.

In detail, **(1) Training:** Given a *calibration set* with  $m$  supervised prompt-label pair  $\{(s^{(i)}, y^{(i)})\}_{i=1}^m$ , where the  $s^{(i)}$ s (*training examples*) are standard ICL prompts with  $k$  demonstrations, and  $y^{(i)}$ s are the ground-truth labels of corresponding  $s^{(i)}$ s’ query, we input each training example  $s^{(i)}$  to LM, and extract the last hidden state  $h^{(i)}$ . Repeating on the whole training example set, we can get a supervised hidden state set

<sup>3</sup>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 label with negative  $A$  components will never be assigned), and usually do not have significant relative values.

$\mathcal{H} = \{(h^{(i)}, y^{(i)})\}_{i=1}^m$ . Then, we calculate the centroids of label  $l$  as:  $\bar{h}_l = \mathbb{E}_{(h^{(i)}, y^{(i)}) \in \mathcal{H}, y^{(i)}=l} [h^{(i)}]$ .

Then, we utilize the calculated centroids in (2) **Inference:** Given a test ICL prompt, we input it into the LM and get the last hidden state  $h$ , then calculate the similarity between  $h$  and every centroid  $\bar{h}_l$  as the centroid-based logits  $\alpha_l$ . In practice, the additive inverse of Euclidean distance is used as the similarity (that is,  $\alpha_l = -\|h - \bar{h}_l\|_2^{\frac{1}{2}}$ ), while Appendix C.1 shows that Hidden Calibration acts equally on cosine similarity. We assign the label with the highest logits as the prediction.

**“Why hidden states?”** Notice that another intuitive solution to the problem in §3.1 is utilizing the logits or probabilities of the *whole vocabulary*, as shown in previous works (Xu et al., 2022; Abbas et al., 2024). However, since the input, hidden states, and logits form a Markov chain, no input-relevant information gain is propagated to the full-vocabulary logits. Moreover, the dimensionality of the full-vocabulary logits is typically significantly larger than the hidden states, therefore we choose the hidden states, a dense and informative precursor of token probabilities, as the classification feature.

**“Why centroid classifier?”** Moreover, more complex classifiers, such as a KNN classifier, or a multi-layer perceptron, can be used on the last hidden states instead of a centroid classifier. However, we choose the centroid classifier as the simplest implementation to avoid attribution confusion, that is, if even a rudimentary classifier on hidden states still outperforms, it is powerful enough to demonstrate that our hypothesis is robust. Also, a centroid classifier has a minimal computation cost to fit the scenario of low-resource (§4.3).

## 4 Experiments & Main Results

In this section, we empirically prove the effectiveness of Hidden Calibration by classification performance on 6 models and 10 datasets. Hidden Calibration outperforms all the baselines and reaches a strong state-of-the-art of ICL with high efficiency in calculation, data, and prompt engineering.

### 4.1 Experimental Settings

**Models.** We use 6 models: OPT-2.7B (Zhang et al., 2022), Llama 2 (Touvron et al., 2023) (6.9B, 13B, 34B), Llama 3 (AI@Meta, 2024) (8B) and GPT2-XL (Radford et al., 2019) (1.6B). Models larger than 10B are quantized.

**Baselines.** We use 6 baselines from the previous



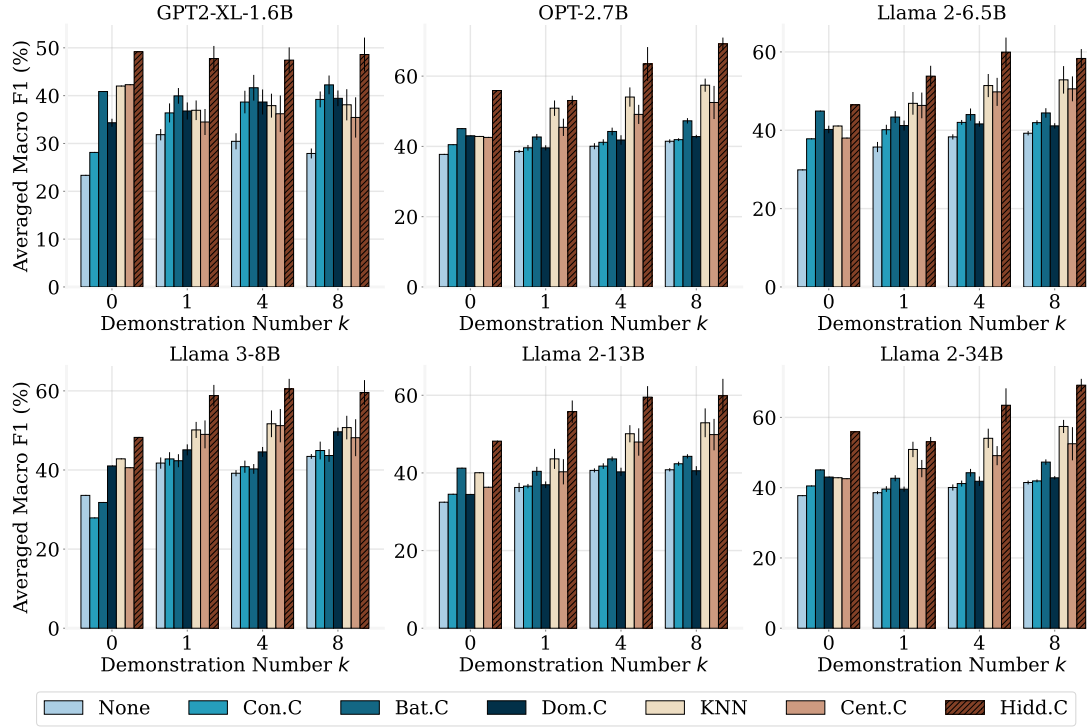


Figure 4: The classification performance (Macro F1(%)) of 6 models averaged on 10 datasets. Hidden Calibration (Hidd.C) is a new state-of-the-art of ICL, where demonstrations consistently improve the performance.

works, with 4 label token-based methods: **Vanilla ICL** (None) (Radford et al., 2019), **Contextual Calibration** (Con.C) (Zhao et al., 2021), **Batch Calibration** (Bat.C) (Zhou et al., 2023), and **Domain Calibration** (Dom.C) (Fei et al., 2023); 2 whole vocabulary probabilities-based methods KNN (Xu et al., 2022) and **Centroid Calibration** (Cent.C), which we propose as a fair comparison with the same processing on the whole output vocabulary probability vectors instead of the hidden states. Details can be found in Appendix A.2.

**Datasets.** We use 10 commonly used classification datasets with some of the overlength data points excluded. See Appendix A.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 A.3. We set  $m = 16|\mathcal{Y}|$  training examples (16 examples per class), and for fairness, every baseline is given equal training 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 tested classification performance of Hidden Calibration and baselines is shown in Fig. 4, where Hidden Calibration (Hidd.C) consistently outper-

forms all the label token-based or vocabulary-based baselines. Comparing to the vanilla ICL (None), Hidden Calibration produces an improvement up to around 100%. In general, compared to the strongest baseline, Hidden Calibration improves the performance by approximately 20%. Detailed numeric and Accuracy results are in Appendix B.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, which confirms our idea that token probability distribution is a less informative classification feature mentioned in §3.2.

## 4.3 Efficiency: Low Complexity towards Time, Space, Data, and Prompting

**Time and Space Complexity.** Intuitively, Hidden Calibration has little additional computational cost compared to the calibration baselines, since they require almost equivalent feedforward calculations, making it competitively efficient as listed in Table 1. 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.

**Training Data Complexity.** Hidden Calibration requires additional annotated data compared to label

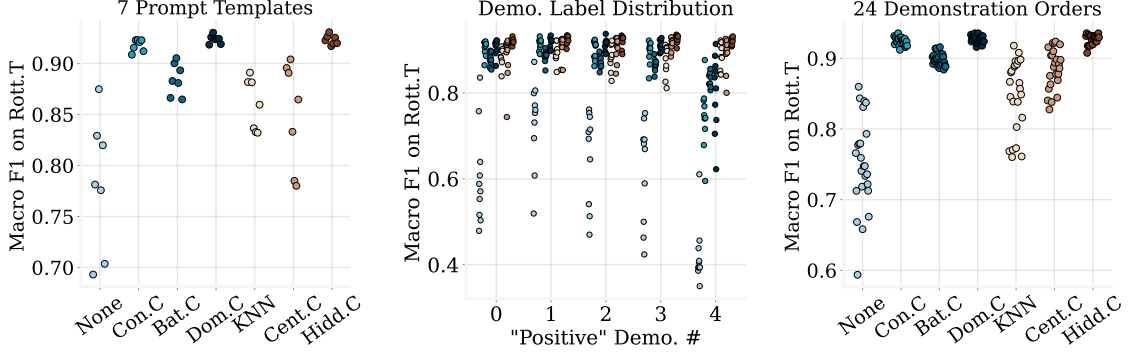


Figure 5: Sensitivities on (left) prompt template, (middle) demonstration label distribution, and (right) demonstration order on Llama 2-6.9B and Rotten\_Tomatoes. Legend is consistent with Fig. 4, omitted.

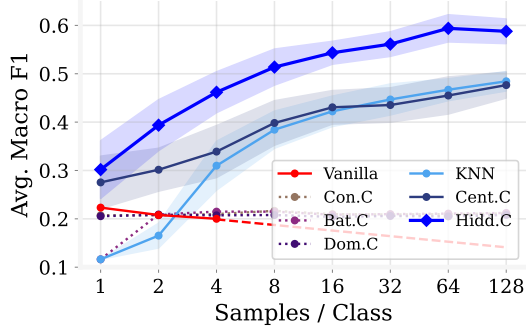


Figure 6: Classification performance against the number of training examples ( $m$ ) of calibrations. As a comparison, we plotted the results of vanilla ICL using equivalent demonstrations, due to the quadratic overhead against the context length, we can test up to  $k = 4|\mathcal{Y}|$ .

token-based calibration methods of the same scale in an acceptable range. In detail, in Con.C and Dom.C of  $k$  demonstrations,  $k$  supervised data is needed with a synthetic query to build a training example, while Hidden Calibration needs a real query **for each label**, requiring one more supervised data. However, for classification tasks, preparing an example for each label can be easily done whether in an industry or laboratory scenario, furthermore, these data and trained centroids can be reused (Appendix C.2) to further reduce the requirement of annotated data.

**Training Sample Efficiency.** Regarding the efficiency of training examples, we repeat the experiments with various  $m$  on OPT-2.7B (see Appendix A.7 for details), from 1 to 128 calibration examples **per class**. Also, for the vanilla ICL, we give equivalent demonstrations for a fair comparison. The results are shown in Fig. 6, which indicate that Hidden Calibration stably benefits from the size of the calibration set, while even one sample per class can still make it outperform. Meanwhile, vanilla ICL and label token-based methods can not benefit from more available data, making

Table 1: 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.  $|\mathcal{V}|$  is the vocabulary size.

Method	Training 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	$O(m \mathcal{V} )$	$O(m)$	$O(m \mathcal{V} )$
Cent.C	$O( \mathcal{Y}  \mathcal{V} )$	$O(m)$	$O( \mathcal{Y}  \mathcal{V} )$
<b>Hidd.C</b>	$O( \mathcal{Y} d)$	$O(m)$	$O( \mathcal{Y} d)$

Hidden Calibration a better practice no matter how much supervised data can be accessed: data can be used to estimate the centroid to improve the classification in a linear cost, rather than increase the demonstrations in a quadratic cost with less benefit.

**Prompting Complexity.** We find Hidden Calibration reduces the pressure on prompt engineering for Hidden Calibration and baselines on Llama 2-6.9B and Rotten\_Tomatoes ( $k = 4$ ) in three aspects: (1) **Prompt template** (Voronov et al., 2024). We select 7 different prompt templates (shown in Appendix A.3) and test ICL performance on them, shown in Fig. 5 (left). (2) **Label distribution in demonstrations.** We construct prompts with various numbers of “positive” demonstrations presented, and test ICL performance shown in Fig. 5 (middle). (3) **Demonstration order** (Lu et al., 2022). We enumerate the full arrangements from a fixed demonstration set of  $k = 4$ , and test the ICL performance using each demonstration arrangement, shown in Fig. 5 (right). All the results show that: compared to baselines, Hidden Calibration keeps narrow and high-performance distribution against all the three variables, i.e., Hidden Calibration stably works for various contexts, providing higher efficiency on prompt designing.

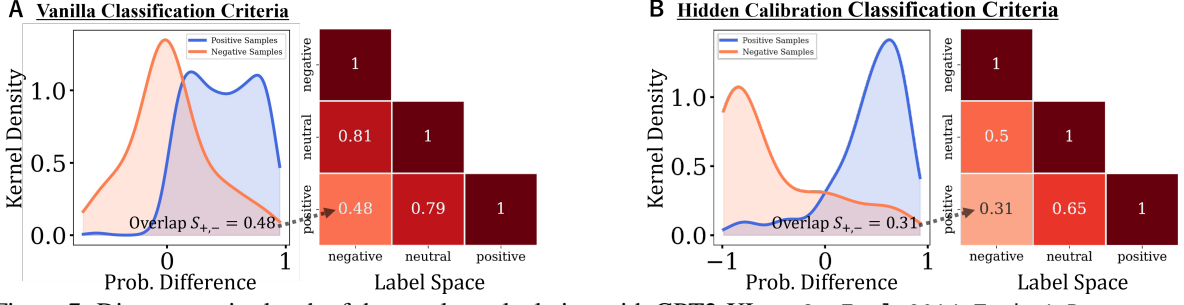


Figure 7: Diagrammatic sketch of the overlap calculation with GPT2-XL on SemEval 2014-Task 4 Restaurant,  $k = 4$ . **Curves:** The kernel density of probability difference of  $l_1$  = “positive” and  $l_2$  = “negative”. **Heatmaps:** The overlap of 2-combinations (we plot the combination with the same label with overlap 1, but omit them in averaging).

