TOKEN-DEPTH AGGREGATION FOR IMPROVED TEXT REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Frozen language models are widely used as text encoders, with a trainable aggregation module that combines their token representations into one vector. Existing aggregation methods, from mean pooling to learned attention and token graphs, read the final layer and are mostly evaluated on benchmarks whose labels reflect the overall content of a sentence. Many uses instead require composing local meaning, such as a negated opinion or a chain of family relations, and aggregating it across a text, such as the majority opinion about one business among reviews of several. In this paper, we ask how the representations of a frozen language model should be combined across depth and across tokens when the label is built from such local information. We introduce a suite of composition and aggregation tasks whose labels are built from local judgments while global cues are removed, and propose Token-Depth Aggregation (ToDA), in which each token attends over its own representations at all layers before a token graph relates the fused tokens. Our experiments show that existing aggregation methods can remain close to chance level when the answer depends on tying opinions to a specific target, whereas ToDA reaches high accuracy. Across frozen encoder and decoder backbones, ToDA achieves the highest accuracy in nearly every setting of the proposed tasks. On the standard GLUE, IMDB and MTEB benchmarks, ToDA also achieves competitive or improved performance while keeping the backbone frozen and training a small number of parameters.

1 INTRODUCTION

Frozen language models (LMs) are commonly used as text encoders. A small trainable module aggregates the contextual token representations of a transformer (Vaswani et al., 2017) into one vector, which is then passed to a classifier or used for retrieval and semantic similarity (Conneau et al., 2017; Reimers & Gurevych, 2019; Kashyap et al., 2024). Existing aggregation methods average or max-pool the tokens, read a special token such as [CLS] (Devlin et al., 2019), learn attention weights over tokens (Brothers, 2025), or relate the tokens through a graph before readout (Mantri et al., 2026). These methods typically read the final layer of the LM, and in text they are commonly evaluated on benchmarks such as GLUE (Wang et al., 2018) and MTEB (Muennighoff et al., 2023), where each label is assigned to a whole text or a pair of texts.

Many uses of a text representation ask for more. The answer may depend on *composing* local meaning, as when a negator reverses an opinion or two family relations combine into a third, and on *aggregating* local judgments across a text, as when one predicts the majority opinion about a business among reviews of several, or counts how many reviews are positive. Such labels are determined by local information that is first interpreted and then combined. Layerwise studies show that different linguistic properties are most accessible at different depths of an LM (Tenney et al., 2019; Jawahar et al., 2019; Liu et al., 2019a), and we observe the same for composition: on a controlled task in which negators reverse the polarity of opinion units, a linear probe on an early layer of a LM can be considerably more accurate than on its final layer (Figure 2a). This raises a basic question: *how should the representations of a frozen LM be combined across depth and across tokens, when the label is built from local information that is composed and aggregated?*

Previous works have taken important steps toward using several layers and the interactions between tokens. ELMo learns a scalar mixture of layers (Peters et al., 2018), SBERT-WK combines the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

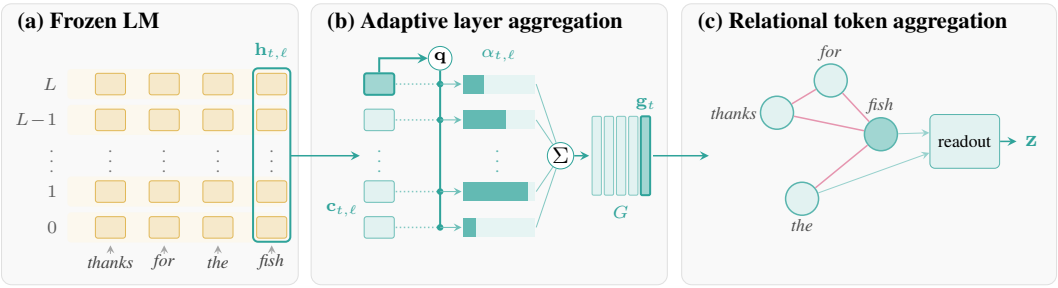


Figure 1: **ToDA combines each token's representations across depth, then aggregates across tokens.** (a) A frozen language model (yellow) produces a representation $\mathbf{h}_{t,\ell}$ of each token t at every accessed layer $\ell \in \mathcal{S}$. (b) The representations are normalized and compressed to $\mathbf{c}_{t,\ell}$; the final-layer state of the token gives the query $\mathbf{q}$, every layer gives a key, and the resulting token-specific weights $\alpha_{t,\ell}$ combine the layers into the fused vector $\mathbf{g}_t$, one column of G. (c) The fused vectors are the nodes of a token graph whose edges (pink) connect similar tokens; a graph neural network updates the tokens and an attention readout combines them into the text vector $\mathbf{z}$, which is passed to the task head. Trainable components are shown in turquoise; weights and edges are illustrative.

representations of each word across the layers of BERT (Wang & Kuo, 2020), GLOT relates tokens through a similarity graph before readout (Mantri et al., 2026), and HistoGraph reads the layer history of graph neural network activations (Galron et al., 2026). These works offer valuable insight into how depth and token interactions can each improve a representation. Nonetheless, the layer mixtures are shared by all tokens, the token graphs are built from a single layer, and the evaluations focus on labels that reflect the overall content of a sentence. It is therefore not yet known how aggregation methods behave when the label requires composition and aggregation, and whether different tokens benefit from different depths. We therefore ask:

When a label is built from local information that is composed and aggregated, how should the representations of a frozen language model be combined across depth and across tokens into one text vector?

Our approach. We identify token-specific access to depth, before the tokens interact, as a desirable property for answering this question. Intuitively, if each token can draw on the layers where its own information is most accessible, then a negated opinion can be read after the negation is composed, a review can be tied to the business named before it, and the relations between tokens can be formed from these fused features instead of the final layer alone. Importantly, a layer mixture shared by all tokens may not be sufficient, because the depth at which this information is most accessible can differ from token to token and from input to input. Guided by this property, we introduce **ToDA** (**To**ken-**D**epth **A**ggregation, Figure 1). For each token, ToDA uses its final-layer representation as a query over its own representations at all accessed layers, and combines them with token-specific weights into one fused vector. The fused vectors then become the nodes of a token graph, a graph neural network relates them, and an attention readout aggregates them into the text vector. To evaluate this design, we build a suite of composition and aggregation tasks from controlled text and from existing datasets. Their labels are built from local judgments, and global cues, such as the overall sentiment of a text or its length, are balanced so that they do not reveal the answer.

Our main contributions are as follows:

- We formulate the evaluation of frozen-LM text representations on composition and aggregation, and introduce seven tasks, from controlled composed polarity to target-specific review aggregation, aspect majority, multi-hop kinship and review counting, whose labels are built from local judgments (Section 4).
- We introduce ToDA, which combines the representations of each token across depth with token-specific attention before relational aggregation over a token graph, with a number of trainable parameters that does not grow with the number of accessed layers (Section 3).

- We conduct an extensive empirical study with multiple frozen encoder and decoder backbones (Section 5). Existing aggregation methods remain close to chance level on target-specific reviews, while ToDA reaches high accuracy; across the proposed tasks, ToDA achieves the highest accuracy in nearly every setting, and token-specific layer weights give its largest gain. ToDA also competitive on the standard GLUE, IMDB and MTEB evaluations.

2 Related Work

Text representations from frozen language models. Sentence encoders read a special token, average the tokens, or learn attention weights over them (Devlin et al., 2019; Reimers & Gurevych, 2019; Brothers, 2025), often with contrastive objectives (Gao et al., 2021). GLOT adds message passing over a similarity graph of the final-layer tokens before readout (Mantri et al., 2026), and parameter-efficient methods instead adapt the backbone through low-rank updates, prompts or adapters (Hu et al., 2021; Lester et al., 2021; Houlsby et al., 2019). These methods are evaluated mainly on GLUE, IMDB and MTEB (Wang et al., 2018; Maas et al., 2011; Muennighoff et al., 2023). In this work, the backbone stays frozen, and we study how its representations should be combined when the label is composed and aggregated from local information.

Reading several layers. Probing studies find that different linguistic properties are most accessible at different depths (Tenney et al., 2019; Jawahar et al., 2019; Liu et al., 2019a). ELMo learns a scalar mixture of layers (Peters et al., 2018), SBERT-WK combines each word's representations across BERT layers without training (Wang & Kuo, 2020), jumping knowledge combines the layers of a graph neural network (Xu et al., 2018), and HistoGraph attends over the layer history of node activations with weights shared across nodes (Galron et al., 2026). Previous works share the layer weights among tokens, while in this work each token computes its own weights, and the fused tokens then determine the token graph.

Composition and aggregation in text. Sentiment composition under negation is a classical test of compositional models (Socher et al., 2013), MAMS pairs sentences with several aspects of different polarities (Jiang et al., 2019), and CLUTRR requires chaining kinship relations (Sinha et al., 2019). These datasets are typically used to evaluate fine-tuned or prompted models. In contrast, we use them, together with constructed labels over Yelp reviews and SST-2 sentences, to evaluate how the representations of a frozen LM are aggregated into one vector.

3 Method

A text representation for composition and aggregation can benefit from letting each token use the depth at which its own information is most accessible, and from letting the tokens interact through these fused features before they are combined. ToDA provides both in two stages, illustrated in Figure 1. First, each token attends over its own representations at all accessed layers, using its final-layer representation as the query. Second, the fused vectors form a graph, a graph neural network (GNN) propagates information between related tokens, and an attention readout combines them. We define the problem in Section 3.1 and describe the two stages in Section 3.2.

3.1 Problem Setup

Given a text x with T tokens, a frozen LM with L layers produces a representation $\mathbf{h}_{t,\ell} \in \mathbb{R}^D$ of every token $t \in \{1, \dots, T\}$ at every layer $\ell \in \{0, \dots, L\}$, where D is the hidden width; layer 0 is the output of the embedding layer, before any transformer layer, and layers 1 to L are the outputs of the transformer layers. We access a set of layers $\mathcal{S} \subseteq \{1, \dots, L\}$ of size $M = |\mathcal{S}|$ that includes the final layer L. An aggregation method Φ_θ with trainable parameters θ maps these representations to one text vector,

$$\mathbf{z} = \Phi_\theta(\{\mathbf{h}_{t,\ell}\}_{t=1,\dots,T;\ \ell \in \mathcal{S}}), \tag{1}$$

which is passed to a task-specific classifier or embedding projection. The LM stays frozen, and we train Φ_θ together with the task-specific head. The existing aggregation methods that we compare in Section 5 use $\mathcal{S} = \{L\}$.

3.2 ToDA

Adaptive layer aggregation. Each token has a sequence of representations across depth, which we combine into one vector per token. Every accessed layer is first compressed by a LayerNorm (LN) and a projection $W_\downarrow \in \mathbb{R}^{r\times D}$, both shared by all layers, $\mathbf{c}_{t,\ell} = W_\downarrow \mathrm{LN}(\mathbf{h}_{t,\ell}) \in \mathbb{R}^r$, where $r < D$ is the latent width. For each of H attention heads, indexed by h, the query comes from the final layer and every accessed layer supplies a key, $\mathbf{q}_t^{(h)} = W_Q^{(h)}(\mathbf{c}_{t,L} + \mathbf{p}_L)$ and $\mathbf{k}_{t,\ell}^{(h)} = W_K^{(h)}(\mathbf{c}_{t,\ell} + \mathbf{p}_\ell)$, with $W_Q^{(h)}, W_K^{(h)} \in \mathbb{R}^{d_k\times r}$ for a query and key width d_k. Here $\mathbf{p}_\ell \in \mathbb{R}^r$ is a fixed sinusoidal encoding (Vaswani et al., 2017) of the position of layer ℓ among the accessed layers, so that each key also carries the depth of its layer. The value $\mathbf{v}_{t,\ell}^{(h)} \in \mathbb{R}^{r/H}$ is the h-th of H equal slices of $\mathbf{c}_{t,\ell}$, without the encoding. The scores, the layer weights and the output of head h are

$$\gamma_{t,\ell}^{(h)} = \frac{\mathbf{q}_t^{(h)\top}\mathbf{k}_{t,\ell}^{(h)}}{\sqrt{d_k}}, \qquad \boldsymbol{\alpha}_t^{(h)} = \mathcal{N}\big(\boldsymbol{\gamma}_t^{(h)}; \tau_s\big) \in \mathbb{R}^M, \qquad \mathbf{o}_t^{(h)} = \sum_{\ell\in\mathcal{S}} \alpha_{t,\ell}^{(h)}\, \mathbf{v}_{t,\ell}^{(h)}, \tag{2}$$

where $\boldsymbol{\gamma}_t^{(h)} \in \mathbb{R}^M$ collects the scores of token t over the accessed layers and $\mathcal{N}$ normalizes them into weights with a learned temperature $\tau_s > 0$. We used $\mathcal{N}(\boldsymbol{\gamma}; \tau_s) = \mathrm{softmax}(\boldsymbol{\gamma}/\tau_s)$, additional details are given in Appendix G). The query therefore determines how much each layer contributes, and the values supply the features that are combined. The head outputs are concatenated ($\|$) and projected back to width D, giving $\tilde{\mathbf{g}}_t = W_\uparrow[\mathbf{o}_t^{(1)}\| \cdots \|\mathbf{o}_t^{(H)}] \in \mathbb{R}^D$ with $W_\uparrow \in \mathbb{R}^{D\times r}$. An optional residual gate additionally mixes in the weighted uncompressed representations $\mathbf{b}_t = \sum_{\ell\in\mathcal{S}} \bar{\alpha}_{t,\ell}\mathbf{h}_{t,\ell}$, where $\bar{\alpha}_{t,\ell} = \frac{1}{H}\sum_h \alpha_{t,\ell}^{(h)}$:

$$\mathbf{g}_t = \boldsymbol{\sigma}_t \odot \tilde{\mathbf{g}}_t + (\mathbf{1} - \boldsymbol{\sigma}_t) \odot \mathbf{b}_t, \qquad \boldsymbol{\sigma}_t = \mathrm{sigmoid}\big(W_g\tilde{\mathbf{g}}_t\big), \tag{3}$$

with $W_g \in \mathbb{R}^{D\times D}$, $\odot$ the elementwise product and $\mathbf{1}$ the all-ones vector; without the gate, $\mathbf{g}_t = \tilde{\mathbf{g}}_t$. Bias terms are omitted for brevity. Because each query is computed from the token's own final-layer representation, the weights $\alpha_{t,\ell}^{(h)}$ can vary from token to token and from input to input. In contrast, the layer-mix baselines in Section 5 combine the layers with weights shared by all tokens and inputs.

