FS-DAG: Few Shot Domain Adapting Graph Networks for Visually Rich Document Understanding

Amit Agarwal[1], Srikant Panda[2], Kulbhushan Pachuri[2]

[1] OCI, Oracle USA, [2] OCI, Oracle India

Correspondence: amit.h.agarwal@oracle.com

Abstract

In this work, we propose Few Shot Domain Adapting Graph (FS-DAG), a scalable and efficient model architecture for visually rich document understanding (VRDU) in few-shot settings. FS-DAG leverages domain-specific and language/vision specific backbones within a modular framework to adapt to diverse document types with minimal data. The model is robust to practical challenges such as handling OCR errors, misspellings, and domain shifts, which are critical in real-world deployments. FS-DAG is highly performant with less than 90M parameters, making it well-suited for complex real-world applications for Information Extraction (IE) tasks where computational resources are limited. We demonstrate FS-DAG's capability through extensive experiments for information extraction task, showing significant improvements in convergence speed and performance compared to state-of-the-art methods. Additionally, this work highlights the ongoing progress in developing smaller, more efficient models that do not compromise on performance.

1 Introduction

Recent advancements of Vision-Language Models (VLMs) (Zhang et al., 2024), Large Multimodal Models (LMMs) (Chen et al., 2024; Li et al., 2024), and Large Language Models (LLMs) (Brown, 2020; Touvron et al., 2023), have significantly enhanced performance across various natural language processing and computer vision tasks. Despite their success, these models are often computationally expensive, requiring substantial resources that are impractical for many real-world industrial applications (Sanh et al., 2019; Kaddour et al., 2023). Furthermore, their ability to adapt to specific domains, especially in the context of visually rich documents (VRDs) remains limited due to the high cost of pre-training and fine-tuning on domain-specific data (Li et al., 2021).

VRDs face challenges stemming from diverse layouts, domain-specific terminology, and text variations in style and size. OCR-free models tend to underperform compared to key-value models that utilize a separate OCR component, and even these models struggle with such variations. Large-scale models, with their monolithic architectures, often rely on vast data for domain adaptation, complicating their deployment. For example, state-of-the-art models like LayoutLM (Xu et al., 2020a) and its successors demand extensive fine-tuning for new domains, making their deployment both costly and time-consuming (Huang et al., 2022).

To address these issues, we introduce FS-DAG, a few-shot learning framework designed for domain-specific document understanding with less than 90M parameters. Few-shot learning methods have gained attention for their ability to train models with limited labeled data, which is crucial in industrial applications where data scarcity is a common challenge. Our approach leverages a modular architecture that integrates domain-specific and language-specific feature extractors, allowing FS-DAG to adapt quickly to new domains with minimal data, thereby overcoming the barriers associated with large-scale models (Lee et al., 2022).

Our approach emphasizes few-shot learning by leveraging Graph Neural Networks (GNNs) (Khemani et al., 2024; Wu et al., 2020) to enable rapid adaptation, robustness to OCR errors, and reduced latency in real-world applications. We provide empirical evidence of the model's performance through extensive experiments, showing significant improvements over larger methods with more than 100M parameters. To summarize, we make the following contributions to VRDU in a few-shot learning environment:

1. A modular framework for few-shot learning that efficiently combines domain-specific and language-specific textual and visual feature extractors in a graph-based architecture.

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 100–114
January 19–24, 2025.

2. We propose shared positional embedding & consistent reading order for GNN along with various training strategies for the model's robustness and effective adaptation with minimal data.

3. We provide comprehensive experimental results demonstrating that FS-DAG achieves state-of-the-art performance and robustness in few-shot learning scenarios while reducing latency and computational costs.

2 Related Work

The development of efficient and scalable NLP models has gained significant attention in recent years, particularly with the rise of LLMs such as GPT-3 (Brown, 2020), LlaMa (Touvron et al., 2023), Mixtrals (Jiang et al., 2024). While these models have achieved remarkable success in various tasks, their application in industrial settings remains challenging due to their high computational demands and difficulty in adapting to domain-specific tasks.

Recent work have focused on enhancing the efficiency of these models through techniques such as model distillation (Sanh et al., 2019), pruning (Cheng et al., 2024), and efficient fine-tuning methods like LoRA (Hu et al., 2022). These approaches aim to reduce the computational cost of LLMs while maintaining their performance, making them more suitable for deployment in resource-constrained environments.

In the context of VRDU, graph-based models have shown promise, particularly in capturing the complex relationships between textual and visual elements in documents. Models such as SDMGR (Sun et al., 2021), DocParser (Rausch et al., 2021), PICK (Yu et al., 2021) and others (Liu et al., 2019; Rastogi et al., 2020; Yao et al., 2021) leverage GNNs to improve IE from documents. However, these models often require large amounts of training data and are not designed for quick adaptation to new domains.

FS-DAG builds on these approaches by introducing a few-shot learning framework that can efficiently adapt to new document types with minimal data. This capability is particularly important in industrial applications, where labeled data is often limited, and the ability to quickly adapt to new domains is crucial. Additionally, FS-DAG addresses practical challenges such as robustness to OCR errors and domain shifts, which are common in real-world deployments.

