PixT3: Pixel-based Table-To-Text Generation

Iñigo Alonso,
HiTZ Center - Ixa,
University of the
Basque Country UPV/EHU
inigoborja.alonso@ehu.eus

Eneko Agirre
HiTZ Center - Ixa,
University of the
Basque Country UPV/EHU
e.agirre@ehu.eus

Mirella Lapata
Institute for Language,
Cognition and Computation,
University of Edinburgh
mlap@inf.ed.ac.uk

Abstract

Table-to-text generation involves generating appropriate textual descriptions given structured tabular data. It has attracted increasing attention in recent years thanks to the popularity of neural network models and the availability of large-scale datasets. A common feature across existing methods is their treatment of the input as a string, i.e., by employing linearization techniques that do not always preserve information in the table, are verbose, and lack space efficiency. We propose to rethink data-to-text generation as a visual recognition task, removing the need for rendering the input in a string format. We present PixT3, a multimodal table-to-text model that overcomes the challenges of linearization and input size limitations encountered by existing models. PixT3 is trained with a new self-supervised learning objective to reinforce table structure awareness and is applicable to open-ended *and* controlled generation settings. Experiments on the ToTTo (Parikh et al., 2020a) and Logic2Text (Chen et al., 2020c) benchmarks show that PixT3 is competitive and, in some settings, superior to generators that operate solely on text.[1]

1 Introduction

Generating text from structured inputs such as tables, tuples, or graphs, is commonly referred to as data-to-text generation (Reiter and Dale, 1997; Covington, 2001; Gatt and Krahmer, 2018). This umbrella term includes several tasks ranging from generating sport summaries based on boxscore statistics (Wiseman et al., 2017), to producing fun facts from superlative Wikipedia tables (Korn et al., 2019), and creating textual descriptions given biographical data (Lebret et al., 2016). From a modeling perspective, data-to-text generation is challenging as it is not immediately obvious how to best describe the given input. For instance, the table in Figure 1 can be verbalized in different ways, depending on the specific content we choose to focus on. In *controlled* data-to-text generation (Parikh et al., 2020a), models are expected to generate descriptions for pre-selected parts of the input (see the *highlighted* cells in Figure 1).

Regardless of the generation setting, numerous approaches have emerged in recent years with different characteristics. A few exploit the structural information of the input (Puduppully et al., 2019; Chen et al., 2020b; Wang et al., 2022), use neural templates (Wiseman et al., 2018), or resort to content planning (Su et al., 2021; Puduppully et al., 2022). While others (Chen et al., 2020a,c; Aghajanyan et al., 2022; Kasner and Dusek, 2022) improve on fluency and generalization by leveraging large-scale pre-trained language models (Devlin et al., 2019; Raffel et al., 2020). A common feature across these methods is their treatment of tabular input as a string, following various linearization methods. As an example, Figure 1 shows the representation of tabular data (top) as a sequence of (Column, Row, Value) tuples (bottom).

Problematically, representing tabular information as a linear sequence results in a verbose representation that often exceeds the context window limit of popular Transformer models (Vaswani et al., 2017). The challenge of processing such long sequences has fostered the development of even more controlled methods which refrain from encoding the table as a whole, concentrating exclusively on highlighted content (e.g., *only* the yellow cells in Figure 1). Unfortunately, models trained on abridged input have difficulty generalizing to new domains while being practically ineffective in scenarios where content selection is not provided.

In this paper we propose to rethink data-to-text generation as a visual recognition task, allowing us to represent and preserve tabular information compactly. Vision Transformers (ViTs; Dosovitskiy et al. 2021) have significantly advanced

[1]Our code, models, and data are available at https://github.com/alonsoapp/PixT3.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6721–6736

Table Title: Shuttle America
Section Title: Fleet

Aircraft	Total	Orders	Passengers				Operated for	Notes
			F	Y+	Y			
Embraer E170	5	-	6	16	48	70	United Express	transferred to Republic Airline
	14	-	9	12		69	Delta Connection Delta Shuttle	2 planes on wet lease from Republic Airline
Embraer E175	15	-	12	12	52	76		
Total	**35**	-						

Linearized Table: <page_title> Shuttle America <page_title> <section_title> Fleet <section_title> <table> <row> <cell> Aircraft <cell> <cell> Total <row_header> Aircraft <row_header> <cell> <cell> Orders <row_header> Aircraft <row_header> <row_header> Total <row_header> <cell> <cell> Passengers <row_header> Aircraft <row_header> <row_header> Total <row_header> <row_header> Orders <row_header> <cell> <cell> Operated For <row_header> Aircraft <row_header> <row_header> Total <row_header> <row_header> Orders <row_header> <row_header> Passengers <row_header> <cell> <cell> Notes <row_header> Aircraft <row_header> . . . . . .

Target Description: Shuttle America operated the E-170 and the larger E-175 aircraft for Delta Air Lines.

Figure 1: Example of table-to-text generation taken from the ToTTo dataset (Parikh et al., 2020a). In the controlled setting, a natural language description is generated only for highlighted (yellow) cells. The table is linearized by encoding each value as a (Column, Row, Value) tuple. We only show the first row, for the sake of brevity.

the field of visual language understanding (Kim et al., 2022; Davis et al., 2022) demonstrating proficiency in various tasks, including language modeling (Rust et al., 2023), visual document understanding (Huang et al., 2022), and visual question answering (Masry et al., 2022). Our work builds on Pix2Struct (Lee et al., 2023), a pretrained image-to-text model which can be fine-tuned for visually-situated language tasks. We recast data-to-text generation as an image-to-text problem and present PixT3, a **Pix**el-based **T**able-**t**o-**T**ext model, which is generally applicable to open-ended and controlled generation settings, overcoming the challenges of linearization and input size limitations encountered by existing models.

Our contributions can be summarized as follows: (a) we introduce the first pixel-based model for table-to-text generation and showcase its robustness across generation settings with varying table sizes; (b) we propose a new training curriculum and self-supervised learning objective to reinforce table structure awareness; (c) automatic and human evaluation results on the ToTTo benchmark (Parikh et al., 2020b) show that PixT3 excels in open-ended generation, leading to improved faithfulness and generation quality, while being competitive with existing methods in controlled scenarios; and (d) we present a new dataset based on Logic2Text (Chen et al., 2020c), which allows us to evaluate generalization capabilities of current table-to-text models.

2 Related Work

The bulk of previous work treats tables as textual objects. Several techniques have been developed to extract accurate information from them (Puduppully et al., 2019; Chen et al., 2020b) using templates (Wiseman et al., 2018), enforcing table structure awareness (Mahapatra and Garain, 2021; Wang et al., 2022), applying contrastive learning (An et al., 2022; Chen et al., 2023b) or focusing on content planning (Su et al., 2021; Puduppully et al., 2022). Other techniques (Chen et al., 2020a,c; Aghajanyan et al., 2022; Kasner and Dusek, 2022) improve fluency and generalization by leveraging large-scale pretrained language models (Devlin et al., 2019; Raffel et al., 2020). Tables are generally linearized, even when special-purpose techniques are developed for encoding table structure (Wang et al., 2022). Dedicated table understanding techniques (Wang et al., 2021; Jin et al., 2023) eschew linearization but have not been integrated with generation tasks.