Relational token aggregation. The fused vectors form the matrix $G = [\mathbf{g}_1, \ldots, \mathbf{g}_T]$, whose columns are the node features of a token graph. Its adjacency matrix $A(G) \in \{0,1\}^{T\times T}$ connects tokens $i \neq j$ whenever the cosine similarity of their edge features exceeds a threshold τ. The edge features, of width d_e, are either the fused vectors themselves ($d_e = D$) or a fixed random Gaussian projection of them ($d_e = 64$) (Johnson et al., 1984), which is used only to build the edges. A graph attention network (Veličković et al., 2017) with J layers and jumping knowledge (Xu et al., 2018) updates the tokens, and a learned attention readout combines them into the text vector:

$$\mathbf{z} = \mathrm{Readout}\big(\mathrm{GNN}\big(G, A(G)\big)\big)\,. \tag{4}$$

The readout computes $\mathbf{z} = \sum_{t=1}^T \beta_t\, \hat{\mathbf{g}}_t$, where $\hat{\mathbf{g}}_t$ is the output of the GNN for token t and $\beta_t \propto \exp\big(\mathbf{w}_\beta^\top \tanh(W_\beta\hat{\mathbf{g}}_t)\big)$ with learned parameters W_β and $\mathbf{w}_\beta$; the sigmoid variant uses $\beta_t = \mathrm{sigmoid}\big(\mathbf{w}_\beta^\top \tanh(W_\beta\hat{\mathbf{g}}_t)\big)$ without normalization. Because the edges are computed from the fused vectors, these vectors determine both the token features and the relations between tokens, so the depth that each token draws on also shapes the graph: two tokens can be connected in the fused space even when their final-layer representations are not similar.

Computational complexity. At fixed D, r, H and d_k, the number of trainable parameters of ToDA does not grow with the number of accessed layers M, because the compression, query, key and output projections are shared across layers. The computation grows linearly with M: the compression costs $O(TMDr)$, the keys cost $O(TMHd_kr)$, the representations of the accessed layers occupy $O(TMD)$ memory, and the gate adds $O(TMD + TD^2)$. Building the token graph costs $O(T^2d_e)$, as for the token-graph baselines. Appendix D reports the parameter counts of every method (Table 22).

4 Composition and Aggregation Tasks

To study the question in Section 1, we build tasks in which the label is determined by local judgments that are first formed and then combined. We construct them from controlled text and from

Table 1: **The composition and aggregation tasks.** u and K give the number of units in one input: opinion units, sentences, reviews per business for target-specific reviews, and the maximum number of reviews for counting.

Task	Operation	Input	Label	Settings	App.
Composed polarity	compose, aggregate	negated opinion units	majority	$u \in \{3, 5, 9\}$	B.1
SST-2 paragraphs	aggregate	SST-2 sentences	majority	$K \in \{3, 5, 9\}$	B.2
Target-specific reviews	select, aggregate	reviews, two businesses	query majority	$K \in \{3, 5, 7\}$	B.3
Aspect majority	aggregate	MAMS sentences	majority	$K \in \{1, 3, 5\}$	B.4
Aspect sentiment	select	aspect, MAMS sentence	3 polarities	one sentence	B.4
Kinship	compose	CLUTRR story	18 relations	2–4 hops	B.5
Review counting	aggregate	Yelp reviews	positive count	$K \in \{5, 9\}$	B.6

existing datasets and summarize them in Table 1; Appendix B gives each construction with a worked example.

Labels built from local judgments. Most of the tasks combine N text units, each with a polarity $s_i \in \{-1, 0, +1\}$ and relevance $\rho_i \in \{0, 1\}$, into the binary label

$$y = \mathbf{1}\left[\sum_{i=1}^{N} \rho_i \, s_i > 0\right], \tag{5}$$

where inputs with a zero sum are excluded. In *composed polarity*, each unit is a sentiment word that may be preceded by a negator, so $s_i \in \{-1, +1\}$ is the polarity after negation and $\rho_i = 1$. In *target-specific reviews*, $2K$ Yelp reviews about two businesses, each preceded by the name of its business, are shuffled after a query naming one of them; $s_i \in \{-1, +1\}$ is the polarity of review i, and $\rho_i = 1$ for the K reviews of the queried business and $\rho_i = 0$ otherwise. *Aspect majority* sums the polarities of all annotated aspects in K MAMS sentences, with $s_i = 0$ for neutral aspects and $\rho_i = 1$, and *SST-2 paragraphs* take the majority over K SST-2 sentences. Three tasks use other labels: *review counting* predicts how many of at most K reviews are positive, *aspect sentiment* predicts the polarity of a queried aspect in a MAMS sentence, and *kinship* predicts one of 18 relations between two people in a CLUTRR story by chaining two to four stated relations. The polarities and relevance indicators define y and are not given to the model.

Global cues are balanced. Each construction removes a shortcut that would otherwise reveal the label. In target-specific reviews, exactly K reviews are positive and K are negative, and the other business has the opposite majority, so the overall sentiment of the text is balanced and the answer depends on associating reviews with businesses. In composed polarity, each unit is negated with probability 0.5, so counting positive words does not give the majority. In review counting, the number of reviews varies, so the fraction of positive reviews does not determine the count.

The proposed tasks leave a large gap for aggregation methods. Before training any method, we check how much of each task the simplest aggregation already solves. We average the frozen token representations, fit a linear probe at every layer, and compare the most accurate layer with a reference accuracy that is known to be reachable: a classifier applied to each review separately, followed by the label rule, for the review tasks; the annotated polarities themselves (100%) for aspect majority; and the fine-tuned BERT of Jiang et al. (2019) for aspect sentiment. A small gap means that averaging already keeps what the label needs, so the task says little about aggregation. This holds for plain majority over Yelp reviews without a queried business, where the gap is at most 5.90 points on all six backbones and for every number of reviews, so we do not use this task. In contrast, the gap is 22.20 to 40.70 points on target-specific reviews and 14.40 to 31.02 points on the aspect tasks (Table 14 in Appendix C).

5 Experiments

ToDA gives each token its own access to depth and then lets the fused tokens interact through a graph. The natural alternatives either keep the final layer or combine the layers with weights shared by all tokens. We compare these designs on the tasks of Section 4, add the components of ToDA one at a time, and finally evaluate ToDA on the standard benchmarks for text representations. Specifically, our experiments address the following questions:

Table 2: **ToDA achieves the highest accuracy on controlled composition and aggregation on every backbone.** Accuracy (%) ↑ on composed-polarity majority with u opinion units, mean over 16 seeds; results for $u = 5$ are in Table 15. The highest result in each column is shown in **bold** and the second highest is underlined.

Backbone →	BERT		RoBERTa		SmolLM2		TinyLlama		LLaMA-3B		Mistral-7B	
Method ↓ / u →	3	9	3	9	3	9	3	9	3	9	3	9
EXISTING AGGREGATION METHODS												
Mean	64.50	61.34	61.03	60.56	58.00	54.41	72.97	65.09	69.56	61.53	76.97	66.63
Max	64.38	57.25	60.72	60.13	56.00	53.28	70.72	59.38	71.38	62.41	75.81	64.44
[CLS]/last token	66.91	60.94	53.69	53.97	60.72	56.72	72.16	63.47	78.53	66.78	79.22	68.38
AdaPool	73.28	69.78	70.38	68.97	65.16	59.75	76.78	66.22	77.97	69.91	83.84	72.16
GLOT	82.38	76.69	79.88	75.66	74.53	66.38	82.81	71.06	81.81	65.56	88.16	70.69
NATURAL BASELINES												
Layer mix + AdaPool	73.78	67.53	76.78	67.72	61.50	58.50	72.13	64.81	79.44	71.50	80.44	70.56
ToDA (Ours)	**94.28**	**85.97**	**96.38**	**86.81**	**87.28**	**68.19**	**99.31**	**94.91**	**98.94**	**91.88**	**98.88**	**92.69**

(RQ1) How do existing aggregation methods, natural multilayer baselines and ToDA compare when the label requires composition and aggregation?

(RQ2) How do the components of ToDA contribute to its accuracy, and how does it weight the layers?

(RQ3) How does ToDA perform on standard text classification and sentence-embedding benchmarks, and at what parameter cost?

5.1 EXPERIMENTAL SETUP

Backbones. We use two encoders, BERT-base (Devlin et al., 2019) and RoBERTa-base (Liu et al., 2019b), and four decoders of increasing size, SmolLM2-135M (Allal et al., 2025), TinyLlama-1.1B (Zhang et al., 2024), LLaMA-3.2-3B (Grattafiori et al., 2024) and Mistral-7B (Jiang et al., 2023). The controlled task uses all six backbones, the natural-text tasks use BERT and the three larger decoders, SST-2 paragraphs use these three decoders, and the standard benchmarks use BERT, TinyLlama and Mistral-7B. The backbones stay frozen, and we train the aggregation method together with its task head.

Compared methods. *Existing aggregation methods* read the final layer: mean and max aggregation, the [CLS] token of encoders or the last token of decoders ([CLS]/last token), the learned attention readout AdaPool (Brothers, 2025), and GLOT (Mantri et al., 2026). *Natural baselines* combine these methods with access to several layers, and test whether that access alone accounts for the improvements of ToDA. *Layer mix* combines the layers with learned weights shared by all tokens and inputs (Peters et al., 2018), followed by mean aggregation (Layer mix + mean) or AdaPool (Layer mix + AdaPool). All relational methods share the same graph network, with four graph attention layers of width 256.

Training and evaluation. On the natural-text tasks, each result is the mean test accuracy over eight data seeds, taken at the epoch with the highest validation accuracy, and each method selects its training configuration among a small set of candidates by mean validation accuracy. Differences between methods come with paired bootstrap intervals over the seeds. The controlled task has no test split, so every method reports its highest validation accuracy, averaged over 16 seeds. Appendix C gives the search spaces, training constants and selection procedure, and Appendix A gives an overview of all supplementary results.

5.2 COMPOSITION AND AGGREGATION

We first address RQ1 on the controlled task, where negation and majority are known exactly, and then on the natural-text tasks.

Table 3: **ToDA achieves the highest accuracy in every natural-text setting.** Test accuracy (%) ↑, averaged over the settings of each task: $K \in \{3, 5, 7\}$ reviews per business, $K \in \{1, 3, 5\}$ sentences, 2 to 4 hops, and at most $K \in \{5, 9\}$ reviews to count; each setting is a mean over eight data seeds (Appendix D). The highest result in each column is shown in **bold** and the second highest is underlined.

Task →	target-specific reviews			Aspect majority			Kinship			Review counting		
Method ↓ / Backbone →	Tiny	3B	7B	Tiny	3B	7B	Tiny	3B	7B	Tiny	3B	7B
EXISTING AGGREGATION METHODS												
Mean	50.94	50.35	52.97	80.49	80.54	80.10	48.98	51.29	63.67	37.46	38.63	42.19
Max	50.05	49.36	50.53	75.90	77.13	78.04	33.97	47.15	50.76	28.24	31.89	31.04
[CLS]/last token	50.18	49.95	51.45	76.85	77.52	75.35	42.40	45.75	50.71	32.35	34.09	34.79
AdaPool	50.29	49.96	50.99	79.50	80.43	78.91	78.13	83.19	91.13	41.94	46.31	38.91
GLOT	71.25	53.42	51.74	80.64	80.90	79.96	88.09	88.05	92.76	41.61	44.56	43.31
NATURAL BASELINES												
Layer mix + AdaPool	59.51	73.96	81.57	78.96	79.80	80.07	75.24	53.68	82.45	46.37	47.78	47.66
ToDA (Ours)	**88.28**	**92.39**	**93.94**	**82.37**	**82.98**	**82.91**	**93.74**	**95.95**	**97.38**	**51.63**	**54.21**	**54.65**

Controlled composition. ToDA achieves the highest accuracy on every backbone and for every number of opinion units (Table 2; all settings in Table 15). On the three larger decoders, its advantage over GLOT grows with the number of units and reaches 22.00 to 26.31 points at $u = 9$, where ToDA attains 91.88 to 94.91%. Against the methods without a token graph (max aggregation, [CLS]/last token and AdaPool), all 54 paired intervals are above zero, with improvements of 8.44 to 42.69 points, and ToDA improves over Layer mix + AdaPool by 9.69 to 30.09 points. Combining layers with shared weights yields accuracy close to final-layer AdaPool. We attribute the gain of ToDA to the composition itself: a negated opinion is resolved at the token where it occurs, which favors depth chosen per token over a single mixture for the whole text.

Natural text. The same trend holds on natural text (Table 3), where ToDA achieves the highest accuracy in all twelve task and backbone pairs. Target-specific reviews show the largest differences between methods. On LLaMA-3B and Mistral-7B, the existing aggregation methods remain close to chance level, as expected when the overall sentiment of each input is balanced, whereas ToDA reaches 92.04 to 94.13% in every setting (Table 17). ToDA improves over Layer mix + AdaPool by 10.59 to 31.03 points, with positive paired intervals in all nine decoder settings. On kinship, ToDA reaches 92.67 to 99.01% and improves over GLOT by 2.26 to 13.21 points in all twelve settings, and on review counting it achieves the highest accuracy in all eight settings. On aspect majority, where each aspect states its polarity locally, ToDA improves over every existing aggregation method and over Layer mix + AdaPool. Aspect sentiment (Table 19) and SST-2 paragraphs (Table 16) show the same trend, and Appendix D reports every setting and backbone, including BERT.