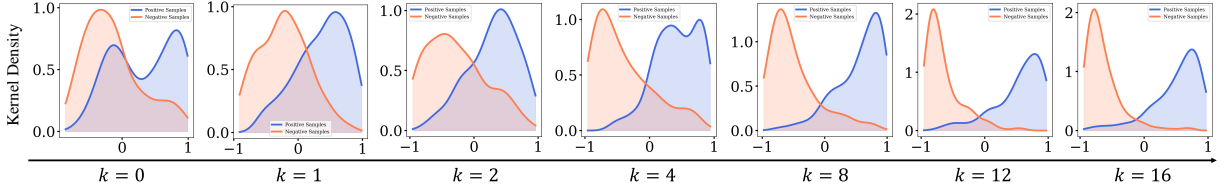


Figure 8: ICL hidden states clustering dynamics visualized on OPT-2.7B and SemEval 2014-Task 4 Laptops (Pontiki et al., 2014). The densities of data points appear in clusters responding to their query labels originally when no demonstrations are given, and gradually converge to the centroid w.r.t. the demonstrations number ( $k$ ).

## 5 Analysis

This section attempts to enhance our understanding of Hidden Calibration through comprehensive observations: (1) Similar to Fig. 2, we measure the inter-class overlapping area on data points projected into classification criteria, to find whether Hidden Calibration maps data into logits with lower inter-class overlap, i.e., better separability. (2) We further investigate why simple linear boundaries can effectively classify ICL hidden states, as observed in Hidden Calibration. We find that LMs provide a primary linear clustering in hidden states responding to query classes, and such clustering is enhanced by more demonstrations.

### 5.1 Effectiveness: Hidden Calibration Finds Criteria with Lower Overlap

In Fig. 2, we projected the data points into the difference of the label logits (*vanilla classification criteria*) on the *oblique coordinate axis*, then a significant *overlap* between the projected data point cloud in two classes can be observed, making it difficult to find suitable classification boundaries vertical to the projection direction. Therefore, such overlap can be used to evaluate classification criteria, so, in this section, we quantify the intuitive observation as the area of overlap serving as a metric for classification criteria.

In detail, we first decompose the multi-way classification dataset into all possible binary classi-

fication combinations w.r.t. the ground-truth labels (for example, in a binary combination with labels “positive” and “negative”, only “positive” and “negative”-labeled data is obtained). Then, for each combination, we build standard ICL prompts with queries labeled with a specific one of the selected binary combination. Input these prompts into the LM and map the last hidden state of the prompts onto the normal vector of the decision boundary formed by the calibration method, we get the mapped coordinate, as what is shown in the *oblique coordinate axis* of Fig. 2. To get a continuous distribution of the distance, we run kernel density estimations on the calculated coordinate, then repeat this processing on the other label in the binary combination, and get two density estimations for both labels in a binary combination, as shown in Fig. 7 (curves). Then, we calculate the overlap area of these two kernel density curves. We repeat such processing for each binary combination as shown in Fig. 7 (heat maps), and the final Averaged Overlap is the macro average of overlap area among all possible binary combinations (operation details are in Appendix A.5).

The overlap area of the two distribution curves is double to the *lower bound* of the classifier’s error rate among these two labels (Appendix A.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,

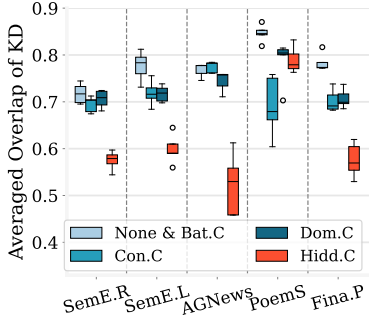


Figure 9: The Averaged Overlap of 4 inference methods on GPT2-XL and 5 datasets.

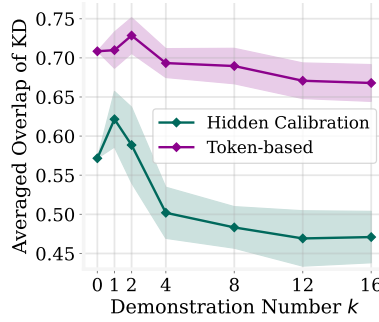


Figure 10: The Averaged Overlap on OPT-2.7B and 5 datasets against the demonstrations number.

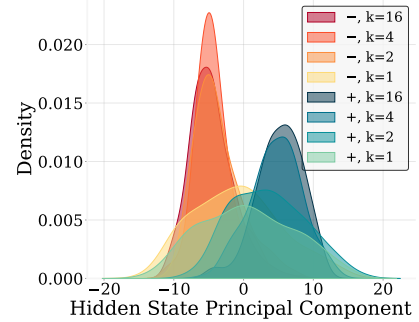


Figure 11: Hidden state clustering w.r.t.  $k$  of Fig. 8 visualized on the direction of principal component II.

resulting in a potential decrease in accuracy.

We measure the Averaged Overlap of 4 un-quantized models on 5 datasets (see Appendix A.5.2 for experimental details). The result on GPT2-XL is shown in Fig. 9 (see Appendix B.2 for other models), where the Averaged Overlaps from token-based methods are consistently high, causing that better classification performance cannot be achieved on such methods, which confirms our hypothesis in §3.1. Meanwhile, the overlaps from Hidden Calibration is much less than from token-based methods, meaning that Hidden Calibration produces better classification criteria with better *possible* classification performance than the token-based methods, even if delicate calibrations transfer or rotate these classification boundaries.

## 5.2 Principle: The Inner Linear-separability

In the practice of Hidden Calibration, simple linear boundaries are used to classify ICL examples, raising curious on the linear separability of hidden states. In this section, We find that LMs primarily produce linearly separable hidden state clusters corresponding to the ground-truth label, and the demonstrations facilitate this process.

As an intuitive visualization, we plot curves the same as the Fig. 7 but with various numbers of demonstrations  $k$  to visualize the *clustering dynamics* of hidden states in Fig. 8, where we find that: (1) the data points have a little linear separability when  $k = 0$ , and (2) such linear separability is being enhanced among the increment of  $k$ , performing increasing intra-class converging tendency.

We further characterize this process. First, we calculate the Averaged Overlap similar to §5.1 against  $k$  in Fig. 10. We find that the token-based 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 as expected, aligning with our observations in §4.2.

More generally, we visualize the distribution of the last hidden states from similar inputs of Fig. 8 on the **second** principal components of hidden states to get an essential observation in Fig. 11, where as  $k$  increases, the hidden state shows more clear intra-class clustering, enabling separability through a linear boundary.

More directly, on the last hidden states, we measure the intra-class standard error and the inter-class averaged centroid distance against  $k$  (see Appendix A.6 for details), both are a first-order moment for a joint measurement of intra-class clustering and inter-class clustering. The results are shown in Fig. 12, where the two curves are both diminishing, showing an obvious **intra-** and **inter-class** clustering trend w.r.t.  $k$ . However, the **inter-class** clustering has weaker and less persistent decreasing trends, presenting only in early demonstrations, or even ascending, which indicates that demonstration enhances **intra-class clustering** stronger than the **inter-class clustering**, which is beneficial to linear classification. Moreover, a model with more parameters shows a stronger difference between these clustering.

## 6 Related Works

Given the topic of enhancing in-context learning, we classify the literature into 3 categories.

**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.,



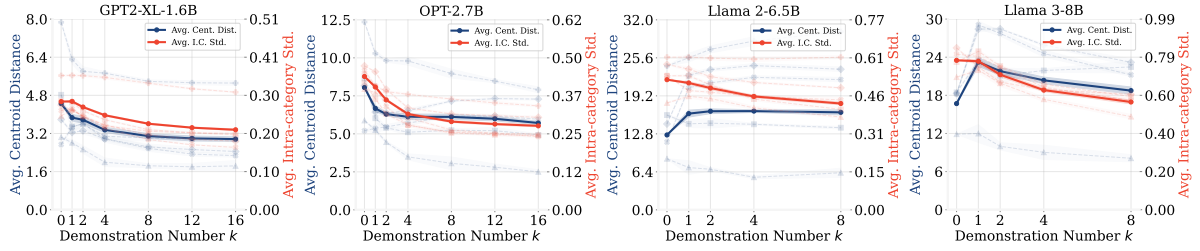


Figure 12: The **averaged intra-class standard error** of data points and the **inter-class averaged centroid distance** against  $k$ . **Solid curves**: means on 5 datasets; **Dashed and pale curves**: Individual results for each dataset.

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 **classification criteria-based method (calibration)**. Such methods focus on re-calculating output label probabilities, keeping the main feed-forward calculation processes and model parameters un-modified. The original motivation for these works is to eliminate prior bias and unfaithful confidence in ICL, by calibrating the output label probabilities (Holtzman et al., 2021; Shi et al., 2022; 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-specific probabilities have also been proposed (Xu et al., 2022; Abbas et al., 2024; Min et al., 2022a).

Also, a careful **design of input prompts** can help improve the ICL performance. **(1) Demonstration selection.** Gonen et al. (2023) finds that selecting the demonstrations with lower perplexity improves the ICL performance, similarly, Kim et al. (2022) generate the demonstrations from pre-trained LMs, etc. **(2) Demonstration ordering.** It is found that the ordering of demonstrations can significantly influence the performance (Lu et al., 2022; Liu et al., 2024; Xu et al., 2024), as also shown in our experiments in Fig. 5. Specifically, Lu et al. (2022) detect the optimal demonstration ordering by some synthetic detecting sequences, while Liu et al. (2024) orders the demonstrations from easy to hard, following a curriculum learning form.

## 7 Discussion

**Conclusion.** In this paper, we analyze the current token-based ICL decision boundaries and point out a limitation of using token probability for ICL pre-

diction. To address such a drawback, We propose Hidden Calibration by decoding the classification logits from centroid classifiers on LM’s last hidden states. Our experiments show that Hidden Calibration is a new state-of-the-art of ICL, with high efficiency on time & space, data, and prompt engineering. Then, we confirm that Hidden Calibration indeed creates better classification logits by reducing the inter-class overlap. Moreover, we discover 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.

**Comparison to Previous Works. (1) Comparison to Probe Methods.** One concern is that our work can be regarded as a degraded linear probe (Abbas et al., 2024) 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. Moreover, compared to fitting a universal approximation (Hornik et al., 1989), our method and settings fully utilize the hidden state convergence on decoder LMs (described in §5.2), making it a true ICL practice. **(2) 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 costly, while our work avoids such an enormous overhead, making it more usable and elegant. **(3) Comparison to Other Calibrations.** Our method can be seen as 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 throughout this paper.

## 8 Limitations

Due to computability limitations, we are not able to 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 argue 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, leaving as future research directions for improving the performance of ICL further.

Other label probability calibrations (e.g. Batch Calibration) 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. 8. Also, more complex prompts can be used. However, due to space constraints, we have not attempted this incremental approach, remaining it for an empirically possible practice.

Observation 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 clustering or Hidden Calibration, which needs to be explained. An explanation of “why such clustering occurs or not”, and “how to enhance the intra-class clustering by fine-tuning or prompt engineering” will be considerably beneficial for understanding ICL.

## Acknowledgments

This work is supported by the Nakajima Foundation.

## 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*.
- AI@Meta. 2024. [Llama 3 model card](#).
- 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.
- Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. [Hate Speech Dataset from a White Supremacy Forum](#). In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.
- 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*.
- 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*.
- Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2023. [Demystifying prompts in language models via perplexity estimation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148.
- 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*.
- Roe Hendel, Mor Geva, and Amir Globerson. 2023. [In-context learning creates task vectors](#). In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. [Surface form competition: Why the highest probability answer isn't always right](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051.
- Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. [Multilayer feedforward networks are universal approximators](#). *Neural networks*, 2(5):359–366.
- Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. [Toward semantics-based answer pinpointing](#). In *Proceedings of the First International Conference on Human Language Technology Research*.
- 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-impl: 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.
- Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2022. [Self-generated in-context learning: Leveraging autoregressive language models as a demonstration generator](#). *arXiv preprint arXiv:2206.08082*.
- Xin Li and Dan Roth. 2002. [Learning question classifiers](#). In *COLING 2002: The 19th International Conference on Computational Linguistics*.
- Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, and Wei Lu. 2024. [Let's learn step by step: Enhancing in-context learning ability with curriculum learning](#). *arXiv preprint arXiv:2402.10738*.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. [Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.
- 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, Xinxu 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](#).
- Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. 2022. [Nearest neighbor zero-shot inference](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3254–3265.