Previous attempts to address table-to-text generation from a visual recognition perspective (Dash et al., 2023; Srihari et al., 2003) have relied on OCR methods which first extract text from the image and then feed it as a string to a generation model. Aside from being noisy, these techniques typically embrace a text-centric point of view, treating the image as a limitation rather than an informative modality. Our work builds on recent visual language understanding models (Kim et al., 2022; Davis et al., 2022; Lee et al., 2023) which are based exclusively on pixels and have managed to outperform OCR methods in several natural language processing tasks (Rust et al., 2023; Huang et al., 2022; Masry et al., 2022; Salesky et al., 2023).

The field of Vision Language Models (VLMs) has also experienced significant growth in recent years (Liu et al., 2023; Ye et al., 2023b; Bai et al., 2023; Wang et al., 2023; Alayrac et al., 2022). While most of them focus primarily on natural images, a few are starting to explore the application of dual encoder architectures to visually represented language (Ye et al., 2023a; Zhang et al., 2023). However, these architectures are not parameter lean (with increased model size of a factor of 40 or more compared to Pix2Struct), and some continue to rely on fixed resolution images which can be particularly problematic when processing tabular data.

A few other efforts have recently explored multimodal approaches to processing tables for various tasks, including table-to-text generation. Dash et al. (2023) convert images into HTML tokens which are subsequently linearized and processed by a traditional text-to-text model. Other work (Chen et al., 2023a) focuses on recognizing the structure of tables from images as an independent task. It also leverages multimodal pretraining and unsupervised table structure learning objectives, but ignores the content of table cells and their relations. To the best of our knowledge, our work is the first to conceptualize data-to-text generation as a visually-situated language understanding problem.

3 Problem Formulation

The task of table-to-text generation aims to take a structured table $\boldsymbol{t}$ as input and output a natural language description $\boldsymbol{y} = [y_1, \ldots, y_k]$ where k is the length of the description. Table $\boldsymbol{t}$ is typically reformatted as a sequence of textual records $\boldsymbol{t} = [t_{1,1}, t_{1,2}, \ldots, t_{i,j}, \ldots, t_{m,n}]$ where m and n respectively denote the number of rows and columns of $\boldsymbol{t}$.

We approach this task from a visual recognition perspective, and expect the input table to be an image $\boldsymbol{x}$. The image is reshaped into a sequence of patches analogous to linguistic tokens. More formally, for an input image $\boldsymbol{x} \in R^{H \times W \times C}$ and patch size p, we create N image patches denoted as $x_p \in R^{N \times (P^2 \cdot C)}$. (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = \frac{HW}{P^2}$ the resulting number of patches, which serves effectively as the input sequence length. Our proposed model learns to autoregressively estimate the conditional probability of a text sequence from

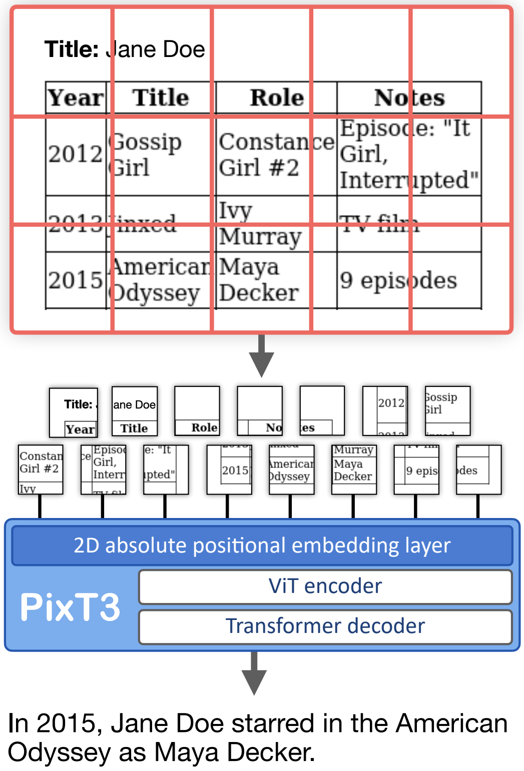

Figure 2: Overview of PixT3 generation model.

a source image as:

$$P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \prod_{i=1}^{n} P(y_i|\boldsymbol{y}_{<i}, \boldsymbol{x}; \boldsymbol{\theta}) \qquad (1)$$

where $\boldsymbol{\theta}$ are transformer parameters and $\boldsymbol{y}_{<i}$ the words decoded thus far.

We further define three generation settings, which manipulate the information provided to the model in terms of content selection (see Appendix B for visualization). In the *tightly-controlled* setting (TControl), the model is given highlighted cells only, ignoring the table. Most recent approaches benchmark model performance in this setting (Wang et al., 2022; An et al., 2022; Chen et al., 2023b; Su et al., 2021; Kale and Rastogi, 2020). In the *loosely controlled* setting (LControl), the model is given highlighted cells *and* the entire table. This is the original setting for which the ToTTo dataset (Parikh et al., 2020a) was constructed. Finally, we introduce the *open-ended* setting (OpenE), where the model is given the table without any highlighting.

4 The PixT3 Model

PixT3 is an image-encoder-text-decoder model based on Pix2Struct (Lee et al., 2023). It expects image rendered tables and generates descriptions thereof (see Figure 2). Pix2Struct is a Vision Transformer model pretrained on 80M screenshots of web pages extracted from URLs in the C4 corpus (Raffel et al., 2020). It splits input images into patches of 16×16 pixels (see Figure 2), linearly embeds each patch, adds position embeddings, and feeds the resulting sequence of vectors to a standard Transformer encoder (Vaswani et al., 2017).

Pix2Struct was first warmed up with a reading curriculum (Rust et al., 2023; Davis et al., 2022), to improve training stability and fine-tuning performance and then pretrained with a screenshot parsing objective; specifically, it generates a simplified version of an HTML subtree that represents a highlighted area of a web page screenshot. It also adds a BART-like (Lewis et al., 2020) learning signal to pretraining by masking 50% of the text in the input and then requiring the model to produce the entire subtree. Importantly for our table-to-text generation task, Pix2Struct supports variable image resolution and multiple aspect ratios. It first re-scales the input (up or down) to extract the maximal number of fixed-size patches that fit within a given sequence length and then replaces the typical 1-dimensional absolute positional embedding with a 2-dimensional one, which adds resolution flexibility and removes any aspect ratio distortion.

We initialize PixT3's model weights with Pix2Struct; we next adopt a curriculum training strategy which instills in our model knowledge about tables and their structure (see Section 4.2); and finally, we fine-tune on table-to-text generation datasets such as ToTTo (Parikh et al., 2020a) with a task-specific supervised objective.