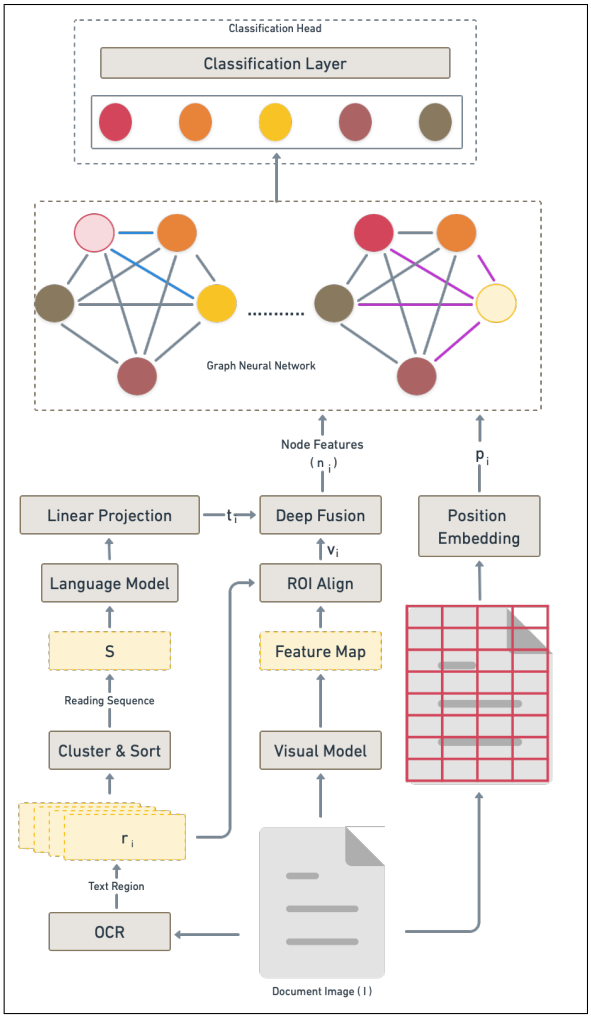

Figure 1: An illustration of the model architecture for FS-DAG. Given a document image (I); its text regions $\{r_i\}$ are extracted using an OCR engine. We cluster and sort the $\{r_i\}$ to create a reading sequence $\{s\}$; textual features $\{t_i\}$ are extracted using a linear projection layer on top of a pre-trained language model processing $\{s\}$. In contrast, visual features $\{v_i\}$ are extracted using ROI-Align on top of the feature map from the Visual Model and $\{r_i\}$. The deep fusion module uses Kronecker product to fuse $\{t_i\}$ and $\{v_i\}$ to initialize the node features $\{n_i\}$. The node features are propagated and aggregated in the GNN during the message passing, which uses positional embedding $\{p_i\}$ and multi-head attention to learn the edge features dynamically. The classification head will finally classify the node features into one of the key-value classes.

3 Our Approach

Figure 1 illustrates our proposed model architecture. FS-DAG formulates the Key Information Extraction (KIE) (Huang et al., 2019) task as a graph node classification problem using pre-trained feature extractors and graph multi-head attention in a few-shot learning environment.

3.1 Model Architecture

The FS-DAG model (Agarwal et al., 2024b) is designed to address the unique challenges associated with VRDU in few-shot learning scenarios. Unlike traditional monolithic models (Yu et al., 2021; Xu et al., 2020b,a; Huang et al., 2022; Xu et al., 2021) that often require large amounts of data and extensive computational resources, FS-DAG employs a modular architecture that efficiently integrates domain-specific and language-specific textual and visual feature extractors with a GNN.

GNNs are particularly well-suited for VRDU tasks due to its ability to capture complex spatial and structural relationships between elements in a document. In FS-DAG, each document is represented as a graph where nodes correspond to these elements representing their textual and visual features, while the edges represent their spatial and semantic relationships. This graph representation allows the model to learn more robust and context-aware representations (Sun et al., 2021; Li et al., 2021). FS-DAG further incorporates shared positional embeddings and a multi-head attention mechanism within the GNN. Shared positional embeddings provide a consistent reference for the spatial location of elements across different document types, while multi-head attention enables dynamic weighting of node connections, thereby improving feature aggregation and learning efficiency.

The FS-DAG architecture allows for the seamless integration of pre-trained domain-specific (Lee et al., 2020; Liu et al., 2021) and language-specific feature extractors. This flexibility enables the model to quickly adapt to new domains with minimal data, significantly reducing the need for extensive retraining. By leveraging both textual and visual backbones tailored to specific domains, FS-DAG achieves superior performance compared to monolithic architectures that lack such adaptability. To further stabilize and boost the model's performance in a low-data setting, we modify the training strategies (Agarwal and Pachauri, 2023) and add augmentations for the graph (Agarwal et al., 2024a) and the visual modules. The individual components of the model are described further in the Appendix A.1.

3.2 Training Strategies

Training strategies are essential in few-shot training as we aim to attain the maximum model performance without overfitting the training dataset. To ensure higher performance and robustness of FS-DAG, we adopt various well-known strategies in the training process.

We include augmentation during training to enable the model to learn faster and be robust to various image and graph orientations. The augmentation technique focuses explicitly on the robustness of the visual embedding and the graph module. We introduce rotation ($\pm$ z degree), perspective transform, affine transform, and scaling and padding as the augmentations in the pipeline. These techniques enable the learning of better positional embeddings, visual embeddings, and node features as they change how the document is perceived and viewed. We also include specific graph augmentation (Agarwal et al., 2024a) which improves the convergence of FS-DAG with minimal data while making it robust to distribution shifts in textual or visual features

The proposed architecture does not support entity-linking currently and relies only on message propagation of the node features for the node classification task. Hence, we eliminate the edge loss function to stabilize the model training with the dedicated task.

Owing to the inductive bias from the pre-trained feature extractors, we introduce Label Smoothing (Müller et al., 2019) to the cross-entropy loss of node classification during training. Finally, to reduce overfitting in a few-shot learning paradigm, we add instance normalization (Ulyanov et al., 2016) over the node features of the graph. These changes enable us to train the model with better robustness and faster convergence.

4 Experiments

FS-DAG is extensively evaluated on multiple datasets against state-of-the-art models based on their official implementations in terms of performance, robustness to OCR errors, and model complexity. The official open-source code base was used to compare the result with other state-of-the-art models, followed by hyper-parameter tuning to get the best results for a fair comparison.

All experiments were conducted thrice on a machine with 16 cores, 32GB of RAM. We trained FS-DAG using a node and edge embedding size of 64 and two GNN layers, with label smoothing set to 0.1. Due to the unavailability of official codebases for tasks, we could not benchmark architectures such as FormNet (Lee et al., 2022) and StrucTexT

(Li et al., 2021). Few-shot techniques like LASER (Wang and Shang, 2022), which do not leverage visual features, were also excluded from the comparison. Additionally, LMMs like LLaVA(Li et al., 2024), Phi-3 (Abdin et al., 2024), and InternVL (Chen et al., 2024) were not benchmarked due to their considerable model size, which posed practical constraints. Other methods, such as (Or and Urbach), were omitted because they make multiple assumptions about the data structure and are not end-to-end trainable. To ensure a fair comparison, we focused on models with a size of less than 500M parameters.

4.1 Datasets & Metrics

For the VRDU task of KIE, publicly available datasets such as SROIE (Huang et al., 2019), CORD (Park et al.), and WildReceipt (Sun et al., 2021) primarily consist of document receipts from restaurants. While datasets like FUNSD (Guillaume Jaume, 2019) and Kleister (Graliński et al., 2020) include various forms and longer documents, they typically focus on high-level key-value pairs. These datasets are valuable for academic research but often fall short of meeting the nuanced requirements of industry-specific data extraction, which demands handling fine-grained classes.

The majority of public datasets are concentrated on receipts, invoices, train tickets, and simple forms, which lack the diversity needed to cover the broad range of use cases in industry domains such as finance, healthcare, and logistics. These datasets also rarely capture documents that require detailed, character-by-character annotations within boxes or placeholders, which are highly relevant in industrial applications. Zilong *et al.*(Wang et al., 2022) highlight these limitations and propose a new benchmark dataset for VRDU in both few-shot (10 and 50 samples) and conventional (100 and 200 samples settings. However, the document types in this dataset are limited to political ad-buys and registration forms, featuring only high-level fields (≤ 10) for extraction, thus not fully addressing the requirements of various industry verticals.