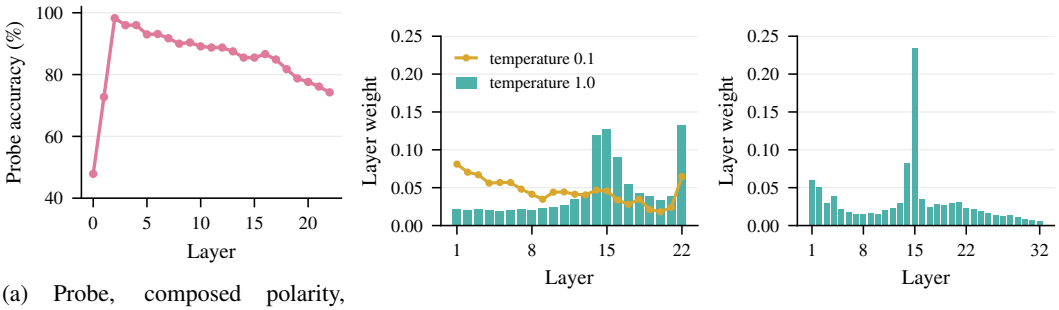
5.3 COMPONENTS AND LEARNED LAYER WEIGHTS

To address RQ2, we add the components of ToDA one at a time on an encoder and a decoder (Table 4), including Non-adaptive ToDA, which applies GLOT to the layer mixture, and we inspect the layer weights that ToDA learns (Figure 2).

Each component adds accuracy, and token-specific layer weights add the most. ToDA achieves the in total highest accuracy in Table 4. Access to all layers with shared weights is most helpful on target-specific reviews, where Layer mix + AdaPool reaches 83.54% on Mistral-7B compared with 50.04% for AdaPool on the final layer, and adding the token graph to the layer mixture further improves aspect majority and target-specific reviews. Replacing the shared layer mixture by token-specific weights then gives the largest and most consistent gain. Over all natural-text settings, ToDA improves over Non-adaptive ToDA with positive paired intervals in all nine target-specific review settings of the three larger decoders and on five of the six backbones for aspect sentiment, and it performs on par with Non-adaptive ToDA on aspect majority (Table 23 in Appendix E). On the

Table 4: **Each component of ToDA contributes, and token-specific layer weights give the largest gain.** Accuracy (%) ↑ on an encoder (BERT) and a decoder (Mistral-7B), at the largest setting of each task: composed polarity with $u = 9$ units (16 seeds), aspect sentiment, aspect majority with $K = 5$ sentences and target-specific reviews with $K = 7$ reviews per business (eight seeds). The columns on the left mark access to all layers, token-specific layer weights instead of weights shared by all tokens, relational aggregation over a token graph, and an attention readout instead of the mean. A dash marks a configuration that was not trained on that task. The highest result in each column is shown in **bold** and the second highest is underlined.

					Polarity		Aspect sent.		Aspect maj.		Reviews	
Method	All layers	Per token	Graph	Attn. readout	BERT	7B	BERT	7B	BERT	7B	BERT	7B
Mean	–	–	–	–	61.34	66.63	56.49	64.98	76.58	78.08	50.43	52.56
AdaPool	–	–	–	✓	69.78	72.16	72.22	70.11	74.61	76.05	50.50	50.04
GLOT	–	–	✓	✓	76.69	70.69	72.79	69.66	78.11	77.50	50.06	53.59
Layer mix + mean	✓	–	–	–	54.81	66.13	57.82	65.70	76.08	79.25	49.78	51.85
Layer mix + AdaPool	✓	–	–	✓	67.53	70.56	75.51	76.20	73.68	78.38	**50.98**	83.54
Non-adaptive ToDA	✓	–	✓	✓	74.66	76.16	74.48	77.73	77.28	80.83	49.99	90.09
ToDA (Ours)	✓	✓	✓	✓	**85.97**	**92.69**	**77.49**	**79.88**	**79.83**	**80.89**	49.18	**94.13**



(a) Probe, composed polarity, TinyLlama

(b) ToDA weights, TinyLlama

(c) ToDA weights, Mistral-7B

Figure 2: **Task information is most accessible at intermediate layers, and ToDA learns to weight them.** (a) Accuracy of a linear probe on mean-aggregated TinyLlama representations for composed polarity with $u = 9$ units, by layer. (b, c) Layer weights $\bar{\alpha}_{t,\ell}$ of ToDA on target-specific reviews with $K = 3$, averaged over tokens and seeds. In (b), the line shows the weights obtained with the initial layer-weighting temperature of 0.1.

controlled task, where negation is resolved per unit, ToDA improves over Layer mix + AdaPool by 18.44 and 22.13 points.

ToDA concentrates its weights where the task information is accessible. On composed polarity, the probe on TinyLlama peaks at layer 2 and declines toward the output (Figure 2a), in agreement with the gains of ToDA over final-layer methods. On target-specific reviews, the weights learned by ToDA on Mistral-7B peak at layer 15, with about seven times the uniform weight, where a linear probe separates the classes almost as accurately as at its best layer, layer 14 (68.40% and 68.50%; Figure 2c). On TinyLlama, the weights concentrate on layers 14 and 15, next to the probe peak at layer 16, and on the final layer, which also supplies the query (Figure 2b). With the initial layer-weighting temperature, the weights stay close to uniform and accuracy is 65.65% instead of 87.58% in the same runs; the higher temperature is selected on validation data in every decoder setting of target-specific reviews. Therefore, the accuracy of ToDA comes with a layer preference that matches where the relevant information is accessible (Appendix E).

Table 5: **ToDA also improves standard classification and sentence-embedding benchmarks.** Top: average score ↑ over nine GLUE tasks and IMDB (0 to 100); baseline scores are those reported by Mantri et al. (2026). Bottom: average MTEB score ↑ ($\times 100$) over 44 evaluation categories after training on MS MARCO, with every method trained and selected under the same protocol. Per-task scores are in Appendix F. The highest result in each column is shown in **bold** and the second highest is underlined.

Method ↓ / Backbone →	BERT	TinyLlama	Mistral-7B	Average
GLUE AND IMDB				
Mean	57.24	54.36	63.20	58.26
Max	53.63	52.37	55.11	53.70
[CLS]/last token	54.17	54.37	60.72	56.42
AdaPool	60.38	53.52	65.70	59.87
GLOT	68.29	62.87	70.02	67.06
ToDA (Ours)	**70.93**	**71.33**	**74.00**	**72.09**
MTEB				
Mean	42.27	46.68	48.53	45.83
Max	39.54	43.22	43.25	42.00
[CLS]/last token	37.80	36.01	39.48	37.76
GLOT	43.36	45.10	46.60	45.02
ToDA (Ours)	**44.65**	**48.30**	**48.61**	**47.19**

5.4 Standard Benchmarks and Parameter Cost

To address RQ3, we evaluate ToDA with the protocols used by GLOT: supervised GLUE tasks and IMDB, and sentence-embedding transfer to MTEB after training on MS MARCO (Bajaj et al., 2016).

ToDA also improves standard benchmarks. Averaged over nine GLUE tasks and IMDB, ToDA improves over the scores reported for GLOT on all three backbones, by 2.63 to 8.46 points, and it is higher in 28 of the 30 task and backbone pairs and equal in one (Table 5), with the largest gains on CoLA, MNLI and SST-2 (Tables 24 and 25). On MTEB, where every method is trained and selected with the same protocol, ToDA achieves the highest average score over 44 evaluation categories on all three backbones. Its improvements are largest on retrieval, semantic similarity and reranking, while mean aggregation remains competitive on classification and clustering (Table 26; protocol in Appendix F.2).

Parameter cost. With Mistral-7B on CoLA, ToDA reaches 66.00 MCC with 19.19M trainable parameters and a frozen backbone. For reference, Mantri et al. (2026) report 53.29 MCC for GLOT with 8.92M parameters, 48.23 for LoRA with 167.8M parameters and 49.63 for full fine-tuning of all 7.11B parameters (Table 29). Measured with their protocol, a training batch of ToDA takes 61.2 ms and 4.15 GB of GPU memory (13.4 ms and 0.42 GB for GLOT), as it reads all 32 layers.

6 Conclusion

We asked how the representations of a frozen language model should be combined when the label is built from local information that is composed and aggregated. In the tasks we study, giving each token its own access to depth before the tokens interact is highly effective. Existing aggregation methods remain close to chance level on target-specific reviews, layer mixtures shared by all tokens recover part of the gap, and ToDA, which weights the layers per token before relational aggregation, achieves the highest accuracy in nearly every task and backbone while also improving GLUE, IMDB and MTEB. The proposed tasks also point to promising directions: counting reviews leaves room for further improvement relative to a per-review reference, and on aspect majority, where each unit states its polarity locally, several multilayer designs perform comparably. We hope that these tasks make it easier to study which operations a text representation performs, and that ToDA provides a strong basis for aggregation methods that compose and count over longer texts.

Reproducibility Statement

Section 5.1 and Appendix C describe the backbones, methods, training configurations, search spaces, model selection and uncertainty estimates. Appendix B specifies the construction of every task with a worked example, and Appendix G gives the configuration of ToDA used in each study. We will publish the full code, including the data construction and the scripts that produce every table and figure, upon acceptance.

AI Use Statement

In the preparation of this paper, generative AI tools were utilized exclusively to support language editing, manuscript formatting, and literature surveying. We emphasize that all core scientific contributions, including experimental designs, mathematical formulations, methodological choices, and data interpretations, are entirely the authors' own work. All AI-assisted text was rigorously reviewed for correctness, and the authors bear complete responsibility for all claims, artifacts, and final content presented in this paper.

Ethics Statement

This work uses publicly available datasets (GLUE, IMDB, MTEB, MS MARCO, SST-2, Yelp reviews, MAMS and CLUTRR) and pretrained language models under their respective licenses. The business names in the target-specific review task are fictitious and assigned at random. We do not foresee specific ethical risks beyond those of the underlying models and datasets.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Greyson Brothers. Robust noise attenuation via adaptive pooling of transformer outputs. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 670–680, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Yaniv Galron, Hadar Sinai, Haggai Maron, and Moshe Eliasof. Learning from historical activations in graph neural networks. *arXiv preprint arXiv:2601.01123*, 2026.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6894–6910, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 3651–3657, 2019.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 6280–6285, 2019.

William B Johnson, Joram Lindenstrauss, et al. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

Abhinav Ramesh Kashyap, Thanh-Tung Nguyen, Viktor Schlegel, Stefan Winkler, See Kiong Ng, and Soujanya Poria. A comprehensive survey of sentence representations: From the bert epoch to the chatgpt era and beyond. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1738–1751, 2024.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 1073–1094, 2019a.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Krishna Sri Ipsit Mantri, Carola-Bibiane Schönlieb, Zorah Lähner, and Moshe Eliasof. Towards improved sentence representations using token graphs. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2603.03389.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL*, pp. 2227–2237, 2018.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4506–4515, 2019.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4593–4601, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Bin Wang and C-C Jay Kuo. Sbert-wk: A sentence embedding method by dissecting bert-based word models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2146–2157, 2020.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pp. 5453–5462. pmlr, 2018.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

A Organization of the Appendix

Table 6 lists the content of each appendix and the part of the main text it supports.

Table 6: **Organization of the appendix.**

Appendix	Content	Supports
B	Construction of every task, with worked examples	Section 4
C	Training, model selection, uncertainty and the probe comparison	Sections 4 and 5.1
D	Full results for every task, setting and backbone; parameter counts	RQ1, Tables 2 and 3
E	Component comparisons over all settings; layer weights	RQ2, Table 4, Figure 2
F	GLUE, IMDB and MTEB per task; CoLA parameter comparison	RQ3, Table 5
G	Layer-weight normalization and configurations of ToDA	Section 3

B Composition and Aggregation Tasks

This appendix supports Section 4. In every task, the language model is frozen and supplies a representation of each token at each accessed layer; a trainable aggregation method combines them into one vector, and a classifier predicts the label. The aggregation methods do not receive the unit labels or target annotations used to construct the answer. Each subsection below gives one worked example of its task (Tables 7, 8, 9, 10, 11, 12 and 13). Pale blue rows contain the model input and pale green rows the expected answer; colors and underlining inside the input explain the example and are not given to the model.

B.1 Composed-polarity majority

Each input contains $u \in \{3, 5, 9\}$ opinion units. A unit samples a positive or negative sentiment word with equal probability, giving a base polarity $p_i \in \{-1, +1\}$, and is preceded by "not" or "never" with probability 0.5. With $n_i \in \{0, 1\}$ indicating the negator, the composed polarity is $s_i = (-1)^{n_i} p_i$, and the label follows Eq. 5 with $N = u$ and $\rho_i = 1$; an odd u excludes ties. The input consists of the units alone, without filler words or markers between units. Each run trains for five epochs on 1,000 generated examples and is evaluated on 200 generated validation examples after every epoch. Table 7 shows an example with $u = 3$.

Table 7: **Composed-polarity majority ($u = 3$).** Example following the generator; the input consists of the units alone. Colors, underlining and explanations are explanatory annotations; the model receives plain text.

Composed-polarity majority ($u = 3$).	
Model input	**not** good bad **never** awful
Answer	**Negative (0)**
Color key	Red: negative; green: positive. Colors show polarity after negation; underlined words reverse the following sentiment word.
Why	"not good" is negative, "bad" is negative, and "never awful" is positive. Two negative units outvote one positive unit. The input contains no unit-boundary markers.

B.2 Majority over SST-2 sentences

We deduplicate the labeled sentences of the official SST-2 training split (Socher et al., 2013) after whitespace normalization and case folding, and partition them once into 80/10/10 train, validation and test sentence banks. Each input joins $K \in \{3, 5, 9\}$ distinct sentences from one bank; the source labels $y_i \in \{0, 1\}$ define $s_i = 2y_i - 1$ with $\rho_i = 1$, and rejection sampling balances the two classes. A sentence can appear in several inputs of the same split but does not cross splits, and sequence length alone does not predict the label (area under the curve at most 0.55). This task aggregates the source annotations and is distinct from standard single-sentence SST-2; Table 8 shows an example with $K = 3$.

Table 8: **SST-2 paragraphs ($K = 3$).** Example built from the saved SST-2 source dataset. Line breaks separate texts for readability; the model receives their concatenation. Colors, underlining and explanations are explanatory annotations; the model receives plain text.

SST-2 paragraphs ($K = 3$).	
Model input	this too-long , spoofy update of shakespeare 's macbeth does n't sustain a high enough level of invention . labute continues to improve . a fascinating case study of flower-power liberation – and the price that was paid for it .
Answer	**Positive (1)**
Color key	Red: negative; green: positive.
Why	The three source texts have labels negative, positive and positive, respectively. Each text gets one vote, so the majority is positive, regardless of text length.