4.1 Table-to-Image Rendering

We parse tables to HTML, and subsequently render them into images. We also render table metadata (e.g., Wikipedia page and section title), if it exists, as part of the image, adding it on top of the table. Tables are rendered into three different images corresponding to the generation settings defined in Section 3 (see Appendix B, Figure 6).

Although Pix2Struct can handle variable resolutions and input patches, very long inputs are nevertheless computationally expensive. Following Lee et al. (2023), we set the maximum input length to 2,048 patches (of 16×16 pixels) which corresponds to a maximum image size of 524,288 pixels. 41.74% of the tables in a dataset like ToTTo (Parikh et al., 2020a) exceed this size (see Figure 5 in Appendix A), with 5% being larger than 8.3M pixels (32,768 patches). Indiscriminately down-scaling *all* images exceeding the maximum input length would negatively affect performance, especially for very big tables, effectively rendering them unreadable (we showcase how image size affects model performance in Figure 4). To avoid this as much as possible, we truncate the image to fit within a maximum down-scaling factor γ. In other words, images are first compressed to $\gamma\%$ of their original size and then truncated from left to right until they fit into 2,048 patches. The optimal value for γ is determined empirically (see Appendix C).

4.2 Structure Learning Curriculum

Pix2Struct is a general-purpose visual language understanding model, and as such it is not particularly knowledgeable about tables and their structure. Tables can be presented in a variety of ways visually, such as spanning multiple columns or rows, with or without horizontal and vertical lines, non-standard spacing and alignment, and text formatting. Aside from presentation, there are various conventions about the underlying semantics of tables and their structure, e.g., each cell is only related to cells in the same column and row. These challenges have led to the development of dedicated table understanding techniques (Jin et al., 2023; Wang et al., 2022) in the domain of text but cannot be readily ported to images.

Instead, we encourage PixT3 to adhere to tabular conventions, by first training it on an intermediate supporting task. This acts as a structure learning curriculum, exposing the model to the rules governing tables. We next elaborate on the intermediate task, its corresponding dataset, and the proposed self-supervised objective.

Dataset for Intermediate Training Existing datasets like ICDAR2021 (Kayal et al., 2021) and TableBank (Li et al., 2019) are representative of the task of parsing table images into their structure and, in theory, could be used for our intermediate training purposes. However, they focus on scientific tables which do not follow the typical distribution of Wikipedia tables found in ToTTo (Parikh et al., 2020a), e.g., in terms of size and cells spanning across rows and columns. We instead propose to create a synthetic image-to-text dataset, making use of the table rendering pipeline described in Section 4.1. Although we generate tables specifically tailored for our use-case, the generation process is

Table:

oY	io	HG	eG2S
Z4ikU	01	aRU	mubk6
URa	dAF		I
I86	GAe	0b	sUr5
L1	3	Vf1	Svaq2

Target:

<<<dAF><<<URa><I>>><<<io><01><GAe><3>><<HG><aRU><0b><Vf1>>>>>

Figure 3: Synthetically generated table with a highlighted cell and corresponding pseudo-HTML target sequence (for self-supervised objective). Cells within the target sequence are highlighted in the table with a colored background. For details on the structure of the target, please refer to Appendix D.

flexible and can be adapted to other domains with different characteristics.

We determine the structure of each table (size, column, and row spans) randomly, following ToTTo's training set distribution. We cap the generation process at a maximum of 20 columns and 75 rows. Table cells are filled with synthetic values consisting of a random combination of one to five random English alphabet characters and digits, functioning as identifiers rather than meaningful values (see Figure 3 for an example). Our dataset contains 135,400 synthetic tables, 120,000 for training, 7,700 for validation, and 7,700 for testing.

Self-supervised Objective While masking is a widely adopted learning objective (Devlin et al., 2019), it does not naturally transfer to our table-to-text generation task; table values are not naturally correlated to neighboring values and thus a masked cell cannot be easily predicted from other cells in its context. Table values could be rearranged so that they correlate to their neighbors, however, early experiments showed that this type of objective does not improve downstream task performance (see Appendix D for details). Another common pretraining objective is table linearization (Chen et al., 2023a), which, however, scales poorly with table size, leading to slow pretraining.

We propose a self-supervised objective that encourages PixT3 to capture the relations between cells within a table while generating a small amount of tokens. Specifically, we highlight a random cell in a synthetically generated table, and train the model to produce a sorted list of cells within the same column and row (see Figure 3). Our objective encapsulates a loose notion of table structure, nudging the model to pay attention to the arrangement of columns and rows around a cell. We follow the same pseudo HTML notation introduced in Pix2Struct to format our output sequence, easing the model's transition from its original screenshot parsing objective to this new one. Note that we consider tables with a heterogeneous structure where cells can span across multiple columns and rows. In such cases, the expected sequence will contain all cells in related rows and columns surrounding the highlighted cell (see Figure 3).

4.3 PixT3 Fine-tuning

The intermediately pre-trained PixT3 is subsequently fine-tuned on an image-rendered dataset (see Section 4.1). In experiments, we use ToTTo (Parikh et al., 2020b), however, our approach is not tied to a particular style of tables. Due to our model's requirement for unimodal input, we treat table-related information (such as its title) as part of the table itself and render them both as one image (see Lee et al. 2023 for a similar approach).

5 Experimental Setup

Model Configuration All our experiments were conducted with the *base* pretrained Pix2Struct[2] model (282M parameters). We trained PixT3 variants for the three table-to-text generation settings defined in Section 3. All PixT3 models were fine-tuned on ToTTo (Parikh et al., 2020a) with tables rendered as images following the procedure outlined in Section 4.1. The maximum down-scaling factor γ was set to 0.39.

PixT3 models were fine-tuned with a batch size of 8 and a gradient accumulation of 32 steps on a single NVIDIA A100 80GB GPU. Checkpoints were selected according to best performance on the validation set. All models used an input sequence length of 2,048 patches and were optimized with AdamW (Loshchilov and Hutter, 2017). We used a learning rate scheduler with a linear warmup of 1,000 steps to 0.0001, followed by cosine decay to 0. The decoder maximum sequence length was set to 50 tokens, which covers 97.49% of the target descriptions in the training data. PixT3 was trained for 1.4k steps with the self-supervised objective described in Section 4.2. Our decoder was not frozen during intermediate training, as initial

[2] https://github.com/google-research/pix2struct

experiments showed that a fully trained model outperformed one with frozen decoder weights. A full list of fine-tuning hyper-parameters can be found in Appendix H.

Datasets We evaluated our model on ToTTo (Parikh et al., 2020a), a large-scale, manually curated dataset representative of several domains and types of tables. We also assessed the generalization capabilities of PixT3 on out-of-distribution tables. We created an out-of-domain benchmark with content selection annotations similar to ToTTo based on Logic2Text (Chen et al., 2020c), an existing dataset which contains a total of 10,161 Wikipedia tables, paired with human-authored descriptions and logical forms. Logic2Text differs from ToTTo in that descriptions are not simple verbalisations of table rows and columns, but require some form of reasoning (e.g., comparisons or counting operations). We were able to automatically trace values mentioned in the logical form back to the cells of the input tables (Alonso and Agirre, 2023), thus obtaining highlighted cell annotations similar ToTTo's (see Appendix E for an example). We report results on the official test set (1,085 examples).

