AliGATr: Graph-based layout generation for form understanding

Armineh Nourbakhsh[1,2], **Zhao Jin**[1], **Siddharth Parekh**[1], **Sameena Shah**[2], **Carolyn P. Rosé**[1]

[1]Language Technologies Institute, Carnegie Mellon University
[2]J.P. Morgan, New York
anourbak@cs.cmu.edu

Abstract

Forms constitute a large portion of layout-rich documents that convey information through key-value pairs. Form understanding involves two main tasks, namely, the identification of keys and values (a.k.a Key Information Extraction or KIE) and the association of keys to corresponding values (a.k.a. Relation Extraction or RE). State of the art models for form understanding often rely on training paradigms that yield poorly calibrated output probabilities and low performance on RE. In this paper, we present AliGATr, a graph-based model that uses a generative objective to represent complex grid-like layouts that are often found in forms. Using a grid-based graph topology, our model learns to generate the layout of each page token by token in a data efficient manner. Despite using 30% fewer parameters than the smallest SotA, AliGATr performs on par with or better than SotA models on the KIE and RE tasks against four datasets. We also show that AliGATr's output probabilities are better calibrated and do not exhibit the over-confident distributions of other SotA models.

1 Introduction

Visually complex forms pose a multimodal challenge to the task of document-grounded reasoning. State of the art approaches have achieved increasingly advanced performance, but their utility in downstream applications remains restricted by a few key challenges we face broadly as a research area, as described in the recent position paper by Nourbakhsh et al. (2024). In this paper, we introduce AliGATr, a new form understanding model that addresses these challenges in order to improve the practical impact of research on document-grounded reasoning for visually complex forms, with potential impact on the broader fields of information extraction and multimodal document understanding.

The two common tasks in form understanding that illustrate the challenges we address include Key Information Extraction (KIE) and Relation Extraction (RE), both of which rely on identifying field names and field values, understanding tabular structures, and distinguishing between headings, main content, and other components, all of which require joint reasoning over the spatial and textual signal on each page.

With a few exceptions, most SotA researches focus on the task of KIE, disregarding RE, whereas in most applications KIE and RE need to be paired in order to identify semantically valid key-value pairs from forms. Without RE, key structural information about the document will not be captured (Luo et al., 2023; Zhang et al., 2021), and open-ended key-value extraction will be difficult (Luo et al., 2023). Despite this, RE remains underexplored in the form understanding literature (Luo et al., 2023; Hong et al., 2021; Li et al., 2021), posing major challenges to downstream applications (Nourbakhsh et al., 2024).

Furthermore, most models require extensive pre-training data and infrastructure to perform at the SotA level. As an example, the most popular pre-training dataset is the IIT-CDIP dataset (Lewis et al., 2006), composed of 11 million images. Lastly, the trade-off between grounding (i.e. providing bounding boxes for each output token such that it can be traced back to the input) and calibration (i.e. producing distributionally robust probabilities) is difficult to balance. Small, efficient, robust, and well-calibrated models remain difficult to obtain for users with limited access to large-scale pre-training data or compute.

In this paper, we introduce **AliGATr**, a new form understanding model that addresses the above challenges by combining a graph-based representation and a layout-generation objective. By focusing on the generation of layout (as opposed to the joint generation of text and layout), our approach leads

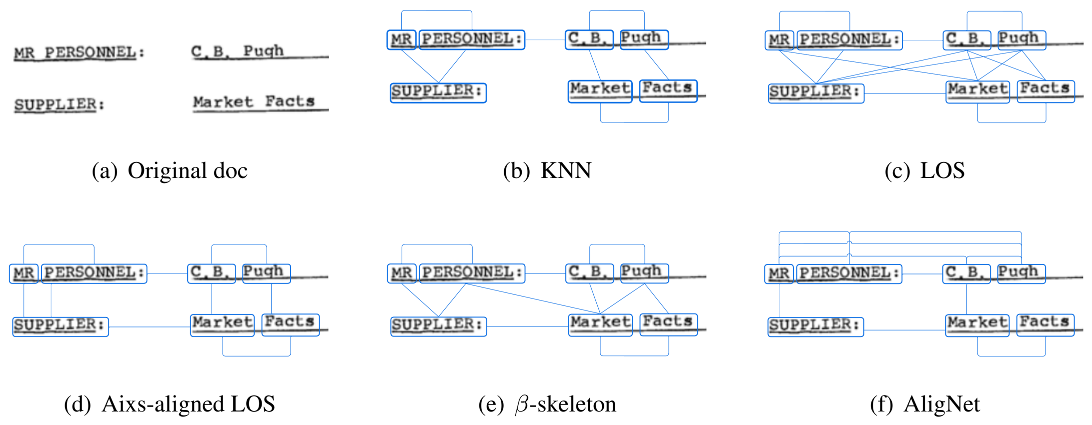


Figure 1: Different graph representations for a given form.

to a model that is more compact and converges using a smaller pre-training dataset. Our proposed graph representation, which we name **AligNet**, enables the model to cover both KIE and RE tasks, leading to SotA performance on the former and exceeding SotA performance on the latter. Even though AliGATr has a generative objective, it samples its output from input tokens, leading to logits that are well-grounded and well-calibrated.

Concretely, our study offers the following contributions to the literature on visually rich form understanding (VrFU):

- We propose AligNet, a graph representation technique for form documents, inspired by the four principles of layout design (Kimball, 2013). AligNet uses soft alignments between tokens to capture short- and long-range spatial dependencies. Its alignment-based structure (compared to the proximity-based structures often used in SotA models) allows it to propagate information more effectively.

- We introduce AliGATr, a GNN-based method inspired by GraphRNN (You et al., 2018), which uses a generative objective to learn layout-aware node representations. The generative objective combines next-node selection with adjacency prediction, allowing the model to recreate the layout of a page token by token. To the best of our knowledge, AliGATr is the first graph-based model to use a generative objective for form understanding.

- With 30% fewer parameters compared to the smallest SotA baseline, and using a small pre-training dataset of 1 million documents, AliGATr performs competitively on the KIE and RE tasks. Furthermore, we show that our model produces better-calibrated output distributions compared to baselines and is not over-confident.

2 Related Work

Research in visually rich document understanding has explored models in two architectural paradigms, namely transformer-based models and graph-based models.

2.1 Transformer-based models

Transformer-based models such as BROS (Hong et al., 2021), Docformer (Appalaraju et al., 2021), and the LayoutLM series (Xu et al., 2020, 2021; Huang et al., 2022) are often inspired by encoder-only architectures and use an adaptation of Masked Language Modeling (MLM) (Devlin et al., 2019) such as Masked Visual Language Modeling (Xu et al., 2020, 2021; Li et al., 2021), Masked Sequence Modeling (Gu et al., 2021), learning to reconstruct (Appalaraju et al., 2021), word-patch alignment (Huang et al., 2022), and vision-language alignment (Gu et al., 2021). A drawback of encoder-based models is that their output probabilities aren't well calibrated (Kumar and Sarawagi, 2019). This means that the output probabilities of these models don't reflect their performance, as the models can be arbitrarily over- or under-confident (Jiang et al., 2021).

In recent years, the adaptation of autoregressive language models to the task of document understanding has produced models that favor a decoder-based architecture and follow generative objectives

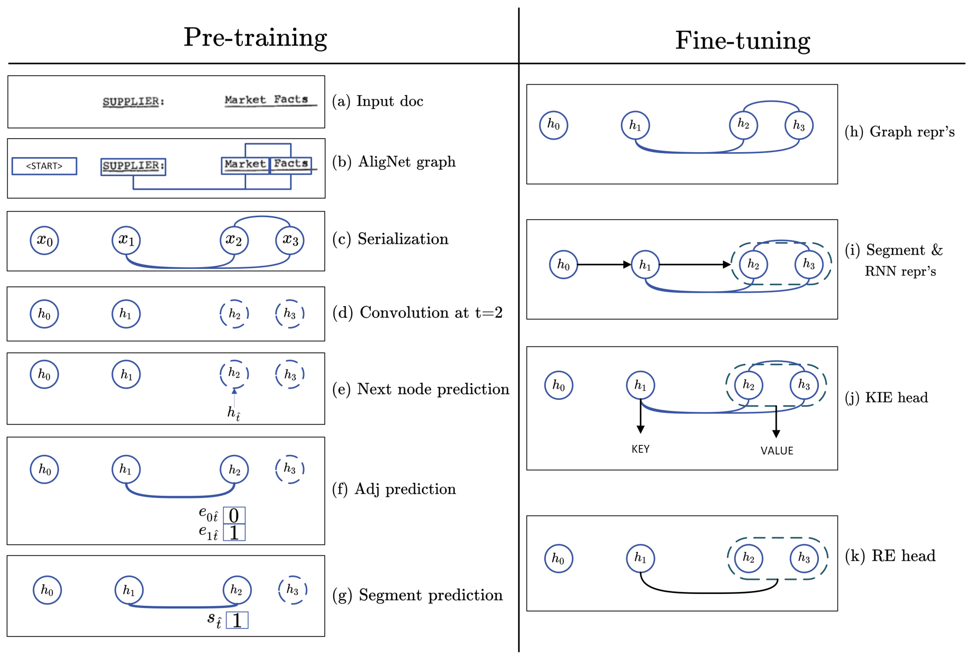


Figure 2: Pre-training and fine-tuning steps in our proposed approach. (a) During pre-training, a form is fed into the model as a set of tokens and bounding boxes. (b) The form is represented as an AligNet graph. (c) The serializer orders the nodes and each node is represented using its text embedding. (d) At step $t = 2$, the graph convolution produces representations $h_0 \cdots h_3$, where any edges adjacent to nodes x_2 and x_3 has been dropped, hence masking their spatial information. (e) The next node predictor uses a pointer mechanism to correctly predict the next node representation to be h_2. (f) The model predicts the adjacency vector between x_2 and the previous nodes as a binary vector. (g) The model predicts a segmentation flag for x_2. (h) During fine-tuning, AliGATr generates graph representations for each node. (i) Using the model's segmentation flags, the graph is split into segments, and an RNN is used to create sequence representations for each segment. (j) For the KIE task, a classification head predicts a class for each segment. (k) For the RE task, a link prediction head predicts the edges between segments.

such as next word prediction (Tang et al., 2023) or block infilling (Wang et al., 2023). While often better calibrated, these models sample their output from the vocabulary (as opposed to the input) and are therefore not guaranteed to produce outputs that can be grounded within the input. This is important for information extraction tasks, where the output should be traceable back to the input (Nourbakhsh et al., 2024).

Another challenge of transformer-based models is their performance on associative tasks. Proper understanding of a form relies on two tasks–the extractive task of KIE, and the associative task of RE. Transformer-based models have consistently underperformed on RE compared to graph-based models such as VisualFudge (Davis et al., 2021), or hybrid graph-transformer models such as GeoLayoutLM (Luo et al., 2023) and RE^2 (Ramu et al., 2024).

2.2 Graph-based models

Through their topology, graphs provide a natural way to encode the grid structure of form documents and allow more control over how information propagates across the nodes. The graph representation in SotA studies captures each token on the page as a node, and the adjacency structure often follows one of the below paradigms (Wang et al., 2022):

In **KNN** graphs, each node is connected to its K closest neighbors on the page (see Figure 1(b)). Due to the dependency on the parameter K, it is difficult to guarantee optimal density (or optimal sparsity) throughout the graph.

Line of Sight (LOS) graphs connect each node to other nodes within its "line of sight" (see Figure 1(c)). This guarantees that nodes that are adjacent on the page are connected, but LOS graphs can still introduce edges that don't carry meaningful information., e.g. the connection between "SUPPLIER" and "Pugh" in Figure 1(c).