- 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*.
- Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. [Mind your format: Towards consistent evaluation of in-context learning improvements](#). *arXiv preprint arXiv:2401.06766*.
- 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*.
- Zhichao Xu, Daniel Cohen, Bei Wang, and Vivek Srikumar. 2024. [In-context example ordering guided by label distributions](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2623–2640.
- 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 Experimental Details

### A.1 Datasets

In this paper, 10 datasets are used as shown in Table 2. 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 the 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.

AGNews and GLUE-RTE have over-length examples. So, in the main experiments, we filter out those examples: for Llama 2-6.9B, when  $k = 8$ , we filter out all the examples with a string length greater than 512 in AGNews and 128 in GLUE-RTE. Also, for Llama 3-8B, when  $k = 8$ , we filter out all the examples with a string length greater than 128 in GLUE-RTE and omit the experiments on AGNews. 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  $k$ .

### A.2 Baselines

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

**Contextual Calibration (Con.C).** Proposed by Zhao et al. (2021), Con.C uses empty queries with normal demonstrations as calibration samples to estimate the calibration term  $A$ . In detail, Con.C inputs  $m$  samples with empty queries into the model and gets the averaged normalized label probabilities  $\bar{p}'$  among  $m$  samples. We take the reciprocal of the probabilities as calibration term  $A = \bar{p}'^{-1}$ , while the  $B = 0$ .



**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  $m$  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, with the difference 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. In detail, 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 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 with a calculation process completely consistent with the Hidden Calibration, except that the usage of the hidden state is replaced by the whole output vocabulary probability distribution consistent with KNN. This method compares with Hidden Calibration to prove that the output probability distribution is not a good classification feature for ICL in a controlled setting.

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.

### A.3 Prompts

In this paper, we use a minimum prompt template shown in Table 3. The separator between demonstrations is “\n”.

To facilitate the replication of label probability-based methods, we limit all the labels in the label space to one token by synonymous conversion. Note that Hidden Calibration does not need to meet such a one-token requirement.

Especially, in §4.3, we use 6 more prompt tem-

Table 2: 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 3: Prompt templates used in this paper.

Dataset	Prompt Template	Verbalizer
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

plates to test the stability of each ICL method against the prompt templates. We list these extra templates in Table 4.

### A.4 Details of Visualization in §3.1

**Principle Component Analysis (PCA).** Given a hidden state set  $\mathcal{H} = \{h^{(i)}\}_{i=1}^n$ , we span all the hidden state vector into a matrix  $H \in \mathbb{R}^{n \times d}$ . The covariance matrix is  $\text{cov}(H) = \frac{1}{n}(H - \bar{H})^T(H - \bar{H})$ , 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  $\text{cov}(H)$  and adjust the dimensions to arrange the eigenvalues  $\Lambda$  in a descending order along the row:

$$\text{cov}(H) = P\Lambda P^T, \quad (1)$$

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:

$$\text{PCA}_{\mathcal{H}}(h) = (h - \bar{h})\tilde{P} = h\tilde{P} - \bar{h}\tilde{P}. \quad (2)$$

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<sup>4</sup>  $h$  and  $E$  in the original space

<sup>4</sup>Due to excessive superscripts, in this section, we omit the superscripts  $U$  in the notation of un-embedding  $E_i^U$ .

Table 4: The 7 prompt templates used in the experiment (on Rotten\_Tomatoes) of Fig. 5 (left).

#	Prompt Template
Original	Input: <x>, Label: <y>
1	Sentence: <x>, Label: <y>
2	sentence: <x>, Label: <y>
3	sentence:\n <x>, Label: <y>
4	Input: <x>, Sentiment: <y>
5	Input: <x>, sentiment: <y>
6	x: <x>, y: <y>

$\mathbb{R}^d$ , producing the dot-product similarity classification criterion  $\alpha$ :

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

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:

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

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

$$= h\tilde{P}\tilde{P}^T E^T - \bar{h}\tilde{P}\tilde{P}^T E^T \quad (6)$$

$$= \alpha - \bar{h}E^T. \quad (7)$$

Notice that we use the mapping direction  $\left( \text{PCA}_{\mathcal{H}}(E)^T - \text{PCA}_{\mathcal{H}}(\mathbf{0})^T \right)$  after the PCA, instead of  $\left( \text{PCA}_{\mathcal{H}}(E)^T - \mathbf{0}^T \right)$ , 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 classes  $l_1$  and  $l_2$  in an non-rotated ICL scenario is:

$$\mathcal{B} = \{h|hE_{l_1}^T - hE_{l_2}^T = C\}. \quad (8)$$

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  $\mathbb{R}^d$  after PCA is:

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

By the aforementioned transformation, we have:

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

That is, the dimensionality-reduced decision boundary  $\tilde{\mathcal{B}}$  is perpendicular to the mapped direction  $(\text{PCA}_{\mathcal{H}}(E_{l_1} - E_{l_2}) - \text{PCA}_{\mathcal{H}}(\mathbf{0}))$ , and biased only by a constant  $(\bar{h}(E_{l_1}^T - E_{l_2}^T) - C)$  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.

## A.5 Details of Experiment in §5.1

### A.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<sup>5</sup>, 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<sup>6</sup> as:

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

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

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

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

<sup>5</sup>The  $\mathbb{C}(m, n)$  is the  $n$ -combination number from  $m$  elements.

<sup>6</sup>Notice that the  $T$  is the prompting function.

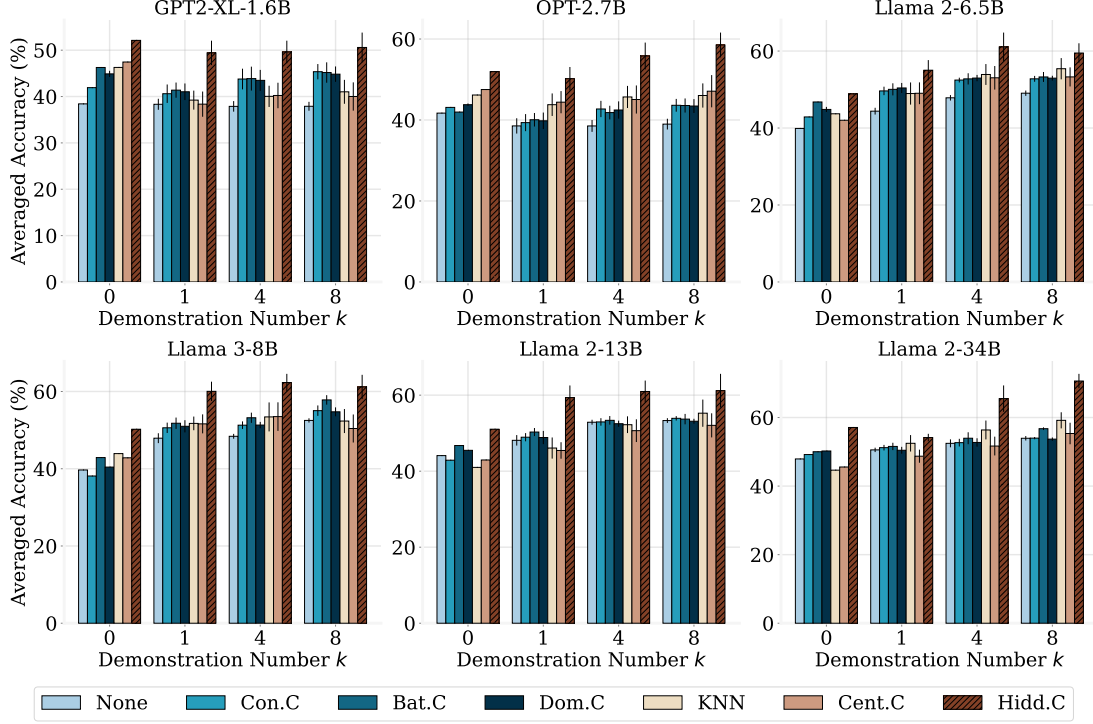


Figure 13: The classification performance (Accuracy(%)) of 3 models averaged on 10 datasets.

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. 7.

Then, we calculate the overlap area of these curves:

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

For each combination<sup>7</sup> in the  $\mathbb{C}(|\mathcal{Y}|, 2)$  2-combinations, we repeat to calculate the  $S_{i,j}$ , and average them as the **Averaged Overlap**  $\bar{S}$ .

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

### A.5.2 Experimental Details in §5.1

We conduct experiments resulting Fig. 9 on 3 models with SemEval 2014-Task 4 Restaurant, SemEval 2014-Task 4 Laptops, AGNews, Poem Sentiment, and financial\_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.

### A.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

<sup>7</sup>Notice that on  $S_{i,j}$ , the labels are rotational symmetry.

$p_{l_1}(x)$  and  $p_{l_2}(x)$  on a criterion  $x \in \mathbb{X}$ , same as the curves in Fig. 7. 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:

$$e(x)_{l_1, l_2} \geq \min [p_{l_1}(x), p_{l_2}(x)]. \quad (15)$$

So, the integral error rate:

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

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

$$= \frac{1}{2} S_{l_1, l_2}. \quad (18)$$

## A.6 Details of Experiment in §5.2

### A.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  $\mathcal{S}_l = \{s^{(c_i)} | y^{(c_i)} = l\}_{i=1}^{n_l}$ , 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  $\mathcal{H}_l = \{h^{(i)}\}_{i=1}^{n_l}$ , and calculate its centroid  $\bar{h}_l$ , as what we do in Hidden

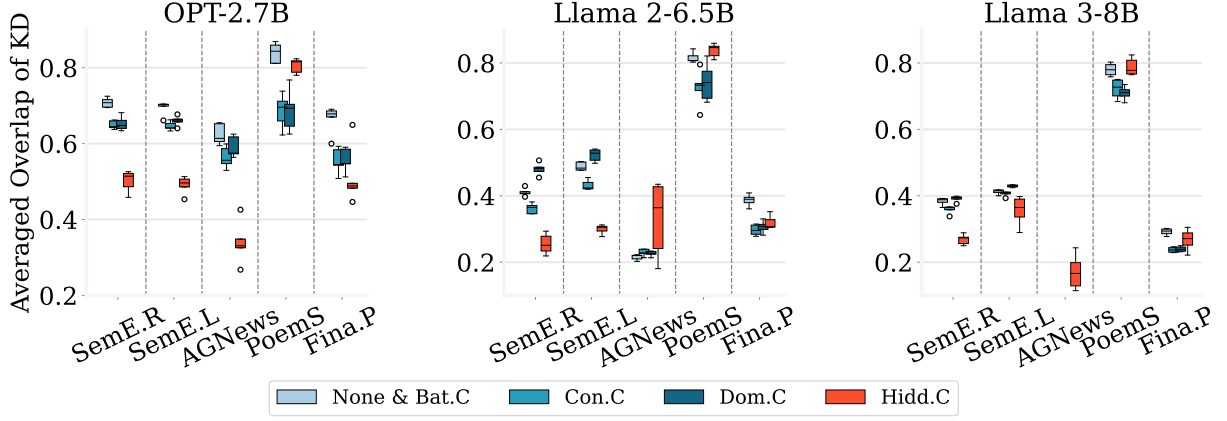


Figure 14: The augmented results on 2 models of Fig. 9.

Table 5: 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**.

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

Table 6: 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. 5

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

Calibration:

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

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:

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

**Averaged Intra-class Standard Error.** Given the hidden state set  $\mathcal{H}_l = \{h^{(i)}\}_{i=1}^{n_l}$  w.r.t. 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  $(H_l - \bar{H}_l)^T (H_l - \bar{H}_l)$ , 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-class standard error metric for label  $l$ . We average all the standard errors from all the classes as the Averaged Intra-class Standard Error:

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

## A.6.2 Experimental Details in §5.2

We conduct experiments resulting Fig. 12 on 4 models with SemEval 2014-Task 4 Restaurant, SemEval 2014-Task 4 Laptops, AGNews, Poem Sentiment, and financial\_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.

## A.7 Experimental Details for Fig. 6

We conduct experiments resulting Fig. 6 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.



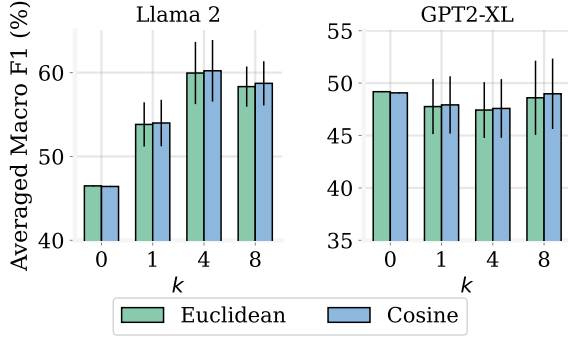


Figure 15: The classification performance (Macro F1(%)) of Hidden Calibration with difference similarity measure.

## B Detailed Results

### B.1 Details of Main Results

Numerical details of Fig. 4 are shown in Table 8, 9, 10, 11, 12 and 13. Accuracy results are shown in Fig. 13.

### B.2 Details of Averaged Overlaps Results

The augmented results on the other 3 models (we skip this experiment on the quantitated model) of Fig. 9 are shown in Fig. 14.