Model Comparison We evaluated PixT3 against several text-only models with similar parameter sizes. These include CoNT (An et al., 2022), the top performant (published) model in the ToTTo leaderboard.[3] CoNT is a text-to-text generation model which makes use of contrastive learning, through improved selection of contrastive examples, a new contrastive loss, and a global decoding strategy. CoNT expects the input table to be converted to a string, and is built on top of T5-base (220M parameters). We also compared against Lattice (Wang et al., 2022), a model which enforces awareness of table layout though pruning the attention flow and encoding cells in a way that is invariant to their relative position in a sequence. This model also uses T5-base and expects linearized input. In addition, we report results with vanilla T5-base which performed competitively on the ToTTo leaderboard without any task specific modifications (Kale and Rastogi, 2020; An et al., 2022). All comparison models and PixT3, were trained on the ToTTo training set in our three generation settings.[4]

[3]A model named SKY appears to slightly outperform CoNT in the leaderboard, however, at the time of writing, we were not able to verify this, i.e., by finding a publication or preprint describing this model.

		Dev		TestN		TestO		Test	
	Model	BL	PR	BL	PR	BL	PR	BL	PR
TControl	T5-base	47.7	57.1	38.9	51.2	55.4	61.1	47.2	56.2
	T5-3B	48.4	57.8	39.3	51.6	55.1	60.7	47.2	56.2
	Lattice	48.0	58.4	40.0	**53.8**	55.9	62.4	48.0	**58.1**
	CoNT	**49.0**	**58.6**	**40.6**	53.7	**56.7**	**62.5**	**48.7**	**58.1**
	PixT3	45.7	55.7	37.5	50.6	53.2	60.4	45.4	55.5
LControl	T5-base	24.5	27.2	19.4	23.9	29.4	30.3	24.5	27.1
	T5-3B	23.6	26.0	18.0	22.4	28.7	29.2	23.4	25.8
	Lattice	24.9	31.0	20.8	27.7	27.5	33.8	24.4	30.8
	CoNT	23.8	29.3	19.2	26.1	28.7	32.3	23.9	29.2
	PixT3	**46.2**	**55.1**	**38.1**	**50.3**	**52.7**	**59.0**	**45.4**	**54.7**
OpenE	T5-base	21.5	23.5	16.8	21.0	26.5	26.5	21.7	23.8
	T5-3B	20.8	22.9	16.7	20.3	25.5	25.5	21.2	22.9
	Lattice	20.9	26.1	17.6	24.3	23.7	27.6	20.8	25.9
	CoNT	21.7	25.8	16.9	23.2	26.3	28.3	21.6	25.8
	PixT3	**24.8**	**28.3**	**20.5**	**26.3**	**28.9**	**30.3**	**24.7**	**28.3**

Table 1: Automatic evaluation results on ToTTo in three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE). We report BLEU (BL) and PARENT (PR) results on the development (Dev) and Test sets, including the overlapping (TestO) and non-overlapping (TestN) test set splits. BLEURT results are in Appendix E.

For our out-of-domain experiments, we also compare against LLaVA-1.5 (Liu et al., 2023), a large pretrained multimodal model (13B parameters) which is built on top of the CLIP visual encoder (Radford et al., 2021) and the Vicuna-7B language model (Zheng et al., 2023), and fine-tuned on vision-language instructions. LLaVA has not been fine-tuned specifically for table-to-text generation, however, it is interesting to see if sufficiently large scale is all it takes to do well on the table-to-text generation task. LLaVA can only handle a *single* image at each forward pass. This limitation prevents it from performing inference in an in-context learning setting, where the model has access to multiple input-output examples at the same time. To approximate in-context learning as closely as possible, we provided LLaVA with an image, an instruction, and three table descriptions as output examples for each generation setting (see Appendix F for details). We summarize the number of parameters for all comparison models in Table 2.

we do still provide a few description examples in our prompt to ensure a fair zero-shot comparison. All prompts used for LLaVA in this evaluation can be found in Appendix F.

[4]Comparison models were trained with the authors' publicly available scripts.

6 Results

PixT3 is the best performing model in loosely controlled and open-ended generation settings. Table 1 summarizes our results on ToTTo in our three generation settings. We evaluated model performance automatically with the same metrics used to rank participant systems in the ToTTo leaderboard. These include BLEU (Papineni et al., 2002) which is as a proxy for fluency, PARENT (Dhingra et al., 2019), a metric proposed specifically for data-to-text evaluation that takes the table into account, serving as a proxy of faithfulness, and BLEURT (Sellam et al., 2020); the latter is a composite metric that takes a reference and model output as input, and returns a score that indicates the extent to which the output is fluent and conveys the meaning of the reference. Note that ToTTo features two splits in the development/test set containing tables whose header values are present (overlapping split) and absent (non-overlapping split) in the training set. Results on the test set, which is not publicly available, were obtained via submitting to the ToTTo leaderboard.

We first discuss our results on the tightly controlled generation setting (TControl) where models are not given the full table, just the highlighted cells. We would not expect PixT3 to excel at this setting, which is better suited to text-to-text models (highlighted cells make for non-descriptive images, see Appendix B, Figure 6). PixT3 is indeed unable to outperform CoNT, Lattice, and related T5 variants, falling 3.5 BLEU points behind on the development set and 3.7 on the test set. However, LControl, the loosely controlled generation setting, better showcases the advantages of PixT3, which in this case demonstrates almost a two times improvement over CoNT and T5 models. Performance degrades drastically for all systems in the open-ended setting (OpenE) which is challenging; models are expected to perform content selection in addition to text generation, and could produce table descriptions which are valid but different from the reference. Automatic metrics based on n-gram overlap are particularly punitive in this case. Nevertheless, PixT3 is superior to CoNT, Lattice, and T5 across evaluation metrics.

PixT3 generalizes to out-of-domain tables which require reasoning skills. We next evaluate whether PixT3 generalizes to unseen tables, outside ToTTo's distribution. Table 2 shows our results on Logic2Text (Chen et al., 2020c), again following the three generation settings. Compared to ToTTo, Logic2Text is a more challenging dataset as most descriptions rely on reasoning over the entire table. This results in poor model performance in the TControl setting which does not include the table as input. Nonetheless, we observe that PixT3 excels at the LControl setting, even though it has to process and reason over the entire table. The OpenE setting is challenging for all models as they are asked to identify interesting cells to talk about in *out-of-domain* tables. PixT3 still maintains an edge over T5 and LLaVA, performing on par with CoNT and Lattice. We observe that LLaVA cannot match the performance of PixT3 and T5-based models. This underscores the importance of task-specific fine-tuning over parameter size. We present output examples in Appendix E.