The **β-skeleton** graph can be thought of as a "ball-of-sight" approach (Wang et al., 2022) that removes some of the edges from LOS by favoring proximity (see Figure 1(e)). This approach has been adopted in line and paragraph-detection models (Wang et al., 2022; Liu et al., 2022) as well as form extraction models (Lee et al., 2021, 2023).

As can be seen in Figure 1(e), even though the β-skeleton graph captures more meaningful relationships compared to KNN and LOS graphs, it can still produce unhelpful edges, e.g. the edge between "PERSONNEL" and "Market". This is because, like KNN and LOS, β-skeleton graphs favor proximity over alignment, whereas alignment is not only one of the core principles of layout design, but is crucial to maintaining the grid structure in forms (Kimball, 2013). An alternative to LOS, namely Axis-aligned LOS (Figure 1(d)) has been proposed to capture alignments, but as Davis et al. (2021) argued, it is not effective for form understanding tasks due to its over-sparsity.

As shown by studies such as Liu et al. (2022), the β-skeleton graph can be enhanced by the addition of redundant (or "multi-hop") edges. We adapt this idea to the Axis-aligned LOS structure, and propose a new graph structure which we name AligNet (see Figure 1(f)). AligNet captures short- and long-range dependencies by adding multi-hop edges to the Axis-aligned LOS structure, which helps the graph honor alignment as well as proximity in modeling the layout of a page. We demonstrate AligNet's ability to capture the global structure of each page using a community detection method. Additionally, when equipped with a graph convolution network, the AligNet structure can route messages between nodes that are meaningfully associated, such as field names and field values. Our proposed graph learning approach, AliGATr, couples the AligNet representation with a layout generation objective, which leads to competitive performance on VrFU tasks, including key information extraction and relation extraction. To balance the calibration and grounding tradeoff, AliGATr uses a generative architecture, but uses a Pointer mechanism (See et al., 2017) to strictly produce output tokens that are extracted from the input. Furthermore, because of its generative objective, AliGATr's logits are better calibrated than encoder-based models.

In summary, AliGATr addresses the previously mentioned shortcomings of SotA approaches using the below solutions: 1) Lack of attention to alignments is resolved using an alignment-based structure (i.e. AligNet). 2) Over-sparsity of alignment-based structures is addressed by the introduction of redundant edges in AligNet. 3) Poor calibration is addressed by following a generative objective. 4) Poor grounding is addressed by using a Pointer mechanism.

The following sections present our methodology and experimental results.

3 Methodology

In this section, we describe our proposed graph representation for documents (AligNet), as well as our proposed model architecture (AliGATr). Figure 2 shows the overall flow of pre-training and fine-tuning steps.

3.1 AligNet

We model each document as an undirected graph $G = (V, E)$, where each node x_i represents a token on the page, and two nodes x_i and x_j are adjacent if their bounding boxes are horizontally or vertically aligned. We define alignment between x_i and x_j as $\exists c \in \{\text{left}, \text{center}, \text{right}, \text{top}, \text{middle}, \text{bottom}\} : |b_i^c - b_j^c| < \mathcal{D}$, where b_i^c represents the coordinates of the bounding box of x_i, and $\mathcal{D}$ is a threshold that is expressed as a percentage of page width/height and can be tuned as a hyperparameter[1]. Figure 2(a) shows a small snippet of a form. In 2(b), the form has been converted into an AligNet graph (see Figure 7(b) for a more substantive example).

We represent each node x_i by it embedding vector $\mathbf{x}_i$ that is generated by a language model such as RoBERTa (Liu et al., 2019). An edge between x_i and x_j is represented by the below attribute vector:

$$\mathbf{e}_{i,j} = [-|b_i^{\text{left}} - b_j^{\text{left}}|, -|b_i^{\text{right}} - b_j^{\text{right}}|, -|b_i^{\text{top}} - b_j^{\text{top}}|, -|b_i^{\text{bottom}} - b_j^{\text{bottom}}|, b_i^{\text{height}} - b_j^{\text{height}}, \frac{b_i^{\text{width}}}{\text{numchars}(x_i)} - \frac{b_j^{\text{width}}}{\text{numchars}(x_j)}]$$

Note that the first four elements show the negative absolute distance (i.e. proximity) between the four coordinates of the bounding boxes[2]. The fifth element shows the difference in the heights of the two bounding boxes, and the last element shows the difference in their average width per character. In order to avoid the need to resample all images to be of the same size, we normalize all coordinates based on the width and height of each page.

In addition to edge attributes, we also assign a label to each edge, which reflects one of the

[1]Alternatively, alignments can be found using more sophisticated clustering-based or convolutional methods, but based on our experiments, the simple threshold-based approach yields comparable performance.

[2]Using the negative distance is a naive but effective way to represent proximity. Our experiments demonstrated that other methods such as using the reciprocal or log-order of distance were not as effective. See Appendix E for more details.

6 possible types of alignment between the adjacent nodes, namely: left, center, right, top, middle, or bottom-aligned. The label also reflects whether the source node is located "before" the target node in the reading order, i.e. whether the source node is to the left or top of the target node. This yield 12 possible classes. In Figure 2(b), the edge between "Market" and "Facts" represents two directed edges: a `bottom-before` edge from "Market" to "Facts", and a `bottom-after` edge from "Facts" to "Market". This means that "Market" and "Facts" are bottom-aligned and "Market" comes before "Facts".

3.2 AliGATr

The AliGATr architecture is composed of three modules, inspired by Lee et al. (2023): a serializer, a GCN, and a decoder. During pre-training, the serializer arranges the nodes into a sequence, and the GCN generates node embeddings using generative objectives. During fine-tuning, the decoder predicts node labels (KIE) or links (RE).

3.2.1 Serialization

The serializer uses a simple heuristic to order the tokens in a sequence. Using the top-left coordinates of each bounding box, the serializer orders the tokens in a left-to-right and top-to-bottom sequence. For English-language documents, this is meant to mimic reading order, even though it is a noisy approximation.[3] Figure 2(c) illustrates the serialized graph for the example in 2(b). This serialized sequence is used to traverse the AligNet graph during pre-training, as described in the next section.

3.2.2 Generative pre-training

After the serializer determines the ordering of nodes, the AligNet graph is fed into a GCN. We use a Relational Graph Attention Network (RGAT) (Busbridge et al., 2019) as the GCN backbone. The model follows an auto-regressive layout-generation objective coupled with a segmentation objective. We describe these objectives below.

Layout generation objectives: First, we add a dummy start node x_0 to the graph that is not connected to any other nodes. This node functions as the `<start>` token for our generative task. At each timestamp t, the model masks the bounding box coordinates of nodes $x_t, x_{t+1}, \cdots, x_T$ as well as any edges adjacent to them. The model then generates representations for nodes $x_0, x_1, \cdots, x_T$, namely, $\mathbf{h}_0, \mathbf{h}_1 \cdots, \mathbf{h}_T \in \mathbb{R}^d$ where d is the hidden dimension. Figure 2(d) shows the node representations at $t = 2$ for the example graph. Nodes with dashed borders have their bounding boxes masked and edges removed. Using these representations, the model optimizes two objectives: 1) Given $\mathbf{h}_{0:t-1}$, the model "picks" the next node $\mathbf{h}_t$ from the set of remaining nodes $\mathbf{h}_{t:T}$, where the ordering is determined by the serializer, and all positional information (i.e. bounding box coordinates) are masked for $\mathbf{h}_{t:T}$. 2) Given the predicted next node $\mathbf{h}_{\hat{t}}$, the model predicts the edges between $\mathbf{h}_{\hat{t}}$ and the subgraph composed of $\mathbf{h}_{0:t-1}$. This is akin to presenting the tokens in a random order to the model, and encouraging the model to put the layout back together token by token, placing each new token in its proper position with regards to previous tokens on the page. Since the model has access to the token identities, it is not performing *text* generation, but *layout* generation. This allows the model to learn layout-aware representations without having to fulfill the text generation objective, which is a data- and parameter-intensive task. Below, we describe the model's two objectives:

First, given the node embeddings $\mathbf{h}_0 \cdots \mathbf{h}_{t-1}$, we use a pointer mechanism (See et al., 2017) to select the next node from the set of remaining nodes. The pointer is implemented as scaled dot-product attention between the sequence embedding $\mathbf{h}_{0:t-1}$ and the remaining embeddings $\mathbf{h}_{t:T}$. The node whose embedding has the highest attention score is predicted as the next node:

$$\begin{aligned}
&\mathbf{h}^{(0:t-1)} = W^{(1)}\mathbf{h}_{0:t-1}^{\top},\ \mathbf{h}^{(t:T)} = W^{(2)}\mathbf{h}_{t:T}^{\top} \\
&\alpha_{\hat{t}} = \text{softmax}\Big(\frac{(\sum_{k=0}^{t-1}\mathbf{h}_k^{(0:t-1)})\mathbf{h}^{(t:T)}}{\sqrt{d}}\Big) \\
&j = \arg\max_i\{\alpha_{\hat{t},i}; i \in \{0, 1, \cdots, T-t-1\}\} \\
&\mathbf{h}_{\hat{t}} = \mathbf{h}_{j+t+1}
\end{aligned}$$

where $W^{(1)}$ and $W^{(2)} \in \mathbb{R}^{d\times d}$ are weight matrices, $\alpha_{\hat{t}}$ are the attention weights, $j+t+1$ is the index of the node with the highest attention weight, and $\mathbf{h}_{\hat{t}}$ is the representation of that node. In Figure 2(e), the next node is correctly picked as $\mathbf{h}_2$.

To calculate the next node prediction loss, we follow See et al. (2017) and use Negative Log Likelihood as pointer loss: $\mathcal{L}_t^{\text{NODE}} = \frac{-\log \alpha_t}{\log(T-t)}$, where

[3]There are many cases where this heuristic won't work, e.g. on multi-column pages. However, using a noisy heuristic yields a more robust model, as it prevents the model from leveraging exact reading order information (Zhang et al., 2023).

α_t is the attention weight of the correct next node x_t, and the $1/\log(T-t)$ factor is used to lower the penalty for cases when the selection of the next node has a higher degree of freedom and is therefore a more difficult task.

Once the next node $\mathbf{h}_{\hat{t}}$ is determined, the model predicts its adjacencies to the subgraph composed of previous nodes. Inspired by You et al. (2018) we model this task as predicting the adjacency vector $a_{\hat{t}}$, which is a binary vector of size t where $a_{\hat{t},k} = 1$ if x_k and $x_{\hat{t}}$ are adjacent, and $a_{\hat{t},k} = 0$ otherwise. The model predicts $a_{\hat{t}}$ based on the attention between $\mathbf{h}_{0:t-1}$ and $\mathbf{h}_{\hat{t}}$. The adjacency loss is calculated using the binary cross entropy between the predicted adjacency vector $a_{\hat{t}}$ and the true vector a_t:

$$\mathbf{h}'^{(0:t-1)} = W^{(3)}\mathbf{h}_{0:t-1}^{\top},\ a_{\hat{t}} = \frac{\mathbf{h}_{\hat{t}}^{\top}\mathbf{h}'^{(0:t-1)}}{\sqrt{d}}$$
$$\mathcal{L}_t^{\mathrm{ADJ}} = \mathrm{BCE}(a_{\hat{t}}, a_t)$$

where $W^{(3)} \in \mathbb{R}^{d\times d}$ is a weight matrix. In Figure 2(f), the adjacency vector of x_2 is predicted as $[0,1]$ indicating that no edge exists between x_0 and x_2, but an edge exists between x_1 and x_2. Note that this adjacency vector only determines the existence of an edge, without sensitivity to directionality. Directionality is only reflected in the attributes and labels of the unmasked edges.