## C Additional Discussion

### C.1 The Similarity Measures Used in Hidden Calibration

In §3.2, we use the Euclidean distance as the similarity measure, while this is not the only option. Intuitively, we can choose other similarity measures as alternatives. Moreover, since we get inspired by observation with dot-product similarity, we should 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 2-6.9B and GPT2-XL.

The results are shown in Fig. 15, 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.

### C.2 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  $k = 4$ ,  $m = 16|\mathcal{Y}|$ . The results are shown in Table 5, 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 6. 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.

### C.3 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.

Table 7: Performance of all 6 models on TREC for a more-way classification, and on Hate\_Speech18 for a biased dataset.  $k = 4$ , top-2 results are in **bold**.

	GPT2-XL-1.6B	OPT-2.7B	Llama 2-6.5B	Llama 3-8B	Llama 2-13B	Llama 2-34B
TREC						
None	13.02	15.85	23.12	16.79	23.22	21.13
Con.C	14.10	8.12	23.72	18.34	23.56	22.35
Bat.C	14.44	17.31	23.26	20.28	22.88	22.15
Dom.C	14.10	8.58	23.14	19.10	23.42	23.23
KNN	<b>27.53</b>	33.80	49.25	43.15	54.76	<b>53.66</b>
Cent.C	23.46	<b>33.80</b>	<b>49.18</b>	<b>46.12</b>	<b>56.79</b>	53.03
Hidd.C	<b>55.91</b>	<b>61.14</b>	<b>64.90</b>	<b>71.59</b>	<b>75.64</b>	<b>68.39</b>
Hate_Speech18						
None	<b>23.96</b>	23.95	<b>23.96</b>	<b>23.96</b>	<b>23.96</b>	23.92
Con.C	21.57	25.47	14.68	<b>23.74</b>	21.49	21.25
Bat.C	20.59	21.25	20.54	18.25	20.66	23.68
Dom.C	21.23	23.68	20.73	23.67	<b>24.36</b>	23.28
KNN	17.30	16.74	16.50	16.26	16.23	<b>27.05</b>
Cent.C	16.32	<b>25.47</b>	15.42	5.46	9.15	22.23
Hidd.C	<b>24.47</b>	<b>27.03</b>	<b>23.44</b>	20.67	20.06	<b>29.94</b>

#### C.4 Applicability on More-way Classification and Biased Dataset

To further verify the applicability of Hidden Calibration on harder tasks, we test the performance of Hidden Calibration and baseline methods on TREC (Li and Roth, 2002; Hovy et al., 2001) for more-way ( $|\mathcal{Y}| = 6$ ) classification and Hate\_Speech18 (de Gibert et al., 2018) for a biased dataset (label frequency distribution:  $[0.87, 0.11, 0.007, 0.015]$ ), shown in Table 7. For results on TREC, Hidden Calibration produces a significant improvement compared to all the baselines on all the models. While, Hidden Calibration outperforms on Hate\_Speech18 in most cases, and sometimes Hidden Calibration is weaker than baseline methods, but consistently produces competitive results. However, we believe that this slightly weaker result cannot be fully attributed to biased datasets, given that there are biased datasets among the 10 standard datasets where Hidden Calibration perform SotA results (e.g. SemEval 2014-Task 4 Restaurant, refer §B.1).

## D Statements

### D.1 Author Contributions Statement

The vast majority of contributions of this paper are attributed to Hakaze Cho (also called Yufeng Zhao), who provided ideas, designed and conducted experiments, collected and described data, wrote papers, and revised them.

Y.S., M.K., K.T., and A.I. participated in our discussion and provided some comments, with se-

quentially decreasing participation and contribution. They also helped write a non-peer-viewed seminar version of this paper in Japanese.

Naoya Inoue is their supervisor, he provides an excellent laboratory environment, funding, necessary guidance, and paper revision.

### D.2 License for Artifacts

**Models.** GPT2-XL and OPT-2.7B is under the MIT license, Llama family 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.

### D.3 AI Agent Usage

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

Table 8: Classification performance (Macro F1(%%)) on GPT2-XL. mean<sub>std</sub>, top-2 results are in **bold**.

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

Table 9: Classification performance (Macro F1(%)) on OPT-2.7B. mean<sub>std</sub>, 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 <sub>0.00</sub>	24.62 <sub>0.00</sub>	31.07 <sub>0.00</sub>	25.00 <sub>0.00</sub>	51.19 <sub>0.00</sub>	24.43 <sub>0.00</sub>	39.31 <sub>0.00</sub>	34.08 <sub>0.00</sub>	47.23 <sub>0.00</sub>	34.49
	Con.C	20.72 <sub>0.00</sub>	20.77 <sub>0.00</sub>	29.33 <sub>0.00</sub>	16.51 <sub>0.00</sub>	34.63 <sub>0.00</sub>	16.25 <sub>0.00</sub>	31.37 <sub>0.00</sub>	30.65 <sub>0.00</sub>	44.88 <sub>0.00</sub>	31.26
	Bat.C	28.58 <sub>0.00</sub>	31.32 <sub>0.00</sub>	34.31 <sub>0.00</sub>	25.44 <sub>0.00</sub>	<b>53.23</b> <sub>0.00</sub>	26.81 <sub>0.00</sub>	43.60 <sub>0.00</sub>	34.34 <sub>0.00</sub>	45.25 <sub>0.00</sub>	38.94
	Dom.C	27.55 <sub>0.06</sub>	20.53 <sub>0.05</sub>	30.90 <sub>0.84</sub>	16.72 <sub>0.40</sub>	45.07 <sub>0.12</sub>	18.01 <sub>0.18</sub>	43.30 <sub>0.12</sub>	31.11 <sub>0.24</sub>	<b>46.16</b> <sub>0.47</sub>	34.56
$k = 1$	KNN	52.50 <sub>0.00</sub>	31.39 <sub>0.00</sub>	<b>44.16</b> <sub>0.00</sub>	25.72 <sub>0.00</sub>	<b>52.08</b> <sub>0.00</sub>	36.15 <sub>0.00</sub>	51.60 <sub>0.00</sub>	<b>39.45</b> <sub>0.00</sub>	42.26 <sub>0.00</sub>	43.25
	Cent.C	<b>55.97</b> <sub>0.00</sub>	<b>31.65</b> <sub>0.00</sub>	43.23 <sub>0.00</sub>	<b>35.86</b> <sub>0.00</sub>	41.98 <sub>0.00</sub>	<b>41.27</b> <sub>0.00</sub>	<b>53.44</b> <sub>0.00</sub>	35.25 <sub>0.00</sub>	37.54 <sub>0.00</sub>	<b>43.48</b>
	<b>Hidd.C</b>	<b>75.01</b> <sub>0.00</sub>	<b>41.94</b> <sub>0.00</sub>	<b>52.14</b> <sub>0.00</sub>	<b>39.92</b> <sub>0.00</sub>	45.64 <sub>0.00</sub>	<b>45.90</b> <sub>0.00</sub>	<b>52.93</b> <sub>0.00</sub>	<b>35.67</b> <sub>0.00</sub>	43.24 <sub>0.00</sub>	<b>49.41</b>
	None	24.16 <sub>0.71</sub>	18.76 <sub>0.50</sub>	24.98 <sub>1.74</sub>	12.23 <sub>1.19</sub>	<b>50.75</b> <sub>3.01</sub>	20.72 <sub>3.33</sub>	<b>51.55</b> <sub>4.28</sub>	23.64 <sub>1.19</sub>	31.40 <sub>1.83</sub>	30.71
$k = 1$	Con.C	22.57 <sub>1.38</sub>	20.15 <sub>1.78</sub>	24.48 <sub>1.88</sub>	13.17 <sub>0.63</sub>	<b>50.75</b> <sub>3.01</sub>	24.30 <sub>1.69</sub>	<b>51.69</b> <sub>4.03</sub>	23.02 <sub>0.00</sub>	28.88 <sub>1.95</sub>	32.46
	Bat.C	27.43 <sub>1.13</sub>	20.48 <sub>0.90</sub>	26.78 <sub>1.17</sub>	14.21 <sub>1.05</sub>	50.25 <sub>3.25</sub>	26.44 <sub>1.41</sub>	50.45 <sub>1.76</sub>	23.34 <sub>2.30</sub>	28.81 <sub>0.24</sub>	33.91
	Dom.C	24.06 <sub>0.94</sub>	19.68 <sub>1.26</sub>	24.24 <sub>1.73</sub>	13.78 <sub>2.27</sub>	<b>50.75</b> <sub>3.01</sub>	23.58 <sub>2.54</sub>	51.54 <sub>3.86</sub>	23.14 <sub>0.83</sub>	28.81 <sub>1.72</sub>	32.96
	KNN	48.15 <sub>2.50</sub>	<b>42.35</b> <sub>4.06</sub>	39.01 <sub>5.24</sub>	25.52 <sub>1.94</sub>	<b>53.07</b> <sub>3.06</sub>	32.62 <sub>3.02</sub>	49.78 <sub>0.58</sub>	<b>31.90</b> <sub>3.64</sub>	36.59 <sub>3.67</sub>	<b>41.63</b>
$k = 1$	Cent.C	<b>49.21</b> <sub>2.81</sub>	39.46 <sub>4.26</sub>	<b>42.30</b> <sub>4.65</sub>	<b>29.60</b> <sub>5.30</sub>	48.48 <sub>3.85</sub>	<b>34.32</b> <sub>2.51</sub>	50.69 <sub>1.21</sub>	31.00 <sub>1.34</sub>	<b>36.69</b> <sub>2.28</sub>	41.54
	<b>Hidd.C</b>	<b>65.18</b> <sub>2.39</sub>	<b>44.91</b> <sub>5.14</sub>	<b>51.62</b> <sub>2.09</sub>	<b>33.72</b> <sub>2.24</sub>	50.40 <sub>1.80</sub>	<b>45.00</b> <sub>2.79</sub>	49.53 <sub>2.92</sub>	<b>35.02</b> <sub>0.47</sub>	<b>48.10</b> <sub>3.33</sub>	<b>48.12</b>
	None	22.91 <sub>1.05</sub>	20.84 <sub>1.16</sub>	25.44 <sub>1.60</sub>	12.46 <sub>1.29</sub>	49.70 <sub>3.22</sub>	14.83 <sub>0.16</sub>	40.68 <sub>0.44</sub>	23.78 <sub>0.99</sub>	28.62 <sub>1.13</sub>	29.30
	Con.C	22.81 <sub>1.04</sub>	20.70 <sub>1.02</sub>	27.74 <sub>1.79</sub>	12.62 <sub>1.32</sub>	<b>50.06</b> <sub>3.86</sub>	21.92 <sub>1.03</sub>	41.21 <sub>0.70</sub>	25.86 <sub>1.65</sub>	36.74 <sub>1.68</sub>	34.32
$k = 4$	Bat.C	25.40 <sub>1.04</sub>	20.07 <sub>1.40</sub>	26.49 <sub>0.86</sub>	11.46 <sub>1.24</sub>	47.61 <sub>2.78</sub>	23.92 <sub>0.96</sub>	45.71 <sub>1.33</sub>	26.91 <sub>1.24</sub>	38.88 <sub>1.70</sub>	34.87
	Dom.C	22.32 <sub>1.17</sub>	20.71 <sub>1.39</sub>	25.96 <sub>1.97</sub>	12.64 <sub>1.24</sub>	<b>50.37</b> <sub>3.22</sub>	21.74 <sub>1.30</sub>	41.30 <sub>0.54</sub>	26.15 <sub>1.41</sub>	38.01 <sub>1.69</sub>	34.32
	KNN	49.36 <sub>2.28</sub>	49.71 <sub>5.26</sub>	47.66 <sub>3.87</sub>	22.18 <sub>1.17</sub>	48.74 <sub>3.76</sub>	<b>32.05</b> <sub>4.50</sub>	<b>47.35</b> <sub>3.18</sub>	<b>30.76</b> <sub>3.92</sub>	40.78 <sub>1.86</sub>	<b>43.49</b>
	Cent.C	<b>49.91</b> <sub>4.46</sub>	<b>51.72</b> <sub>2.80</sub>	<b>48.96</b> <sub>3.00</sub>	<b>21.63</b> <sub>3.55</sub>	47.02 <sub>2.37</sub>	29.84 <sub>3.43</sub>	46.99 <sub>2.05</sub>	25.39 <sub>3.05</sub>	40.23 <sub>8.32</sub>	42.75
$k = 4$	<b>Hidd.C</b>	<b>69.56</b> <sub>3.62</sub>	<b>59.59</b> <sub>1.97</sub>	<b>62.77</b> <sub>3.08</sub>	<b>25.42</b> <sub>0.79</sub>	47.94 <sub>3.16</sub>	<b>47.47</b> <sub>5.86</sub>	<b>49.46</b> <sub>1.80</sub>	<b>35.61</b> <sub>3.45</sub>	<b>65.91</b> <sub>2.01</sub>	<b>54.32</b>
	None	21.52 <sub>1.03</sub>	21.24 <sub>1.48</sub>	24.11 <sub>1.51</sub>	12.84 <sub>0.73</sub>	48.05 <sub>3.65</sub>	14.54 <sub>0.00</sub>	37.54 <sub>0.03</sub>	21.85 <sub>1.07</sub>	28.47 <sub>1.48</sub>	29.00
	Con.C	22.26 <sub>0.33</sub>	23.84 <sub>1.26</sub>	30.27 <sub>1.27</sub>	13.02 <sub>0.89</sub>	47.58 <sub>3.27</sub>	23.06 <sub>1.10</sub>	37.80 <sub>0.48</sub>	25.91 <sub>0.79</sub>	37.67 <sub>1.44</sub>	34.48
	Bat.C	24.38 <sub>1.49</sub>	22.28 <sub>1.04</sub>	28.05 <sub>1.65</sub>	13.31 <sub>0.50</sub>	47.50 <sub>2.86</sub>	24.46 <sub>0.50</sub>	49.30 <sub>1.63</sub>	27.96 <sub>1.51</sub>	40.39 <sub>1.53</sub>	36.16
$k = 8$	Dom.C	22.38 <sub>0.70</sub>	21.33 <sub>1.57</sub>	26.63 <sub>1.36</sub>	13.18 <sub>1.08</sub>	47.48 <sub>3.87</sub>	23.63 <sub>1.08</sub>	37.59 <sub>0.25</sub>	25.25 <sub>1.11</sub>	39.27 <sub>2.20</sub>	34.34
	KNN	47.45 <sub>1.64</sub>	55.08 <sub>1.89</sub>	48.16 <sub>3.38</sub>	22.61 <sub>2.20</sub>	<b>49.56</b> <sub>2.68</sub>	<b>34.80</b> <sub>3.18</sub>	49.60 <sub>2.26</sub>	<b>30.77</b> <sub>2.79</sub>	38.73 <sub>3.52</sub>	44.34
	Cent.C	<b>47.50</b> <sub>4.46</sub>	<b>55.83</b> <sub>4.46</sub>	<b>51.06</b> <sub>8.42</sub>	<b>22.95</b> <sub>2.22</sub>	48.22 <sub>2.35</sub>	33.57 <sub>4.12</sub>	<b>49.66</b> <sub>2.73</sub>	29.18 <sub>4.37</sub>	<b>43.87</b> <sub>2.57</sub>	<b>44.93</b>
	<b>Hidd.C</b>	<b>65.27</b> <sub>1.64</sub>	<b>63.95</b> <sub>2.51</sub>	<b>66.48</b> <sub>2.53</sub>	<b>23.70</b> <sub>1.76</sub>	<b>49.86</b> <sub>2.86</sub>	<b>52.24</b> <sub>2.39</sub>	<b>50.57</b> <sub>3.28</sub>	<b>41.94</b> <sub>6.42</sub>	<b>85.78</b> <sub>4.39</sub>	<b>57.09</b>