	Model	Size	BLEU	PARENT
TControl	LLaVA	13B	12.6	34.36
	T5-base	220M	16.8	55.97
	T5-3B	3B	17.7	52.75
	Lattice	220M	19.8	61.05
	CoNT	220M	18.8	61.73
	PixT3	282M	**20.6**	**61.86**
LControl	LLaVA	13B	5.9	23.18
	T5-base	220M	11.5	40.02
	T5-3B	3B	10.9	35.45
	Lattice	220M	11.5	40.02
	CoNT	220M	11.8	43.25
	PixT3	282M	**21.5**	**56.45**
OpenE	LLaVA	13B	6.7	20.14
	T5-base	220M	7.9	30.67
	T5-3B	3B	9.5	29.47
	Lattice	220M	**11.7**	**38.12**
	CoNT	220M	11.0	36.94
	PixT3	282M	11.4	35.68

Table 2: Automatic evaluation results on Logic2Text in three generation settings: tightly controlled (LControl), loosely controlled (LControl), and open-ended (OpenE). All models (except LLAVA) were fine-tuned on ToTTo and tested on Logic2Text. BLEURT results are in Appendix E.

PixT3 is robust against table input size. In Figure 4, we analyze the effect of table size on model performance. As can be seen, T5, Lattice, and CoNT are severely affected: the bigger the table, the less accurate the generated description. PixT3 is evidently more robust, showing degradation in performance only for very big tables. We also examined whether PixT3 has an edge because of its ability to encode longer inputs. Recall that CoNT, Lattice, and T5-base utilize a fixed input length

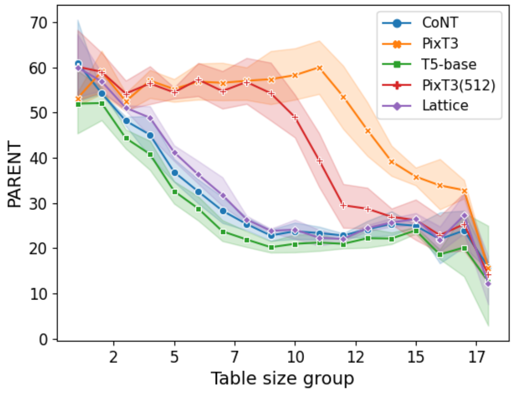

Figure 4: Model performance (CoNT, T5, PixT3, Lattice, and PixT3 with 512 patch input size) in the LControl setting across 18 table size groups (logarithmic scale). Upper and lower bounds in shaded areas correspond to results for the overlapping and non-overlapping ToTTo splits, while central points correspond to results overall. We report results with PARENT, other metrics show similar tendencies. We refer to Appendix A for further details.

of 512 tokens, while PixT3 uses 2,048 patches. We thus trained a PixT3 variant with input length set to 512 patches. As shown in Figure 4, the more constrained PixT3 model is slightly worse and more likely to degrade with increased table size but consistently outperforms CoNT, Lattice, and T5.

The structure learning curriculum improves generation quality across metrics. In Table 3 we perform an ablation study comparing PixT3 with and without our structure learning curriculum and self-supervised objective (Section 4.2). For both models we follow the same fine-tuning process: we render tables into images, identify the optimal point of image compression and truncation (see Section 4.1), and perform hyper-parameter search to optimize Pix2Struct-base for our task. Vanilla PixT3 (second row in Table 3) shows a substantial improvement over an out-of-the-box Pix2Struct model which achieves a BLEU score of 0.2 and PARENT score of 0.6 on the ToTTo development set. Adding the intermediate training curriculum (second row in Table 3) slightly improves vanilla PixT3 across evaluation metrics.

Manual inspection of the descriptions produced by the two PixT3 model variants reveals they are often semantically equivalent to the target (43% of the time). Nevertheless, the intermediate training curriculum substantially reduces structure-based faithfulness errors, especially in the OpenE setting. On a sample of 200 outputs (randomly selected

	Dev			Test		
Models	BL	PR	BRT	BL	PR	BRT
Pix2Struct	0.2	0.6	-1.433	—	—	—
PixT3 (W/o SLC)	38.7	46.0	-0.003	38.3	45.6	0.001
PixT3 (With SLC)	**39.2**	**46.5**	**0.008**	**38.7**	**46.3**	**0.007**

Table 3: PixT3 with and without structure learning curriculum (SLC); we report results on the ToTTo development (Dev) and Test set with BLEU (BL), PARENT (PR), and BLEURT (BRT), averaged across the three generation settings.

from the development set), we found that 23% of the descriptions produced by vanilla PixT3 disregard or misinterpret the structure of the table. Structural faithfulness errors reduce to 7% when PixT3 is trained with our structure learning curriculum.

PixT3 is most faithful in loosely controlled and open-ended generation settings. We further conducted a human evaluation study to quantify the extent to which the generated descriptions are faithful to the table. We evaluated PixT3, and the two best performing text-only systems (CoNT, and Lattice) on two sets of 100 randomly selected table-description pairs from ToTTo (development set) and Logic2Text (test set), in the three generation settings. Crowdworkers were presented with an uncompressed image of a table, its page and section title, and a model generated description. As an upper bound, we also elicited judgments for the human curated reference descriptions for the same ToTTo and Logic2Text examples. Participants were asked to determine whether a description was "True" or "False" based on the information provided in the table and/or its title and subtitle (see instructions in Appendix G). Overall we elicited 7,200 judgments (100 examples $\times$ 3 generation settings $\times$ 4 model descriptions $\times$ 3 participants $\times$ 2 datasets). Crowdworkers were recruited using the online platform Prolific.[5]

Table 4 shows the results of the human evaluation, specifically the proportion of descriptions deemed faithful. As expected, the human authored Reference description is consistently faithful across generation settings. CoNT is more faithful in TControl but deteriorates in the LControl and OpenE settings. We further examined whether differences among systems are statistically significant using paired bootstrap resampling. PixT3 is significantly worse ($p < 0.05$) than the Reference in TControl

[5] https://www.prolific.com

	Model	TControl	LControl	OpenE
ToTTo	Reference	87	84	89
	Lattice	**79**	16	20
	CoNT	76	16	35
	PixT3	69	**72**	**78**
L2T	Reference	81	87	86
	Lattice	34	3	16
	CoNT	**35**	3	26
	PixT3	32	**40**	**60**

Table 4: Human evaluation results on ToTTo and Logic2Text (L2T). Proportion of descriptions rated as faithful for PixT3, CoNT, and Reference in three generation settings: tightly controlled (LControl), loosely controlled (LControl), and open-ended (OpenE).

but not CoNT or Lattice. In LControl all differences between systems are statistically significant ($p < 0.05$). In OpenE, PixT3 is significantly different ($p < 0.05$) from CoNT and Lattice but not from the Reference. Inter-rater agreement was moderate with a Fleiss' Kappa coefficient of 0.55 (Fleiss, 1971).