Segmentation objective: In addition to the node and adjacency prediction objectives, the model predicts the boundaries of various segments on the page. This is to encourage the model to find groupings of tokens that correspond to an entity. Most OCR engines provide segment boundaries based on the spacing between the tokens on the page. The model uses this information to predict whether a given token marks the beginning of a new segment[4]. The loss is modeled as a simple binary cross entropy: $\mathcal{L}_t^{\mathrm{SEG}} = -(s_t \log s_{\hat{t}} + (1-s_t)\log(1-s_{\hat{t}}))$, where $s_{\hat{t}} = w^{(4)}\mathbf{h}_{\hat{t}}^{\top}$ is the predicted binary segmentation flag for $x_{\hat{t}}$, s_t is the true flag, and $w^{(4)} \in \mathbb{R}^{1\times d}$ is a weight vector. Figure 2(g) shows that the segmentation flag for x_2 has been predicted as 1, indicating that it signals the start of a new segment.

The total pretraining loss at step t is calculated as the sum of node prediction, adjacency prediction, and segment boundary prediction losses: $\mathcal{L}_t = \mathcal{L}_t^{\mathrm{NODE}} + \mathcal{L}_t^{\mathrm{ADJ}} + \mathcal{L}_t^{\mathrm{SEG}}$. AliGATr uses these objectives to learn layout-aware representations for each node x_t.

3.2.3 Fine-tuning

At fine-tuning time, we use the segmentation flags learned by the model to identify the boundaries of each entity. This reduces the complexity of the downstream KIE and RE tasks since they both rely on entity-grouping to produce accurate output. Figures 2 (h) and (i) show the pre-segmentation and post-segmentation stages of the example graph, respectively.

We implement two fine-tuning heads, each corresponding to one of the two target tasks, i.e. KIE and RE. KIE from forms can be modeled as a node classification problem, and RE can be modeled as a link prediction problem.

The KIE node classification head uses the ordering created by the serializer to generate a sequence representation using an RNN (Hochreiter and Schmidhuber, 1997) (Figure 2(i)). The sequence can then be used to predict I-O-B tags for each token. Finally, the KIE classification loss, $\mathcal{L}^{\mathrm{CLF}}$, can be calculated as cross entropy loss between the predicted and true classes. The introduction of the RNN is important as it models the sequentiality of the input more effectively than the graph. However, if its representations deviate too much from the those created by the graph, they can "unlearn" certain semantic information. Inspired by Yao et al. (2024), we introduce an auxiliary co-distillation loss that keeps the RNN representations ($\mathbf{h}^{\mathrm{RNN}}$) and the graph representations ($\mathbf{h}^{\mathrm{GNN}}$) close to each other:

$$\mathcal{L}^{\mathrm{CoD}} = \sum_{i=1}^{N} \mathrm{CL}(\mathbf{h}_i^{\mathrm{GNN}}, \tilde{\mathbf{h}}_i^{\mathrm{RNN}}) + \mathrm{CL}(\mathbf{h}_i^{\mathrm{RNN}}, \tilde{\mathbf{h}}_i^{\mathrm{GNN}})$$
$$\mathcal{L}^{\mathrm{KIE}} = \mathcal{L}^{\mathrm{CLF}} + \mathcal{L}^{\mathrm{KIE}} + \mathcal{L}^{\mathrm{SEC}}$$

where CL stands for the contrastive loss described in Tian et al. (2020) and $\tilde{\cdot}$ is the stop-gradient operator, which freezes the corresponding representation. The model continues to learn segmentation during fine-tuning via the segmentation loss $\mathcal{L}^{\mathrm{SEG}}$. Figure 2(j) shows the final output of the KIE classification head.

The RE link prediction head does not require serialization, as it simply uses the dot product of two node representations $\mathbf{h}_i^{\mathrm{GNN}}$ and $\mathbf{h}_j^{\mathrm{GNN}}$ to predict whether an edge exists between them. The RE loss, $\mathcal{L}^{\mathrm{RE}}$ is calculated based on the binary cross

[4]Studies such as Huang et al. (2022) and Luo et al. (2023) also use segment-level information. Note that AliGATr only uses segment-level information at training time and does not expect this information during inference.

Dataset	# Train	# Test	Tasks	# Classes
FUNSD (Jaume et al., 2019)	149	50	KIE, RE	4
SROIE (Huang et al., 2019)	626	347	KIE	4
CORD (Park et al., 2019)	800	100	KIE, RE	30
BuDDIE (Zmigrod et al., 2024)	1,172	332	KIE	69

Table 1: Statistics about four datasets that cover KIE and RE tasks. Note that we only list the tasks that are used in our experiments. "# Classes" indicates the number of entity classes used in the KIE task.

entropy between predicted and true edges. Figure 2 (k) shows the output of the RE head. Note that the RE head would be able to identify relations between nodes and segments, even if they are not aligned. The alignment edges are only used during pre-training to create layout-aware node representations. See Figure 5 for a examples of unaligned RE results.

4 Experiments

In this section we describe the datasets and baselines used in our experiments. Other experimental settings are described in Appendix A.

4.1 Datasets

We use four multimodal form understanding datasets that cover KIE and RE tasks. **CORD** and **SROIE** are collections of retail receipts. **FUNSD** includes research and advertising forms sampled from the RVL-CDIP dataset (Harley et al., 2015), and **BuDDIE** is a collection of business entity filings collected from various US states. Table 1 shows high-level statistics about each dataset.

4.2 Baselines

We use four SotA baselines in multimodal form understanding. **LayoutLMv3** (Huang et al., 2022) is a transformer-based model that uses vision, spatial, and text signal to model multimodal documents. By abandoning a complex Region-Proposal Network in favor of a simple patch-based vision encoder, LayoutLMv3 reduces the number of parameters compared to LayoutLMv2 (Xu et al., 2021), while achieving superior performance on the KIE task[5]. **GraphLayoutLM** (Li et al., 2023) enhances LayoutLMv3 with a graph component that maps the relative positioning of various nodes with regards to each other, improving performance on KIE. **GeoLayoutLM** (Luo et al., 2023) adds geometric constraints to LayoutLMv3 and demonstrates SotA performance on both the KIE and RE tasks. Lastly, **FormNetv2**[6] (Lee et al., 2023) uses a β-skeleton graph and a Graph Convolution Network to model visually rich forms. In contrast to previous models, FormNetv2 does not rely on segment-level bounding boxes, and relies entirely on token-level presentations. The model outperforms LayoutLMv3 on KIE despite a 44% reduction in model size.

[5]We do not cover the performance on tasks that are out of scope for AliGATr, such as document classification and visual question answering

5 Results and discussion

5.1 Performance on KIE and RE tasks

Table 2 shows the performance of AliGATr and four baselines on the multimodal form datasets. As mentioned in Section 4.2 three of the four baseline models rely on segment-level bounding boxes, while FormNetv2 and AliGATr do not rely on segment-level bounding boxes during inference, and only use token-level bounding boxes. To make the comparisons consistent across all models, we have reported the performances using token as well as segment bounding boxes (see caption for more detail). Despite a 30% reduction in size compared to the smallest baseline (FormNetv2), AliGATr performs on par with or better than the SotA models on the KIE task. The model falls short of SotA on BuDDIE, which has the largest number of classes and is composed of denser documents (business entity filings).

Table 3 shows the performance of AliGATr and two other baselines on the RE task. Once again, AliGATr matches or outperforms SotA models despite having 60% fewer parameters than the smaller baseline (LayoutLMv3LARGE).

5.2 Calibration

There are two aspects of calibration that facilitate straight-through-processing of documents in downstream applications. The first is the confidence of the model with regards to the output. Under-confidence and over-confidence are both problem-

[6]Note that we do not include Multimodal LLMs such as UReader (Ye et al., 2023) or DocLLM (Wang et al., 2023) because they have not yet achieved SotA performance on form processing tasks. OCR-free models such as UDOP (Tang et al., 2023) and mPLUG-DocOwl1.5 (Hu et al., 2024) are excluded for the same reason.

[7]The BuDDIE dataset does not provide segment level bboxes. Therefore only token-level results are reported.

[8]Some experimental results are missing for GraphLayoutLM because the authors were not able to recreate the baselines reported by Li et al. (2023), and are therefore only reporting the numbers disclosed in the original paper.

[9]FormNetv2 is not Open Source. Therefore only the results reported in Lee et al. (2023) are included.

Model	**Modalities**	**# Params**	**Pre-training dataset size**	**FUNSD**	**CORD**	**SROIE**	**BuDDIE**[7]
LayoutLMv3LARGE	T+L+I	357M	11M	82.53/92.08	95.92/97.46	94.96/98.63	83.42
GraphLayoutLMLARGE[8]	T+L+I	372M	11M	-/**94.39**	-/97.75	-/-	-
GeoLayoutLM	T+L+I	399M	11M	84.40/92.86	96.57/97.71	95.04/98.70	**84.86**
FormNetv2[9]	T+L+I	204M	11M	**86.35**/92.51	97.37/97.70	98.31/-	-
AliGATr	T+L	145M	1M	86.31/92.95	**97.48/97.83**	**98.57/98.78**	81.85

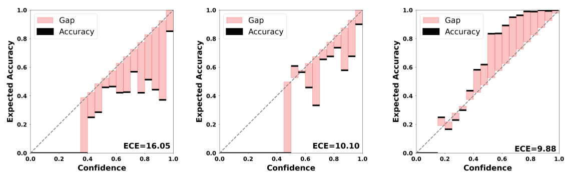
Table 2: Performance on the KIE task. "T", "L", and "I" stand for text, layout, and image. The performance is reported as `token/segment`, where `segment` indicates performance when segment-level bounding boxes are available at test time, and `token` indicates performance when only token-level bounding boxes are available.

Model	**Modalities**	**# Params**	**FUNSD**	**CORD**
LayoutLMv3LARGE	T+L+I	357M	80.35	99.64
GeoLayoutLM	T+L+I	399M	89.45	**100.00**
AliGATr	T+L	145M	**89.50**	**100.00**

Table 3: Performance on the RE task. The numbers reported for the CORD dataset correspond to the "REaKV" task mentioned in Luo et al. (2023).

atic as they do not reflect the model's true performance. The second, and arguably more important aspect is the consistency of the confidence gap. If a model is consistently over or under-confident, it is much easier to set a fixed threshold beyond which the model's outputs can be trusted.

Figure 3 shows the confidence versus performance plot for LayoutLMv3LARGE, GeoLayoutLM, and AliGATr, when finetuned on the FUNSD dataset. As the Figure shows, AliGATr's output probabilities are better calibrated, and do not exhibit the over-confident trend that is observed in the baselines. As indicated by the lower ECE, AliGATr is also more consistent in its confidence gap, and a confidence threshold of 0.8 and above yields near perfect performance.



(a) LayoutLMv3 (b) GeoLayoutLM (c) AliGATr

Figure 3: Calibration plots and ECE measures for AliGATr versus two baselines. All models have been finetuned for the KIE task on the FUNSD dataset.

6 Ablation and sensitivity studies

In this section, we investigate how three components of our proposed pipeline contribute to downstream performance. Due to infrastructure limitations, all of the studies reported here are based on a toy pre-training dataset of 30K examples sampled from OCR-IDL (Biten et al., 2022). The gains/drops in performance are statistically significant at $p < 0.005$, based on the paired-bootstrap test proposed by Berg-Kirkpatrick et al. (2012), with $b = 10^2$. Therefore we expect the trends to hold for larger pre-training datasets.