Table 10: Classification performance (Macro F1(%)) on Llama 2-6.9B. mean<sub>std.</sub>, 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 <sub>0.00</sub>	44.36 <sub>0.00</sub>	32.48 <sub>0.00</sub>	16.23 <sub>0.00</sub>	35.88 <sub>0.00</sub>	17.01 <sub>0.00</sub>	46.06 <sub>0.00</sub>	30.59 <sub>0.00</sub>	19.64 <sub>0.00</sub>	29.89
	Con.C	22.76 <sub>0.00</sub>	44.50 <sub>0.00</sub>	42.83 <sub>0.00</sub>	10.53 <sub>0.00</sub>	43.88 <sub>0.00</sub>	24.06 <sub>0.00</sub>	42.42 <sub>0.00</sub>	33.32 <sub>0.00</sub>	54.92 <sub>0.00</sub>	37.82
	Bat.C	37.26 <sub>0.00</sub>	49.03 <sub>0.00</sub>	<b>54.33</b> <sub>0.00</sub>	19.70 <sub>0.00</sub>	<b>47.34</b> <sub>0.00</sub>	<b>31.64</b> <sub>0.00</sub>	<b>51.06</b> <sub>0.00</sub>	<b>39.82</b> <sub>0.00</sub>	<b>55.76</b> <sub>0.00</sub>	<b>44.90</b>
	Dom.C	28.11 <sub>1.43</sub>	47.88 <sub>0.45</sub>	49.74 <sub>1.40</sub>	<b>26.58</b> <sub>1.27</sub>	39.99 <sub>0.13</sub>	23.00 <sub>1.11</sub>	37.48 <sub>0.00</sub>	33.33 <sub>0.21</sub>	<b>61.03</b> <sub>0.76</sub>	40.21
	KNN	<b>40.24</b> <sub>0.00</sub>	47.65 <sub>0.00</sub>	49.15 <sub>0.00</sub>	20.56 <sub>0.00</sub>	<b>50.93</b> <sub>0.00</sub>	26.65 <sub>0.00</sub>	42.35 <sub>0.00</sub>	33.63 <sub>0.00</sub>	46.44 <sub>0.00</sub>	41.09
	Cent.C	34.36 <sub>0.00</sub>	<b>50.87</b> <sub>0.00</sub>	46.36 <sub>0.00</sub>	17.57 <sub>0.00</sub>	42.15 <sub>0.00</sub>	25.09 <sub>0.00</sub>	42.31 <sub>0.00</sub>	28.21 <sub>0.00</sub>	43.81 <sub>0.00</sub>	38.01
	<b>Hidd.C</b>	<b>62.46</b> <sub>0.00</sub>	<b>50.90</b> <sub>0.00</sub>	<b>55.77</b> <sub>0.00</sub>	<b>22.49</b> <sub>0.00</sub>	46.94 <sub>0.00</sub>	<b>34.06</b> <sub>0.00</sub>	<b>47.37</b> <sub>0.00</sub>	<b>34.29</b> <sub>0.00</sub>	54.84 <sub>0.00</sub>	<b>46.49</b>
	None	13.97 <sub>1.15</sub>	51.57 <sub>0.30</sub>	51.50 <sub>0.47</sub>	22.28 <sub>0.98</sub>	36.26 <sub>0.80</sub>	27.80 <sub>1.58</sub>	40.16 <sub>0.83</sub>	30.78 <sub>2.77</sub>	27.89 <sub>2.18</sub>	35.72
	Con.C	13.29 <sub>1.62</sub>	52.36 <sub>0.58</sub>	54.12 <sub>0.50</sub>	24.16 <sub>1.29</sub>	38.53 <sub>0.92</sub>	27.77 <sub>1.16</sub>	40.63 <sub>1.52</sub>	37.39 <sub>1.86</sub>	37.23 <sub>0.68</sub>	40.14
	Bat.C	22.05 <sub>1.07</sub>	52.24 <sub>0.44</sub>	53.77 <sub>1.27</sub>	<b>24.75</b> <sub>1.51</sub>	<b>49.77</b> <sub>3.25</sub>	28.94 <sub>0.90</sub>	47.72 <sub>1.19</sub>	<b>40.25</b> <sub>2.30</sub>	38.43 <sub>1.91</sub>	43.37
$k = 1$	Dom.C	11.89 <sub>1.01</sub>	51.80 <sub>0.38</sub>	53.98 <sub>0.45</sub>	24.23 <sub>1.13</sub>	<b>49.33</b> <sub>3.82</sub>	29.17 <sub>1.44</sub>	38.23 <sub>0.85</sub>	39.02 <sub>1.72</sub>	34.95 <sub>1.35</sub>	41.20
	KNN	<b>52.50</b> <sub>3.58</sub>	56.19 <sub>5.59</sub>	61.24 <sub>2.42</sub>	21.91 <sub>0.91</sub>	48.82 <sub>1.90</sub>	<b>30.66</b> <sub>3.56</sub>	<b>49.65</b> <sub>3.15</sub>	34.86 <sub>1.11</sub>	<b>40.26</b> <sub>3.40</sub>	<b>46.87</b>
	Cent.C	46.17 <sub>0.93</sub>	<b>62.09</b> <sub>3.04</sub>	<b>64.50</b> <sub>1.76</sub>	23.29 <sub>2.57</sub>	46.66 <sub>3.16</sub>	27.80 <sub>2.36</sub>	45.63 <sub>2.33</sub>	33.10 <sub>2.42</sub>	38.99 <sub>3.56</sub>	46.34
	<b>Hidd.C</b>	<b>61.88</b> <sub>0.68</sub>	<b>64.83</b> <sub>1.85</sub>	<b>69.05</b> <sub>2.29</sub>	<b>24.26</b> <sub>1.37</sub>	48.32 <sub>2.35</sub>	<b>42.77</b> <sub>2.38</sub>	<b>51.55</b> <sub>1.27</sub>	<b>40.72</b> <sub>1.91</sub>	<b>63.79</b> <sub>0.93</sub>	<b>53.82</b>
	None	9.57 <sub>0.26</sub>	53.45 <sub>0.34</sub>	53.76 <sub>0.62</sub>	19.82 <sub>0.51</sub>	35.90 <sub>0.18</sub>	29.30 <sub>1.47</sub>	37.56 <sub>0.00</sub>	36.55 <sub>0.97</sub>	37.86 <sub>0.78</sub>	38.31
	Con.C	9.35 <sub>0.23</sub>	53.49 <sub>0.33</sub>	54.21 <sub>0.30</sub>	<b>27.01</b> <sub>1.15</sub>	37.65 <sub>1.74</sub>	<b>31.37</b> <sub>1.07</sub>	37.56 <sub>0.00</sub>	39.34 <sub>0.47</sub>	39.28 <sub>0.55</sub>	42.01
	Bat.C	19.33 <sub>1.51</sub>	52.58 <sub>0.64</sub>	54.40 <sub>0.69</sub>	24.62 <sub>0.99</sub>	46.81 <sub>5.22</sub>	31.26 <sub>1.58</sub>	48.81 <sub>2.93</sub>	38.47 <sub>0.33</sub>	37.05 <sub>0.60</sub>	44.00
	Dom.C	9.49 <sub>0.32</sub>	52.12 <sub>0.79</sub>	54.14 <sub>0.28</sub>	<b>26.47</b> <sub>0.87</sub>	39.59 <sub>2.03</sub>	30.75 <sub>1.06</sub>	37.56 <sub>1.43</sub>	<b>39.52</b> <sub>0.22</sub>	34.71 <sub>0.58</sub>	41.62
	KNN	<b>66.47</b> <sub>6.82</sub>	60.00 <sub>0.92</sub>	62.87 <sub>2.54</sub>	21.76 <sub>3.20</sub>	54.41 <sub>4.95</sub>	29.96 <sub>2.19</sub>	49.46 <sub>2.22</sub>	31.90 <sub>3.97</sub>	<b>49.09</b> <sub>5.20</sub>	<b>51.41</b>
	Cent.C	60.62 <sub>4.62</sub>	<b>63.61</b> <sub>2.06</sub>	<b>64.55</b> <sub>3.77</sub>	21.42 <sub>4.73</sub>	<b>55.50</b> <sub>5.61</sub>	28.41 <sub>2.51</sub>	<b>50.72</b> <sub>3.29</sub>	21.12 <sub>5.70</sub>	42.48 <sub>2.81</sub>	49.80
$k = 4$	<b>Hidd.C</b>	<b>68.22</b> <sub>6.82</sub>	<b>65.64</b> <sub>1.38</sub>	<b>71.38</b> <sub>2.00</sub>	25.47 <sub>2.10</sub>	<b>58.00</b> <sub>4.38</sub>	<b>48.48</b> <sub>4.32</sub>	<b>51.83</b> <sub>3.03</sub>	<b>49.44</b> <sub>3.20</sub>	<b>68.79</b> <sub>9.71</sub>	<b>59.95</b>
	None	8.03 <sub>0.00</sub>	53.90 <sub>0.31</sub>	54.12 <sub>0.38</sub>	20.47 <sub>0.49</sub>	35.98 <sub>0.00</sub>	30.77 <sub>1.50</sub>	37.56 <sub>0.00</sub>	38.59 <sub>1.26</sub>	39.25 <sub>1.04</sub>	39.24
	Con.C	8.15 <sub>0.26</sub>	53.72 <sub>0.30</sub>	54.26 <sub>0.37</sub>	<b>28.72</b> <sub>1.91</sub>	35.98 <sub>0.00</sub>	31.40 <sub>1.94</sub>	37.56 <sub>0.00</sub>	39.33 <sub>0.53</sub>	38.80 <sub>0.46</sub>	41.94
	Bat.C	17.58 <sub>1.19</sub>	53.06 <sub>0.52</sub>	54.15 <sub>0.28</sub>	25.62 <sub>0.85</sub>	51.85 <sub>3.70</sub>	30.66 <sub>1.48</sub>	48.45 <sub>2.63</sub>	38.44 <sub>0.46</sub>	36.28 <sub>0.37</sub>	44.42
	Dom.C	8.03 <sub>0.00</sub>	52.26 <sub>0.43</sub>	54.26 <sub>0.47</sub>	26.50 <sub>0.99</sub>	36.44 <sub>1.05</sub>	29.28 <sub>2.06</sub>	37.56 <sub>0.00</sub>	<b>39.72</b> <sub>0.41</sub>	35.21 <sub>0.86</sub>	41.12
	KNN	<b>69.41</b> <sub>4.38</sub>	59.12 <sub>4.16</sub>	62.69 <sub>3.74</sub>	23.15 <sub>2.09</sub>	<b>54.86</b> <sub>4.00</sub>	<b>34.89</b> <sub>2.06</sub>	48.44 <sub>3.62</sub>	36.27 <sub>4.36</sub>	<b>50.27</b> <sub>5.38</sub>	<b>52.87</b>
	Cent.C	<b>60.04</b> <sub>8.07</sub>	<b>61.80</b> <sub>1.44</sub>	<b>66.16</b> <sub>2.05</sub>	23.17 <sub>1.31</sub>	53.64 <sub>3.23</sub>	32.16 <sub>2.62</sub>	<b>51.74</b> <sub>1.94</sub>	30.29 <sub>3.91</sub>	36.55 <sub>6.71</sub>	50.57
	<b>Hidd.C</b>	59.17 <sub>4.08</sub>	<b>65.88</b> <sub>0.84</sub>	<b>70.07</b> <sub>2.32</sub>	<b>27.85</b> <sub>2.56</sub>	<b>55.79</b> <sub>3.05</sub>	<b>46.70</b> <sub>5.37</sub>	<b>53.95</b> <sub>2.36</sub>	<b>43.90</b> <sub>0.48</sub>	<b>67.69</b> <sub>2.29</sub>	<b>58.33</b>

Table 11: Classification performance (Macro F1(%)) on Llama 3-8B. mean<sub>std</sub>, top-2 results are in **bold**.