7 Conclusion

In this paper, we leverage the capabilities of Vision Transformers to recast table-to-text generation as a visual recognition task, removing the need for rendering the input in a string format. Our model, PixT3, introduces a new training curriculum and self-supervised learning objective in order to capture the structure and semantics of tables. Experiments across constrained and open-ended generation settings show it is robust to different table sizes, performing competitively and often better than state-of-the-art models. PixT3 is also able to handle new domains with unseen tables, as evidenced by our results on Logic2Text, a new dataset which we propose for assessing the generalization capabilities of table-to-text generation models.

Avenues for future research are many and varied. There are several downstream tasks which stand to benefit from a pixel-based view of textual information, including multilingual table-to-text generation, and semantic parsing. We would also like to investigate additional objectives and inductive biases that can better capture the structure of tables and inter-cell dependencies.

8 Limitations

While PixT3 shows promising results, its performance is affected by the dimension of the input tables (for instance, 16% of the Wikipedia tables in ToTTo remain too big for PixT3 to represent effectively). It would be interesting to look into alternative ways of preprocessing very large tables, e.g., by rendering them via multiple images. While our proposed intermediate training methodology mitigates faithfulness errors, the model still struggles with hallucinations, falling short of human-level performance.

Finally, PixT3, as well as other comparison systems, have limited reasoning capabilities, e.g., they cannot infer information which is not explicitly stated in the table or make logical connections between concepts. PixT3's superior performance in terms of faithfulness on Logic2Text (see Table 4) is due to generating simpler sentences rather than superior reasoning skills. Thus, aside from new training objectives, a promising direction would be to combine the visual representations with an intermediate planning component that encourages the model to reason about the input while generating the output.

Acknowledgements

We thank the meta-reviewer and anonymous reviewers for their constructive feedback. The authors also thank Ander Salaberria for his insightful comments on earlier versions of this work. We gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1), the Basque Government (Research group funding IT-1805-22), MCIN/AEI/10.13039/501100011033 project AWARE (TED2021-131617B-I00), European Union NextGenerationEU/PRTR, and the LUMINOUS project (HORIZON-CL4-2023-HUMAN-01-21-101135724).

References

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2022. HTLM: Hyper-text pre-training and prompting of language models. In *International Conference on Learning Representations*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022.

Flamingo: a visual language model for few-shot learning.

Iñigo Alonso and Eneko Agirre. 2023. Automatic logical forms improve fidelity in table-to-text generation. *Expert Systems with Applications*, page 121869.

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 2197–2210. Curran Associates, Inc.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Leiyuan Chen, Chengsong Huang, Xiaoqing Zheng, Jinshu Lin, and Xuanjing Huang. 2023a. TableVLM: Multi-modal pre-training for table structure recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2437–2449, Toronto, Canada. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020b. KGPT: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online. Association for Computational Linguistics.

Xi Chen, Xinjiang Lu, Haoran Xin, Wenjun Peng, Haoyang Duan, Feihu Jiang, Jingbo Zhou, and Hui Xiong. 2023b. A table-to-text framework with heterogeneous multidominance attention and self-evaluated multi-pass deliberation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 607–620, Singapore. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020c. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111, Online. Association for Computational Linguistics.

Michael A Covington. 2001. Building natural language generation systems. *Language*, 77(3):611–612.

Amanda Dash, Melissa Cote, and Alexandra Branzan Albu. 2023. Weathergov+: A table recognition and summarization dataset to bridge the gap between document image analysis and natural language generation. In *Proceedings of the ACM Symposium on Document Engineering 2023*, DocEng '23, New York, NY, USA. Association for Computing Machinery.

Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. 2022. End-to-end document recognition and understanding with dessurt. In *European Conference on Computer Vision*, pages 280–296. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

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*, MM '22, page 4083–4091, New York, NY, USA. Association for Computing Machinery.

Rihui Jin, Jianan Wang, Wei Tan, Yongrui Chen, Guilin Qi, and Wang Hao. 2023. TabPrompt: Graph-based pre-training and prompting for few-shot table understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7373–7383, Singapore. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of* *the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Zdeněk Kasner and Ondrej Dusek. 2022. Neural pipeline for zero-shot data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3914–3932, Dublin, Ireland. Association for Computational Linguistics.

Pratik Kayal, Mrinal Anand, Harsh Desai, and Mayank Singh. 2021. Icdar 2021 competition on scientific table image recognition to latex. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16*, pages 754–766. Springer.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*.

Flip Korn, Xuezhi Wang, You Wu, and Cong Yu. 2019. Automatically generating interesting facts from wikipedia tables. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 349–361, New York, NY, USA. Association for Computing Machinery.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2019. Tablebank: A benchmark dataset for table detection and recognition.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Joy Mahapatra and Utpal Garain. 2021. Exploring structural encoding for data-to-text generation. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 404–415, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020a. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020b. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of EMNLP*.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Ratish Puduppully, Yao Fu, and Mirella Lapata. 2022. Data-to-text generation with variational sequential planning. *Transactions of the Association for Computational Linguistics (to appear)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023. Multilingual pixel representations for translation and effective cross-lingual transfer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13845–13861, Singapore. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Sargur N. Srihari, Ajay Shekhawat, and Stephen W. Lam. 2003. *Optical Character Recognition (OCR)*, page 1326–1333. John Wiley and Sons Ltd., GBR.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048, Seattle, United States. Association for Computational Linguistics.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. Cogvlm: Visual expert for pretrained language models.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1780–1790, New York, NY, USA. Association for Computing Machinery.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Brussels, Belgium. Association for Computational Linguistics.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, Qin Jin, Liang He, Xin Lin, and Fei Huang. 2023a. UReader: Universal OCR-free visually-situated language understanding with multimodal large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2841–2858, Singapore. Association for Computational Linguistics.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023b. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2023. Llavar: Enhanced visual instruction tuning for text-rich image understanding.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

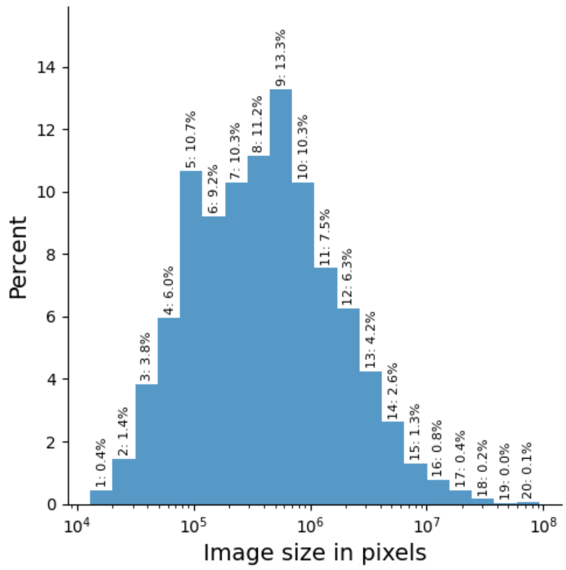

Figure 5: Proportion of ToTTo examples (development set) per table size (shown in logarithmic scale).

