Dual Modalities of Text: Visual and Textual Generative Pre-Training

Anonymous ACL submission

Abstract

Harnessing visual texts represents a burgeoning frontier in the evolution of language modeling. In this paper, we introduce a novel pre-training framework for a suite of pixel-based autoregressive language models, pre-training on a corpus of over 400 million document images. Our approach is characterized by a dual-modality training regimen, engaging both visual data through next patch prediction with a regression head and/or textual data via next token prediction with a classification head. This study is particularly focused on investigating the synergistic interplay between visual and textual modalities of language. Our comprehensive evaluation across a diverse array of benchmarks reveals that the confluence of visual and textual data substantially augments the efficacy of pixel-based language models. Notably, our findings show that a unidirectional pixel-based model, *devoid* of textual data during training, can match the performance levels of advanced bidirectional pixel-based models on various language understanding benchmarks. This work highlights the considerable untapped potential of integrating visual and textual information for language modeling purposes. We will release our code, data, and checkpoints to inspire further research advancement.

1 Introduction

The landscape of large language models (LLMs) is undergoing a significant transformation, with advancements that extend the boundaries of language assistant (Touvron et al., 2023a), code generation (Lozhkov et al., 2024; Chai et al., 2023), and multimodal comprehension (OpenAI, 2023; Anil et al., 2023). These models traditionally tokenize input data into discrete elements, treating them as sequences of identifiers, thereby enabling diverse applications. However, this approach often struggles with visually enriched textual content, such as PDFs, where direct parsing into text incurs significant information loss. Traditional methodologies typically employ pre-trained optical character recognition (OCR) tools for extracting information from such visual texts, but these methods are inherently limited by the fidelity of text extraction.

In response to these challenges, a novel paradigm of pixel-based language modeling has emerged, offering a direct pathway to learning from text as visual data (images), transcending the constraints of textual modality (Rust et al., 2023; Tschannen et al., 2023). This approach promises to surmount the *vocabulary bottleneck* issue (Rust et al., 2023)—a trade-off inherent in balancing input encoding granularity against the computational feasibility of vocabulary probability estimation in conventional language models.

In the previous literature, the development of pixel-based language models has been bifurcated into encoder-based (Rust et al., 2023; Tschannen et al., 2023) or encoder-decoder architectures (Salesky et al., 2023), encompassing models that either employ bidirectional mechanisms akin to MAE (He et al., 2022) or utilize an encoder-decoder framework, where a pixel-based model serves as the encoder, paired with a unidirectional language decoder. Despite these advancements, the exploration of pixel-based models employing a decoder-centric approach remains in its infancy.

Moreover, current research often processes visual text as 8-bit grayscale (Rust et al., 2023) or 2-bit binary images (Tai et al., 2024). This approach restricts the representation of color, critical for elements like emojis and font highlights, and diverges from the natural image format in RGB. Notably, there appears to be a lack of studies pre-training on RGB images, which could more accurately reflect the complexities of visual text.

This research aims to fill these gaps by offering a comprehensive examination of the effects of pixel-based versus text-based pre-training within an autoregressive language modeling context. Our study is steered by three critical research questions:

RQ1: Feasibility of tokenization-free autogressive pre-training on visual text images. Can an autoregressive language model trained solely on raw images of visual texts achieve competitive performance?

RQ2: Impact of autoregressive pixel pre-training on multilingual tasks. We explore whether autoregressive pixel pre-training can overcome the *vocabulary bottleneck* in multilingual contexts, assessing its effectiveness in generalizing linguistic features across languages.

RQ3: Synergistic effects of multimodal pre-training. How do pixel-based and text-based pre-training synergize, and in what ways does this multimodal strategy enhance the model's performance on language understanding tasks and its cross-lingual applicability?

Contributions #1) We empirically demonstrate the substantial potential of integrating visual text images for enhanced language model training, proposing the first tokenization-free autoregressive language models on *real-valued* pixels and indicating promising directions for future scaling.