Approach		**KIE**	**RE**
Graph Structure	β-Skeleton	50.89	64.52
	AligNet	**51.30**	**73.12**
Serialization	No node prediction	50.44	70.01
	No edge labels	49.29	68.43
	Order-invariant labels	51.03	71.26
	Order-sensitive labels	**51.30**	**73.12**
Full Gen.	$N = 1$	51.30	**73.12**
Skip Gen.	$N = 5$	50.94	65.14
	$N = 10$	50.39	64.97
	$N = 20$	50.50	64.09
Chunk Gen.	$M = 20$	51.28	73.07
	$M = 50$	**51.31**	72.98
	$M = 100$	51.29	73.01

Table 4: The impact of graph representation, edge representation, and generation methods on the KIE and RE tasks (F1 performance on the FUNSD dataset).

6.1 Graph structure

As mentioned in Section 2.2, β-skeleton graphs are a common choice in graph-based models. The top segment of Table 4 shows the performance of the β-skeleton graph against the AligNet structure. The β-skeleton graph slightly underperforms AligNet on the KIE task, but has an even larger gap on the RE task. The latter is expected, as alignments often play a major role in indicating semantic correspondence between field names and values. This demonstrates the effectiveness of the AligNet structure in modeling form understanding tasks. For further analysis on this topic, see Appendix C.

6.2 Serialization

As discussed in Section 3.2.1, the serializer orders the nodes in left-to-right and top-to-bottom fashion. This has an impact on two components of

AliGATr, namely the next node predictor (which is designed to predict the next node in the sequence according to the serializer's ordering), and the edge labels (which are determined based on the relative position of two nodes on the page).

The second segment of Table 4 shows the impact of ablating these components. Without node prediction, both KIE and RE tasks suffer. Removing edge labels has an even bigger impact on performance, even though order-invariant labels recover some of the performance. The best performance belongs to a model that has order-sensitive edge labels (i.e. 12 classes, as described in Section 3.1), which is therefore the model used in our final experiments. For a deeper analysis on how edge representations can impact downstream performance, see Appendix E.

6.3 Generation regime

Lastly, we analyze the impact of the generation regime on downstream performance. In the default auto-regressive setting, every token is generated one by one. This can be costly if the number of tokens on a page is large. SotA models such as LayoutLMv3LARGE cap the sequence length at 512 tokens which poses a risk for text-heavy pages. Furthermore during pre-training the model might not be exposed to sections that usually appear at the bottom of the page, e.g. footers or page numbers. Instead of truncating the input during pre-training, we experiment with two alternatives. In **Skip Generation**, the model generates every N tokens. In **Chunk Generation**, the model generates a randomly sampled subsequence of length M from each page. The last segment of Table 4 shows the model's performance in these settings. In the default setting (titled "Full Gen."), the model has the highest performance on RE and close to highest performance on KIE. The performance suffers when switching to Skip Generation, especially for RE. This may be attributed to the disjointedness of generations, because skipping over $N-1$ tokens can obscure the relationship between neighboring tokens. This problem is largely addressed by Chunk Generation, as is evident from the model's performance, even when M is small. Given the competitive performance of Chunk Generation with $M = 20$, we selected this setting to perform pre-training. A possible risk of Chunk Generation is that the robustness of output probabilities might be undermined, but, as presented in Section 5.2, the model has better calibrated output than baselines. The effectiveness of Chunk Generation further demonstrates the robustness and efficiency of AliGATr's learning objectives.

7 Conclusion and future work

In this paper, we presented AliGATr, a layout generation technique for form understanding that is competitive with SotA on Key Information Extraction and Relation Extraction tasks, using 30% fewer parameters and 11x fewer training examples. We showed how, despite using the spatial and textual modalities alone, and relying on subsequence generation, the model produces better-calibrated probabilities. In future studies, we hope to investigate AliGATr's effectiveness in other adjacent tasks that lend themselves to graph-based representations, such as document classification, page segmentation, and structure extraction.

8 Limitations

Any methodology that relies on alignment-based signal, such as the AligNet structure, is at risk of failing to recognize noisy alignments, e.g. on skewed or tilted pages. We rely on the accuracy of OCR software to recognize the angle at which the document is presented, which may not always be reliable. However, as with segment/line detection, the rotation detection capability of modern OCR software has substantially improved.

As mentioned in Section 3.2.1, we use a serializer that orders the nodes in a left-to-right and top-to-bottom fashion to mimic reading order. The ordering might not work for documents with complex layouts (such as multi-column pages), but since the ordering is only needed for pre-training, the risk is minimal at inference time. More consequentially, the ordering might not generalize to many non-English languages.

Lastly, the pre-training dataset is sampled from the IDL collection[10], which covers enterprise documents from a limited set of industries. As discussed in Nourbakhsh et al. (2024), this can lead to poor OOD performance without further fine-tuning or continued pre-training.

9 Acknowledgements

The authors would like to thank Ivan Brugere, Hao-Ren Yao, Pranav Shetty, Natraj Raman, and Jill Fain Lehman for their insightful feedback.

Armineh Nourbakhsh's work is supported by JP-Morgan Chase & Co. This paper was prepared

[10]https://www.industrydocuments.ucsf.edu

References

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 993–1003.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Ali Furkan Biten, Rubèn Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. 2022. Ocr-idl: Ocr annotations for industry document library dataset.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *International Conference on Learning Representations*.

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. 2019. Relational graph attention networks. *arXiv preprint arXiv:1904.05811*.

Brian L. Davis, B. Morse, Brian L. Price, Chris Tensmeyer, and Curtis Wigington. 2021. Visual fudge: Form understanding via dynamic graph editing. In *IEEE International Conference on Document Analysis and Recognition*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. 2021. Unidoc: Unified pretraining framework for document understanding. In *Advances in Neural Information Processing Systems*.

Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2021. BROS: A layout-aware pre-trained language model for understanding documents. *CoRR*, abs/2108.04539.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. 2024. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Miles A Kimball. 2013. Visual design principles: An empirical study of design lore. *Journal of Technical Writing and Communication*, 43(1):3–41.

David G. Kirkpatrick and John D. Radke. 1985. A framework for computational morphology. *Machine Intelligence and Pattern Recognition*, 2:217–248.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Andrea Lancichinetti and Santo Fortunato. 2009. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5).

Chen-Yu Lee, Chun-Liang Li, Chu Wang, Renshen Wang, Yasuhisa Fujii, Siyang Qin, Ashok Popat, and Tomas Pfister. 2021. ROPE: Reading order equivariant positional encoding for graph-based document information extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 314–321, Online. Association for Computational Linguistics.

Chen-Yu Lee, Chun-Liang Li, Hao Zhang, Timothy Dozat, Vincent Perot, Guolong Su, Xiang Zhang, Kihyuk Sohn, Nikolay Glushnev, Renshen Wang, Joshua Ainslie, Shangbang Long, Siyang Qin, Yasuhisa Fujii, Nan Hua, and Tomas Pfister. 2023. FormNetV2: Multimodal graph contrastive learning for form document information extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9011–9026, Toronto, Canada. Association for Computational Linguistics.

David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 665–666.

Qiwei Li, Zuchao Li, Xiantao Cai, Bo Du, and Hai Zhao. 2023. Enhancing visually-rich document understanding via layout structure modeling. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia Workshops*, MMAsia '23 Workshops, New York, NY, USA. Association for Computing Machinery.

Yulin Li, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. 2021. Structext: Structured text understanding with multi-modal transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 1912–1920, New York, NY, USA. Association for Computing Machinery.

Shuang Liu, Renshen Wang, Michalis Raptis, and Yasuhisa Fujii. 2022. Unified line and paragraph detection by graph convolutional networks. In *Document Analysis Systems*, pages 33–47, Cham. Springer International Publishing.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. 2023. Geolayoutlm: Geometric pre-training for visual information extraction. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Armineh Nourbakhsh, Sameena Shah, and Carolyn Rose. 2024. Towards a new research agenda for multimodal enterprise document understanding: What are we missing? In *Proceedings of the Findings of the 62nd Annual Meeting of the Association for Computational Linguistics*, Bangkok, Thailand. Association for Computational Linguistics.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: a consolidated receipt dataset for post-ocr parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.

Pritika Ramu, Sijia Wang, Lalla Mouatadid, Joy Rimchala, and Lifu Huang. 2024. re^2: Region-aware relation extraction from visually rich documents. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8731–8747, Mexico City, Mexico. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive representation distillation. In *International Conference on Learning Representations*.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong

Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2023. Docllm: A layout-aware generative language model for multimodal document understanding. *arXiv preprint arXiv:2401.00908*.

R. Wang, Y. Fujii, and A. C. Popat. 2022. Post-ocr paragraph recognition by graph convolutional networks. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2533–2542, Los Alamitos, CA, USA. IEEE Computer Society.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online. Association for Computational Linguistics.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1192–1200, New York, NY, USA. Association for Computing Machinery.

Hao-Ren Yao, Luke Breitfeller, Aakanksha Naik, Chunxiao Zhou, and Carolyn Rose. 2024. Multi-scale contrastive knowledge co-distillation for event temporal relation extraction.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. 2023. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv preprint arXiv:2310.05126*.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pages 5708–5717. PMLR.

Chong Zhang, Ya Guo, Yi Tu, Huan Chen, Jinyang Tang, Huijia Zhu, Qi Zhang, and Tao Gui. 2023. Reading order matters: Information extraction from visually-rich documents by token path prediction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13716–13730, Singapore. Association for Computational Linguistics.

Yue Zhang, Zhang Bo, Rui Wang, Junjie Cao, Chen Li, and Zuyi Bao. 2021. Entity relation extraction as dependency parsing in visually rich documents. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2759–2768, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ran Zmigrod, Dongsheng Wang, Mathieu Sibue, Yulong Pei, Petr Babkin, Ivan Brugere, Xiaomo Liu, Nacho Navarro, Antony Papadimitriou, William Watson, Zhiqiang Ma, Armineh Nourbakhsh, and Sameena Shah. 2024. Buddie: A business document dataset for multi-task information extraction.

A Experimental settings

For AligNet, we set the alignment parameter $\mathcal{D} = 0.01$. This means that if the horizontal or vertical distance between a pair of nodes is smaller than 1% of the width or height of the page, the two nodes are considered aligned (and thus adjacent).

Our GCN backbone is a 2-layer RGAT, implemented by the `Pytorch Geometric` library.[11] We use a 2-layer unidirectional LSTM (Hochreiter and Schmidhuber, 1997) as the RNN module.

We use a sample of 1 million documents from the OCR-IDL dataset (Biten et al., 2022) to pre-train the model. During pre-training, we initialize the token embeddings using RoBERTaBASE(Liu et al., 2019). We use a batch size of 1, a learning rate of $5e{-}6$, the AdamW optimizer (Loshchilov and Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.999)$, and train the model for 1 epoch. During fine-tuning for the KIE and RE tasks, we use a batch size of 16, learning rate of $1e{-}5$, the AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, and train the model for 1000 epochs. We set the negative sampling rate for the co-distillation loss as 5.