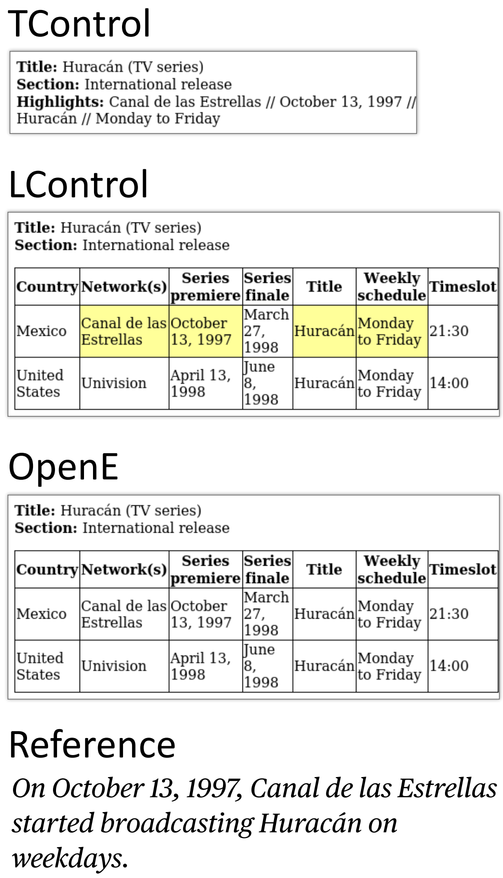

Figure 6: PixT3 input image examples (and reference) in three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE).

A Table Size Distribution in ToTTo

We measure the size of a table by the total amount of pixels in its corresponding rendered image. We then calculate the distribution of each size, and group tables into 20 buckets accordingly. Each bucket covers a logarithmically increasing amount of table sizes. Figure 5 shows the resulting buckets and the proportion of ToTTo examples in each (development set). The quality of descriptions generated within each group, are evaluated in Section 6, see Figure 4.

B Table-to-Text Generation Settings

Figure 6 illustrates how the image input to PixT3 differs according to three generation settings: tightly controlled (the model is given only highlighted cells, no table), loosely controlled (the model is given the table and highlighted cells), and open-ended (the model is given the table without any highlighting).

C Image Truncation and Down-scaling

We explored the impact of down-scaling on model performance and its tradeoff with truncation. We conducted a series of experiments wherein PixT3 models were trained on versions of ToTTo with varying down-scaling factor γ: 0.87, 0.58, 0.39, 0.26, and 0.00. Note that γ=0.00 corresponds to a setting where no truncation takes place, only down-scaling. According to the results shown in Table 5, it is best to combine truncation with down-scaling, none of the extreme settings (no truncation vs too much truncation) are beneficial. The optimal γ value is 0.39.

D Intermediate Training

Synthetic Dataset Generation In this section we provide a more detailed description regarding the generation of synthetic tables for intermediate training. As our goal was to generate tables with a structure similar to ToTTo, we first measured the probability distribution of columns, rows, column spans and row spans for the tables in the training set to avoid over-fitting and contamination. We observed that the distribution of columns (up to 20 columns) remained almost constant across tables, and did not affect the probability distribution of rows. As a result, we aggregated row numbers across columns and computed a single distribution for rows to simplify our generation task, using discrete probability distributions. In order to limit the size of the generated tables we cap the number of columns and rows to 20 and 75, respectively. For

Epoch \ γ	0.00	0.26	0.39	0.57	0.87
16	28.71	29.13	29.47	**29.58**	27.47
17	28.99	29.53	**29.99**	29.70	27.69
18	29.67	30.04	**30.55**	30.21	28.13
19	29.98	30.04	**30.63**	30.54	28.33
20	29.83	30.21	**30.68**	30.53	29.39

Table 5: Evaluation results (BLUE) for PixT3 model in tightly controlled generation setting for different γ down-scaling factors. We show the Last five epochs on the ToTTo training set.

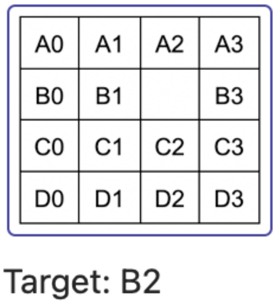

Figure 7: Synthetically generated table with masked cell. Filled cell values denote position in the table.

		Dev Set (All)	Test Set (Non)	Test Set (Over)	Test Set (All)
	Model	BLEURT	BLEURT	BLEURT	BLEURT
TControl	T5-base	0.233	0.106	0.354	0.230
	T5-3B	0.228	0.104	0.344	0.224
	Lattice	0.226	0.103	0.348	0.226
	CoNT	**0.240**	0.116	0.364	0.240
	PixT3	0.178	0.044	0.312	0.178
LControl	T5-base	−0.298	−0.395	−0.191	-0.293
	T5-3B	−0.309	−0.416	−0.194	-0.305
	Lattice	−0.287	−0.382	−0.195	-0.288
	CoNT	−0.293	−0.387	−0.190	-0.289
	PixT3	**0.169**	**0.047**	**0.287**	**0.167**
OpenE	T5-base	−0.371	−0.458	−0.278	-0.368
	T5-3B	−0.385	−0.456	−0.301	-0.378
	Lattice	−0.377	−0.451	−0.302	-0.377
	CoNT	−0.370	−0.452	−0.281	-0.366
	PixT3	**−0.332**	**−0.414**	**−0.258**	**−0.336**

Table 6: BLEURT results on ToTTo for T5, PixT3, Lattice, and CoNT in three generation settings: tightly controlled (LControl), loosely controlled (LControl), and open-ended (OpenE). In the TControl setting, T5 results are taken from Kale and Rastogi (2020) and CoNT results from An et al. (2022). This table complements results reported in Table 1.

the synthetic text within the cells, we randomly generated digits in the [1–5] range and character sequences from [A–Z, a–z] which gave us a total of 776,520,240 permutations of possible unique cell values.

Overall, we generated 120K tables accompanied with target pseudo HTML descriptions. The latter were on average 121 tokens long, with the longest sequences containing 877 tokens. In experiments, we observed that text size affects mainly the average count of tokens, whereas the number of table columns and rows influences the length of the target sequences. The sequences follow a hierarchical structure defined by the characters < and >. In the first hierarchical level, one container can be found for each highlighted cell in the table. Each container includes, in the following order, the highlighted cell, the cells in all related columns, and all cells in all related rows. This structure can represent multiple related columns and rows per highlighted cell, as well as multiple highlighted cells per table.