B.3 TARGET-SPECIFIC REVIEWS

The source bank contains 27,586 deduplicated Yelp reviews (Zhang et al., 2015) of 5 to 20 words: 9,666 negative (one or two stars) and 17,920 positive (four or five stars); three-star reviews are excluded. A fixed partition assigns the reviews to disjoint train, validation and test banks in an 80/10/10 ratio. For each example, two fictitious business names are drawn from a common pool. We sample K positive and K negative reviews, assign a majority of the requested polarity to the queried business and the rest to the distractor, prefix each review with its business name, and shuffle all $2K$ reviews after the query "Reviews of [target]." Writing each review as (e_i, s_i) with business e_i and polarity $s_i \in \{-1, +1\}$, the label is

$$y = \mathbf{1}\left[\sum_{i=1}^{2K} \mathbf{1}[e_i = e^\star]\, s_i > 0\right], \qquad \sum_{i=1}^{2K} \mathbf{1}[e_i = e^\star] = K, \qquad \sum_{i=1}^{2K} s_i = 0, \tag{6}$$

where $e^\star$ is the queried business, so that $\rho_i = \mathbf{1}[e_i = e^\star]$ in Eq. 5. For a positive example, the number of positive reviews of the queried business is drawn uniformly from $\{(K+1)/2, \ldots, K\}$, and for a negative example from $\{0, \ldots, (K-1)/2\}$; the distractor receives the complementary reviews. We use $K \in \{3, 5, 7\}$; at $K = 9$, the 99th percentile of the input length on TinyLlama exceeds 512 tokens. Table 9 shows an example with $K = 3$.

Table 9: **Target-specific review aggregation ($K = 3$).** Example from the saved Yelp construction. Business names are assigned by the task; source spelling is retained. Line breaks separate reviews for readability; the model receives their concatenation. Colors, underlining and explanations are explanatory annotations; the model receives plain text.

Target-specific review aggregation ($K = 3$).	
Model input	Reviews of **Maple House**. Blue Heron: Everything is bad but the $2 Miller lites **Maple House**: Rude Representative on the phone!!! I can't believe that she hang up on me!! **Maple House**: Awsome good, the bread and herbs rock, and I love the steak marsala. Great service! Blue Heron: Very dishonest people and terrible workmanship would never take my vehicle here! **Maple House**: Nothing negative about this place! Blue Heron: I love this neighborhood gem!
Answer	**Positive (1)**
Color key	Red: negative; green: positive. Underlined business names identify the query and the reviews that count.
Why	The Maple House reviews count: two are positive and one is negative. Blue Heron has the opposite majority. Counting all reviews would give a tie, so the model has to associate each review with the queried business.

B.4 ASPECT MAJORITY AND ASPECT SENTIMENT

Aspect majority. We join $K \in \{1, 3, 5\}$ distinct MAMS sentences (Jiang et al., 2019) and predict whether their annotated aspects are mostly positive or mostly negative. With $a_{ij} \in \{-1, 0, +1\}$ the polarity of aspect j in sentence i, the label is $y = \mathbf{1}[\sum_i \sum_j a_{ij} > 0]$ and ties are excluded. Neutral aspects stay in the text but do not vote, and every aspect votes separately, so even $K = 1$ can require combining several opinions. Examples are balanced between the two classes and drawn from the official sentence splits (Table 10).

Table 10: **MAMS aspect majority ($K = 3$).** Example from the saved MAMS construction. Line breaks separate sentences for readability. Colors, underlining and explanations are explanatory annotations; the model receives plain text.

MAMS aspect majority ($K = 3$).	
Model input	The **atmosphere**'s a bit grungy, yes, but the **music** and the **coffee** are good. For the **main course**, try out the **Beef Negimaki**. Tried it again for **brunch**, when the **service** was worse.
Answer	**Positive (1)**
Color key	Red: negative; green: positive. Gray: neutral (no vote). Each bold aspect is one unit.
Why	Music, coffee and Beef Negimaki are positive; atmosphere and service are negative. Main course and brunch are neutral and do not vote. Three positive aspects outvote two negative aspects; sentences do not receive one vote each.

Aspect sentiment. The model predicts the positive, negative or neutral polarity of one queried aspect, which is placed before the original sentence. We use all examples of the official aspect-term splits (Table 11).

Table 11: **MAMS aspect-term sentiment.** Source sentence and aspect annotation from MAMS. Colors, underlining and explanations are explanatory annotations; the model receives plain text.

MAMS aspect-term sentiment.	
Model input	Aspect: **atmosphere**. The **atmosphere**'s a bit grungy, yes, but the **music** and the **coffee** are good.
Answer	**Negative**
Color key	Red: negative; green: positive. The underlined aspect is the prediction target; other aspects provide context.
Why	The queried aspect is atmosphere, described as grungy. Music and coffee are positive, but they are not the target. Changing the prefix to "Aspect: coffee." changes the answer to positive while leaving the sentence unchanged.

B.5 KINSHIP

We use CLUTRR v1 (Sinha et al., 2019). Given a story and two queried names, the model predicts one of 18 kinship relations, which requires chaining two, three or four stated relations (hops). We place the two names before the story, combine the training and validation stories of the CLUTRR v1 release whose training stories require two to four hops, and create separate 80/10/10 splits for each hop count: 4,050/506/507 stories for two hops, 3,997/500/499 for three and 4,019/502/503 for four. Models are trained and tested at the same hop count on different stories. Uniform guessing gives an accuracy of $1/18 \approx 5.6\%$, and Table 12 shows an example with two hops.

Table 12: **Kinship classification (two hops).** Source story, query and label from CLUTRR. Colors, underlining and explanations are explanatory annotations; the model receives plain text.

Kinship classification (two hops).	
Model input	**Dale** and **Rosetta**: Tony is husband of Rosetta and son of Dale.
Answer	**Daughter-in-law**
Color key	Red: spouse relationship; green: parent–child relationship. Underlined names form the query. Colors distinguish the two links, not sentiment.
Why	The query asks how Rosetta is related to Dale. Tony is Dale's son, and Rosetta is Tony's wife. Combining these two relationships makes Rosetta Dale's daughter-in-law; this relation is not stated directly.

B.6 Review counting

The input is a sequence of Yelp reviews from the splits of Appendix B.3, and the label is the number of positive reviews. With at most $K \in \{5, 9\}$ reviews, the counts $y \in \{0, \ldots, K\}$ are balanced: for label y, the number of reviews is drawn uniformly from $\{\max(2, y), \ldots, K\}$, and y positive reviews are shuffled with the remaining negative ones. The fraction of positive reviews therefore does not determine the label. The two settings have six and ten classes, so uniform guessing gives 16.7% and 10.0%; Table 13 shows an example.

Table 13: **Counting positive reviews ($K = 5$).** Example built from the saved Yelp source dataset. Here the maximum is five reviews, but the input contains three. Line breaks are for readability. Colors, underlining and explanations are explanatory annotations; the model receives plain text.

Counting positive reviews ($K = 5$).	
Model input	Terrible service. The water tastes like dirt. Good bartenders, good people, good times. EBs is where its at. Freshest fish in the east valley!
Answer	**2**
Color key	Red: negative; green: positive.
Why	The first review is negative; the second and third are positive. The target is the number of positive reviews, so the answer is 2, rather than a majority-sentiment label.

C Training and Evaluation Protocol

This appendix supports Sections 4 and 5.1. Appendix C.1 covers the natural-text tasks and Appendix C.2 the controlled task and SST-2 paragraphs.

C.1 Natural-text tasks

Data. The constructed tasks use 2,000/500/1,000 training, validation and test examples per data seed, drawn from disjoint source banks; aspect sentiment and kinship use their full splits. Data seeds 42 to 49 change the sampled examples of the constructed tasks and the example order of the full-split tasks, while the parameter initialization is fixed. Inputs are truncated at 256, 384 and 512 tokens for target-specific reviews with $K = 3, 5, 7$; at 128, 256 and 320 for aspect majority with $K = 1, 3, 5$; at 128 for aspect sentiment; at 128, 160 and 192 for kinship with two to four hops; and at 192 and 320 for counting with $K = 5, 9$. A length audit finds that at most 0.16% of the inputs of any setting exceed these limits (17 of 10,500 target-specific review inputs with $K = 5$ on TinyLlama); aspect sentiment and kinship with three or four hops have none.

Training. We train the aggregation method and the classifier with Adam, no learning-rate schedule and no weight decay, for 20 epochs (10 for aspect sentiment). The batch size is 16, reduced to 8 on LLaMA-3B and 4 on Mistral-7B. Decoder weights are loaded in float16 and trained with bfloat16 mixed precision; BERT uses float32. Within a training run, all methods on one backbone share the frozen forward pass and the batch order. The multilayer methods access layers 1 to 12 of BERT, 1 to 22 of TinyLlama, 1 to 28 of LLaMA-3B and 1 to 32 of Mistral-7B. Each run uses a single NVIDIA RTX 6000 Pro, NVIDIA RTX 6000, RTX 3090 or RTX 4090 GPU.

Configurations and selection. Every method has an initial configuration: learning rate 2×10^{-4} and, for ToDA, latent width 64, initial layer-weighting temperature 0.1, four heads with query and key width 16, the residual gate and softmax normalization. The relational methods use four graph attention layers of width 256 with max jumping knowledge, and relation thresholds $\{0.5, 0.9\}$. Tuned configurations were searched on target-specific reviews with $K = 3$ on TinyLlama and Mistral-7B with three seeds. ToDA searched 32 configurations (learning rate $\{5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}\}$, gate bias $\{0, -2\}$, latent width $\{64, 256\}$ and two thresholds), and a separate diagnostic added layer-weighting temperature 1.0; Layer mix + AdaPool searched eight configurations (four learning rates and scorer widths $\{128, 256\}$). Every other method has a tuned candidate with learning rate 5×10^{-5}. BERT and LLaMA-3B reuse the configurations selected on Mistral-7B, and kinship and counting reuse the tuned configurations without a new search. Within each run, we keep the epoch with the highest validation accuracy; among the candidates of a method, we select the one with the highest mean validation accuracy over the eight seeds, and report its mean test accuracy. A candidate is eligible when all eight seeds are complete. The search spaces differ across methods, and the tuned configurations also benefit the baselines, for example raising Layer mix + AdaPool by 26.06 points on Mistral-7B at $K = 3$.

Probe comparison. Before training, a logistic-regression classifier (L2 regularization, $C = 1$, standardized features) on mean-aggregated representations was fitted at every layer on 2,000 training examples, and its most accurate layer was compared with a task-specific reference: a per-review probe followed by the label rule for the Yelp tasks, the annotation ceiling for aspect majority and the published fine-tuned BERT result of Jiang et al. (2019) for aspect sentiment. A task was used when the reference exceeded the most accurate probe layer by at least 10 points. The probes used for these decisions were scored on 1,000 validation examples (all validation stories for kinship), and Table 14 reports the validation scores.

Uncertainty. Standard deviations are computed over the eight seeds. Paired intervals are 95% percentile intervals of 10,000 bootstrap resamples of the seed-matched differences between two selected methods; they condition on the selected configurations.

C.2 Controlled task and SST-2 paragraphs

On the controlled task, every method uses the same 16 seeds (42 to 57), 1,000 training and 200 validation examples, and five epochs. Each run is scored on validation data after every epoch and keeps its highest-scoring epoch; relational methods additionally select the relation threshold from

Table 14: **The proposed tasks leave a large gap for aggregation methods, whereas plain review majority leaves little room.** Gap in percentage points between a reference accuracy and the accuracy of a linear probe on mean-pooled representations at its most accurate layer, measured on validation examples; each cell gives the smallest and largest gap over the settings of the task (K), and aspect sentiment has a single setting. Plain review majority did not reach the threshold of 10 points in any setting and was not used.

Task ↓ / Backbone →	BERT	RoBERTa	SmolLM2	TinyLlama	LLaMA-3B	Mistral-7B
Plain review majority	1.00 to 3.60	1.40 to 5.90	1.20 to 3.70	2.30 to 5.90	2.90 to 5.30	2.50 to 5.40
Target-specific reviews	36.20 to 38.00	36.20 to 40.70	34.20 to 38.00	34.20 to 36.80	27.50 to 30.00	22.20 to 27.00
Aspect majority	20.00 to 28.50	17.20 to 25.60	16.00 to 26.10	14.50 to 23.00	14.40 to 22.60	14.70 to 22.20
Aspect sentiment	31.02	30.72	28.52	24.92	23.12	20.12

$\{0.3, 0.5, 0.7, 0.9\}$ by the 16-seed mean. No test split is generated, so the reported values are the highest validation accuracies, obtained in the same way for every method; paired intervals resample the 16 seeds with these choices fixed. Majority over SST-2 sentences uses 16 data draws, 30 epochs and 256 tokens, and reports test accuracy at the epoch with the highest validation accuracy. In both studies, every method is trained with Adam at learning rate 2×10^{-4}, without a learning-rate schedule or weight decay. The batch size is 32, except on LLaMA-3B and Mistral-7B: there, ToDA, GLOT and mean aggregation use 64 and 32, respectively, whereas the remaining methods, which were trained on GPUs with less memory, use 16 and 8 on the controlled task and 32 and 16 on SST-2 paragraphs. In both studies, ToDA uses four heads with query and key width 16 and initial layer-weighting temperature 0.1.

D Full Results on the Composition and Aggregation Tasks

This appendix supports RQ1 and complements Tables 2 and 3 with every setting, standard deviations and parameter counts. Methods are named as in Section 5.1. For SST-2 paragraphs (Table 16), we additionally report BERT, RoBERTa and SmolLM2, and for the natural-text tasks (Tables 17 to 21), RoBERTa and SmolLM2, all trained and selected with the same protocol.

Controlled composition. Table 15 reports every backbone and number of opinion units. ToDA improves over GLOT in all 18 settings, with paired intervals above zero in 17 of them; in the remaining one, SmolLM2 at $u = 9$, ToDA reaches 68.19% compared with 66.38% for GLOT.