Llama 3	AGNews	SemE.R	SemE.L	PoemS	RTE	TEE	TEH	TES	Fin.P	Rott.T	Average
$k = 0$	None	19.55 <sub>0.00</sub>	27.61 <sub>0.00</sub>	24.38 <sub>0.00</sub>	11.25 <sub>0.00</sub>	43.68 <sub>0.00</sub>	26.33 <sub>0.00</sub>	46.69 <sub>0.00</sub>	30.67 <sub>0.00</sub>	44.45 <sub>0.00</sub>	33.58
	Con.C	23.20 <sub>0.00</sub>	17.16 <sub>0.00</sub>	14.55 <sub>0.00</sub>	2.33 <sub>0.00</sub>	<b>47.50</b> <sub>0.00</sub>	21.66 <sub>0.00</sub>	33.34 <sub>0.00</sub>	29.66 <sub>0.00</sub>	28.78 <sub>0.00</sub>	27.89
	Bat.C	26.59 <sub>0.00</sub>	43.52 <sub>0.00</sub>	38.62 <sub>0.00</sub>	17.40 <sub>0.00</sub>	47.25 <sub>0.00</sub>	<b>32.02</b> <sub>0.00</sub>	<b>49.12</b> <sub>0.00</sub>	<b>38.17</b> <sub>0.00</sub>	<b>50.32</b> <sub>0.00</sub>	41.02
	Dom.C	19.17 <sub>0.00</sub>	25.92 <sub>0.00</sub>	24.36 <sub>0.00</sub>	3.97 <sub>0.00</sub>	32.86 <sub>0.00</sub>	26.46 <sub>0.00</sub>	39.35 <sub>0.00</sub>	<b>37.95</b> <sub>0.00</sub>	<b>48.39</b> <sub>0.00</sub>	31.79
	KNN	<b>49.70</b> <sub>0.00</sub>	51.20 <sub>0.00</sub>	47.55 <sub>0.00</sub>	20.74 <sub>0.00</sub>	47.25 <sub>0.00</sub>	28.97 <sub>0.00</sub>	51.08 <sub>0.00</sub>	31.26 <sub>0.00</sub>	47.80 <sub>0.00</sub>	<b>42.82</b>
$k = 1$	Cent.C	39.40 <sub>0.00</sub>	<b>53.41</b> <sub>0.00</sub>	<b>48.67</b> <sub>0.00</sub>	<b>23.68</b> <sub>0.00</sub>	<b>47.41</b> <sub>0.00</sub>	27.00 <sub>0.00</sub>	<b>49.58</b> <sub>0.00</sub>	25.19 <sub>0.00</sub>	40.38 <sub>0.00</sub>	40.58
	<b>Hidd.C</b>	<b>81.52</b> <sub>0.00</sub>	<b>52.75</b> <sub>0.00</sub>	<b>49.43</b> <sub>0.00</sub>	<b>27.67</b> <sub>0.00</sub>	44.53 <sub>0.00</sub>	<b>34.04</b> <sub>0.00</sub>	44.53 <sub>0.00</sub>	37.20 <sub>0.00</sub>	47.74 <sub>0.00</sub>	<b>48.25</b>
	None	19.97 <sub>0.70</sub>	53.47 <sub>0.13</sub>	53.75 <sub>0.74</sub>	<b>25.41</b> <sub>1.06</sub>	48.60 <sub>2.81</sub>	17.02 <sub>1.72</sub>	<b>50.68</b> <sub>2.19</sub>	28.30 <sub>1.50</sub>	39.28 <sub>1.36</sub>	41.75
	Con.C	15.57 <sub>1.02</sub>	53.37 <sub>0.69</sub>	54.42 <sub>0.44</sub>	20.37 <sub>5.23</sub>	48.60 <sub>3.06</sub>	19.27 <sub>1.28</sub>	49.71 <sub>3.07</sub>	37.32 <sub>1.11</sub>	39.07 <sub>0.33</sub>	<b>90.32</b> <sub>0.74</sub>
	Bat.C	27.50 <sub>1.91</sub>	52.87 <sub>0.17</sub>	55.71 <sub>0.60</sub>	23.73 <sub>0.65</sub>	49.23 <sub>2.95</sub>	26.48 <sub>0.99</sub>	49.80 <sub>2.77</sub>	<b>39.54</b> <sub>2.26</sub>	39.06 <sub>0.76</sub>	45.09
$k = 4$	Dom.C	12.96 <sub>1.62</sub>	52.23 <sub>1.07</sub>	54.41 <sub>0.56</sub>	19.94 <sub>4.35</sub>	<b>49.37</b> <sub>3.03</sub>	20.24 <sub>0.86</sub>	49.73 <sub>3.05</sub>	37.86 <sub>1.05</sub>	36.40 <sub>0.36</sub>	42.34
	KNN	<b>61.64</b> <sub>0.94</sub>	54.58 <sub>3.37</sub>	61.94 <sub>1.25</sub>	21.49 <sub>1.40</sub>	48.15 <sub>2.60</sub>	34.73 <sub>2.96</sub>	48.13 <sub>2.61</sub>	39.11 <sub>1.54</sub>	<b>43.88</b> <sub>1.85</sub>	<b>50.13</b>
	Cent.C	60.16 <sub>4.68</sub>	<b>56.81</b> <sub>5.45</sub>	<b>62.72</b> <sub>1.87</sub>	22.74 <sub>3.75</sub>	45.77 <sub>4.80</sub>	<b>35.10</b> <sub>3.40</sub>	44.59 <sub>5.76</sub>	35.95 <sub>2.01</sub>	38.15 <sub>2.52</sub>	49.00
	<b>Hidd.C</b>	<b>83.37</b> <sub>2.76</sub>	<b>60.59</b> <sub>5.42</sub>	<b>64.12</b> <sub>1.73</sub>	<b>27.82</b> <sub>1.99</sub>	<b>49.24</b> <sub>1.88</sub>	<b>53.37</b> <sub>3.10</sub>	<b>51.77</b> <sub>2.59</sub>	<b>41.87</b> <sub>1.07</sub>	<b>65.62</b> <sub>5.94</sub>	<b>58.79</b>
	None	9.75 <sub>0.22</sub>	53.90 <sub>0.25</sub>	54.36 <sub>0.52</sub>	<b>24.45</b> <sub>1.23</sub>	37.09 <sub>1.14</sub>	14.54 <sub>0.00</sub>	39.42 <sub>1.76</sub>	33.23 <sub>1.55</sub>	40.28 <sub>0.18</sub>	39.18
$k = 8$	Con.C	9.17 <sub>0.00</sub>	53.72 <sub>0.64</sub>	54.42 <sub>0.51</sub>	<b>23.58</b> <sub>4.38</sub>	37.20 <sub>1.21</sub>	15.56 <sub>0.72</sub>	43.43 <sub>4.03</sub>	38.31 <sub>1.40</sub>	40.17 <sub>1.84</sub>	40.83
	Bat.C	23.64 <sub>0.50</sub>	52.70 <sub>0.67</sub>	54.35 <sub>0.33</sub>	22.95 <sub>1.60</sub>	46.73 <sub>5.45</sub>	27.62 <sub>1.01</sub>	<b>51.46</b> <sub>0.44</sub>	<b>38.65</b> <sub>0.99</sub>	37.11 <sub>0.41</sub>	44.57
	Dom.C	9.17 <sub>0.01</sub>	52.51 <sub>0.47</sub>	54.35 <sub>0.38</sub>	21.85 <sub>1.37</sub>	37.52 <sub>1.69</sub>	15.94 <sub>0.73</sub>	43.01 <sub>5.37</sub>	38.04 <sub>1.01</sub>	37.76 <sub>0.82</sub>	40.27
	KNN	<b>74.94</b> <sub>3.56</sub>	<b>54.73</b> <sub>5.26</sub>	<b>59.53</b> <sub>2.92</sub>	23.01 <sub>3.28</sub>	<b>49.54</b> <sub>5.90</sub>	<b>40.01</b> <sub>1.41</sub>	49.43 <sub>3.14</sub>	37.00 <sub>4.50</sub>	42.38 <sub>2.07</sub>	<b>51.69</b>
	Cent.C	73.26 <sub>3.26</sub>	54.31 <sub>3.12</sub>	61.26 <sub>4.08</sub>	21.02 <sub>2.43</sub>	45.64 <sub>10.22</sub>	38.73 <sub>5.55</sub>	50.74 <sub>5.16</sub>	37.05 <sub>3.14</sub>	<b>44.61</b> <sub>2.96</sub>	51.21
$k = 8$	<b>Hidd.C</b>	<b>84.53</b> <sub>1.35</sub>	<b>61.73</b> <sub>4.56</sub>	<b>69.28</b> <sub>1.43</sub>	23.03 <sub>1.82</sub>	<b>53.49</b> <sub>3.61</sub>	<b>54.70</b> <sub>2.03</sub>	<b>52.96</b> <sub>3.84</sub>	<b>40.71</b> <sub>2.35</sub>	<b>72.18</b> <sub>3.19</sub>	<b>60.48</b>
	None	-	53.77 <sub>0.42</sub>	54.82 <sub>0.12</sub>	<b>27.10</b> <sub>1.62</sub>	35.98 <sub>0.00</sub>	14.54 <sub>0.00</sub>	37.69 <sub>0.27</sub>	35.96 <sub>1.51</sub>	42.65 <sub>0.85</sub>	39.09
	Con.C	-	53.52 <sub>0.53</sub>	54.42 <sub>0.53</sub>	<b>25.07</b> <sub>4.81</sub>	35.98 <sub>0.00</sub>	14.54 <sub>0.00</sub>	47.99 <sub>9.84</sub>	38.74 <sub>1.35</sub>	41.19 <sub>2.12</sub>	40.43
	Bat.C	-	52.40 <sub>0.72</sub>	54.82 <sub>0.35</sub>	24.68 <sub>0.67</sub>	<b>60.38</b> <sub>5.04</sub>	31.50 <sub>0.44</sub>	<b>55.00</b> <sub>0.54</sub>	38.59 <sub>0.78</sub>	37.95 <sub>0.35</sub>	44.69
	Dom.C	-	52.03 <sub>0.35</sub>	54.11 <sub>0.66</sub>	21.95 <sub>0.83</sub>	35.98 <sub>0.00</sub>	14.67 <sub>0.22</sub>	45.15 <sub>9.11</sub>	38.61 <sub>1.13</sub>	37.30 <sub>1.39</sub>	39.29
$k = 8$	KNN	-	54.47 <sub>2.60</sub>	<b>58.17</b> <sub>2.84</sub>	21.49 <sub>1.69</sub>	50.75 <sub>4.92</sub>	<b>47.22</b> <sub>2.50</sub>	51.75 <sub>2.65</sub>	<b>41.66</b> <sub>3.19</sub>	<b>46.47</b> <sub>4.53</sub>	<b>45.65</b>
	Cent.C	-	<b>56.03</b> <sub>3.55</sub>	49.78 <sub>5.61</sub>	22.15 <sub>1.36</sub>	48.30 <sub>9.65</sub>	45.41 <sub>6.52</sub>	48.34 <sub>4.57</sub>	38.65 <sub>1.60</sub>	39.58 <sub>7.11</sub>	43.35
	<b>Hidd.C</b>	-	<b>63.34</b> <sub>1.61</sub>	<b>67.48</b> <sub>1.00</sub>	23.84 <sub>1.22</sub>	<b>60.89</b> <sub>3.38</sub>	<b>56.67</b> <sub>7.36</sub>	<b>53.45</b> <sub>4.92</sub>	<b>39.74</b> <sub>2.66</sub>	<b>79.28</b> <sub>4.45</sub>	<b>53.63</b>