In this study, we use WildReceipt as a representative dataset from the existing public datasets, given its applicability to real-world receipt processing tasks. Additionally, we incorporate an industry-specific dataset[1] that better reflects the characteristics needed across multiple domains, as outlined in Table 1. This dataset includes document types filled character-by-character and features fine-grained key-value pair annotations at the word level, making it more aligned with the demands of industrial applications. We compare state-of-the-art models under the same few-shot setting on these datasets and conduct an extensive ablation study on the proposed methods. Performance on the given datasets is evaluated using the F1 score, as defined by the ICDAR 2019 robust challenge (Huang et al., 2019), with the averaged F1 score over all classes being reported.

Dataset Category	Dataset Name	# of classes
1	Ecommerce Invoice	34
	Adverse Reaction Health Form	46
	Medical Invoice	33
	University Admission Form	65
	Visa Form (Immigration)	45
2	Medical Authorization	34
	Personal Bank Account	94
	Equity Mortgage	70
	Corporate Bank Account	40
	Online Banking Application	28
	Medical Tax Returns	52
	Medical Insurance Enrollment	68

Table 1: Highlights the number of key-value classes across the each document type in the two categories.

4.2 Results and Discussions

We extensively conduct experiments with the two industrial dataset categories, owing to their diversity and industry relevance compared to publicly existing datasets. For benchmarking the models, we used five documents for training, while the remaining documents were used for testing. The split pattern was consistent across all the document types in both dataset categories. All the experiments for FS-DAG and other state-of-the-art models were run thrice, and the average results of the three runs are reported. We report the average F1 score across the document types in each dataset

[1] https://github.com/oracle-samples/fs-dag

Model	Params	Avg. Training Time	Avg. Inference Time	Category 1 Dataset			Category 2 Dataset		
				w/o OCR Error	**w/ OCR Error**	**Perf. Drop**	**w/o OCR Error**	**w/ OCR Error**	**Perf. Drop**
$BERT_{BASE}$	110M	27 mins	959 ms	89.84	64.60	25.24	92.03	58.97	33.06
Distill-BERT	65M	25 mins	**565 ms**	90.50	59.12	31.38	93.63	55.71	37.91
SDMGR	5M	28 mins	1207 ms	89.14	87.03	2.11	98.03	94.65	3.38
$LayoutLMv2_{BASE}$	200M	44 mins	1907 ms	94.03	74.57	19.46	93.26	89.71	3.55
$LayoutLMv3_{BASE}$	125M	35 mins	1363 ms	97.24	91.40	5.84	99.31	95.77	3.54
FS-DAG (ours)	81M	**21 mins**	773 ms	**98.89**	**97.96**	**0.93**	**99.93**	**99.02**	**0.91**

Table 2: Summary of model complexity, performance, robustness, and computational efficiency across five document types in the Category 1 dataset and seven document types in the Category 2 dataset. The best performance is highlighted in bold, and the second-best is underlined.

category.

Few-shot Key Information Extraction (KIE) Task. Column "w/o OCR Error" of Category 1 & Category 2 Datasets of Table 2 summarises the average F1-score results for both the dataset categories mentioned in Table 1 when the input OCR results of the document has no detection or recognition errors. It can be seen that FS-DAG outperforms its peer models with a high-performance gap. It can also be seen that LayoutLMv3 outperforms LayoutLMv2 while reducing the model complexity. LayoutLMv3 has very competitive results with FS-DAG but has higher model complexity. FS-DAG's performance can be attributed to the pre-trained models plugged in as feature extractors and position embeddings in the graph layer. It is also observed that the performance of FS-DAG and LayoutLMv3 are similar though the model complexity differs. FS-DAG outperforms SDMG-R by 9.75% and 1.9% for category 1 and 2 datasets, respectively. It highlights that the proposed changes over other graph models enable FS-DAG to have competitive performance with other larger multi-model models. The detailed experiment results are presented in Appendix B.

Model Robustness. KIE models often depend on OCR engines to extract text, which are then used as input. Despite improvements, OCR engines still produce errors, particularly with poor-quality documents. Some LMMs (e.g., Donut, LLaVa) incorporate OCR capabilities but suffer from similar limitations while significantly increasing model size beyond 500M parameters. We assess model robustness to OCR and misspelling errors by measuring performance drops due to misclassification. A robust model shows minimal performance decline, while models heavily reliant on text modality exhibit a more significant drop.

To evaluate robustness, we train models with ground-truth OCR data but introduce standard OCR errors with a probability of 0.1 during inference using nlpaug (Ma, 2019) (details in Appendix B). The average F1-scores under these conditions are shown in Column "w/ OCR Error" of Table 2, with the performance drop reported in Column "Perf. Drop".

FS-DAG demonstrates consistent robustness to OCR and misspelling errors with a performance drop of less than 1%, enhancing its reliability for real-world applications. Notably, SDMG-R also shows a lower performance drop compared to other models, underscoring the advantage of graph-based models in effectively integrating a document's modalities, as opposed to transformer-based models that heavily rely on textual sequences and tokenization.

Model Complexity. Table 2 also compares the model parameters, training and inference time across models. FS-DAG has substantially higher parameters compared graph-based SDMG-R owing to the pluggable pre-trained backbones. However, FS-DAG has almost 60-40% fewer parameters than other pre-trained transformer-based models like LayoutLMv2 or LayoutLMv3. LayoutLMv3 has competitive results with FS-DAG but with 64% additional model parameters.

The "Avg. Training Time" is reported against both the dataset categories for all the models. SDMG-R requires longer training because it's trained from scratch, unlike other models that are only fine-tuned. Additionally, training time increases with model size.

The "Avg. Inference Time" is reported against both the dataset categories for all the models. DistilBERT demonstrates the lowest latency but also exhibits lower performance across the datasets.

Model	Params	Avg. Perf. (%) (F1-Score)
$\text{BERT}_{\text{BASE}}$	110M	82.80
Distill-BERT	65M	80.70
SDMG-R	5M	82.80
$\text{LayoutLMv2}_{\text{BASE}}$	200M	86.00
$\text{LayoutLMv3}_{\text{BASE}}$	125M	<u>87.14</u>
FS-DAG	81M	**93.90**

Table 3: Summary of the average F1-Score (%) across the 25 classes in the WildReceipt dataset. The best performance is highlighted in bold, while the second-best performance is underlined.

FS-DAG achieves low latency while maintaining higher performance. Meanwhile, LayoutLMv3 has a latency that is 76% higher than FS-DAG, offering competitive performance but with reduced robustness. The lower model complexity reduces the cost of adopting the proposed model for the industry while outperforming other models.

Wildreceipt KIE Task. Table 3 shows the average F1-score on the publicly available dataset WildReceipt (Sun et al., 2021), which extracts key-value pairs (25 classes) from restaurant receipts from various restaurants. The results reported here take an average of all the 25 classes in the dataset compared to the 12 classes reported by Sun *etal* (Sun et al., 2021). The results show that FS-DAG outperforms the graph-based model by 11.1% while outperforming the LayoutLMv2 by 7.9% and LayoutLMv3 by 6.76% . These results demonstrate that FS-DAG is not only effective for a few-shot setting for a given document type but can scale across datasets with multiple-document types given sufficient training data with lesser model complexity.