SST-2 paragraphs. ToDA achieves the highest accuracy in 16 of the 18 settings (Table 16), shared with Layer mix + AdaPool on SmolLM2 at $K = 5$. On TinyLlama, LLaMA-3B and Mistral-7B, it improves over GLOT by 2.10 to 6.66 points with positive paired intervals throughout, and it improves over max aggregation, the last token, AdaPool and Layer mix + AdaPool in all 36 comparisons with positive intervals. On BERT, RoBERTa and SmolLM2, it improves over the same four methods with positive intervals in 33 of 36 comparisons. The two remaining settings are on BERT, where GLOT at $K = 3$ and mean aggregation at $K = 5$ are slightly more accurate, with paired intervals that include zero.

Target-specific reviews. On TinyLlama, LLaMA-3B and Mistral-7B, ToDA achieves the highest accuracy in all nine settings (Table 17). On BERT, RoBERTa and SmolLM2, this task remains challenging for all compared methods, which stay close to chance level (at most 57.41%); we therefore view target-specific reviews as an open challenge for these backbones.

Aspect majority and aspect sentiment. On aspect majority (Table 18), ToDA improves over GLOT with positive intervals in 15 of 18 settings and over Layer mix + AdaPool in 16. On aspect sentiment, ToDA achieves the highest accuracy on all six backbones (Table 19).

Kinship and counting. On kinship, ToDA achieves the highest accuracy in all 18 settings (Table 20), improving over GLOT, which is already strong on this task. On counting, ToDA achieves the highest accuracy in all twelve settings (Table 21). A reference that classifies each review with a probe and then counts reaches 62.50 to 84.20% at $K = 5$ and 47.40 to 75.60% at $K = 9$, which leaves room for future aggregation methods. For ToDA, validation selects a softmax readout in all twelve settings over a sigmoid readout that sums the weighted tokens.

Parameter counts. Table 22 lists the trainable parameters on target-specific reviews. Most of the parameters of ToDA belong to the residual gate, whose projection has D^2 entries.

Table 15: **Composed-polarity majority for every number of opinion units.** Accuracy (%) ↑, mean and standard deviation over 16 seeds; for GLOT and ToDA, at the relation threshold selected by the 16-seed mean. Within each backbone, methods are listed as existing aggregation methods, natural baselines and ToDA; the highest result in each column of a backbone is shown in **bold** and the second highest is underlined.

Backbone	Method ↓ / u →	3	5	9
BERT	Mean	64.50±2.5	63.72±5.0	61.34±3.6
	Max	64.38±3.2	60.41±2.9	57.25±3.0
	[CLS]/last token	66.91±3.4	65.06±3.4	60.94±2.5
	AdaPool	73.28±5.5	71.81±4.0	69.78±3.0
	GLOT	82.38±2.6	79.44±2.8	76.69±2.6
	Layer mix + AdaPool	73.78±4.1	71.66±3.8	67.53±2.8
	ToDA (Ours)	**94.28**±4.0	**93.53**±5.0	**85.97**±3.7
RoBERTa	Mean	61.03±3.8	60.13±4.6	60.56±5.2
	Max	60.72±3.6	60.44±5.1	60.13±4.4
	[CLS]/last token	53.69±5.6	51.94±5.0	53.97±5.7
	AdaPool	70.38±2.9	72.06±2.6	68.97±3.9
	GLOT	79.88±3.5	78.72±3.2	75.66±3.9
	Layer mix + AdaPool	76.78±2.8	71.06±2.6	67.72±4.0
	ToDA (Ours)	**96.38**±3.1	**93.63**±3.5	**86.81**±7.1
SmolLM2	Mean	58.00±2.8	57.25±3.1	54.41±3.4
	Max	56.00±3.2	56.72±2.4	53.28±3.0
	[CLS]/last token	60.72±3.2	57.59±3.2	56.72±3.3
	AdaPool	65.16±3.7	62.38±3.9	59.75±3.5
	GLOT	74.53±3.7	67.25±3.0	66.38±3.3
	Layer mix + AdaPool	61.50±4.5	60.09±3.1	58.50±2.9
	ToDA (Ours)	**87.28**±4.7	**77.34**±8.3	**68.19**±4.2
TinyLlama	Mean	72.97±3.5	69.09±2.4	65.09±3.0
	Max	70.72±4.1	63.88±3.6	59.38±3.6
	[CLS]/last token	72.16±3.8	65.88±3.6	63.47±3.8
	AdaPool	76.78±3.6	70.66±3.4	66.22±2.2
	GLOT	82.81±3.3	75.63±2.5	71.06±3.4
	Layer mix + AdaPool	72.13±3.6	70.03±3.9	64.81±3.3
	ToDA (Ours)	**99.31**±0.7	**98.16**±1.7	**94.91**±3.3
LLaMA-3B	Mean	69.56±2.6	66.50±3.0	61.53±2.8
	Max	71.38±3.4	66.66±2.5	62.41±2.6
	[CLS]/last token	78.53±2.9	71.72±2.7	66.78±2.8
	AdaPool	77.97±3.5	73.31±3.6	69.91±4.8
	GLOT	81.81±3.6	71.59±3.3	65.56±2.9
	Layer mix + AdaPool	79.44±4.4	74.16±3.8	71.50±3.6
	ToDA (Ours)	**98.94**±1.0	**97.66**±1.9	**91.88**±4.6
Mistral-7B	Mean	76.97±2.7	72.72±2.5	66.63±1.7
	Max	75.81±2.9	68.63±3.5	64.44±4.1
	[CLS]/last token	79.22±2.5	71.78±3.4	68.38±3.1
	AdaPool	83.84±3.0	77.81±3.0	72.16±3.5
	GLOT	88.16±2.6	77.13±2.1	70.69±2.3
	Layer mix + AdaPool	80.44±4.0	74.06±2.6	70.56±2.9
	ToDA (Ours)	**98.88**±1.3	**98.03**±1.0	**92.69**±5.0

Table 16: **Majority over K SST-2 sentences.** Test accuracy (%) ↑, mean and standard deviation over 16 data draws; for GLOT and ToDA, at the relation threshold selected on validation data. Within each backbone, methods are listed as existing aggregation methods, natural baselines and ToDA; the highest result in each column of a backbone is shown in **bold** and the second highest is underlined.

Backbone	Method ↓ / K →	3	5	9
BERT	Mean	73.70±1.8	**72.00**±2.5	69.89±2.4
	Max	68.73±2.0	64.98±2.0	59.53±2.3
	[CLS]/last token	72.59±1.8	69.85±2.2	66.54±1.4
	AdaPool	71.04±2.0	68.78±2.8	64.89±2.8
	GLOT	**74.25**±1.8	71.46±2.9	69.95±1.9
	Layer mix + AdaPool	71.46±1.4	67.30±2.0	64.66±1.8
	ToDA (Ours)	73.66±2.4	71.54±2.0	**71.43**±2.0
RoBERTa	Mean	73.55±2.7	73.08±2.0	70.84±2.8
	Max	65.14±2.5	62.70±3.0	59.08±3.0
	[CLS]/last token	70.98±2.4	70.66±1.9	69.34±3.4
	AdaPool	69.43±3.4	68.19±1.9	65.61±2.3
	GLOT	73.41±1.9	71.60±2.3	69.83±2.5
	Layer mix + AdaPool	71.05±2.2	70.30±2.5	67.64±2.6
	ToDA (Ours)	**75.28**±2.0	**75.01**±2.3	**73.08**±2.4
SmolLM2	Mean	70.44±2.6	69.35±2.1	69.03±2.0
	Max	60.95±2.8	58.04±2.8	55.34±1.8
	[CLS]/last token	68.46±2.0	66.24±2.5	63.19±2.7
	AdaPool	66.73±2.5	63.73±2.9	62.14±2.7
	GLOT	69.31±2.7	69.91±2.7	69.96±2.4
	Layer mix + AdaPool	72.88±2.1	70.78±2.4	68.60±3.2
	ToDA (Ours)	**72.99**±1.7	**70.80**±2.8	**70.59**±2.6
TinyLlama	Mean	74.58±1.9	73.64±1.9	72.18±1.6
	Max	66.96±1.9	63.35±2.3	60.39±2.2
	[CLS]/last token	69.73±1.7	65.81±2.8	64.25±2.3
	AdaPool	68.59±2.1	66.89±1.7	65.23±4.5
	GLOT	74.19±1.7	72.78±1.8	71.13±2.0
	Layer mix + AdaPool	71.91±2.8	69.14±2.2	68.00±2.3
	ToDA (Ours)	**76.54**±2.1	**74.88**±1.9	**74.61**±1.8
LLaMA-3B	Mean	74.75±1.7	73.83±2.0	72.55±1.9
	Max	69.23±2.0	66.95±2.1	63.15±2.9
	[CLS]/last token	70.81±1.7	67.00±2.2	64.64±2.4
	AdaPool	71.91±2.0	69.74±2.9	68.65±3.0
	GLOT	73.93±2.0	72.54±2.7	70.30±2.6
	Layer mix + AdaPool	73.89±2.7	70.81±2.5	68.83±2.3
	ToDA (Ours)	**77.65**±2.7	**76.26**±2.1	**74.84**±1.6
Mistral-7B	Mean	74.29±1.4	73.20±1.8	71.89±2.1
	Max	67.64±1.8	66.43±2.4	63.14±2.4
	[CLS]/last token	68.14±2.1	64.94±2.5	62.04±2.8
	AdaPool	73.35±1.6	72.20±2.1	68.48±2.2
	GLOT	73.08±1.6	69.93±1.7	67.93±2.2
	Layer mix + AdaPool	74.19±2.0	70.89±2.0	70.24±2.8
	ToDA (Ours)	**77.85**±2.4	**76.54**±2.3	**74.59**±2.2

Table 17: **target-specific review aggregation with K reviews per business.** Test accuracy (%) ↑, mean and standard deviation over eight data seeds. Within each backbone, methods are listed as existing aggregation methods, natural baselines and ToDA; a dash marks a configuration without a complete eight-seed run. Within each backbone, the highest result in each column is shown in **bold** and the second highest is underlined.

Backbone	Method ↓ / K →	3	5	7
BERT	Mean	49.58±1.3	50.69±1.1	50.43±1.1
	Max	49.56±0.9	50.04±1.3	49.91±1.1
	[CLS]/last token	50.59±1.2	49.89±1.9	49.75±1.1
	AdaPool	49.94±1.6	50.34±1.2	50.50±1.5
	GLOT	49.79±1.4	50.85±1.2	50.06±1.0
	Layer mix + AdaPool	**50.86**±1.5	50.25±1.5	**50.98**±1.3
	ToDA (Ours)	49.98±0.9	**51.19**±1.8	49.18±0.8
RoBERTa	Mean	49.96±0.5	50.65±1.9	50.90±1.3
	Max	50.09±1.4	50.96±1.2	49.43±1.4
	[CLS]/last token	49.73±2.1	49.96±1.6	50.41±0.8
	AdaPool	50.40±1.3	**51.30**±1.0	50.45±1.6
	GLOT	52.01±4.1	50.95±1.9	52.79±2.1
	Layer mix + AdaPool	**53.73**±8.3	49.71±1.7	53.23±5.2
	ToDA (Ours)	53.46±1.4	50.49±1.2	**57.19**±2.1
SmolLM2	Mean	49.26±1.6	50.58±1.4	49.88±1.3
	Max	50.44±1.3	50.19±1.1	50.33±0.9
	[CLS]/last token	49.89±1.8	50.34±1.6	49.43±1.6
	AdaPool	49.90±1.3	49.86±1.0	49.73±1.9
	GLOT	49.73±1.3	50.19±1.7	49.14±1.0
	Layer mix + AdaPool	51.03±1.7	51.34±1.6	50.54±1.3
	ToDA (Ours)	**54.66**±4.9	**57.41**±3.9	**57.38**±5.9
TinyLlama	Mean	51.26±1.8	50.76±1.8	50.80±1.1
	Max	49.60±1.1	50.69±1.0	49.85±1.2
	[CLS]/last token	50.68±1.0	49.93±2.2	49.93±1.1
	AdaPool	50.76±1.0	50.44±1.4	49.66±1.3
	GLOT	66.71±4.7	65.19±13.2	81.85±5.4
	Layer mix + AdaPool	56.45±9.2	59.31±14.5	62.78±16.1
	ToDA (Ours)	**87.48**±1.1	**87.83**±2.0	**89.54**±0.6
LLaMA-3B	Mean	50.78±1.1	50.28±1.9	49.99±1.5
	Max	48.70±1.5	49.90±1.3	49.48±0.8
	[CLS]/last token	49.50±0.9	50.66±1.5	49.70±0.7
	AdaPool	50.76±1.8	49.46±1.4	49.65±1.4
	GLOT	56.90±8.6	52.76±5.9	50.60±4.4
	Layer mix + AdaPool	72.40±4.1	73.90±4.6	75.58±1.6
	ToDA (Ours)	**93.04**±0.7	**92.04**±0.8	**92.09**±0.6
Mistral-7B	Mean	52.93±1.9	53.41±1.5	52.56±2.2
	Max	51.11±1.0	49.38±1.3	51.10±0.7
	[CLS]/last token	51.20±2.0	51.24±1.7	51.91±1.9
	AdaPool	51.73±1.2	51.21±1.8	50.04±1.9
	GLOT	50.51±1.5	51.11±1.5	53.59±10.6
	Layer mix + AdaPool	79.35±2.6	81.83±1.1	83.54±1.2
	ToDA (Ours)	**94.10**±1.2	**93.59**±0.9	**94.13**±0.7

Table 18: **Aspect majority over K MAMS sentences.** Test accuracy (%) ↑, mean and standard deviation over eight data seeds. Within each backbone, methods are listed as existing aggregation methods, natural baselines and ToDA; a dash marks a configuration without a complete eight-seed run. Within each backbone, the highest result in each column is shown in **bold** and the second highest is underlined.