Table 12: Classification performance (Macro F1(%)) on Llama 2-13B. mean<sub>std</sub>. 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	9.61 <sub>0.00</sub>	47.67 <sub>0.00</sub>	51.12 <sub>0.00</sub>	8.01 <sub>0.00</sub>	36.04 <sub>0.00</sub>	31.92 <sub>0.00</sub>	37.56 <sub>0.00</sub>	46.19 <sub>0.00</sub>	30.94 <sub>0.00</sub>	35.40
	Con.C	22.36 <sub>0.00</sub>	48.44 <sub>0.00</sub>	<b>57.96</b> <sub>0.00</sub>	15.71 <sub>0.00</sub>	41.45 <sub>0.00</sub>	23.57 <sub>0.00</sub>	37.56 <sub>0.00</sub>	39.59 <sub>0.00</sub>	44.97 <sub>0.00</sub>	37.42
	Bat.C	30.84 <sub>0.00</sub>	<b>52.16</b> <sub>0.00</sub>	<b>61.45</b> <sub>0.00</sub>	20.50 <sub>0.00</sub>	43.54 <sub>0.00</sub>	<b>37.88</b> <sub>0.00</sub>	43.73 <sub>0.00</sub>	<b>47.31</b> <sub>0.00</sub>	<b>50.83</b> <sub>0.00</sub>	<b>44.95</b>
	Dom.C	23.21 <sub>0.00</sub>	46.54 <sub>0.00</sub>	51.51 <sub>0.00</sub>	8.88 <sub>0.00</sub>	41.87 <sub>0.00</sub>	29.11 <sub>0.00</sub>	37.56 <sub>0.00</sub>	<b>47.22</b> <sub>0.00</sub>	31.98 <sub>0.00</sub>	37.85
	KNN	<b>46.44</b> <sub>0.00</sub>	42.06 <sub>0.00</sub>	43.18 <sub>0.00</sub>	<b>24.03</b> <sub>0.00</sub>	<b>50.33</b> <sub>0.00</sub>	28.36 <sub>0.00</sub>	<b>55.76</b> <sub>0.00</sub>	45.74 <sub>0.00</sub>	51.95 <sub>0.00</sub>	42.31
$k = 1$	Cent.C	43.48 <sub>0.00</sub>	44.64 <sub>0.00</sub>	41.05 <sub>0.00</sub>	21.23 <sub>0.00</sub>	34.02 <sub>0.00</sub>	30.41 <sub>0.00</sub>	49.02 <sub>0.00</sub>	30.44 <sub>0.00</sub>	43.17 <sub>0.00</sub>	39.02
	<b>Hidd.C</b>	<b>69.78</b> <sub>0.00</sub>	<b>54.08</b> <sub>0.00</sub>	50.67 <sub>0.00</sub>	<b>33.01</b> <sub>0.00</sub>	<b>54.41</b> <sub>0.00</sub>	<b>41.23</b> <sub>0.00</sub>	<b>51.14</b> <sub>0.00</sub>	<b>48.87</b> <sub>0.00</sub>	57.61 <sub>0.00</sub>	<b>49.63</b>
	None	9.17 <sub>0.00</sub>	50.67 <sub>0.80</sub>	53.51 <sub>0.08</sub>	15.49 <sub>0.67</sub>	35.11 <sub>0.00</sub>	33.79 <sub>1.09</sub>	37.56 <sub>0.00</sub>	37.11 <sub>0.78</sub>	34.73 <sub>0.90</sub>	38.88
	Con.C	9.17 <sub>0.00</sub>	53.38 <sub>0.30</sub>	<b>54.04</b> <sub>0.20</sub>	<b>25.20</b> <sub>0.64</sub>	35.11 <sub>0.00</sub>	33.61 <sub>0.46</sub>	37.52 <sub>0.05</sub>	<b>38.00</b> <sub>0.53</sub>	35.73 <sub>0.58</sub>	40.92
	Bat.C	24.81 <sub>2.72</sub>	49.21 <sub>0.87</sub>	53.72 <sub>0.14</sub>	23.39 <sub>0.33</sub>	47.24 <sub>1.16</sub>	<b>35.68</b> <sub>0.19</sub>	<b>51.66</b> <sub>1.75</sub>	36.21 <sub>0.81</sub>	34.35 <sub>0.39</sub>	44.23
$k = 4$	Dom.C	9.55 <sub>0.38</sub>	51.02 <sub>0.55</sub>	52.92 <sub>0.29</sub>	19.66 <sub>1.43</sub>	35.11 <sub>0.00</sub>	32.55 <sub>2.39</sub>	37.56 <sub>0.00</sub>	38.43 <sub>0.72</sub>	35.55 <sub>0.68</sub>	39.88
	KNN	<b>58.35</b> <sub>3.83</sub>	<b>54.98</b> <sub>5.17</sub>	48.86 <sub>0.44</sub>	24.03 <sub>1.43</sub>	<b>47.50</b> <sub>2.83</sub>	32.11 <sub>1.50</sub>	50.95 <sub>3.28</sub>	33.33 <sub>4.95</sub>	<b>37.93</b> <sub>3.72</sub>	<b>47.02</b>
	Cent.C	51.96 <sub>0.88</sub>	49.83 <sub>4.34</sub>	46.51 <sub>2.10</sub>	25.20 <sub>2.94</sub>	43.15 <sub>3.01</sub>	27.11 <sub>1.80</sub>	47.81 <sub>6.69</sub>	31.46 <sub>6.15</sub>	31.22 <sub>4.39</sub>	43.65
	<b>Hidd.C</b>	<b>76.45</b> <sub>5.43</sub>	<b>64.70</b> <sub>1.97</sub>	<b>65.70</b> <sub>0.16</sub>	<b>30.22</b> <sub>3.11</sub>	<b>49.42</b> <sub>1.78</sub>	<b>48.97</b> <sub>0.32</sub>	<b>52.81</b> <sub>4.66</sub>	<b>41.05</b> <sub>4.32</sub>	<b>59.29</b> <sub>4.43</sub>	<b>57.26</b>
	None	9.17 <sub>0.00</sub>	49.99 <sub>0.32</sub>	53.69 <sub>0.44</sub>	19.29 <sub>0.86</sub>	35.11 <sub>0.00</sub>	<b>36.23</b> <sub>2.22</sub>	37.56 <sub>0.00</sub>	37.63 <sub>0.58</sub>	36.02 <sub>0.42</sub>	40.65
$k = 8$	Con.C	9.37 <sub>0.22</sub>	53.12 <sub>0.62</sub>	<b>54.25</b> <sub>0.26</sub>	<b>26.98</b> <sub>1.34</sub>	35.11 <sub>0.00</sub>	34.64 <sub>2.69</sub>	37.56 <sub>0.00</sub>	37.64 <sub>1.28</sub>	36.38 <sub>0.15</sub>	41.76
	Bat.C	19.64 <sub>1.36</sub>	49.33 <sub>0.12</sub>	53.69 <sub>0.13</sub>	23.93 <sub>0.41</sub>	46.56 <sub>3.27</sub>	34.29 <sub>1.84</sub>	46.95 <sub>1.79</sub>	35.92 <sub>0.83</sub>	34.14 <sub>0.46</sub>	43.61
	Dom.C	9.56 <sub>0.31</sub>	49.61 <sub>0.38</sub>	53.79 <sub>0.35</sub>	16.85 <sub>0.80</sub>	35.11 <sub>0.00</sub>	33.62 <sub>1.59</sub>	37.56 <sub>0.00</sub>	<b>38.20</b> <sub>1.38</sub>	35.95 <sub>0.35</sub>	40.27
	KNN	<b>76.62</b> <sub>0.69</sub>	<b>53.18</b> <sub>1.01</sub>	53.38 <sub>4.35</sub>	22.73 <sub>1.17</sub>	47.94 <sub>1.65</sub>	35.09 <sub>5.00</sub>	<b>50.17</b> <sub>1.01</sub>	38.26 <sub>1.25</sub>	40.40 <sub>2.69</sub>	<b>50.09</b>
	Cent.C	74.72 <sub>7.17</sub>	52.52 <sub>2.90</sub>	48.20 <sub>3.83</sub>	22.08 <sub>2.73</sub>	<b>48.59</b> <sub>2.55</sub>	28.15 <sub>3.24</sub>	49.53 <sub>1.53</sub>	29.35 <sub>4.83</sub>	<b>43.69</b> <sub>5.11</sub>	47.96
$k = 8$	<b>Hidd.C</b>	<b>80.76</b> <sub>4.75</sub>	<b>66.91</b> <sub>1.22</sub>	<b>68.44</b> <sub>2.54</sub>	<b>25.39</b> <sub>1.27</sub>	<b>52.69</b> <sub>3.96</sub>	<b>53.41</b> <sub>3.40</sub>	<b>52.29</b> <sub>4.93</sub>	<b>44.47</b> <sub>4.05</sub>	<b>59.48</b> <sub>1.42</sub>	<b>59.52</b>
	None	9.17 <sub>0.00</sub>	50.01 <sub>0.25</sub>	53.82 <sub>0.37</sub>	19.54 <sub>0.37</sub>	35.11 <sub>0.00</sub>	36.93 <sub>1.76</sub>	37.56 <sub>0.00</sub>	36.64 <sub>0.75</sub>	36.49 <sub>0.04</sub>	40.81
	Con.C	9.17 <sub>0.00</sub>	<b>52.89</b> <sub>0.18</sub>	<b>54.31</b> <sub>0.29</sub>	<b>29.11</b> <sub>0.49</sub>	35.11 <sub>0.00</sub>	36.44 <sub>1.99</sub>	37.56 <sub>0.00</sub>	38.24 <sub>0.79</sub>	38.10 <sub>0.70</sub>	42.36
	Bat.C	17.94 <sub>0.26</sub>	48.95 <sub>0.69</sub>	53.64 <sub>0.45</sub>	24.35 <sub>0.44</sub>	<b>52.35</b> <sub>3.27</sub>	34.94 <sub>2.46</sub>	47.18 <sub>3.05</sub>	36.11 <sub>0.40</sub>	35.04 <sub>0.09</sub>	44.27
	Dom.C	9.17 <sub>0.00</sub>	50.55 <sub>0.37</sub>	53.95 <sub>0.41</sub>	18.12 <sub>1.11</sub>	35.11 <sub>0.00</sub>	34.14 <sub>2.53</sub>	37.56 <sub>0.00</sub>	38.43 <sub>0.17</sub>	35.70 <sub>0.27</sub>	40.56
$k = 8$	KNN	<b>79.57</b> <sub>4.10</sub>	52.03 <sub>2.71</sub>	51.68 <sub>8.67</sub>	23.48 <sub>0.62</sub>	50.35 <sub>3.07</sub>	<b>39.90</b> <sub>6.24</sub>	<b>53.54</b> <sub>4.10</sub>	<b>41.95</b> <sub>2.23</sub>	50.11 <sub>3.56</sub>	<b>52.91</b>
	Cent.C	76.52 <sub>2.63</sub>	51.46 <sub>5.58</sub>	42.86 <sub>5.16</sub>	22.13 <sub>4.60</sub>	49.28 <sub>0.85</sub>	34.97 <sub>5.39</sub>	53.09 <sub>4.32</sub>	33.61 <sub>7.13</sub>	<b>50.29</b> <sub>2.71</sub>	49.87
	<b>Hidd.C</b>	<b>80.47</b> <sub>2.80</sub>	<b>66.49</b> <sub>2.19</sub>	<b>62.69</b> <sub>8.43</sub>	<b>25.58</b> <sub>3.44</sub>	<b>53.02</b> <sub>4.91</sub>	<b>50.88</b> <sub>5.15</sub>	<b>57.19</b> <sub>7.28</sub>	<b>41.99</b> <sub>3.69</sub>	<b>69.04</b> <sub>5.48</sub>	<b>59.87</b>
	None	9.17 <sub>0.00</sub>	50.01 <sub>0.25</sub>	53.82 <sub>0.37</sub>	19.54 <sub>0.37</sub>	35.11 <sub>0.00</sub>	36.93 <sub>1.76</sub>	37.56 <sub>0.00</sub>	36.64 <sub>0.75</sub>	36.49 <sub>0.04</sub>	40.81
	Con.C	9.17 <sub>0.00</sub>	<b>52.89</b> <sub>0.18</sub>	<b>54.31</b> <sub>0.29</sub>	<b>29.11</b> <sub>0.49</sub>	35.11 <sub>0.00</sub>	36.44 <sub>1.99</sub>	37.56 <sub>0.00</sub>	38.24 <sub>0.79</sub>	38.10 <sub>0.70</sub>	42.36
$k = 8$	Bat.C	17.94 <sub>0.26</sub>	48.95 <sub>0.69</sub>	53.64 <sub>0.45</sub>	24.35 <sub>0.44</sub>	<b>52.35</b> <sub>3.27</sub>	34.94 <sub>2.46</sub>	47.18 <sub>3.05</sub>	36.11 <sub>0.40</sub>	35.04 <sub>0.09</sub>	44.27
	Dom.C	9.17 <sub>0.00</sub>	50.55 <sub>0.37</sub>	53.95 <sub>0.41</sub>	18.12 <sub>1.11</sub>	35.11 <sub>0.00</sub>	34.14 <sub>2.53</sub>	37.56 <sub>0.00</sub>	38.43 <sub>0.17</sub>	35.70 <sub>0.27</sub>	40.56
	KNN	<b>79.57</b> <sub>4.10</sub>	52.03 <sub>2.71</sub>	51.68 <sub>8.67</sub>	23.48 <sub>0.62</sub>	50.35 <sub>3.07</sub>	<b>39.90</b> <sub>6.24</sub>	<b>53.54</b> <sub>4.10</sub>	<b>41.95</b> <sub>2.23</sub>	50.11 <sub>3.56</sub>	<b>52.91</b>
	Cent.C	76.52 <sub>2.63</sub>	51.46 <sub>5.58</sub>	42.86 <sub>5.16</sub>	22.13 <sub>4.60</sub>	49.28 <sub>0.85</sub>	34.97 <sub>5.39</sub>	53.09 <sub>4.32</sub>	33.61 <sub>7.13</sub>	<b>50.29</b> <sub>2.71</sub>	49.87
	<b>Hidd.C</b>	<b>80.47</b> <sub>2.80</sub>	<b>66.49</b> <sub>2.19</sub>	<b>62.69</b> <sub>8.43</sub>	<b>25.58</b> <sub>3.44</sub>	<b>53.02</b> <sub>4.91</sub>	<b>50.88</b> <sub>5.15</sub>	<b>57.19</b> <sub>7.28</sub>	<b>41.99</b> <sub>3.69</sub>	<b>69.04</b> <sub>5.48</sub>	<b>59.87</b>