B Qualitative examples

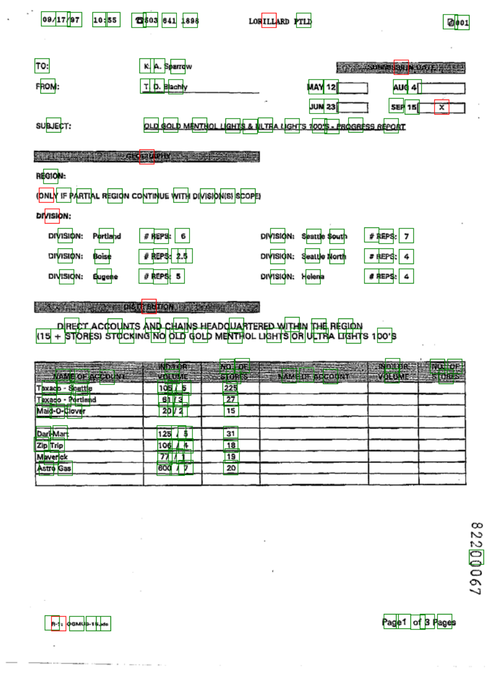
Figure 4 shows the performance of AliGATr on the KIE task on two samples from the FUNSD dataset. As the figure shows, AliGATr struggles with tokens that do not have a clear alignment with other elements of a similar class. This can be attributed to AliGATr's weaker text backbone compared to other SotA models.

Figure 5 shows the performance of AliGATr on the RE task on two samples from the FUNSD dataset. AliGATr recovers all edges that correspond to aligned elements. The performance is lower for elements that are not horizontally or vertically aligned. Notably, the rate of false negatives is higher than false positives.

C Constructing β-skeleton graphs

As mentioned in Section 2.2, the β-skeleton graph is favored in many graph-based form understand-

[11]https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.RGATConv.html


09/17/97 10:55 ☎503 641 1898 LORILLARD PTLD ⎙001

TO: K. A. Sparrow

FROM: T. D. Blachly

SUBMISSION DATE:

MAY 12		AUG 4	
JUN 23		SEP 15	X

SUBJECT: OLD GOLD MENTHOL LIGHTS & ULTRA LIGHTS 100'S - PROGRESS REPORT

GEOGRAPHY

REGION:

(ONLY IF PARTIAL REGION CONTINUE WITH DIVISION(S) SCOPE)

DIVISION:

DIVISION: Portland	# REPS: 6		DIVISION: Seattle South	# REPS: 7
DIVISION: Boise	# REPS: 2.5		DIVISION: Seattle North	# REPS: 4
DIVISION: Eugene	# REPS: 5		DIVISION: Helena	# REPS: 4

DISTRIBUTION

DIRECT ACCOUNTS AND CHAINS HEADQUARTERED WITHIN THE REGION
(15 + STORES) STOCKING NO OLD GOLD MENTHOL LIGHTS OR ULTRA LIGHTS 100'S

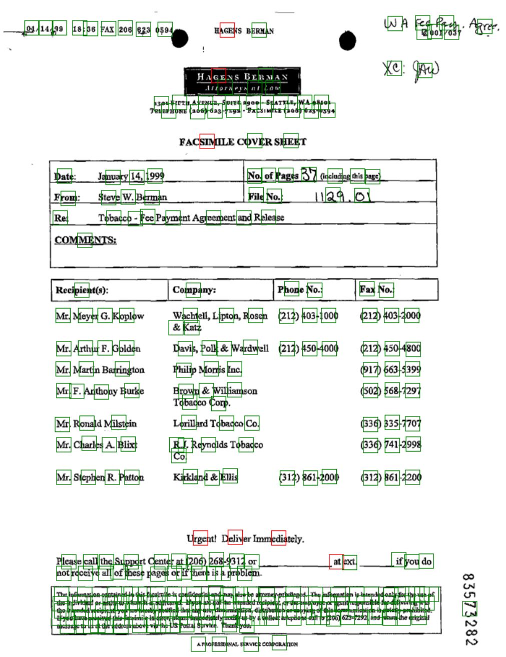
NAME OF ACCOUNT	IND/TOR VOLUME	NO. OF STORES	NAME OF ACCOUNT	IND/TOR VOLUME	NO. OF STORES
Taxaco - Seattle	105 / 5	225			
Taxaco - Portland	61 / 3	27			
Maid-O-Clover	20 / 2	15			
Dari-Mart	125 / 5	31			
Zip Trip	106 / 4	18			
Maverick	77 / 1	19			
Astro Gas	600 / 7	20			

82200067

R-1: OGMUD-15.xls

Page1 of 3 Pages

(a) KIE results on form with tabular segments


01/14/99 18:36 FAX 206 623 0594 HAGENS BERMAN ⎙001/037

WA Fee Pay. Agree.

XC: JAW

HAGENS BERMAN
Attorneys at Law
1301 FIFTH AVENUE, SUITE 2900 · SEATTLE, WA 98101
TELEPHONE (206) 623-7292 · FACSIMILE (206) 623-0594

FACSIMILE COVER SHEET

Date:	January 14, 1999	No. of Pages 37 (including this page)	
From:	Steve W. Berman	File No.:	1129.01
Re:	Tobacco - Fee Payment Agreement and Release		
COMMENTS:			

Recipient(s):	Company:	Phone No.:	Fax No.:
Mr. Meyer G. Koplow	Wachtell, Lipton, Rosen & Katz	(212) 403-1000	(212) 403-2000
Mr. Arthur F. Golden	Davis, Polk & Wardwell	(212) 450-4000	(212) 450-4800
Mr. Martin Barrington	Philip Morris Inc.		(917) 663-5399
Mr. F. Anthony Burke	Brown & Williamson Tobacco Corp.		(502) 568-7297
Mr. Ronald Milstein	Lorillard Tobacco Co.		(336) 335-7707
Mr. Charles A. Blixt	R.J. Reynolds Tobacco Co.		(336) 741-2998
Mr. Stephen R. Patton	Kirkland & Ellis	(312) 861-2000	(312) 861-2200

Urgent! Deliver Immediately.

Please call the Support Center at (206) 268-9312 or ______________ at ext. ______ if you do not receive all of these pages or if there is a problem.

The information contained in this facsimile is confidential and may also be attorney-privileged. The information is intended only for the use of the individual or entity to whom it is addressed. If you are not the intended recipient, or the employee or agent responsible for delivering it to the intended recipient, you are hereby notified that any use, dissemination, distribution or copying of this communication is strictly prohibited. If you have received this facsimile in error, please immediately notify us by a collect telephone call to (206) 623-7292, and return the original message to us at the address above via the US Postal Service. Thank you.

A PROFESSIONAL SERVICE CORPORATION

83573282

(b) KIE results on sparse form

Figure 4: KIE results on two samples from the FUNSD dataset. Green boxes show correct predictions and red boxes show incorrect predictions.

ing models. Consistent with Lee et al. (2023), we set $\beta = 1$, making our graph a Gabriel graph—a subset of Delaunay triangulation (Kirkpatrick and Radke, 1985). Unlike typical point-based β-skeleton graphs, our approach involves bounding box β-skeleton graphs. We use each token's four coordinates (top-left, top-right, bottom-left, bottom-right) as vertices and employ Delaunay triangulation from `scipy.spatial`[12] to construct the graph, as shown in figure 6(a). We then remove all internal connections within a bounding box. While a strict β-skeleton graph would exclude any edges with vertices inside the circle formed by those edges, this results in excessive sparsity due to token proximity. To address this, we maintain all edges but simplify by collapsing the four corners to the center of each bounding box, as demonstrated in figure 6(b).

D Community based self-supervision

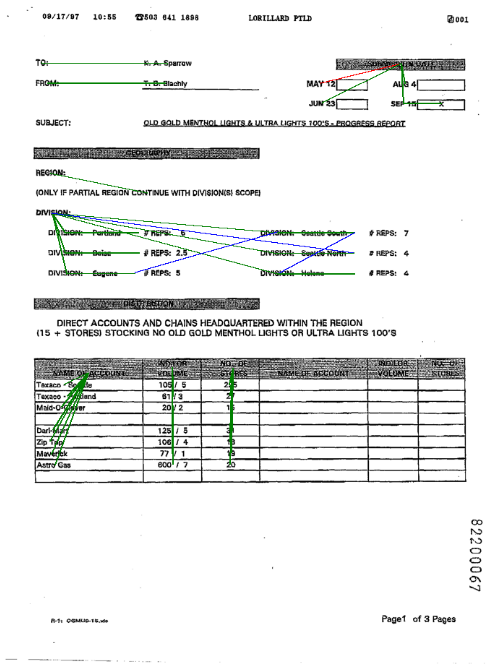
D.1 Community detection

The AligNet representation can be used to segment the page based on alignments, using a graph segmentation algorithm. These algorithms are designed to find cliques, partitions, or communities (i.e. locally dense segments) within the graph. Among such algorithms, the Leiden community detection method (Traag et al., 2019) is a particularly useful approach, because: 1) It focuses on maximizing the modularity of a network, which is defined as the density of intra-community edges compared to inter-community edges. This is congruent with the segmentation objective in AligNet, since high-density areas of a page can indicate a segment (see Figure 7(c)). 2) The greedy implementation of the Leiden method leads to log-linear complexity in most experimental settings (Lancichinetti and Fortunato, 2009), which offers a runtime advantage. 3) The recursive nature of the Leiden method exempts it from requiring a pre-determined number of communities. The algorithm stops when the overall modularity of the network can no longer be improved beyond a minimum threshold.

We use the implementation of the algorithm offered by the `NetworkX` python library[13]. Edges

[12]https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Delaunay.html

[13]https://networkx.org/documentation/networkx-3.1/reference/algorithms/generated/networkx.algorithms.community.louvain.louvain_communities.html


09/17/97 10:55 ☎503 641 1898 LORILLARD PTLD 001

TO: K. A. Sparrow

FROM: T. B. Blachly

SUBMISSION DATE

MAY 12		AUG 4	
JUN 23		SEP 15	X

SUBJECT: OLD GOLD MENTHOL LIGHTS & ULTRA LIGHTS 100'S - PROGRESS REPORT

GEOGRAPHY

REGION:

(ONLY IF PARTIAL REGION CONTINUE WITH DIVISION(S) SCOPE)

DIVISION:

DIVISION: Portland	# REPS: 6	DIVISION: Seattle South	# REPS: 7
DIVISION: Boise	# REPS: 2.5	DIVISION: Seattle North	# REPS: 4
DIVISION: Eugene	# REPS: 5	DIVISION: Helena	# REPS: 4

DISTRIBUTION

DIRECT ACCOUNTS AND CHAINS HEADQUARTERED WITHIN THE REGION
(15 + STORES) STOCKING NO OLD GOLD MENTHOL LIGHTS OR ULTRA LIGHTS 100'S

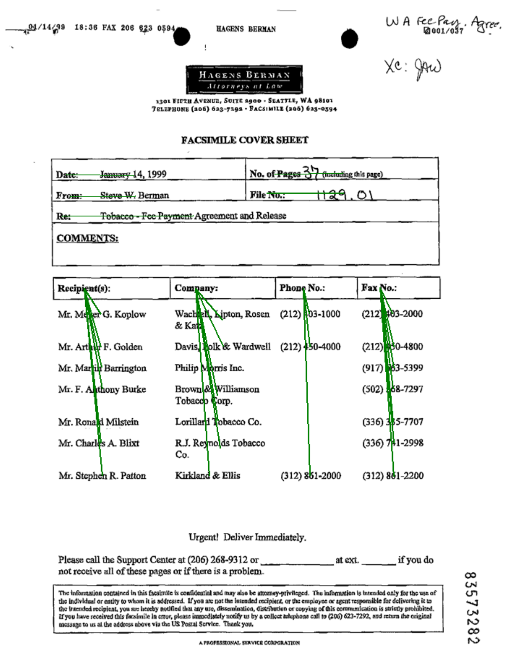
NAME OF ACCOUNT	IND/LOR VOLUME	NO. OF STORES	NAME OF ACCOUNT	IND/LOR VOLUME	NO. OF STORES
Texaco - Seattle	105 / 5	25			
Texaco - Portland	61 / 3	27			
Maid-O-Clover	20 / 2	15			
Dari-Mart	125 / 5	31			
Zip Trip	106 / 4	18			
Maverick	77 / 1	19			
Astro Gas	600 / 7	20			

82200067

R-1: OGMUS-15.xls

Page1 of 3 Pages

(a) RE results on form with tabular segments


01/14/99 18:36 FAX 206 623 0594 HAGENS BERMAN 001/037

W A Fee Pay. Agree.