Backbone	Method ↓ / K →	1	3	5
BERT	Mean	82.50±1.4	76.31±1.3	76.58±1.0
	Max	80.25±1.4	72.93±1.3	71.71±1.8
	[CLS]/last token	84.05±1.0	74.10±1.6	73.54±1.2
	AdaPool	82.40±2.2	74.68±1.2	74.61±1.6
	GLOT	**84.48**±1.6	77.33±1.9	78.11±1.2
	Layer mix + AdaPool	83.10±1.1	73.51±2.3	73.68±1.4
	ToDA (Ours)	84.09±1.9	**79.34**±1.4	**79.83**±2.0
RoBERTa	Mean	81.69±2.2	76.88±1.1	76.58±1.2
	Max	76.50±1.9	71.31±1.6	69.69±1.3
	[CLS]/last token	80.89±1.5	75.41±1.6	75.23±1.0
	AdaPool	81.10±2.5	73.66±1.0	72.28±2.3
	GLOT	84.93±2.4	79.15±0.7	79.05±0.9
	Layer mix + AdaPool	84.88±0.7	75.20±2.2	74.93±1.8
	ToDA (Ours)	**85.75**±1.7	**80.90**±1.4	**81.38**±1.4
SmolLM2	Mean	80.39±2.3	75.83±0.9	76.25±1.5
	Max	78.61±1.9	70.01±1.1	69.13±2.3
	[CLS]/last token	83.35±1.6	72.93±1.1	71.00±1.0
	AdaPool	83.28±1.9	75.65±1.5	74.21±1.9
	GLOT	83.23±1.4	75.59±1.5	76.85±1.9
	Layer mix + AdaPool	84.55±2.3	77.04±1.6	76.31±0.9
	ToDA (Ours)	**85.74**±0.9	**78.44**±1.6	**78.96**±0.9
TinyLlama	Mean	84.24±2.1	78.36±1.0	78.86±1.4
	Max	80.46±2.0	74.51±0.8	72.74±1.1
	[CLS]/last token	83.21±1.0	74.74±1.4	72.59±1.1
	AdaPool	84.90±2.3	77.19±1.6	76.41±0.9
	GLOT	85.31±2.9	78.24±1.4	78.38±1.5
	Layer mix + AdaPool	85.43±2.0	75.51±1.8	75.95±1.7
	ToDA (Ours)	**86.73**±1.8	**80.19**±1.7	**80.19**±1.2
LLaMA-3B	Mean	85.36±1.9	78.01±0.8	78.24±1.3
	Max	81.75±2.5	75.06±1.6	74.56±0.8
	[CLS]/last token	83.09±2.3	75.60±1.3	73.88±1.1
	AdaPool	85.94±2.6	78.09±1.8	77.26±1.9
	GLOT	85.20±1.1	79.24±1.2	78.26±1.1
	Layer mix + AdaPool	84.14±2.1	78.16±1.4	77.09±1.3
	ToDA (Ours)	**86.51**±1.9	**81.00**±1.8	**81.44**±0.9
Mistral-7B	Mean	84.43±2.1	77.80±1.7	78.08±1.5
	Max	83.50±2.2	75.46±1.3	75.16±0.7
	[CLS]/last token	82.10±2.5	72.34±1.5	71.61±1.6
	AdaPool	84.78±3.4	75.91±1.5	76.05±1.5
	GLOT	84.70±1.8	77.68±1.0	77.50±1.1
	Layer mix + AdaPool	83.86±3.1	77.96±1.7	78.38±1.2
	ToDA (Ours)	**87.01**±1.9	**80.83**±1.3	**80.89**±1.8

Table 19: **Aspect-term sentiment classification on MAMS.** Test accuracy (%) ↑, mean and standard deviation over eight data-order seeds on the official splits. The highest result in each column is shown in **bold** and the second highest is underlined.

Method ↓ / Backbone →	BERT	RoBERTa	SmolLM2	TinyLlama	LLaMA-3B	Mistral-7B
EXISTING AGGREGATION METHODS						
Mean	56.49±0.4	55.19±0.5	59.71±0.6	64.54±1.0	64.41±0.8	64.98±0.7
Max	53.54±0.8	53.27±0.8	54.77±0.5	58.84±0.8	61.17±0.6	59.56±1.5
[CLS]/last token	58.36±0.3	56.39±0.8	57.43±0.7	59.26±1.1	59.83±1.1	58.30±0.7
AdaPool	72.22±0.6	72.76±1.5	66.84±0.9	71.44±0.7	69.68±0.7	70.11±0.7
GLOT	72.79±0.5	74.68±0.7	69.27±0.5	71.53±0.6	70.67±0.9	69.66±0.8
NATURAL BASELINES						
Layer mix + AdaPool	75.51±0.5	77.59±0.7	73.37±0.5	74.51±0.9	77.45±0.7	76.20±0.8
ToDA (Ours)	**77.49**±0.6	**78.97**±0.6	**74.55**±1.1	**78.66**±0.4	**79.30**±0.5	**79.88**±1.3

Table 20: **Kinship classification on CLUTRR by number of hops.** Test accuracy (%) ↑, mean and standard deviation over eight data seeds. Within each backbone, methods are listed as existing aggregation methods, natural baselines and ToDA; a dash marks a configuration without a complete eight-seed run. Within each backbone, the highest result in each column is shown in **bold** and the second highest is underlined.

Backbone	Method ↓ / Hops →	2	3	4
BERT	Mean	30.74±0.6	24.37±0.6	22.64±0.2
	Max	30.74±1.2	22.02±1.2	25.47±1.2
	[CLS]/last token	24.56±0.9	19.91±0.8	19.48±0.5
	AdaPool	50.89±0.8	49.65±0.6	50.30±3.3
	GLOT	91.69±0.8	87.80±1.8	91.23±1.1
	Layer mix + AdaPool	55.30±2.2	55.61±0.8	61.78±0.5
	ToDA (Ours)	**94.23**±1.5	**93.64**±0.3	**94.28**±0.4
RoBERTa	Mean	28.21±0.2	19.99±0.2	18.41±0.6
	Max	26.45±1.0	15.23±1.1	14.44±0.9
	[CLS]/last token	23.64±0.5	15.86±0.1	17.47±0.3
	AdaPool	47.46±0.5	43.74±2.6	40.01±1.2
	GLOT	81.93±2.1	72.04±3.2	76.54±1.5
	Layer mix + AdaPool	52.91±6.7	49.90±0.5	50.75±0.7
	ToDA (Ours)	**94.28**±2.0	**91.01**±2.3	**91.82**±0.8
SmolLM2	Mean	41.22±0.3	34.94±0.5	36.28±0.6
	Max	31.53±0.6	22.29±0.5	26.19±0.8
	[CLS]/last token	43.54±0.5	32.79±0.7	30.64±0.9
	AdaPool	56.80±1.0	51.55±1.1	71.72±0.7
	GLOT	64.42±1.4	72.97±2.6	85.44±2.8
	Layer mix + AdaPool	86.44±1.0	78.21±0.5	88.17±0.5
	ToDA (Ours)	**87.13**±1.0	**89.53**±0.9	**92.77**±0.7
TinyLlama	Mean	55.18±1.0	46.24±0.5	45.53±0.9
	Max	43.12±2.6	27.30±1.5	31.49±1.2
	[CLS]/last token	50.15±1.1	40.68±0.9	36.36±0.8
	AdaPool	67.83±1.0	80.46±2.6	86.11±2.7
	GLOT	83.21±2.1	90.66±1.2	90.41±0.6
	Layer mix + AdaPool	73.94±3.7	73.65±2.1	78.13±4.6
	ToDA (Ours)	**95.09**±2.1	**93.46**±2.6	**92.67**±0.9
LLaMA-3B	Mean	60.53±1.0	45.97±1.3	47.37±1.5
	Max	53.97±1.3	43.14±2.0	44.36±2.3
	[CLS]/last token	56.83±0.7	43.26±1.4	37.15±1.3
	AdaPool	78.65±3.6	85.60±5.0	85.34±2.5
	GLOT	85.53±2.6	88.85±1.3	89.76±2.2
	Layer mix + AdaPool	55.30±1.0	41.33±0.7	64.41±5.4
	ToDA (Ours)	**98.74**±0.7	**94.59**±1.0	**94.53**±1.0
Mistral-7B	Mean	79.61±1.0	55.69±1.2	55.72±1.0
	Max	62.11±3.5	44.36±1.7	45.83±2.5
	[CLS]/last token	61.88±1.4	46.69±1.6	43.56±1.7
	AdaPool	92.95±1.9	89.23±1.0	91.20±1.0
	GLOT	94.35±1.1	91.41±1.0	92.52±1.4
	Layer mix + AdaPool	78.92±4.1	83.99±5.3	84.44±6.0
	ToDA (Ours)	**99.01**±0.4	**96.32**±1.1	**96.79**±0.5

Table 21: **Counting positive reviews among at most K reviews.** Test accuracy (%) ↑, mean and standard deviation over eight data seeds. Within each backbone, methods are listed as existing aggregation methods, natural baselines and ToDA; a dash marks a configuration without a complete eight-seed run. Within each backbone, the highest result in each column is shown in **bold** and the second highest is underlined.

Backbone	Method ↓ / K →	5	9
BERT	Mean	40.13±1.3	24.76±0.8
	Max	33.10±1.3	18.43±1.1
	[CLS]/last token	36.61±1.9	24.63±1.4
	AdaPool	42.86±1.3	29.66±1.5
	GLOT	45.88±2.1	31.84±1.4
	Layer mix + AdaPool	46.39±1.5	30.23±0.9
	ToDA (Ours)	**55.26**±1.2	**40.06**±1.0
RoBERTa	Mean	42.03±1.0	27.65±0.8
	Max	32.35±2.0	18.93±0.8
	[CLS]/last token	38.53±1.2	26.05±0.9
	AdaPool	47.66±2.4	31.06±1.9
	GLOT	53.53±1.5	37.96±2.3
	Layer mix + AdaPool	50.28±2.7	31.36±1.3
	ToDA (Ours)	**59.88**±2.0	**42.94**±1.1
SmolLM2	Mean	39.10±1.5	24.35±1.1
	Max	27.21±1.4	15.64±1.2
	[CLS]/last token	37.06±0.7	21.81±1.6
	AdaPool	22.60±1.1	21.16±1.4
	GLOT	44.58±1.9	28.61±1.2
	Layer mix + AdaPool	49.29±1.8	30.86±1.2
	ToDA (Ours)	**51.55**±1.2	**35.68**±1.4
TinyLlama	Mean	45.95±0.9	28.96±1.6
	Max	35.91±1.3	20.58±1.4
	[CLS]/last token	40.79±1.0	23.91±1.1
	AdaPool	49.79±2.3	34.09±2.0
	GLOT	50.69±1.2	32.54±1.2
	Layer mix + AdaPool	57.48±1.8	35.26±1.0
	ToDA (Ours)	**60.16**±1.1	**43.10**±1.1
LLaMA-3B	Mean	48.31±1.0	28.94±1.6
	Max	40.13±1.0	23.66±1.3
	[CLS]/last token	42.59±1.6	25.59±0.7
	AdaPool	53.95±1.3	38.68±1.9
	GLOT	52.73±2.1	36.39±0.6
	Layer mix + AdaPool	54.68±2.6	40.89±1.0
	ToDA (Ours)	**62.80**±2.0	**45.63**±1.0
Mistral-7B	Mean	50.96±2.6	33.41±1.4
	Max	39.76±0.7	22.33±1.8
	[CLS]/last token	42.63±2.0	26.95±1.0
	AdaPool	48.98±3.0	28.84±0.7
	GLOT	51.94±2.5	34.69±1.5
	Layer mix + AdaPool	53.33±2.0	42.00±1.2
	ToDA (Ours)	**62.90**±1.8	**46.40**±1.1

Table 22: **Trainable parameters on target-specific reviews ($K = 3$)**, including the task classifier.

Method ↓ / Backbone →	TinyLlama	LLaMA-3B	Mistral-7B
EXISTING AGGREGATION METHODS			
Mean	4.1K	6.1K	8.2K
Max	4.1K	6.1K	8.2K
[CLS]/last token	4.1K	6.1K	8.2K
AdaPool	0.53M	0.79M	1.06M
GLOT	0.76M	1.02M	1.28M
NATURAL BASELINES			
Layer mix + AdaPool	0.53M	0.79M	1.06M
ToDA (Ours)	6.04M	12.08M	20.21M

Table 23: **ToDA minus Non-adaptive ToDA: the effect of token-specific layer weights.** Difference in test accuracy (points) with its paired 95% bootstrap interval over eight seeds; the three settings are $K = 3, 5, 7$ reviews per business for target-specific reviews and $K = 1, 3, 5$ sentences for aspect majority, and aspect sentiment has a single setting. Differences whose interval lies above zero are shown in **bold**. A dash marks a setting without a complete eight-seed run of the baseline.

Backbone	Task	Setting 1	Setting 2	Setting 3
BERT	Target-specific reviews	–	+1.20 [-0.05, +2.53]	-0.81 [-1.99, +0.50]
	Aspect majority	-0.85 [-1.99, +0.31]	**+1.89 [+0.98, +2.80]**	**+2.55 [+1.01, +3.81]**
	Aspect sentiment	**+3.01 [+2.45, +3.56]**		
RoBERTa	Target-specific reviews	+3.53 [-1.85, +12.76]	+0.15 [-0.81, +1.26]	**+7.29 [+0.30, +16.44]**
	Aspect majority	-0.89 [-2.64, +0.94]	**+1.75 [+0.93, +2.68]**	**+1.34 [+0.41, +2.20]**
	Aspect sentiment	+0.19 [-0.30, +0.53]		
SmolLM2	Target-specific reviews	+1.78 [-2.10, +6.04]	**+6.70 [+1.10, +13.15]**	**+6.53 [+0.96, +11.81]**
	Aspect majority	+0.01 [-1.17, +1.10]	+0.01 [-1.19, +1.21]	-0.05 [-0.74, +0.71]
	Aspect sentiment	**+0.52 [+0.08, +1.07]**		
TinyLlama	Target-specific reviews	**+18.51 [+15.38, +22.71]**	**+18.58 [+8.39, +29.01]**	**+12.25 [+3.55, +22.56]**
	Aspect majority	+0.30 [-0.71, +1.34]	-0.23 [-1.23, +0.66]	-0.34 [-0.99, +0.30]
	Aspect sentiment	**+1.73 [+1.08, +2.40]**		
LLaMA-3B	Target-specific reviews	**+6.90 [+5.69, +8.13]**	**+5.45 [+3.91, +7.13]**	**+4.45 [+3.06, +5.91]**
	Aspect majority	-0.81 [-1.70, +0.01]	+0.41 [-0.81, +1.60]	**+0.58 [+0.18, +0.97]**
	Aspect sentiment	**+0.77 [+0.26, +1.33]**		
Mistral-7B	Target-specific reviews	**+9.34 [+7.62, +10.94]**	**+5.78 [+4.16, +7.34]**	**+4.04 [+3.34, +4.71]**
	Aspect majority	**+1.24 [+0.13, +2.41]**	+0.75 [-0.29, +1.80]	+0.06 [-1.35, +1.48]
	Aspect sentiment	**+2.15 [+1.19, +3.02]**		

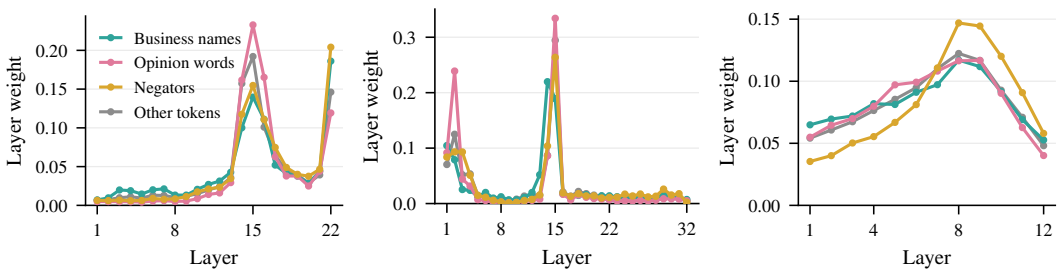
E Component Comparisons and Layer Weights

This appendix supports RQ2 and complements Table 4 and Figure 2.