Effect of Domain-Specific Language Model: We swap the pre-trained language model backbone (Distill-BERT) of FS-DAG with domain-specific language models for some of the datasets. The results in Table 4 and 5 showcase that using a language model which is better adapted to the finance and medical domain enables FS-DAG to perform better than using a generic language model as a textual feature extractor. Thus, the proposed modular architecture design enables higher performance in domain-specific use cases.

4.3 Ablation Study

We performed an ablation study on the industrial dataset to evaluate the effects of the architectural and training modifications, as detailed in Table 6. The starting point for each experiment is the skeleton FS-DAG architecture (row #1), with node and edge dimensions as 64. From rows #2s to #2e in Table 6, we study the individual contribution of the proposed changes in the few-shot setting. The results show that each component individually leads to a performance gain between 2%-6%. From rows #3 to #5 in Table 6, we combine the individual component and observe a performance gain increasing from 4% to 10% against row #1. The experiments conclusively show the importance and impact of the proposed changes and training for FS-DAG.

Base Architecture	Langauge Model used	# of Params (FS-DAG)	Ecommerce Invoice
FS-DAG (proposed)	Distill-BERT	81M	95.1
	$\text{BERT}_{\text{BASE}}$	110M	96.26
	ProsusAI/finbert	125M	**98.63**

Table 4: Results of replacing DistilBERT in FS-DAG with BERT and finance-domain-specific models on the eCommerce Invoice.

Base Architecture	Langauge Model used	# of Params (FS-DAG)	Adverse Reaction Health Form
FS-DAG (proposed)	Distill-BERT	81M	96.53
	$\text{BERT}_{\text{BASE}}$	110M	97.13
	BiomedVLP-CXR-BERT-general	125M	**98.98**

Table 5: Results of replacing DistilBERT in FS-DAG with BERT and medical-domain-specific models on the medical form.

Effect of Pre-trained Language Model: We use Distill-BERT as the pluggable pre-trained language model for all the experiments for extracting textual features. Adding a pre-trained language model and using the first sub-token to represent a text region $\{r_i\}$ improves the F1-score by 0.95% on average (Table 6: From #1 vs. #2a). Further pooling all the sub-token representations of a text region $\{r_i\}$ to get the token representation, we see the performance improves by 3.30% on average (Table 6: From #1 vs. #2b). It highlights that pooling the sub-token representation to represent a text region $\{r_i\}$ gives a better and richer representation that enables the model to learn in a few-shot setting.

Effect of Pretrained Visual Model: We use UNET with a Resnet-18 backbone pre-trained on PubTabnet (Smock et al., 2022) for extracting the visual features. The model F1-score increases by

Model	#	Architectural Changes Proposed					Avg Perf. (%) (F1 Score)	Perf. Gain (%) (F1 Score)
		Pre-trained LM w/ first token embedding	Pre-trained LM w/ pooling token embeddings	Pre-trained Visual Model	Position Embedding in GNN	Training Strategies		
FS-DAG	1						88.31	NA
	2a	✓					89.26	0.95
	2b		✓				91.61	3.30
	2c			✓			91.33	3.02
	2d				✓		93.64	5.33
	2e					✓	93.86	5.55
	3		✓	✓			92.43	4.12
	4		✓	✓	✓		97.37	9.06
	5		✓	✓	✓	✓	**98.89**	**10.58**

Table 6: The detailed ablation study results on different components and training of FS-DAG are reported for the Category 1 dataset. We observe that each proposed change has a significant positive impact on the model performance. The final proposed architecture of FS-DAG configuration is shown in experiment row **#5**.

3.02% (Table 6: From #1 vs. #2c) on average across the five few-shot datasets. It highlights that using a pre-trained visual feature extractor enables FS-DAG to learn better in a few-shot setting. However, it can also be seen that the impact of pre-trained visual features is lesser than the textual features.

Effect of Position Embedding: We introduce learnable position embedding in the GNN layer of the model. The model F1-score increases by 5.33% (Table 6: From #1 vs. #2d) on average across the five datasets, showcasing that the position embedding plays an essential role in the GNN layers learning, helping it to adapt to the given document type.

Effect of Training Strategies: Apart from the model architecture changes, the training strategy for models in a few-shot learning environment plays an important role. The proposed training strategies for FS-DAG led to an increase of F1-score of 5.55% (Table 6: From #1 vs. #2e) on average across the five datasets.

Finally, combining the different components shows an improvement (Table 6: From #3 to #5), showcasing that the proposed components complement each other and leading to an overall average gain of 9.28% for the proposed model in a few-shot setting.

5 Conclusion

FS-DAG presents a compelling alternative to large-scale models like VLMs, LMMs and LLMs, particularly for visually rich document understanding tasks in industrial applications like document classification, key value extraction, entity-linking and graph classification. By focusing on efficiency, scalability, and practical deployment, FS-DAG addresses the key limitations of these larger models, including their high computational cost and the challenges associated with training and running them in resource-constrained environments.

This work demonstrates FS-DAG's technical strengths and emphasizes its practical application in real-world environments, where its robustness, customizability, and low computational demands significantly lower operational costs, making advanced models more accessible across various industries. Currently, FS-DAG is adopted by over 50+ customers and provided through hyperscale cloud providers with over 1M+ API calls monthly.

Future research will focus on extending FS-DAG's capabilities to zero-shot learning and enhancing its adaptability to a broader range of industrial scenarios.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

AMIT AGARWAL. 2021. Evaluate generalisation & robustness of visual features from images to video.

Amit Agarwal and Kulbhushan Pachauri. 2023. Pseudo labelling for key-value extraction from documents. US Patent 11,823,478.

Amit Agarwal, Kulbhushan Pachauri, Iman Zadeh, and Jun Qian. 2024a. Techniques for graph data structure augmentation. US Patent 11,989,964.

Amit Agarwal, Srikant Panda, Deepak Karmakar, and Kulbhushan Pachauri. 2024b. Domain adapting

graph networks for visually rich documents. US Patent App. 18/240,480.

Amit Agarwal, Srikant Panda, and Kulbhushan Pachauri. 2024c. Synthetic document generation pipeline for training artificial intelligence models. US Patent App. 17/994,712.

Amit Agarwal, Priyaranjan Pattnayak, Bhargava Kumar, Hitesh Patel, Srikant Panda, and Tejaswini Kumar. 2024d. Enhancing document ai data generation through graph-based synthetic layouts. *International Journal of Engineering Research & Technology (IJERT)*, 13(10).

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. 2024. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2021. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*.

Filip Graliński, Tomasz Stanisławek, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2020. Kleister: A novel task for information extraction involving long documents with complex layout. *arXiv preprint arXiv:2003.02356*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Jean-Philippe Thiran Guillaume Jaume, Hazim Kemal Ekenel. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *Accepted to ICDAR-OST*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Himanshu. 2019. Detectron2. https://github.com/hpanwar08/detectron2.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Bharti Khemani, Shruti Patil, Ketan Kotecha, and Sudeep Tanwar. 2024. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, 11(1):18.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. 2022. Formnet: Structural encoding beyond sequential modeling in form document information extraction. *arXiv preprint arXiv:2203.08411*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2024. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36.