Xc: JAW

HAGENS BERMAN
Attorneys at Law

1301 FIFTH AVENUE, SUITE 2900 · SEATTLE, WA 98101
TELEPHONE (206) 623-7292 · FACSIMILE (206) 623-0594

FACSIMILE COVER SHEET

Date: January 14, 1999	No. of Pages 37 (including this page)
From: Steve W. Berman	File No.: 1129.01
Re: Tobacco - Fee Payment Agreement and Release	
COMMENTS:	

Recipient(s):	Company:	Phone No.:	Fax No.:
Mr. Meyer G. Koplow	Wachtell, Lipton, Rosen & Katz	(212) 403-1000	(212) 403-2000
Mr. Arthur F. Golden	Davis, Polk & Wardwell	(212) 450-4000	(212) 450-4800
Mr. Martin Barrington	Philip Morris Inc.		(917) 663-5399
Mr. F. Anthony Burke	Brown & Williamson Tobacco Corp.		(502) 568-7297
Mr. Ronald Milstein	Lorillard Tobacco Co.		(336) 335-7707
Mr. Charles A. Blixt	R.J. Reynolds Tobacco Co.		(336) 741-2998
Mr. Stephen R. Patton	Kirkland & Ellis	(312) 861-2000	(312) 861-2200

Urgent! Deliver Immediately.

Please call the Support Center at (206) 268-9312 or ______________ at ext. ______ if you do not receive all of these pages or if there is a problem.

The information contained in this facsimile is confidential and may also be attorney-privileged. The information is intended only for the use of the individual or entity to whom it is addressed. If you are not the intended recipient, or the employee or agent responsible for delivering it to the intended recipient, you are hereby notified that any use, dissemination, distribution or copying of this communication is strictly prohibited. If you have received this facsimile in error, please immediately notify us by a collect telephone call to (206) 623-7292, and return the original message to us at the address above via the US Postal Service. Thank you.

A PROFESSIONAL SERVICE CORPORATION

83573282

(b) RE results on sparse form

Figure 5: RE results on two samples from the FUNSD dataset. Green links show correct predictions. Red links show false negatives. Blue links show false positives.

are weighted according to the following distance calculation:

$$w_{e_{ij}} = \frac{\mathcal{W}(e_{ij})}{\sqrt{(b_i^{\text{center}} - b_j^{\text{center}})^2 + (b_i^{\text{middle}} - b_j^{\text{middle}})^2}} \quad (1)$$

where $\mathcal{W}(e_{ij})$ is a weighing hyperparameter. This weighing scheme allows the Leiden algorithm to consider distance and proximity when identifying the segments. Once the algorithm converges, each node in the AligNet graph v_i is assigned a community label c_i.

D.2 Community-aware GAT

The classic GAT model (Veličković et al., 2018) learns the representation of each node by convolving its original representation with those of its neighbors. In the GATv2 convolution (Brody et al., 2022), this is designed as:

$$\mathbf{h}_i' = \alpha_{i,i}\mathbf{\Theta}_s\mathbf{h}_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\mathbf{\Theta}_t\mathbf{h}_j \quad (2)$$

The attention score $\alpha_{i,j}$ is calculated as:

$$\alpha_{i,j} = \quad (3)$$

$$\frac{\exp(\mathbf{f}^\top \mathrm{LeakyReLU}(\mathbf{\Theta}_s\mathbf{h}_i + \mathbf{\Theta}_t\mathbf{h}_j + \mathbf{\Theta}_e\mathbf{e}_{i,j}))}{\sum_{k\in\mathcal{N}(i)\cup\{i\}} \exp(\mathbf{f}^\top \mathrm{LeakyReLU}(\mathbf{\Theta}_s\mathbf{h}_i + \mathbf{\Theta}_t\mathbf{h}_k + \mathbf{\Theta}_e\mathbf{e}_{i,k}))} \quad (4)$$

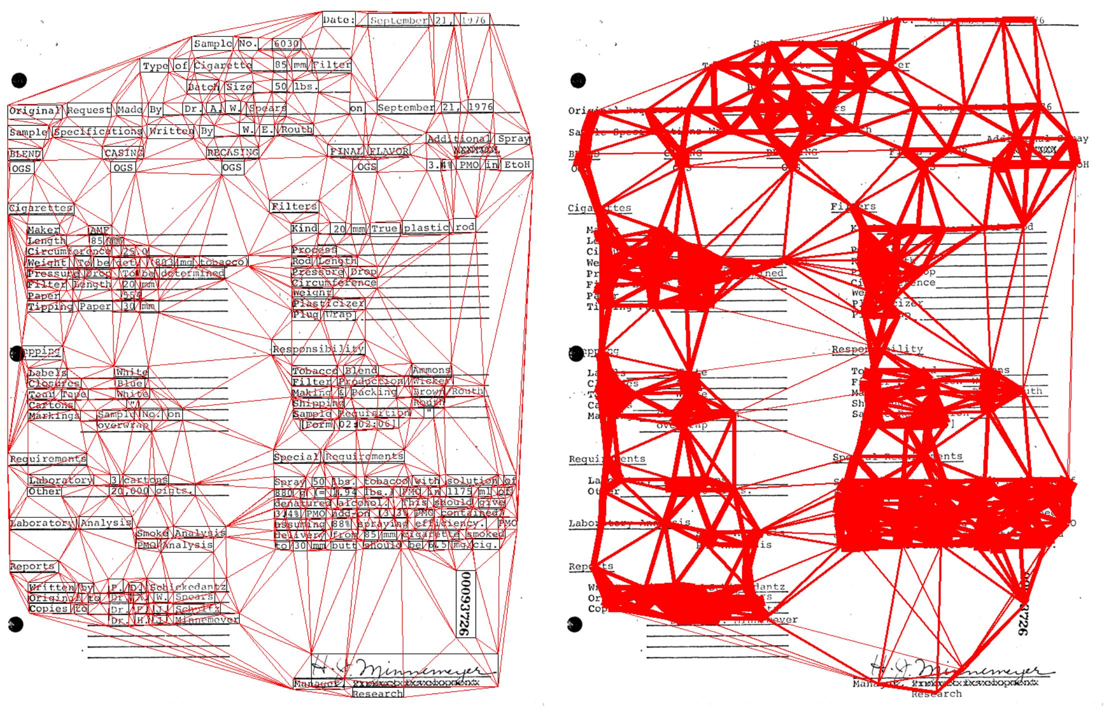
where $\mathbf{f}$ is an affine parameter, $\mathcal{N}(i)$ represents the set of nodes adjacent to x_i and $\mathbf{\Theta}_s$, $\mathbf{\Theta}_t$, and $\mathbf{\Theta}_e$ are weight parameters corresponding to source, target, and edge representations, respectively.

For Leiden community detection, we set $\mathcal{W}(e_{ij})$ to 1 for horizontal edges and to $\frac{1}{16}$ for vertical edges. This encourages the algorithm to prioritize the merging of nodes along horizontal edges, which leads to the creation of horizontally-aligned segments. This is consistent with the general reading order of English-language documents (left-to-right, then top-to-bottom), but can be adjusted for other languages.

In a similar fashion, a community detection algorithm can be used to segment the β-skeleton graph (see Figures 10 and 11).

D.3 Graph Representations and Number of Communities

In this section, we investigate the effect of splitting each page into a predetermined set of communities. Our ablation experiments aim to compare different graph representations, focusing on β-skeleton graphs and AligNet graphs. All experiments use $\beta = 1$ (Gabriel graph) and a very small pre-training dataset of 149 documents from FUNSD (Jaume



(a) Point-based β-skeleton graph on bbox coordinates (b) β-skeleton graph merging internal bbox connections

Figure 6: Construction of a β-skeleton graph on a sample form. First, create a point-based β-skeleton graph with the 4 corners of each bounding box as vertices (a). Next, remove internal connections within each bounding box and merge the 4 vertices into the centroid (b). The width of the edges in (b) indicates the edge weight: shorter edges have higher weights.

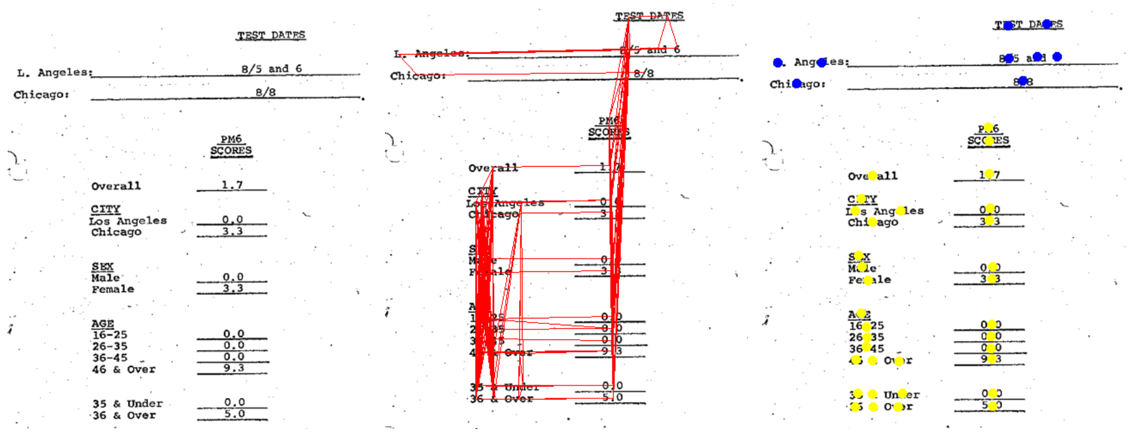
	Graph Structure	
# of communites	β-skeleton	AligNet
baseline	16.56	**19.53**
1	0.0	11.10
2	**20.50**	13.45
4	16.68	17.03
8	15.48	-
16	16.24	-

Table 5: Ablation results on graph representations and community numbers. For the β-skeleton graph, 2 communities per document yield the best results. For the AligNet graph, the baseline with the Louvian algorithm's optimal community number performs best. All numbers reflect F1 performance on the FUNSD dataset.

et al., 2019). We also explored the effects of different community detection configurations using the Leiden method, focusing on variations in the resolution parameter γ and explicitly setting community sizes.

Adjusting the resolution parameter γ allowed us to control community detection granularity, impacting both community count and size. Higher γ values led to more but smaller communities. However, γ adjustments did not yield consistent results across different graph structures. Therefore, we predetermined the number of communities by initially assigning nodes to a set number of groups, allowing the algorithm to refine these into fixed community counts without exceeding the predefined limits.