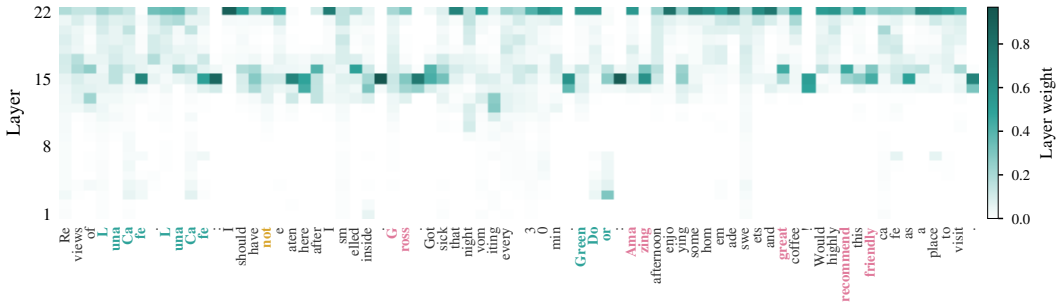
Token-specific layer weights over all settings. Non-adaptive ToDA replaces the token-specific weights of ToDA by weights shared by all tokens and inputs, before a relational stage with the same graph network. Table 23 gives the difference for every setting. ToDA is more accurate with positive intervals in all nine target-specific review settings of TinyLlama, LLaMA-3B and Mistral-7B and in three of the six settings of RoBERTa and SmolLM2, and on five of the six backbones for aspect sentiment. On aspect majority, six differences are positive and the other twelve have intervals that include zero.

Layer weights. The weights in Figures 2b and 2c come from training runs on target-specific reviews with $K = 3$ and relation threshold 0.9, with four seeds on TinyLlama and three on Mistral-7B, trained for 40 epochs. For each run, the weights are averaged over heads, tokens and validation batches at the final epoch, and then averaged over seeds. These runs reach 87.58% (TinyLlama) and 95.47% (Mistral-7B) test accuracy with temperature 1.0, and 65.65% and 93.17% with the initial temperature 0.1. The probe peaks marked in the figure are the most accurate layers of the linear probe of Appendix C at $K = 3$, layer 15 on both backbones. Temperature 1.0 is selected on validation data in all nine decoder settings of target-specific reviews and in eight of twelve decoder settings of the aspect tasks.

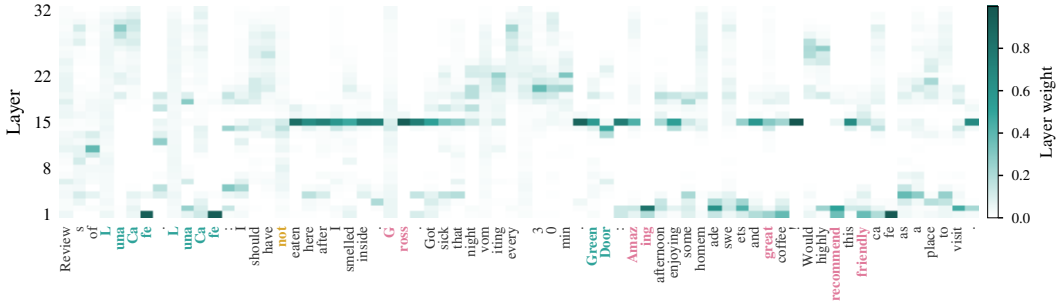
Per-token layer weights. To compare the weights of individual tokens, we repeated the runs of Figures 2b and 2c with the same seeds and configurations, added BERT with seeds 42 to 45, and recorded $\bar{\alpha}_{t,\ell}$ for every token of the first 64 test inputs after the final epoch. With temperature 1.0, these runs reach 87.08% (TinyLlama), 95.17% (Mistral-7B) and 50.68% (BERT) test accuracy. Figure 3 groups the tokens into business names, opinion words, negators and other tokens, using the business names of the task construction and fixed word lists. On TinyLlama, opinion words place 0.23 of their weight on layer 15 and 0.12 on the final layer, whereas business names place 0.14 on layer 15 and 0.19 on the final layer. On Mistral-7B, opinion words place 0.33 of their weight on layer 15 and business names 0.19, with a peak of 0.22 at layer 14, and every category places less than 0.01 on the final layer. On BERT, which remains at chance level on this task, the four categories receive



(a) Token categories, TinyLlama (b) Token categories, Mistral-7B (c) Token categories, BERT



(d) Example input, TinyLlama



(e) Example input, Mistral-7B

Figure 3: **Layer weights differ between the tokens of an input.** (a to c) Layer weights $\bar{\alpha}_{t,\ell}$ of ToDA on target-specific reviews with $K = 3$ and temperature 1.0, averaged over the tokens of each category within an input, then over 64 test inputs and the seeds. (d, e) Weights of the individual tokens of one test input, the first that both decoders classify correctly and whose first two reviews contain a negator, shown up to the end of the second review; tick labels are colored by category (business names turquoise, opinion words pink, negators yellow).

similar weights. Figures 3d and 3e show one test input as a heatmap: each column is a token, each row is a layer, and a darker cell means that the token gives more weight to that layer; the weights of each token sum to one. The input asks for the majority opinion about Luna Cafe, and the part shown contains the query, a negative review of Luna Cafe and a positive review of Green Door. On TinyLlama, 68 of the 73 tokens shown place their largest weight on layer 15, 16 or the final layer. On Mistral-7B, the tokens of the Luna Cafe review place 0.34 of their weight on layers 14 and 15 and 0.10 on layers 1 to 3, whereas the tokens of the Green Door review place 0.29 and 0.34, and the final layer receives almost no weight. Different tokens of the same input therefore draw on different layers, which a layer mixture shared by all tokens cannot express.

F STANDARD BENCHMARKS

This appendix supports RQ3 and complements Table 5.

F.1 GLUE AND IMDB

For every backbone and task, ToDA is trained on the GLUE training split (Wang et al., 2018) or IMDB (Maas et al., 2011). As in Mantri et al. (2026), QQP, QNLI and MNLI use a subsample of 20,000 training examples, and inputs are truncated at 128 tokens (512 for IMDB). Each entry is the best run of Bayesian hyperparameter searches over settings that include the relation threshold, the initial layer-weighting temperature, the learning rate, weight decay and optimizer, the depth, width and jumping knowledge of the graph network, the projection width, the batch size, and the latent width, number of heads and query and key width of ToDA; most runs train for five epochs, and each entry reports the epoch with the highest score. No separate development split is held out, so the official validation split both selects and reports each entry; Mantri et al. (2026) likewise report one run per task, whose configuration was chosen by a grid search. On MNLI mismatched, ToDA reaches 57.66, 57.76 and 59.96% for BERT, TinyLlama and Mistral-7B, compared with 54.47, 49.34 and 53.22% reported for GLOT. The baseline scores are those reported by Mantri et al. (2026), obtained with their training and selection procedure; the comparison therefore places ToDA next to the published results. Tables 24 and 25 give every task. ToDA improves over GLOT in 28 of 30 task and backbone pairs, is equal in one, and improves on the average of every backbone, with the largest gains on CoLA, MNLI and SST-2.

F.2 MTEB

All five methods are trained on MS MARCO passage pairs (Bajaj et al., 2016) with a contrastive objective and a learned output projection, using cached bfloat16 hidden states of the frozen backbone. Each method receives a 32-trial search (40 trials for ToDA on BERT, see below) on 20,000 pairs with 128 tokens, followed by a single retraining of the selected configuration on 100,000 pairs. The configuration and the epoch are selected on six development tasks (STSBenchmark, SprintDuplicateQuestions, NFCorpus, Banking77Classification, StackOverflowDupQuestions and TwentyNewsgroupsClustering), which are excluded from the reported score. The score averages the twelve CQADupstack subcorpora within one category and then averages uniformly over the remaining 44 evaluation categories (Muennighoff et al., 2023). The searches of mean, max and last-token aggregation and GLOT vary the contrastive temperature $\{0.05, 0.07\}$, the batch size $\{32, 64, 128\}$ and the learning rate (log-uniform in $[10^{-4}, 6 \times 10^{-4}]$) with 12 epochs; GLOT additionally varies the graph width $\{128, 256\}$, the depth $\{4, 6\}$ and the relation threshold (uniform in $[0.3, 0.6]$), with concatenated jumping knowledge. The search of ToDA varies the same temperatures, batch sizes and learning rates, hard negatives (used or not), the latent width $\{256, 512\}$ and the graph width and depth as for GLOT, with 30 epochs on BERT and 12 on the other backbones; its relation threshold stays at 0.3. On BERT, the selected ToDA configuration comes from an earlier 40-trial search with temperatures $\{0.01, 0.03, 0.05, 0.07\}$, batch sizes $\{128, 256, 512\}$ and learning rates up to 2×10^{-3}. Every method uses an output projection of width 256, and the retraining lasts six epochs. The selected ToDA configurations use temperature 0.05 or 0.07 and batch sizes from 64 to 256. Each entry is a single retraining with one seed, so the variability across training seeds is not measured. Table 26 groups the categories by task family, and Tables 27 and 28 list every category for BERT and TinyLlama and for Mistral-7B.

F.3 PARAMETER COST ON COLA

Table 29 places the Mistral-7B CoLA result of ToDA next to the full fine-tuning, LoRA (Hu et al., 2021) and GLOT results reported by Mantri et al. (2026). Our result comes from a separate search of 128 trials, selected on a development split held out from the training data: the frozen Mistral-7B backbone supplies layers 1 to 32 at a maximum length of 128 tokens, and the selected configuration uses eight heads, latent width 128, query and key width 32, the residual gate, four graph attention layers of width 256 with max jumping knowledge, a width-64 Gaussian projection for edge construction. It trains for five epochs with batch size 64 and AdamW with cosine decay. The selected

Table 24: **GLUE and IMDB results (part 1).** Scores on a 0 to 100 scale ↑. Baseline scores are those reported by Mantri et al. (2026). ToDA reports the best run of its hyperparameter searches on the official validation split. The highest result in each column is shown in **bold** and the second highest is underlined.

Method ↓ / Backbone →	BERT	TinyLlama	Mistral-7B
CoLA (MCC) ↑			
Mean	19.55	5.93	38.61
Max	15.79	2.76	10.78
[CLS]/last token	22.66	8.33	38.63
AdaPool	29.20	4.63	48.00
GLOT	47.49	17.61	54.30
ToDA (Ours)	**51.92**	**55.93**	**67.74**
SST-2 (ACCURACY) ↑			
Mean	82.91	73.85	89.91
Max	80.73	70.87	85.89
[CLS]/last token	83.83	73.85	92.55
AdaPool	87.72	59.92	93.00
GLOT	90.25	80.73	94.38
ToDA (Ours)	**92.32**	**94.27**	**97.13**
STS-B (SPEARMAN) ↑			
Mean	74.96	61.29	77.96
Max	74.12	63.99	70.72
[CLS]/last token	61.08	64.63	72.36
AdaPool	80.01	69.53	79.55
GLOT	83.86	71.77	80.51
ToDA (Ours)	**84.93**	**83.85**	**84.24**
MRPC (F1) ↑			
Mean	80.28	80.67	77.22
Max	81.64	81.45	65.61
[CLS]/last token	79.58	80.31	76.32
AdaPool	77.99	81.04	81.12
GLOT	82.58	82.54	82.83
ToDA (Ours)	**83.01**	**82.94**	**83.11**
QQP (F1) ↑			
Mean	29.01	41.46	57.44
Max	29.58	39.64	54.39
[CLS]/last token	19.70	41.46	51.68
AdaPool	40.15	30.17	49.07
GLOT	62.19	59.92	64.07
ToDA (Ours)	**62.44**	**62.66**	**66.67**

trial has a development score of 68.18 and a validation score of 66.00, and its 19,191,716 trainable parameters include the task head.

Table 25: **GLUE and IMDB results (part 2).** Scores on a 0 to 100 scale ↑. Baseline scores are those reported by Mantri et al. (2026). ToDA reports the best run of its hyperparameter searches on the official validation split. The highest result in each column is shown in **bold** and the second highest is underlined.