Table 13: Classification performance (Macro F1(%)) on Llama 2-34B. mean<sub>std.</sub>, 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	18.28 <sub>0.00</sub>	50.09 <sub>0.00</sub>	51.30 <sub>0.00</sub>	15.50 <sub>0.00</sub>	39.75 <sub>0.00</sub>	24.91 <sub>0.00</sub>	31.05 <sub>0.00</sub>	34.67 <sub>0.00</sub>	71.67 <sub>0.00</sub>	37.73
	Con.C	23.06 <sub>0.00</sub>	50.75 <sub>0.00</sub>	<b>58.71</b> <sub>0.00</sub>	8.81 <sub>0.00</sub>	35.11 <sub>0.00</sub>	30.96 <sub>0.00</sub>	34.93 <sub>0.00</sub>	48.68 <sub>0.00</sub>	69.32 <sub>0.00</sub>	40.49
	Bat.C	26.52 <sub>0.00</sub>	<b>54.51</b> <sub>0.00</sub>	<b>60.19</b> <sub>0.00</sub>	18.91 <sub>0.00</sub>	48.47 <sub>0.00</sub>	31.62 <sub>0.00</sub>	<b>40.37</b> <sub>0.00</sub>	<b>53.45</b> <sub>0.00</sub>	70.69 <sub>0.00</sub>	<b>45.06</b>
	Dom.C	22.58 <sub>0.00</sub>	52.91 <sub>0.00</sub>	55.96 <sub>0.00</sub>	24.06 <sub>0.00</sub>	47.61 <sub>0.00</sub>	31.18 <sub>0.00</sub>	36.45 <sub>0.00</sub>	50.42 <sub>0.00</sub>	<b>71.77</b> <sub>0.00</sub>	43.02
	KNN	31.79 <sub>0.00</sub>	45.85 <sub>0.00</sub>	45.03 <sub>0.00</sub>	<b>27.38</b> <sub>0.00</sub>	<b>53.04</b> <sub>0.00</sub>	31.64 <sub>0.00</sub>	34.10 <sub>0.00</sub>	46.57 <sub>0.00</sub>	57.60 <sub>0.00</sub>	<b>42.83</b>
$k = 1$	Cent.C	<b>35.28</b> <sub>0.00</sub>	50.83 <sub>0.00</sub>	51.72 <sub>0.00</sub>	23.44 <sub>0.00</sub>	51.15 <sub>0.00</sub>	<b>32.02</b> <sub>0.00</sub>	35.33 <sub>0.00</sub>	44.25 <sub>0.00</sub>	48.47 <sub>0.00</sub>	42.56
	<b>Hidd.C</b>	<b>65.31</b> <sub>0.00</sub>	<b>56.22</b> <sub>0.00</sub>	54.83 <sub>0.00</sub>	<b>31.82</b> <sub>0.00</sub>	<b>54.76</b> <sub>0.00</sub>	<b>50.49</b> <sub>0.00</sub>	<b>45.46</b> <sub>0.00</sub>	<b>69.22</b> <sub>0.00</sub>	65.14 <sub>0.00</sub>	<b>55.90</b>
	None	9.59 <sub>0.03</sub>	53.48 <sub>0.49</sub>	54.97 <sub>0.04</sub>	22.51 <sub>0.39</sub>	35.53 <sub>0.18</sub>	<b>33.32</b> <sub>1.18</sub>	36.38 <sub>1.24</sub>	35.41 <sub>0.18</sub>	81.78 <sub>0.31</sub>	40.05
	Con.C	10.38 <sub>0.91</sub>	53.97 <sub>0.80</sub>	54.97 <sub>0.55</sub>	25.44 <sub>0.20</sub>	35.11 <sub>2.42</sub>	32.92 <sub>0.56</sub>	37.08 <sub>2.18</sub>	38.73 <sub>0.28</sub>	<b>85.60</b> <sub>0.25</sub>	41.17
	Bat.C	21.41 <sub>0.28</sub>	51.86 <sub>0.43</sub>	54.81 <sub>0.10</sub>	23.47 <sub>1.50</sub>	52.97 <sub>1.18</sub>	33.02 <sub>1.00</sub>	48.06 <sub>1.46</sub>	35.23 <sub>0.21</sub>	84.63 <sub>0.44</sub>	44.25
$k = 4$	Dom.C	10.00 <sub>0.29</sub>	53.46 <sub>0.90</sub>	54.96 <sub>0.11</sub>	<b>29.23</b> <sub>2.93</sub>	37.32 <sub>0.40</sub>	32.70 <sub>0.82</sub>	37.21 <sub>1.47</sub>	39.96 <sub>0.16</sub>	<b>85.80</b> <sub>1.51</sub>	41.82
	KNN	<b>65.38</b> <sub>1.25</sub>	<b>58.91</b> <sub>1.13</sub>	<b>61.28</b> <sub>2.68</sub>	29.09 <sub>4.68</sub>	<b>62.63</b> <sub>3.40</sub>	30.21 <sub>1.64</sub>	54.69 <sub>0.85</sub>	<b>68.78</b> <sub>1.13</sub>	68.77 <sub>3.17</sub>	<b>54.05</b>
	Cent.C	63.03 <sub>3.16</sub>	55.45 <sub>2.61</sub>	54.41 <sub>4.82</sub>	24.23 <sub>1.29</sub>	62.47 <sub>0.04</sub>	31.16 <sub>0.39</sub>	<b>56.13</b> <sub>2.05</sub>	55.08 <sub>3.23</sub>	58.61 <sub>3.26</sub>	49.11
	<b>Hidd.C</b>	<b>79.39</b> <sub>0.48</sub>	<b>67.23</b> <sub>2.91</sub>	<b>69.03</b> <sub>0.53</sub>	<b>33.12</b> <sub>0.35</sub>	<b>74.89</b> <sub>1.33</sub>	<b>52.50</b> <sub>4.09</sub>	<b>46.63</b> <sub>1.04</sub>	<b>76.83</b> <sub>2.01</sub>	73.14 <sub>0.06</sub>	<b>63.46</b>
	None	9.59 <sub>0.55</sub>	53.48 <sub>0.39</sub>	54.97 <sub>0.64</sub>	22.51 <sub>0.69</sub>	35.53 <sub>0.06</sub>	<b>33.32</b> <sub>3.35</sub>	37.56 <sub>0.00</sub>	35.41 <sub>0.51</sub>	81.78 <sub>0.55</sub>	40.05
$k = 8$	Con.C	10.38 <sub>0.01</sub>	53.97 <sub>0.16</sub>	54.97 <sub>0.61</sub>	25.44 <sub>0.06</sub>	35.11 <sub>0.00</sub>	32.92 <sub>3.78</sub>	37.08 <sub>0.97</sub>	38.73 <sub>0.55</sub>	<b>85.60</b> <sub>2.37</sub>	41.17
	Bat.C	21.41 <sub>1.08</sub>	51.86 <sub>0.34</sub>	54.81 <sub>0.44</sub>	23.47 <sub>0.33</sub>	52.97 <sub>0.10</sub>	33.02 <sub>4.05</sub>	48.06 <sub>2.65</sub>	35.23 <sub>0.77</sub>	84.63 <sub>1.97</sub>	44.25
	Dom.C	10.00 <sub>0.55</sub>	53.46 <sub>0.36</sub>	54.96 <sub>0.66</sub>	<b>29.23</b> <sub>0.90</sub>	37.32 <sub>1.28</sub>	32.70 <sub>4.18</sub>	37.56 <sub>0.00</sub>	39.96 <sub>0.04</sub>	<b>85.80</b> <sub>2.09</sub>	41.82
	KNN	<b>65.38</b> <sub>3.35</sub>	<b>58.91</b> <sub>0.83</sub>	<b>61.28</b> <sub>0.90</sub>	29.09 <sub>2.60</sub>	<b>62.63</b> <sub>2.71</sub>	30.21 <sub>3.04</sub>	54.69 <sub>2.18</sub>	<b>68.78</b> <sub>1.44</sub>	68.77 <sub>8.22</sub>	<b>54.05</b>
	Cent.C	63.03 <sub>10.02</sub>	55.45 <sub>0.33</sub>	54.41 <sub>2.47</sub>	24.23 <sub>0.35</sub>	62.47 <sub>6.66</sub>	31.16 <sub>2.87</sub>	<b>56.13</b> <sub>0.93</sub>	55.08 <sub>1.63</sub>	58.61 <sub>1.20</sub>	49.11
$k = 8$	<b>Hidd.C</b>	<b>79.39</b> <sub>3.49</sub>	<b>67.23</b> <sub>1.13</sub>	<b>69.03</b> <sub>9.36</sub>	<b>33.12</b> <sub>2.59</sub>	<b>74.89</b> <sub>1.75</sub>	<b>52.50</b> <sub>2.17</sub>	<b>46.63</b> <sub>5.67</sub>	<b>76.83</b> <sub>9.16</sub>	73.14 <sub>11.06</sub>	<b>63.46</b>
	None	9.18 <sub>0.01</sub>	53.56 <sub>0.99</sub>	55.06 <sub>0.70</sub>	26.48 <sub>0.16</sub>	35.11 <sub>0.00</sub>	35.06 <sub>0.43</sub>	37.56 <sub>0.00</sub>	36.12 <sub>0.27</sub>	88.65 <sub>2.13</sub>	41.47
	Con.C	11.00 <sub>1.59</sub>	52.04 <sub>0.51</sub>	54.55 <sub>0.11</sub>	27.01 <sub>0.23</sub>	35.11 <sub>0.00</sub>	34.60 <sub>0.17</sub>	37.56 <sub>0.00</sub>	38.81 <sub>0.29</sub>	<b>89.87</b> <sub>0.41</sub>	41.92
	Bat.C	18.25 <sub>0.02</sub>	52.17 <sub>0.44</sub>	54.91 <sub>0.11</sub>	25.89 <sub>0.60</sub>	<b>66.77</b> <sub>0.63</sub>	35.08 <sub>0.61</sub>	57.00 <sub>0.26</sub>	35.85 <sub>0.25</sub>	90.17 <sub>1.13</sub>	47.28
	Dom.C	9.39 <sub>0.30</sub>	53.02 <sub>1.61</sub>	55.18 <sub>0.52</sub>	<b>32.92</b> <sub>1.16</sub>	35.11 <sub>0.00</sub>	33.10 <sub>2.08</sub>	37.56 <sub>0.00</sub>	41.86 <sub>1.38</sub>	<b>92.08</b> <sub>0.41</sub>	42.80
$k = 8$	KNN	<b>63.23</b> <sub>1.12</sub>	<b>60.79</b> <sub>0.34</sub>	<b>61.09</b> <sub>4.12</sub>	30.84 <sub>0.06</sub>	63.67 <sub>1.20</sub>	<b>37.53</b> <sub>0.81</sub>	55.26 <sub>5.26</sub>	<b>42.30</b> <sub>4.48</sub>	78.58 <sub>0.41</sub>	<b>57.42</b>
	Cent.C	57.73 <sub>6.27</sub>	57.02 <sub>0.78</sub>	54.73 <sub>5.85</sub>	26.35 <sub>2.44</sub>	62.27 <sub>5.14</sub>	35.94 <sub>1.68</sub>	<b>59.23</b> <sub>6.62</sub>	58.83 <sub>10.95</sub>	74.64 <sub>0.80</sub>	52.50
	<b>Hidd.C</b>	<b>83.00</b> <sub>0.52</sub>	<b>68.71</b> <sub>0.42</sub>	<b>77.09</b> <sub>0.85</sub>	<b>31.59</b> <sub>8.39</sub>	<b>78.91</b> <sub>0.40</sub>	<b>60.97</b> <sub>0.90</sub>	<b>50.52</b> <sub>0.85</sub>	<b>91.98</b> <sub>0.17</sub>	84.78 <sub>3.82</sub>	<b>69.17</b>