For the β-skeleton graph, setting all nodes into a single community prevented the model from converging. As shown in Figure 8, models with two communities achieved higher and more stable F1 scores throughout the epochs. Conversely, increasing the number of communities to 4, 8, or 16 generally decreased performance. Notably, models with 4 or 8 communities showed slower convergence rates, whereas configurations with 16 communities unexpectedly improved convergence compared to the previous two. Further analysis of community size distributions, shown in Figure 9, reveals that setting the community number to 16 aligns the distribution closely with the baseline model, resulting in similar performance trends. Moreover, Figure 9 also indicates that the maximum viable number of communities for the β-skeleton graph is 12, high-



	TEST DATES
L. Angeles:	8/5 and 6
Chicago:	8/8

	PM6 SCORES
Overall	1.7
CITY	
Los Angeles	0.0
Chicago	3.3
SEX	
Male	0.0
Female	3.3
AGE	
16-25	0.0
26-35	0.0
36-45	0.0
46 & Over	9.3
35 & Under	0.0
36 & Over	5.0

(a) Raw form (b) AligNet graph (c) Leiden communities

Figure 7: Visual illustration of how the AligNet representation can enable page segmentation. The example document is excerpted from FUNSD (Jaume et al., 2019).

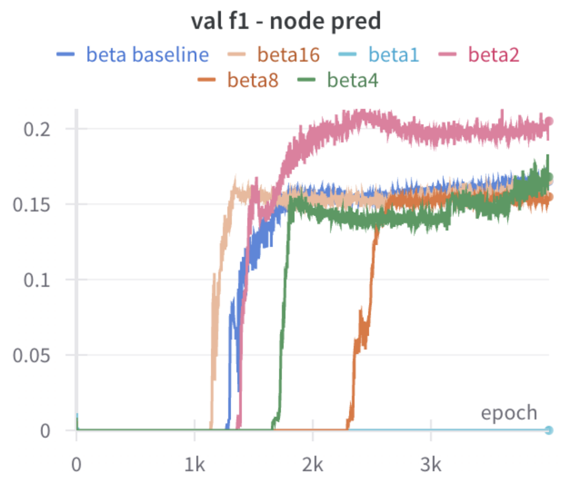


Figure 8: Training curves for β-skeleton with different community sizes. The number after "beta" in the legend indicates the number of communities per document. β-skeleton with 2 communities yields the best results. Graphs with 4 and 8 communities have lower convergence rates compared to the baseline and the graph with 16 communities, despite similar F1.

lighting the graph's limitations due to sparsity.

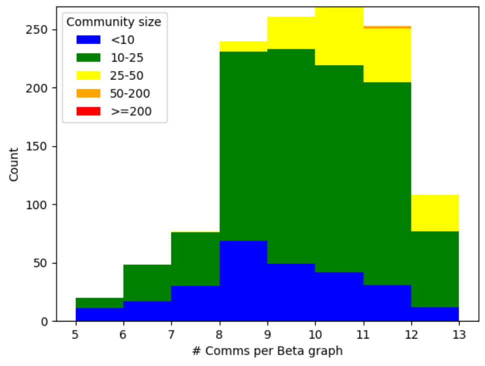
For AligNet, the results in Table 5 show a different pattern compared to the β-skeleton graph. The baseline model, which uses the Leiden algorithm to determine the optimal community numbers, achieves the best F1 scores. However, explicitly setting a lower number of communities results in lower F1 scores. The lowest F1 score is observed when all nodes are grouped into a single community. These findings suggest that AligNet performs optimally with multiple, smaller-sized communities.

Our findings indicate that community information enhances modeling for both AligNet and β-skeleton graphs, each benefiting from different community configurations. The β-skeleton graph performs optimally with larger communities, effectively utilizing extensive neighboring information, while the AligNet graph is more effective with finer community granularity. For the β-skeleton graph, smaller communities do not ensure accurate separation into distinct blocks in uniformly dense documents, as shown in Figure 10(a). Conversely, utilizing larger communities reduces the focus on smaller clusters and enhances the separation of specific tokens like "questions" and "answers", thereby improving the task performance as shown in Figure 11.

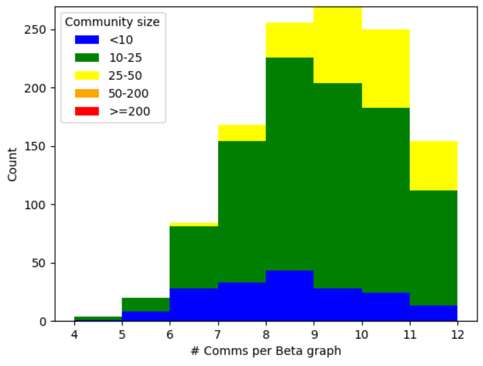
E Order Sensitive Edge Representations

Building on the findings of Lee et al. (2021), we explored the impact of reading order on graph structures in our ablation experiments. All experiments in this section are base on a small pre-training dataset of 149 documents from FUNSD (Jaume et al., 2019).

Raw Distance: We modified our approach by using raw distances instead of the original edge definition shown in the edge representation described in Section 3.1. The distance between nodes x_i and



(a) β-skeleton community distribution, baseline



(b) β-skeleton community distribution, $\#$ of comm = 16

Figure 9: Cumulative counts for community size for β-skeleton graphs. In (a), for graphs with 5 communities (left-most column), approximately half of them have less than 10 nodes (blue segment at the bottom), while the other half have 10-25 nodes (green segment). The typical community size for β-skeleton graphs is 10-25 nodes. In (b), when explicitly setting the maximum community size to 16, the distribution trend is similar to (a).

x_j is represented as:

$$\mathbf{e}_{i,j} = [b_i^{\text{left}} - b_j^{\text{left}}, b_i^{\text{right}} - b_j^{\text{right}}, b_i^{\text{top}} - b_j^{\text{top}}, b_i^{\text{bottom}} - b_j^{\text{bottom}}] \quad (5)$$

We hypothesize that this raw distance can implicitly convey reading order, with negative values suggesting x_i precedes x_j, and positive values indicating the reverse.

Order-Sensitive Edge Label: We initially defined alignment edge labels as

$$\exists c \in \{\text{left}, \text{center}, \text{right}, \text{top}, \text{middle}, \text{bottom}\}$$

as described in Section 3.1. For our ablation experiments, we expanded these into twelve labels:

$$\exists c \in \{\text{left}_{pre}, \text{center}_{pre}, \text{right}_{pre}, \text{top}_{pre}, \text{middle}_{pre}, \text{bottom}_{pre}, \text{left}_{post}, \text{center}_{post}, \text{right}_{post}, \text{top}_{post}, \text{middle}_{post}, \text{bottom}_{post}\}$$

The label is determined by the summation of vectors in 5: negative sums result in one of the first six labels (which x_i precedes x_j), while positive sums assign one of the latter six, indicating x_i follows x_j. This adjustment aims to further encode the reading order into the graph.

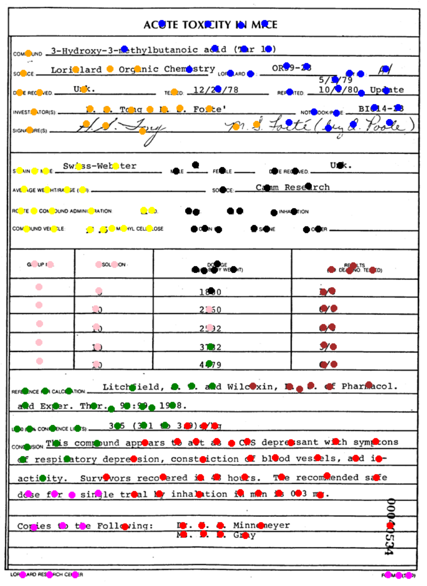
	w/ Raw Distance	w/o Raw Distance
Order-invariant labels	16.49	15.08
Order-sensitive labels	**18.29**	12.82

Table 6: Ablation study results for edge representations in AligNet, showing F1 performance on the FUNSD dataset. Utilizing raw distance and explicit order-sensitive labels improves model performance.

Incorporating raw distance as implicit order-sensitive edge representations improves model performance, as shown in Table 6 and Figure 12. However, adding explicit order-sensitive edge labels did not consistently enhance performance. The edge labels did improve convergence when combined with raw distance, but using them alone resulted in slower convergence.

F Edge Types

Of the two graph structures, we also perform ablation studies to determine which edge types are useful for the task. In this section, all experiments use segment loss instead of community loss, and are based on a small pre-training dataset consisting of 149 documents from FUNSD (Jaume et al., 2019). In AligNet, we classified edges into four categories: horizontal-long, horizontal-short, vertical-long, and vertical-short. We set a threshold $\lambda = 0.3$ for short edges, including those shorter than 30% of the page width or height, and $\lambda = 0.5$ for long edges, which are longer than 50% of the page dimensions. Edges not meeting these criteria were excluded. For the β-skeleton graph, which primar-



ACUTE TOXICITY IN MICE

COMPOUND	3-Hydroxy-3-methylbutanoic acid (Tar 10)
SOURCE	Lorillard Organic Chemistry — LORILLARD #: OR79-28 — A/
DATE RECEIVED	Unk. — TESTED: 12/2[illegible]/78 — REPORTED: 5/1/79, 10/[illegible]/80 Update
INVESTIGATOR(S)	H. J. Tong M. J. Forte' — NOTEBOOK/PAGE: BI[illegible]14-[illegible]
SIGNATURE(S)	[illegible] M. J. Forte (by J. Poole)
STRAIN NAME	Swiss-Webster — MALE — FEMALE — DATE RECEIVED: Unk.
AVERAGE WEIGHT/RANGE (GM)	SOURCE: Camm Research
ROUTE OF COMPOUND ADMINISTRATION	[illegible] — INHALATION
COMPOUND VEHICLE	[illegible] METHYL CELLULOSE — CORN OIL — SALINE — OTHER

GROUP NO.	SOLUTION	DOSAGE (MG/KG BODY WEIGHT)	RESULTS (NO. DEAD/NO. TESTED)
[illegible]	[illegible]	18[illegible]0	[illegible]/[illegible]
[illegible]	[illegible]0	2[illegible]50	[illegible]/[illegible]
[illegible]	[illegible]0	2[illegible]92	[illegible]/[illegible]
[illegible]	[illegible]0	37[illegible]2	[illegible]/[illegible]
[illegible]	[illegible]0	4[illegible]79	[illegible]/[illegible]

REFERENCE FOR CALCULATION: Litchfield, J. T. and Wilcoxin, F. J. of Pharmacol. and Exper. Ther. 9[illegible]:99 1948.

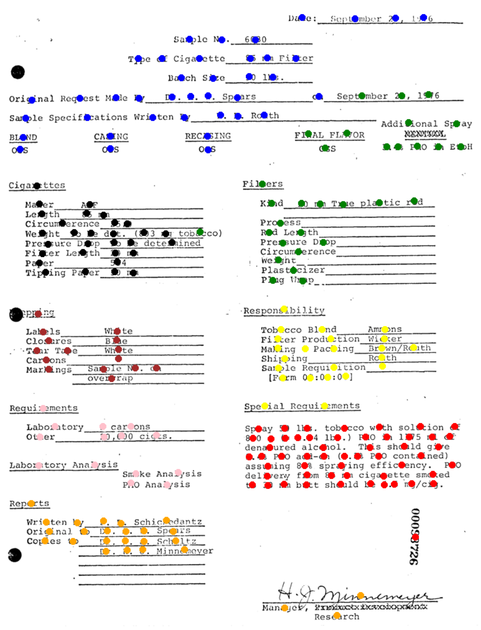
LD50 (95% CONFIDENCE LIMITS): 3[illegible]5 (3[illegible]1 to 3[illegible]9) g/kg

CONCLUSION: This compound appears to act as a CNS depressant with symptoms of respiratory depression, constriction of blood vessels, and inactivity. Survivors recovered in 48 hours. The recommended safe dose for a single trial by inhalation in man is 0[illegible]3 mg.