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*, pages 1912–1920.

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019. Graph convolution for multimodal information extraction from visually rich documents. *arXiv preprint arXiv:1903.11279*.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 4513–4519.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Nerya Or and Shlomo Urbach. Few-shot learning for structured information extraction from form-like documents using a diff algorithm.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: A consolidated receipt dataset for post-ocr parsing.

Hitesh Laxmichand Patel, Amit Agarwal, Bhargava Kumar, Karan Gupta, and Priyaranjan Pattnayak. 2024. Llm for barcodes: Generating diverse synthetic data for identity documents. *arXiv preprint arXiv:2411.14962*.

Mouli Rastogi, Syed Afshan Ali, Mrinal Rawat, Lovekesh Vig, Puneet Agarwal, Gautam Shroff, and Ashwin Srinivasan. 2020. Information extraction from document images via fca-based template detection and knowledge graph rule induction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 558–559.

Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. 2021. Docparser: Hierarchical document structure parsing from renderings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4328–4338.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Brandon Smock, Rohith Pesala, and Robin Abraham. 2022. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4634–4642.

Hongbin Sun, Zhanghui Kuang, Xiaoyu Yue, Chenhao Lin, and Wayne Zhang. 2021. Spatial dual-modality graph reasoning for key information extraction. *arXiv preprint arXiv:2103.14470*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Zilong Wang and Jingbo Shang. 2022. Towards few-shot entity recognition in document images: A label-aware sequence-to-sequence framework. *arXiv preprint arXiv:2204.05819*.

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2022. A benchmark for structured extractions from complex documents. *arXiv preprint arXiv:2211.15421*.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020b. 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*, pages 1192–1200.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *arXiv preprint arXiv:2104.08836*.

Minghong Yao, Zhiguang Liu, Liangwei Wang, Houqiang Li, and Liansheng Zhuang. 2021. One-shot key information extraction from document with deep partial graph matching. *arXiv preprint arXiv:2109.13967*.

Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao. 2021. Pick: processing key information extraction from documents using improved graph learning-convolutional networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4363–4370. IEEE.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE.

A Appendix

A.1 Details of Model Architecture

A.1.1 Text Embeddings

Training language models from scratch are resource-intensive, time-consuming, and needs to generalize better in a few-shot learning environment. Hence, we designed our architecture to have a pluggable language model. It enables choosing multi-lingual domain-specific language models like BioBERT(Lee et al., 2020), BiomedNLP-PubMedBERT (Gu et al., 2021), FinBERT (Liu et al., 2021), for various use cases requiring fine-grained features, like in the medical, finance, or law domain, while also helping choose regional or multi-lingual language-based models. Standard use cases can rely on models like BERT (Devlin et al., 2018), Distill-BERT (Sanh et al., 2019), and Alberta (Lan et al., 2019) based on the performance and latency requirement of the model.

As shown in Figure 1, a document image I is parsed via an OCR engine (word-level) to extract text regions $\{r_i\}$. Formally, for a document with a total number of words L we have the i-th $(0 < i \leq L)$ text region as the i-th word in the document. We then cluster and sort the $\{r_i\}$ to get a consistent reading sequence $\{s\}$ for the document, which later enables us to extract contextual text representation using a pre-trained language model. The reading sequence $\{s\}$ is the document's reading order to ensure consistent feature extraction during training and inference.

The reading sequence $\{s\}$ is then passed through a language model which tokenizes and decodes the sequence to return a sequence of token embedding, where $y_j \in R^{D_t}$ is the text-embedding for the token in $\{s\}$, D_t is the dimension of the text embedding. The language model tokenizes the words/text regions $\{r_i\}$ within $\{s\}$ into multiple tokens for which we get the text embedding $\{y_j\}$. Hence, we pool text embeddings of the tokens belonging to a particular $\{r_i\}$ to get the textual embedding of the document's original word/text region. During the model training, the language model weights are frozen, and the extracted textual embedding of the words/text regions $\{r_i\}$ is projected over linear layers to adapt them as per the document type. Formally, for a sequence of length L, we have the i-th text embedding as:

$$t_i = MLP_1(\text{LangModelEmb}(r_i)) \quad (1)$$

MLP_1 is a learnable multi-layer perceptron that fine-tunes the textual embedding of a word/text region from the Language Model. The LangModelEmb layer clusters and sorts the text regions $\{r_i\}$ to create the reading sequence $\{s\}$ and extracts and pools the token embeddings $\{y_j\}$ to create the textual embedding of the given word/text regions $\{r_i\}$.

A.1.2 Visual Embeddings

Text in documents is designed to capture human attention based on the text's color, font size, texture, and appearance. Hence to extract the visual features (AGARWAL, 2021), we use a UNET (Ronneberger et al., 2015) with a Resnet-18 (He et al., 2016) backbone as a visual feature extractor. The Resnet-18 backbone is pre-trained on the document's dataset (Zhong et al., 2019; Himanshu, 2019) and can be swapped with any other feature extractor based on the document type. Since visual features in VRDs are very extensive and document type dependent, we do not freeze weights of the visual backbone, letting it adapt in the few-shot setting during end-to-end training.

As shown in Figure 1, a document image I is passed through the pre-trained visual model to extract feature maps. The RoI Align layer (Sun et al., 2021; He et al., 2017) extracts the visual embedding v_i for every text region $\{r_i\}$ using the bounding box coordinates on the output feature maps of the visual model.

$$v_i = RoIAlign(VisFeatMap\,(I), r_i) \quad (2)$$

The VisFeatMap layer extracts the visual feature map based on the feature extractor backbone used. The RoIAlign layer extracts $\{v_i\}$, where $v_i \in R^{D_v}$ based on the $\{r_i\}$ bounding box co-ordinates, and D_v is the dimension of the visual embedding.

A.1.3 Node & Edge Embeddings

The graph nodes $\{n_i\}$ are initialized by fusing the textual features $\{t_i\}$ and visual features $\{v_i\}$ in the deep fusion block as shown in Figure 1. The deep fusion block uses the Kronecker product as per (Sun et al., 2021) and projects the result on linear layers as :

$$n_i = MLP_2\left(t_i \otimes v_i\right) \tag{3}$$

$\otimes$ is the Kronecker product operation, while MLP_2 is a learnable multi-layer perceptron, where $n_i \in R^{D_n}$ and D_n is the dimension of the visual embedding.