Method ↓ / Backbone →	BERT	TinyLlama	Mistral-7B
MNLI-M (ACCURACY) ↑			
Mean	43.86	39.50	47.86
Max	38.60	36.88	38.77
[CLS]/last token	43.86	39.33	48.18
AdaPool	48.57	42.69	50.72
GLOT	54.39	48.04	51.66
ToDA (Ours)	**56.71**	**56.50**	**60.92**
QNLI (ACCURACY) ↑			
Mean	56.43	57.51	53.46
Max	53.79	55.29	58.70
[CLS]/last token	54.75	56.19	50.82
AdaPool	58.04	57.71	55.75
GLOT	61.08	**63.77**	60.93
ToDA (Ours)	**63.34**	63.23	**62.35**
RTE (ACCURACY) ↑			
Mean	51.62	49.58	53.07
Max	51.98	50.90	53.07
[CLS]/last token	50.90	47.29	50.90
AdaPool	51.62	46.20	54.87
GLOT	59.21	57.40	59.21
ToDA (Ours)	**62.82**	**61.73**	**64.62**
WNLI (ACCURACY) ↑			
Mean	52.11	45.07	42.25
Max	49.26	46.48	48.70
[CLS]/last token	45.07	45.07	40.85
AdaPool	45.07	50.70	49.30
GLOT	54.93	53.52	**56.34**
ToDA (Ours)	**59.15**	**56.34**	**56.34**
IMDB (ACCURACY) ↑			
Mean	81.64	88.72	94.21
Max	60.78	75.45	64.43
[CLS]/last token	80.23	87.27	84.86
AdaPool	85.45	92.61	95.66
GLOT	86.93	93.38	95.95
ToDA (Ours)	**92.61**	**95.84**	**96.92**

Table 26: **MTEB scores by task family.** Average score ↑ (×100) over BERT, TinyLlama and Mistral-7B for classification, clustering, pair classification, reranking, retrieval, semantic textual similarity and summarization; the number of evaluation categories per family is given in parentheses. ToDA leads on retrieval, semantic similarity and reranking, while mean aggregation leads on classification and clustering. The highest result in each column is shown in **bold** and the second highest is underlined.

Method ↓ / Family →	Class. (11)	Clust. (10)	Pair (2)	Rerank. (3)	Retr. (8)	STS (9)	Summ. (1)	All (44)
EXISTING AGGREGATION METHODS								
GLOT	45.80	38.06	60.79	52.02	34.26	57.42	27.93	45.02
Mean	**48.02**	**39.78**	**65.50**	52.33	31.63	57.79	29.43	45.83
Max	45.57	36.76	62.94	49.98	23.36	54.18	29.01	42.00
[CLS]/last token	43.48	34.82	55.35	48.16	17.29	45.99	27.72	37.76
ToDA (Ours)	46.94	37.37	62.29	**53.25**	**38.71**	**62.49**	**29.85**	**47.19**

Table 27: **MTEB scores per evaluation category for BERT and TinyLlama.** Score ↑ (×100). Within each backbone, the highest score in each row is shown in **bold** and the second highest is underlined.

	BERT					TinyLlama				
Task ↓ / Method →	GLOT	Mean	Max	CLS	ToDA (Ours)	GLOT	Mean	Max	CLS	ToDA (Ours)
CLASSIFICATION										
AmazonCounterfactual	56.87	**62.16**	61.41	58.02	58.07	58.07	**63.23**	60.47	56.47	55.06
AmazonPolarity	60.54	64.43	63.83	**65.79**	59.08	61.15	64.67	62.73	**69.56**	59.67
AmazonReviews	25.10	25.36	24.97	24.52	**25.58**	28.32	**31.76**	27.07	28.55	27.49
Emotion	**39.93**	37.13	32.99	33.05	39.28	37.22	34.55	29.24	29.45	**38.99**
Imdb	58.20	59.92	57.24	**60.64**	57.15	60.18	**61.21**	59.44	56.68	58.30
MTOPDomain	50.52	47.82	48.31	44.54	**53.23**	64.85	67.78	65.76	50.30	**71.07**
MTOPIntent	27.28	25.20	27.10	22.70	**31.26**	40.57	**45.85**	43.45	25.64	41.60
MassiveIntent	19.25	16.61	19.55	15.14	**23.29**	28.13	29.49	29.72	20.48	**33.55**
MassiveScenario	24.75	22.03	24.43	20.79	**28.60**	34.48	36.14	34.67	28.04	**40.65**
ToxicConversations	63.90	**67.00**	61.26	63.07	61.76	59.83	**62.09**	60.62	60.31	59.66
TweetSentimentExtraction	47.44	**48.37**	47.23	47.62	47.75	48.03	**51.29**	47.47	49.07	47.50
CLUSTERING										
ArxivClusteringP2P	36.41	**36.86**	34.18	36.42	34.88	38.86	**46.15**	42.89	26.63	41.58
ArxivClusteringS2S	28.73	**29.58**	28.54	27.11	26.96	31.72	**36.39**	33.51	35.48	34.99
BiorxivClusteringP2P	28.20	**30.08**	29.11	30.04	28.34	31.47	**35.05**	33.44	22.54	32.00
BiorxivClusteringS2S	24.57	**25.64**	25.28	23.86	22.57	26.91	**30.47**	29.37	29.02	28.94
MedrxivClusteringP2P	29.03	**29.11**	26.40	28.03	28.17	31.22	**33.28**	31.77	21.59	31.79
MedrxivClusteringS2S	**26.34**	26.32	25.15	23.41	25.30	27.98	29.98	28.96	25.63	**30.47**
Reddit	**44.24**	43.42	39.44	39.93	42.31	46.39	48.50	42.31	**49.74**	48.32
RedditClusteringP2P	**48.25**	46.53	42.30	44.44	46.47	52.05	**54.32**	51.38	40.88	52.27
StackExchange	52.43	**53.16**	51.97	51.81	50.44	60.17	61.21	58.27	**62.43**	58.36
StackExchangeClusteringP2P	32.04	28.58	23.75	27.13	**32.41**	34.49	34.72	28.92	26.05	**34.84**
PAIR CLASSIFICATION										
TwitterSemEval2015	50.45	**56.25**	54.78	53.57	52.14	40.28	**49.70**	46.73	45.55	42.13
TwitterURLCorpus	79.38	78.42	77.59	73.61	**79.83**	77.52	79.02	75.93	51.63	**79.14**
RERANKING										
AskUbuntuDupQuestions	51.58	49.36	47.76	46.24	**52.58**	53.82	52.64	51.90	46.49	**57.25**
MindSmall	**30.80**	30.41	30.49	29.92	30.45	32.30	**33.28**	32.15	31.40	31.70
SciDocsRR	69.28	**70.45**	67.35	63.22	68.87	70.19	73.13	67.80	68.14	**74.16**
RETRIEVAL										
ArguAna	**41.39**	37.72	20.08	26.87	40.47	30.43	**44.77**	28.22	14.56	38.63
CQADupstack (mean)	21.04	13.07	6.48	5.01	**23.01**	24.21	20.28	11.70	4.21	**27.69**
FiQA2018	14.93	7.75	3.84	4.28	**16.45**	15.92	13.28	7.30	4.16	**20.63**
Quora	78.91	75.73	70.24	68.43	**80.01**	74.83	70.68	65.61	59.92	**80.28**
SCIDOCS	10.56	7.49	4.38	3.63	**11.63**	9.32	9.65	5.61	2.15	**14.29**
SciFact	38.87	30.73	25.86	15.93	**45.97**	44.39	41.21	29.12	10.71	**50.37**
TRECCOVID	28.04	30.76	19.23	18.18	**44.01**	56.28	42.49	32.52	9.76	**67.38**
Touche2020	15.88	12.17	6.19	8.30	**21.32**	15.78	12.85	9.65	4.95	**19.31**
STS										
BIOSSES	69.19	62.89	66.75	57.28	**71.85**	61.09	**70.51**	69.76	44.28	68.01
SICK-R	64.77	**66.51**	63.87	56.22	65.29	53.28	60.60	61.17	58.88	**61.42**
STS12	57.41	51.44	46.87	35.93	**61.41**	53.62	52.07	49.86	34.56	**60.19**
STS13	73.13	72.10	66.33	61.51	**76.56**	67.30	65.69	63.74	53.93	**71.39**
STS14	62.59	61.53	56.32	47.35	**66.65**	58.99	58.61	57.30	39.13	**63.69**
STS15	70.59	71.16	69.95	62.10	**76.42**	70.15	70.65	67.67	54.79	**75.19**
STS16	72.81	71.30	70.14	68.96	**73.81**	68.10	66.38	66.04	64.12	**69.18**
STS17	**29.64**	23.73	22.06	15.95	28.04	37.79	37.50	34.72	26.55	**48.92**
STS22	23.44	22.97	20.66	22.73	**25.37**	39.87	40.39	36.61	12.67	**47.22**
SUMMARIZATION										
SummEval	29.38	**30.73**	28.27	29.69	29.60	26.67	**30.47**	29.16	27.56	29.85

Table 28: **MTEB scores per evaluation category for Mistral-7B.** Score ↑ (×100). The highest score in each row is shown in **bold** and the second highest is underlined.

	Mistral-7B				
Task ↓ / Method →	GLOT	Mean	Max	CLS	ToDA (Ours)
CLASSIFICATION					
AmazonCounterfactual	57.63	**64.09**	59.03	55.54	53.65
AmazonPolarity	63.43	60.92	62.54	**64.71**	58.24
AmazonReviews	28.91	**33.05**	28.99	31.76	28.49
Emotion	35.22	33.79	27.65	30.69	**36.42**
Imdb	61.00	**62.06**	59.67	55.00	56.72
MTOPDomain	63.17	70.87	61.82	61.19	**71.57**
MTOPIntent	39.18	**48.60**	41.15	34.12	41.25
MassiveIntent	28.38	33.33	31.17	26.45	**36.65**
MassiveScenario	34.83	40.62	36.81	34.53	**44.43**
ToxicConversations	58.13	**62.86**	58.48	60.90	56.59
TweetSentimentExtraction	47.06	**50.28**	47.57	49.37	46.43
CLUSTERING					
ArxivClusteringP2P	43.07	**47.49**	42.99	27.81	37.42
ArxivClusteringS2S	34.40	**39.02**	33.35	37.50	33.10
BiorxivClusteringP2P	32.44	**35.90**	34.00	24.20	31.16
BiorxivClusteringS2S	28.30	**31.49**	29.17	30.93	28.80
MedrxivClusteringP2P	31.95	**34.27**	31.96	23.44	32.11
MedrxivClusteringS2S	28.32	30.67	28.74	26.02	**30.98**
Reddit	51.65	52.36	45.12	**56.79**	48.48
RedditClusteringP2P	60.55	**62.09**	58.55	49.00	55.99
StackExchange	64.66	64.76	61.63	**69.99**	56.12
StackExchangeClusteringP2P	35.05	**36.03**	30.29	26.87	35.57
PAIR CLASSIFICATION					
TwitterSemEval2015	39.39	49.60	45.31	**50.70**	41.77
TwitterURLCorpus	77.70	**80.00**	77.30	57.04	78.70
RERANKING					
AskUbuntuDupQuestions	55.59	53.21	52.03	47.95	**56.15**
MindSmall	31.88	**33.32**	32.20	31.62	32.93
SciDocsRR	72.73	**75.15**	68.13	68.43	75.13
RETRIEVAL					
ArguAna	37.68	**49.20**	34.19	16.00	41.50
CQADupstack (mean)	26.87	22.85	12.56	5.25	**27.25**
FiQA2018	20.34	17.01	8.97	7.76	**21.27**
Quora	77.56	74.41	67.48	72.81	**80.62**
SCIDOCS	12.26	11.15	7.03	2.41	**14.50**
SciFact	51.99	**53.70**	35.86	13.37	53.62
TRECCOVID	58.22	47.48	39.13	30.14	**67.80**
Touche2020	16.43	12.62	9.45	6.17	**20.94**
STS					
BIOSSES	67.15	**76.57**	71.05	47.63	70.54
SICK-R	59.67	60.81	56.11	59.30	**62.40**
STS12	53.21	48.77	43.00	35.78	**57.83**
STS13	66.28	65.21	61.54	59.23	**72.27**
STS14	59.60	58.83	55.11	45.01	**63.78**
STS15	71.33	70.13	63.06	59.14	**73.73**
STS16	67.73	67.15	63.21	67.56	**67.91**
STS17	36.34	40.29	31.80	34.57	**56.35**
STS22	35.29	46.42	28.22	16.44	**51.77**
SUMMARIZATION					
SummEval	27.74	27.09	29.60	25.91	**30.08**

Table 29: **Trainable parameters, cost and CoLA accuracy with Mistral-7B.** Matthews correlation coefficient ($\times 100$) $\uparrow$; peak GPU memory and runtime of one training batch (batch size 32, frozen backbone with precomputed representations). The rows of existing methods are reported by Mantri et al. (2026) on an NVIDIA A6000; ToDA (Ours) is measured with the same protocol on an NVIDIA RTX 6000 Ada, with the runtime averaged over 10 batches.

Method $\downarrow$	Trainable parameters $\downarrow$	Memory (GB) $\downarrow$	Time per batch (ms) $\downarrow$	MCC $\uparrow$
EXISTING ADAPTATION METHODS				
Full fine-tuning	7.11B	32.59	1318.8	49.63
LoRA ($r = 64$)	167.8M	33.50	1454.6	48.23
EXISTING AGGREGATION METHODS				
GLOT	8.92M	0.42	13.4	53.29
ToDA (Ours)	19.19M	4.15	61.2	**66.00**

G IMPLEMENTATION DETAILS

This appendix complements Section 3.

Configurations. In all studies, the layer aggregation is computed in the compressed space of width r. The relational stage follows GLOT (Mantri et al., 2026): it builds the token graph by thresholding cosine similarities, applies graph attention layers with jumping knowledge and aggregates the nodes with a learned attention readout. On the natural-text tasks, the controlled task and the SST-2 paragraphs, ToDA uses four heads, the residual gate, four graph layers and max jumping knowledge, with latent width 64 (or 256 after tuning on the natural-text tasks). On GLUE and IMDB, these settings are part of the search. On MTEB, ToDA uses one head with query and key width equal to the latent width, which is 256, no gate and concatenated jumping knowledge.

Natural baselines. Layer mix computes $\sum_{\ell \in \mathcal{S}} w_\ell \mathbf{h}_{t,\ell}$ with $\mathbf{w} = \mathrm{softmax}(\boldsymbol{\lambda})$ and one learned vector $\boldsymbol{\lambda} \in \mathbb{R}^M$ shared by all tokens and inputs, followed by mean aggregation (Layer mix + mean), AdaPool (Layer mix + AdaPool) or Non-adaptive ToDA.