Alternative Objectives We conducted a set of experiments to identify the best self-supervised objective for our structure learning curriculum. In addition to the objective presented in Section 4.2, we also experimented with a masking objective. Specifically, given a randomly generated table, we filled each cell with text indicative of its position in the table. We then masked random cells and the model was trained to predict the missing cell values (see Figure 7 for an example). We empirically observed that this objective led to worse performance compared to PixT3, even though it resulted in relatively fast training, since the table can be converted into a sequence with a small number of tokens. We hypothesize that this objective only weakly enforces table structure learning as the model does not need to pay attention to all the cells in a column and row to guess the missing value but simply rely on its closest neighbors. We also experimented with a combination of the masking objective discussed here and the structure learning objective described in Section 4.2. However, this model still lagged behind PixT3.

E Additional Results and Examples

In addition to BLEU and PARENT reported in Tables 1 and 2, we also present results with BLEURT in Table 6 and Table 7. We further show example output on the Logic2Text dataset (zero-shot setting) in Figure 8. In the TControl setting, CoNT struggles to produce a coherent sentence, while PixT3 generates a faithful but not very informative one. This is not surprising as the models receive nothing but the title and highlighted cells, making it extremely difficult to generate the target sentence. In LControl, both models have access to the entire table; however, they still produce a false statement,

	Model	BLEURT
TControl	LLaVA	−1.230
	T5-base	−1.086
	T5-3B	−1.079
	Lattice	**−1.060**
	CoNT	−1.103
	PixT3	−1.104
LControl	LLaVA	−1.189
	T5-base	−1.147
	T5-3B	−1.167
	Lattice	−1.147
	CoNT	−1.159
	PixT3	**−1.073**
OpenE	LLaVA	**−1.184**
	T5-base	−1.237
	T5-3B	−1.196
	Lattice	−1.231
	CoNT	−1.231
	PixT3	−1.213

Table 7: Automatic evaluation results on Logic2Text in three generation settings: tightly controlled (LControl), loosely controlled (LControl), and open-ended (OpenE). All models (except LLAVA) were fine-tuned on ToTTo and tested on the Logic2Text. This table complements results reported in Table 2.

most likely a consequence of the zero-shot nature of our generation task. Finally, in the less constrained OpenE setting, PixT3 generates a coherent and faithful sentence. While CoNT also produces a fluent sentence, it incurs a faithfulness error when mentioning "(+5)" instead of "(-5)". This is likely due to the performance degradation this model experiences when provided with the full table.

F LLaVA promts

As mentioned in Section 5, our zero-shot experiments involved comparisons against LLaVA-1.5 (Liu et al., 2023), a large pretrained multimodal model (13B parameters). We devised the following prompts for each generation setting:

TControl "Here are some descriptions based on other highlights of other tables 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description based on the following highlighted cells extracted form a table."

LControl "Here are some descriptions based on the highlights of other tables not present in the input: 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description based on the highlighted cells in this table following the same style as the example descriptions."

OpenE "Here are some descriptions from other tables not present in the input: 'chilawathurai had the 2nd lowest population density among main towns in the mannar district .', 'zhou mi only played in one bwf super series masters finals tournament .', 'tobey maguire appeared in vanity fair later than mike piazza in 2003 .'. Now write a short description stating something from this table following the same style as the example descriptions."

G Human Evaluation Guidelines

We provide the full set of instructions presented to crowdworkers for the human evaluation study. Our participants were native English speakers from the United Kingdom and the United States of America, with a 50/50 equal gender split between male and female.

Thank you for taking part in our experiment! You will be presented with a table and a computer-generated description of its content. Your task is to determine whether each description is "True" or "False" based on the information provided in the table and/or its title and subtitle (you will see examples later-on). No expert knowledge is required to perform this task. You should evaluate the descriptions given the information presented in the table, without taking any other information into account (e.g., based on your own knowledge or the web).

Here are some guidelines to help you with your evaluation:

Acronyms: tables often have acronyms which the descriptions might spell out. For example, if the table mentions "TD" and the description correctly spells it out as "touch down," you should not consider this "False" (although the description might be false for other reasons).

Implicit information: the description might mention information that can be inferred but is not explicitly spelled-out in the table. For example, it could mention "steam engines" when the table lists theirs names without explicitly

Title: 1973 u.s. open (golf)

place	player	country	score	to par
1	gary player	south africa	67 + 70 = 137	- 5
2	jim colbert	united states	70 + 68 = 138	- 4
t3	jack nicklaus	united states	71 + 69 = 140	- 2
t3	johnny miller	united states	71 + 69 = 140	- 2
t3	bob charles	new zealand	71 + 69 = 140	- 2
t6	gene borek	united states	77 + 65 = 142	e
t6	julius boros	united states	73 + 69 = 142	e
t6	tom weiskopf	united states	73 + 69 = 142	e
t6	arnold palmer	united states	71 + 71 = 142	e
t6	lee trevino	united states	70 + 72 = 142	e

- **Reference:** Jim Colbert has the second best number of strokes to par.
- **CoNT (TControl):** Jim Colbert led the 1973 U.S. open (golf course) with a score of to par.
- **PixT3 (TControl):** Jim Colbert took part in the 1973 U.S. open (golf) tournament.
- **CoNT (LControl):** At the 1973 U.S. open (golf), Jim Colbert shot a record of 267 (+1) and finished four strokes ahead of runner-up Lee Janzen.
- **PixT3 (LControl):** Jim Colbert had a score of 142.
- **CoNT (OpenE):** Gary Player scored 137 (+5) and finished five strokes ahead of runner-up Jim Colbert.
- **PixT3 (OpenE):** Gary Player won the 1973 U.S. Open (golf) with a score of 137.

Figure 8: Logic2Text table and model output in three generation settings: tightly controlled (TControl), loosely controlled (LControl), and open-ended (OpenE).

talking about steam engines. In this case, the description should not be considered "False".

- You should evaluate each description independently.

- If the description does not make sense and is impossible to evaluate (usually when summarizing very large tables), you should consider it as "False".

We suggest starting by reading the description and then referring to the table to verify if it aligns with its claims.

This data elicitation study is performed by researchers at [REDACTED]. If you have any questions, feel free to contact [REDACTED]. Participation in this research is voluntary. You have the right to withdraw from the experiment at any time. The collected data will be used for research purposes only. We will not collect any personal information. Your responses will be linked to your anonymous Prolific ID for the exclusive purpose of conducting our experiment.

H PixT3 Fine-tuning Hyper-parameters

PixT3 models across all three settings (TControl, LControl, OpenE) were fine-tuned using the same hyper-parameters. To prevent over-fitting, we employed early stopping based on the BLEU score computed on the validation set every 250 steps. Table 8 enumerates the specific hyper-parameter values used in PixT3, with all remaining parameters set to the default values defined in Pix2Struct (Lee et al., 2023).

Hyperparameter	Value
Optimizer	AdamW
Learning rate	0.0001
Warm-up steps	1000
Max. input patches	2048
Shuffle train data	False
Epochs	30
Train batch size	8
Gradient accum. steps	32
Mixed precision	fp16
Evaluation batch size	32
Eval freq. steps	250
Inf. beam search	8 beams

Table 8: Hyperparameters used in PixT3.