Copies to the Following: Dr. [illegible]. [illegible]. Minnemeyer
Mr. [illegible]. [illegible]. Gray

0004[illegible]534

LORILLARD RESEARCH CENTER — FORM [illegible]



Date: September 2[illegible], 1976

Sample No. 6[illegible]80

Type of Cigarette [illegible]5 mm Filter

Batch Size [illegible]0 lbs.

Original Request Made By Dr. [illegible]. [illegible]. Spears on September 2[illegible], 1976

Sample Specifications Written By [illegible]. [illegible]. Routh

Additional Spray ~~XXXXXXX~~

BLEND	CASING	RECASING	FINAL FLAVOR	
OCS	OCS	OCS	OCS	[illegible] PHO in EtOH

Cigarettes

Maker	AMF
Length	85 mm
Circumference	25 [illegible]
Weight	To be det. (8[illegible]3 mg tobacco)
Pressure Drop	To be determined
Filter Length	[illegible]0 mm
Paper	5[illegible]4
Tipping Paper	[illegible]0 mm

Filters

Kind	[illegible]0 mm True plastic rod
Process	
Rod Length	
Pressure Drop	
Circumference	
Weight	
Plasticizer	
Plug Wrap	

Shipping

Labels	White
Closures	Blue
Tear Tape	White
Cartons	
Markings	Sample No. on overwrap

Responsibility

Tobacco Blend	Ammons
Filter Production	Wicker
Making & Packing	Brown/Routh
Shipping	Routh
Sample Requisition	

[Form 0[illegible]:0[illegible]:0[illegible]]

Requirements

Laboratory	[illegible] cartons
Other	[illegible]0,000 cigts.

Laboratory Analysis

Smoke Analysis
PHO Analysis

Special Requirements

Spray [illegible] lbs. tobacco with solution of 8[illegible]0 [illegible] ([illegible].[illegible]4 lbs.) PHO in 1[illegible]75 ml of denatured alcohol. This should give [illegible].[illegible]% PHO add-on ([illegible].[illegible]% PHO contained) assuming 8[illegible]% spraying efficiency. PHO delivery from 85 mm cigarette smoked to [illegible] mm butt should be [illegible].[illegible] mg/cig.

Reports

Written by	[illegible]. [illegible]. Schickedantz
Original to	Dr. [illegible]. [illegible]. Spears
Copies to	Dr. [illegible]. [illegible]. Schultz
	Dr. [illegible]. [illegible]. Minnemeyer

0008[illegible]726

H. J. Minnemeyer
Manager, ~~Product Development~~
Research

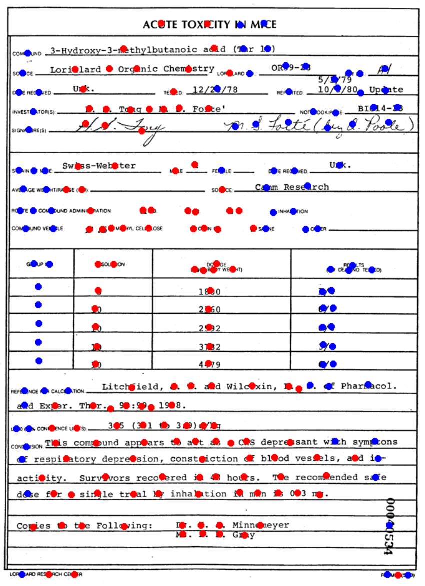
(a) β-skeleton communities for a form with tabular segments
(b) β-skeleton communities for form with nested segments

Figure 10: β-skeleton communities baseline. In uniformly dense documents, communities are not properly separated (a), whereas distinct communities are formed in less dense and more structured documents (b).

ily comprises short edges, we categorized edges into three groups: horizontal, vertical, and others, using the same threshold criteria as AligNet for orientation determination.

In our analysis, we used the AligNet graph as the baseline for comparison. The results, depicted in Table 7 (Experiment 1-7), indicate that horizontal short edges significantly outperform all other types. Vertical short edges from AligNet notably reduced performance relative to the baseline, while vertical long edges and horizontal long edges showed performance similar to the baseline. For the β-skeleton graph, horizontal edges provided a slight improvement over the baseline.

We further tested combinations of edge types, maintaining the same experimental settings and considering the union of overlapping edge types. As also shown in Table 7 (Experiment 8-11), the best results within the AligNet graph were achieved by combining horizontal short and horizontal long edges, with performance trends similar to those of horizontal short edges alone. Conversely, combinations of horizontal long and vertical short edges yielded the poorest results, indicating their limited utility. In the β-skeleton graph (Experiment 12-14), adding horizontal short edges enhanced performance. Those results show the importance of horizontal short edges in effectively connecting segment information in this task.



ACUTE TOXICITY IN MICE

COMPOUND	3-Hydroxy-3-methylbutanoic acid (Tar 10)
SOURCE	Lorillard Organic Chemistry
LORILLARD	OR79-28
DATE RECEIVED	Unk.
TESTED	12/2?/78
REPORTED	5/?/79 10/?/80 Update
INVESTIGATOR(S)	H. S. Tong M. D. Forte'
NOTEBOOK/PAGE	BI014-28
SIGNATURE(S)	[illegible]
STRAIN NAME	Swiss-Webster
DATE RECEIVED	Unk.
SOURCE	Camm Research
ROUTE OF COMPOUND ADMINISTRATION	INHALATION
COMPOUND VEHICLE	METHYL CELLULOSE CORN OIL SALINE OTHER

GROUP NO.	SOLUTION	DOSAGE (MG/KG BODY WEIGHT)	RESULTS (NO. DEAD/NO. TESTED)
	5	1800	1/?
	10	2160	0/?
	10	2592	0/?
	10	3732	3/?
	10	4479	6/?

REFERENCE FOR CALCULATION: Litchfield, J. T. and Wilcoxin, F. J. of Pharmacol. and Exper. Ther. 96:99, 1948.

LD50 (95% CONFIDENCE LIMITS): 3?5 (3?1 to 3?9) g/kg

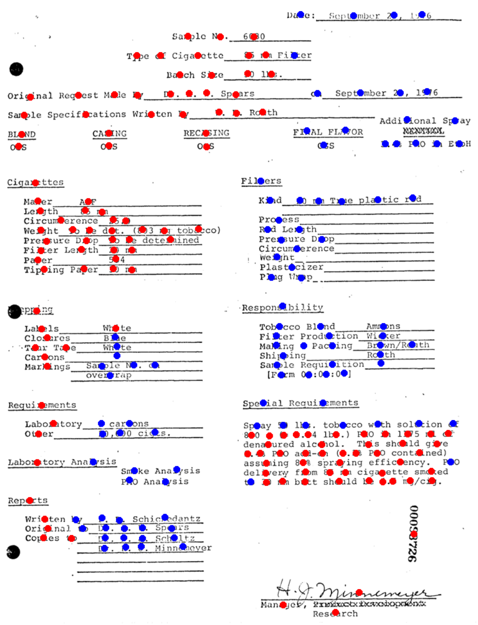
CONCLUSION: This compound appears to act as a CNS depressant with symptoms of respiratory depression, constriction of blood vessels, and inactivity. Survivors recovered in 48 hours. The recommended safe dose for a single trial by inhalation in man is 0.03 mg.

Copies to the Following: Dr. ?. ?. Minnemeyer
Mr. ?. ?. Gray

0004534

LORILLARD RESEARCH CENTER FORM ?(5-?0)

(a) β-skeleton communities with #comms = 2 for a form with tabular segments



Date: September 2?, 1976

Sample No. 6?80

Type of Cigarette 85 mm Filter

Batch Size ?0 lbs.

Original Request Made By Dr. ?. ?. Spears on September 2?, 1976

Sample Specifications Written By ?. ?. Roth

BLEND	CASING	RECASING	FINAL FLAVOR	Additional Spray ~~XXXXXXX~~
OCS	OCS	OCS	OCS	?.?% PMO in EtOH

Cigarettes

Maker	AMF
Length	85 mm
Circumference	25.0
Weight	To be det. (8?3 mg tobacco)
Pressure Drop	To be determined
Filter Length	20 mm
Paper	5?4
Tipping Paper	50 mm

Filters

Kind	20 mm Tube plastic rod
Process	
Rod Length	
Pressure Drop	
Circumference	
Weight	
Plasticizer	
Plug Wrap	

Shipping

Labels	White
Closures	Blue
Tear Tape	White
Cartons	
Markings	Sample No. on overwrap

Responsibility

Tobacco Blend	Ammons
Filter Production	Wicker
Making & Packing	Brown/Roth
Shipping	Roth
Sample Requisition	

[Form 0?:0?:0?]

Requirements

Laboratory	? cartons
Other	?0,000 cigs.

Laboratory Analysis

Smoke Analysis
PMO Analysis

Special Requirements

Spray 50 lbs. tobacco with solution of 800 g (? ?.?4 lbs.) PMO in 1175 ml of denatured alcohol. This should give ?.?% PMO add-on (?.?% PMO contained) assuming 80% spraying efficiency. PMO delivery from 85 mm cigarette smoked to 30 mm butt should be ?.? mg/cig.

Reports

Written by	?. ?. Schickedantz
Original to	Dr. ?. ?. Spears
Copies to	Dr. ?. ?. Schultz
	Dr. ?. ?. Minnemeyer

00058726

H. J. Minnemeyer
Manager, ~~Product Development~~
Research

(b) β-skeleton communities with #comms = 2 for a form with nested segments

Figure 11: β-skeleton communities with number of communities = 2. The visualizations show the communities generally separating out the "question" type token on the left and the "answer" type tokens in the middle.

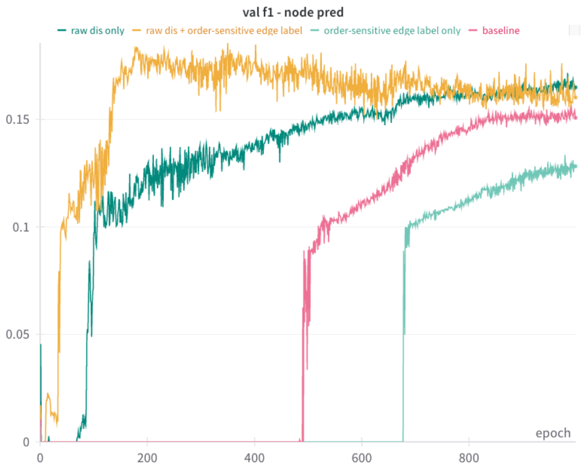


Figure 12: Training curves comparing different edge representations show that incorporating raw distance improves model performance and accelerates convergence compared to the baseline. Additionally, utilizing order-sensitive labels along with raw distance leads to even faster convergence. However, using order-sensitive labels alone results in slower convergence.

#	Edge Types							F1
	horizontal short	**horizontal long**	**vertical short**	**vertical long**	**beta horizontal**	**beta vertical**	**beta other**	
1	✓							**42.99**
2		✓						38.21
3			✓					30.05
4				✓				36.82
5					✓			39.17
6						✓		37.74
7							✓	35.99
8	✓	✓						41.24
9	✓		✓					34.53
10		✓	✓					29.38
11	✓			✓				35.46
12	✓				✓			36.92
13	✓				✓	✓	✓	37.92
14					✓	✓	✓	35.51

Table 7: Ablation results on edge types. The baseline F1 score is 35.81, using all edges in AligNet. Experiments 1-7 use single edge types, with horizontal short edges performing best. Experiments 8-11 test combinations of edge types for AligNet. Experiments 12-14 evaluate the effect of adding horizontal short edges to the β-skeleton graph.