The spatial relation $\{s_{ij}\}$ between the two connecting nodes $\{n_i\}$ and $\{n_j\}$, where $0 < i, j \leq L$, is defined by calculating the relative distance between the nodes using the bounding box coordinates $< x_0, y_0, x_1, y_1 >$ as described in (Sun et al., 2021; Agarwal et al., 2024d).The spatial relation s_{ij} is normalized after passing it through linear projection layers to initialize the edge embedding e'_{ij} as follows:

$$e'_{ij} = N_{l2}(MLP_3\left(s_{ij}\right)) \tag{4}$$

MLP_3 is a learnable multi-layer perceptron that transforms the spatial relation information s_{ij} into e'_{ij}, where $e_{ij} \in R^{D_e}$ and D_e dimension of the edge embedding. N_{l2} is the l_2 normalization operation. In the GNN layer, e'_{ij} interacts with the node and position embeddings to refine the edge embedding and interaction between nodes using multi-head attention.

A.1.4 Position Embeddings & Multi-head Attention

We divide the entire document in a K x K grid as shown in Figure 1, and all the text regions $\{r_i\}$ in a particular grid, share the same positional embedding. The positional embedding enables the graph module to learn more about a node's absolute positioning and neighbors. The size of the grid K becomes a hyper-parameter that can be updated based on the document type. In our experiments, we found the value of K=25 to work consistently well across all the datasets.

Given a text region $\{r_i\}$, with the bounding box coordinates as $< x_0, y_0, x_1, y_1 >$, the individual horizontal and vertical position embedding are computed as:

$$Pos_{hor} = PosEmb_{hor}\left(x_0\right) \,||\, PosEmb_{hor}\left(x_1\right) \tag{5}$$

$$Pos_{ver} = PosEmb_{ver}\left(y_0\right) \,||\, PosEmb_{ver}\left(y_1\right) \tag{6}$$

We separately learn the horizontal and vertical positional embedding. Finally, the positional embedding $\{p_i\}$, where $p_i \in R^{D_p}$ for a given node concatenates the horizontal and vertical positional embeddings and passes it through a non-linear function TanH as suggested in (Dwivedi et al., 2021).

$$p_i = TanH(Pos_{hor} \,||\, Pos_{ver}) \tag{7}$$

The positional embedding is integrated and trained during the message propagation along the edges and multi-head attention. The different attention heads focus on the groups and segments within the nodes that strongly influence each other. The attention scores enable dynamic weighing of the edge connections to enable better node feature aggregation along various positional grids.

$$e^h_{ij} = MLP_4(\, n_i \,||\, p_i \,||\, e'_{ij} \,||\, n_j \,||\, p_j) \tag{8}$$

$$\mathbf{e^h_{ij}} = MLP_5(e^h_{ij}) \tag{9}$$

We concatenate the node embeddings $\{n_i\}$ and $\{n_j\}$ with their corresponding positional embedding $\{p_i\}$ and $\{p_j\}$ before concatenating it with the initial edge embedding e'_{ij} between them. MLP_4 is a multi-layer perceptron that transforms the concatenated embeddings for each attention head. $e^h_{ij} \in R^{D_{ne}\ X\ D_h\ X\ D_n}$, where D_{ne} represents the number of edges in the graph, D_h represents the number of heads in the network and D_n represents the node embedding dimension. MLP_5 is a multi-layer perceptron that transforms e^h_{ij} into a scaler for each of the edges, where $\mathbf{e^h_{ij}} \in R^{D_{ne}\ X\ D_h\ X\ 1}$. Finally, we refine the node features $\{n_i\}$ of the graph module K times as follows :

$$n^{k+1}_i = n^k_i + \sigma(N_{IN}(MLP^k_6(\Big|\Big|_h \sum_{j \neq i} \alpha^{kh}_{ij} \mathbf{e^{kh}_{ij}}))) \tag{10}$$

where $n^k_i \in R^{D_n}$ indicates the features of the ith graph node at time step k. α^{kh}_{ij} is normalized edge weight at time step k for a particular attention head. e^{kh}_{ij} is the transformed concatenated representation

of a particular attention ahead at time step k as described in Equation 9. MLP_6^k is a linear transformation at time step k. N_{IN} is the instance norm over the embeddings before passing it through σ, which is the non-linear activation function ReLU. α_{ij}^{kh} is the learnable normalized weights between nodes i and j for every attention head h at time step k. It is given by :

$$\alpha_{ij}^{kh} = \frac{\exp(\mathbf{e_{ij}^h})}{\sum_{j \neq i} \exp(\mathbf{e_{ij}^h})} \tag{11}$$

B Experiments, Extended

B.1 Dataset & Metrics, Extended

In Table 1 we share the class distribution of the various document types proposed in the dataset. Sample images for each document type (Agarwal et al., 2024c) in Category 1 are highlighted in Figure 2. In Figure 3, we highlight the sample images for each document type in Category 2. It can be seen that document types visually in Category 2 are fundamentally different from documents in Category 1 in how they are generated and filled with capturing necessary information for the business. These document types capture relevant information within specific placeholders, mostly filled character-by-character by a human or digital application. Document types in Category 2 datasets are still actively used worldwide, and more publicly available datasets for such documents must be available to steer research and evaluation of models. The released dataset will thus help further push boundaries for different document types in a few-shot setting.

B.2 Results, Extended

The main paper reports average results across the different datasets for various state-of-the-art models. Here, we present the results on individual document types across both the dataset category for fine-grained analysis.

Model Robustness. To stimulate real-world misspelling or OCR errors in documents (Agarwal et al., 2024c; Patel et al., 2024), we use nlpaug (Ma, 2019) to introduce text recognition errors during the inference of models. Table 7 showcases the most common errors observed across various human misspellings and available OCR engines. The benchmarking of all the document types across the dataset categories when input errors are introduced during inference are detailed in Table 9 and 12.

Character	**Common OCR Errors**
1	l(lower case of L), I (Upper case of i)
l (lowercase of L)	I (Upper case of i)
6	b
5	S
,	.
Sample Augmentation	
Original	**OCR Error Text**
The quick brown fox ate 5 chocolates	The quick brown fox ate S chocoIates

Table 7: Highlights most common OCR errors across popular OCR engines, along with a sample augmentation using nlpaug.

Finally, we observe the drop in performance for individual document types across the two dataset categories in Table 10 and 13. The observations are discussed in the following sections.

Category 1 Dataset (KIE Task). Table 8 shows the F1-score results of FS-DAG on the five industry document types from the category 1 dataset while comparing it to other state-of-the-art models. All the models are trained and tested in this benchmark with ground-truth annotations. We can observe that FS-DAG outperforms most of its peers by a considerable margin. At the same time, LayoutLMv3 has very similar performance compared to FS-DAG, and the best model varies based on the dataset with a small margin. In Table 9, we report the F1-score when the training has been done with ground-truth OCR annotations. At the same time, during inference, misspelling and OCR errors are introduced at the word level with a probability of 0.1. Table 10 reports the drop in performance when the model is tested under the two different scenarios as represented in Table 8 and 9. Models which are robust to input errors or less dependent on textual modality show a lower drop in performance.

It is observed that language models like $\text{BERT}_{\text{BASE}}$ and Distill-BERT have the maximum drop in performance as they rely entirely on textual modality. Multimodal model like LayoutLMv2 shows a higher performance drop than LayoutLMv3, suggesting that LayoutLMv2 is more dependent on the textual features. FS-DAG has the least fall in performance, followed by SDMG-R, implying better robustness to misspelling or OCR errors. The best-performing model for different document types vary and is highlighted in bold in

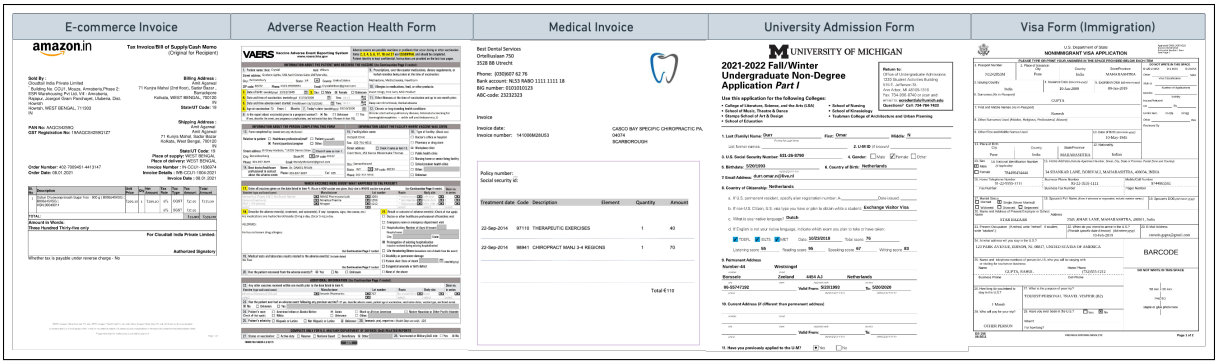

Figure 2: Sample images from each of the five document types released as part of the Category 1 dataset.

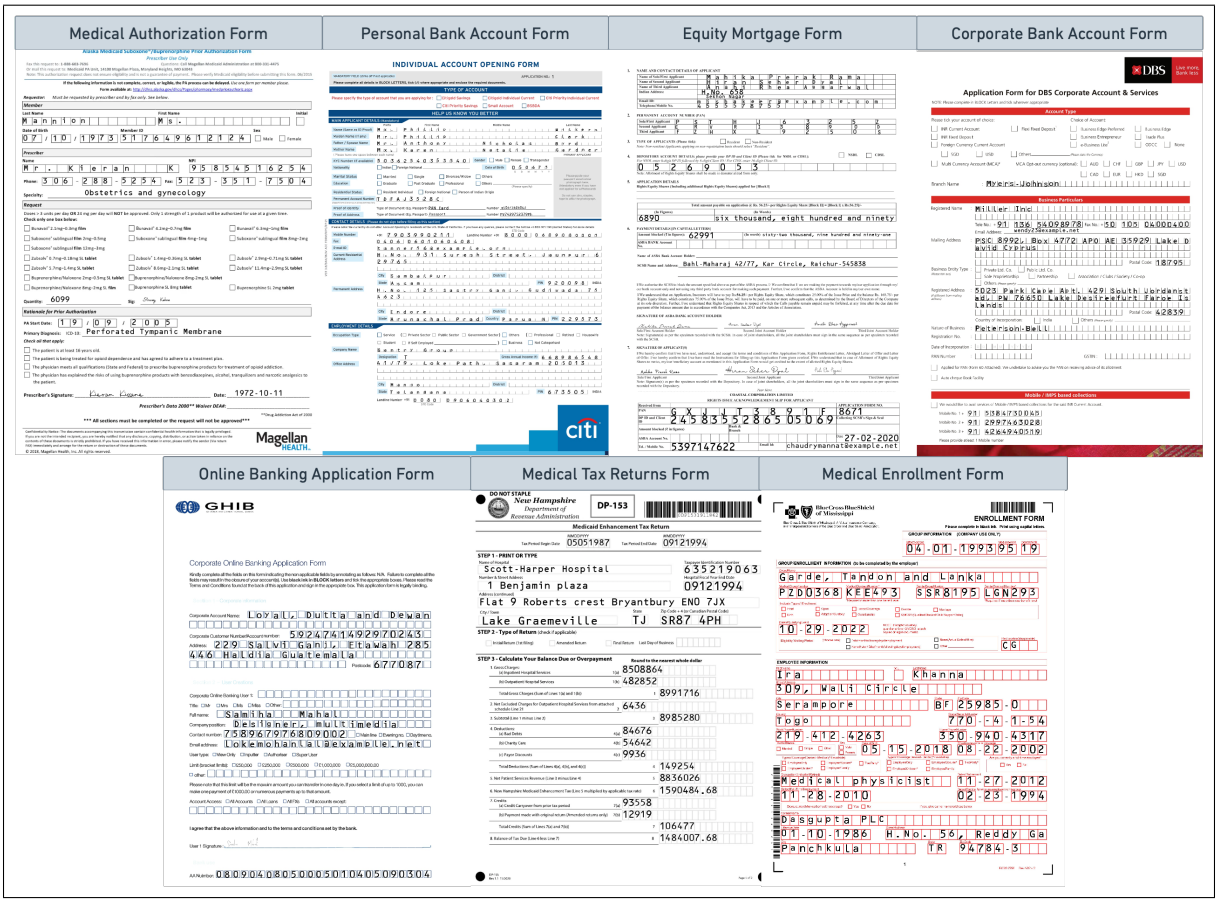

Figure 3: Sample images from each of the seven document types released as part of the Category 2 dataset.

Table 8. However, FS-DAG outperforms its peers with the most consistent performance with lesser model complexity.

Category 2 Dataset (KIE Task). Table 11 shows the F1-score results of FS-DAG on the seven industry document types from the category 2 dataset while comparing it to other state-of-the-art models. All the models are trained and tested in this benchmark with ground-truth OCR annotations. We can observe that FS-DAG outperforms most of its peers by a considerable margin, while LayoutLMv3 has a similar performance. In Table 12, we report the F1-score when the training has been done with ground-truth annotations. At the same time, during inference, misspelling and OCR errors are introduced at the word level with a probability of 0.1. Table 13 reports the drop in performance when the model is tested under the two different scenarios as represented in Table 11 and 12. SDMG-R, LayoutLM Series have performance drop in similar range which is higher compared to FS-DAG. The best-performing model for different document types vary and is highlighted in bold in Table 11. FS-DAG outperforms its peers with

Model	Params	F1- Score across Category 1 Dataset (Inference without OCR Errors)					
		Ecommerce Invoice	**Adverse Reaction Health Form**	**Medical Invoice**	**University Admission Form**	**Visa Form (Immigration)**	**Avg Perf.**
$BERT_{BASE}$	110M	91.60	81.00	98.60	86.20	91.80	89.84
Distill-BERT	65M	90.30	82.50	99.20	90.70	89.80	90.50
SDMGR	5M	90.58	89.86	90.15	90.10	85.01	89.14
$LayoutLMv2_{BASE}$	200M	97.20	88.60	100.00	95.97	88.40	94.03
$LayoutLMv3_{BASE}$	125M	95.80	95.00	**100.00**	97.20	98.20	97.24
FS-DAG (ours)	81M	**98.30**	**98.51**	99.90	**98.40**	**99.34**	**98.89**

Table 8: Reports the field-level F1 scores for the KIE task in a few-shot learning setting for the five domain-specific document types from the category 1 dataset are reported. The best performance is highlighted in bold, while the second-best performance is underlined.

Model	F1 Score across Category 1 Dataset (Inference with OCR errors)					
	Ecommerce Invoice	**Adverse Reaction Health Form**	**Medical Invoice**	**University Admission Form**	**Visa Form (Immigration)**	**Avg Performance**
$BERT_{BASE}$	83.20	36.30	84.90	60.40	58.20	64.60
Distill-BERT	78.60	38.70	84.70	46.30	47.30	59.12
SDMGR	90.00	86.50	87.67	87.00	84.00	87.03
$LayoutLMv2_{BASE}$	93.80	42.30	93.74	85.00	58.02	74.57
$LayoutLMv3_{BASE}$	95.40	81.20	99.20	89.60	91.60	91.40
FS-DAG (ours)	**98.01**	**97.93**	**99.50**	**96.80**	**97.56**	**97.96**

Table 9: Reports the field-level F1 scores for the KIE tasks when the models are trained with ground-truth OCR (without any errors), and testing happens with words having OCR errors with a probability of 0.1. FS-DAG outperforms the competitor models with a substantial performance gap, highlighting the generalizability and robustness of the model.The best performance is highlighted in bold, while the second-best performance is underlined.

Model	Drop in F1 Score across Category 1 Dataset (Table 2 - Table 3)					
	Ecommerce Invoice	**Adverse Reaction Health Form**	**Medical Invoice**	**University Admission Form**	**Visa Form (Immigration)**	**Avg Perf. Drop**
$BERT_{BASE}$	8.40	44.70	13.70	25.80	33.60	25.24
Distill-BERT	11.70	43.80	14.50	44.40	42.50	31.38
SDMGR	0.58	3.36	2.48	3.10	1.01	2.11
$LayoutLMv2_{BASE}$	3.40	46.30	6.26	10.97	30.38	19.46
$LayoutLMv3_{BASE}$	0.40	13.80	0.80	7.60	6.60	5.84
FS-DAG (ours)	**0.29**	**0.58**	**0.40**	**1.6**	**1.78**	**0.93**

Table 10: Highlights the fall in model performance (difference between results in Table 2 vs. Table 3) when the test document has misspelling or OCR errors with a probability of 0.1. FS-DAG shows the minimum drop in performance overall and consistently higher performance compared to other models. The best performance is highlighted in bold, while the second-best performance is underlined

the most consistent performance with lesser model complexity. It is observed that language models like $BERT_{BASE}$ and Distill-BERT have the maximum drop in performance (comparatively higher than document types in Category 1) as they rely entirely on textual features.

Models	Params	F1- Score across Category 2 Dataset (Inference without OCR Errors)							Avg Perf.
		Medical Authorization	Personal Bank Account	Equity Mortage	Corporate Bank Account	Online Banking Application	Medical Tax Returns	Medical Insurance Enrollment	
$BERT_{BASE}$	110M	96.1	95.3	87.4	92.4	89.2	89.1	94.7	92.03
Distill-BERT	65M	95.7	97	92.3	92	91.1	90.2	97.1	93.63
SDMGR	5M	95.67	99.13	95.67	99.7	98.3	99	98.77	98.03
$LayoutLMv2_{BASE}$	200M	96.9	88.1	94.1	96.4	87.5	97.9	91.9	93.26
$LayoutLMv3_{BASE}$	125M	96.9	99.9	**100**	99.9	100	100	98.5	99.31
FS-DAG	81M	**100**	**100**	99.9	**100**	**100**	**100**	**99.6**	**99.93**

Table 11: Reports the field-level F1 scores for the KIE task in a few-shot learning setting for the seven domain-specific document types from the category 2 dataset are reported. The best performance is highlighted in bold, while the second-best performance is underlined.

Models	Params	F1- Score across Category 2 Dataset(Inference with OCR errors)							Avg Perf.
		Medical Authorization	Personal Bank Account	Equity Mortage	Corporate Bank Account	Online Banking Application	Medical Tax Returns	Medical Insurance Enrollment	
$BERT_{BASE}$	110M	50.60	40.80	67.40	58.90	75.30	69.00	50.80	58.97
Distill-BERT	65M	40.30	42.60	64.90	50.70	77.70	66.00	47.80	55.71
SDMGR	5M	88.27	90.70	95.23	98.37	99.10	98.47	92.40	94.65
$LayoutLMv2_{BASE}$	200M	93.24	80.19	97.28	91.39	89.43	91.12	85.31	89.71
$LayoutLMv3_{BASE}$	125M	88.60	98.00	**99.45**	95.37	98.49	99.84	90.61	95.77
FS-DAG	81M	**98.40**	**98.50**	99.09	**99.43**	**99.5**	**99.67**	**96.57**	**99.02**

Table 12: Reports the field-level F1 scores for the KIE tasks when the models are trained with ground-truth OCR (without any errors), and testing happens with words having OCR errors with a probability of 0.1. FS-DAG outperforms the competitor models with a substantial performance gap, highlighting the generalizability and robustness of the model.The best performance is highlighted in bold, while the second-best performance is underlined.

Models	Params	Drop in F1 Score across Category 2 Dataset (Table 4 - 5)							Avg Perf. Drop
		Medical Authorization	Personal Bank Account	Equity Mortgage	Corporate Bank Account	Online Banking Application	Medical Tax Returns	Medical Insurance Enrollment	
$BERT_{BASE}$	110M	45.50	54.50	20.00	33.50	13.90	20.10	43.90	33.06
Distill-BERT	65M	55.40	54.40	27.40	41.30	13.40	24.20	49.30	37.91
SDMGR	5M	7.40	8.43	**0.44**	1.33	0.80	0.53	6.37	3.39
$LayoutLMv2_{BASE}$	200M	3.66	7.91	3.18	5.01	1.93	6.78	6.59	3.55
$LayoutLMv3_{BASE}$	125M	8.30	1.90	0.55	4.53	1.51	**0.16**	7.89	3.55
FS-DAG	81M	**1.60**	**1.50**	0.81	**0.57**	**0.50**	0.33	**1.03**	**0.91**

Table 13: Highlights the fall in model performance (difference between results in Table 4 vs. Table 5) when the test document has misspelling or OCR errors with a probability of 0.1. FS-DAG shows the minimum drop in performance overall and consistently higher performance compared to other models.