#2) We systematically explore autoregressive pre-training on both visual text images and plain text modalities, demonstrating the potential of causal language models to effectively learn from visual text images and highlighting the interplay between different modalities.

#3) We show that pre-training decoder-only transformers on visual images can match or slightly underperform compared to text-based inputs but achieve competitive results with bidirectional PIXEL models (Rust et al., 2023). This illustrates the potential for scaling trends to eventually surpass text-based pre-trained models.

#4) We construct a comprehensive visual text dataset of over 400 million documents for pixel-based pre-training, equivalent to roughly 236 billion text tokens. We will release the fine-tuning datasets for language understanding and multilingual evaluation, facilitating further research in this emerging field.

2 Related Work

Pixel Representations for Text Advances in pixel-based language modeling have increasingly focused on exploiting the orthographic and typographic properties of text through visual representations. PIXEL (Rust et al., 2023) utilizes masked auto-encoders to address the vocabulary bottleneck by reconstructing pixels in masked text images. Moreover, CLIPPO (Tschannen et al., 2023) demonstrates enhanced language comprehension using a unified encoder for both image and text modalities. Further research by Lotz et al. (2023) evaluates the impact of rendering techniques on the efficacy of pixel-based encoders. These studies primarily utilize bidirectional encoders and process text as grayscale images.

In contrast, our approach leverages RGB imaging to render text, employing a 24-bit color depth to enrich the visual data interpretation. This enhancement allows for handling of elements like emojis and colored text, prevalent in digital communications. Concurrent work by Tai et al. (2024) explores *binary* image rendering and binary cross-entropy loss in discrete space, whereas we implement a mean square error loss in continuous pixel space for finer reconstruction granularity. Moreover, research such as OCR-free visually-rich document understanding (Kim et al., 2022), which focuses on direct learning from visual document images, shares similarities with our approach. However, our work distinctively explores rendered text, expanding the potential for comprehensive multimodal text pre-training.

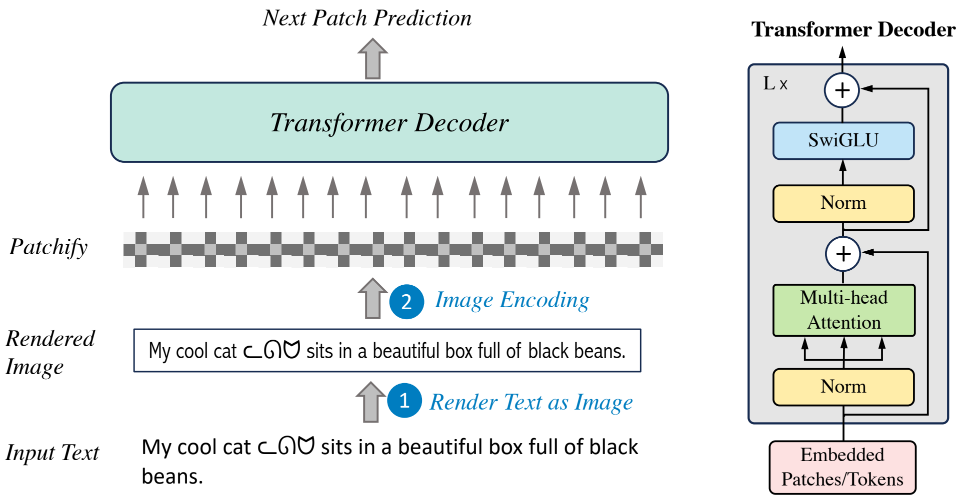
Autoregressive Pre-training on Pixels Existing methods in pixel-based autoregressive pre-training divide into vector quantization techniques—transforming continuous images into discrete tokens—and direct pixel prediction. These approaches include VQ-VAE (Van Den Oord et al., 2017) and VQGAN (Esser et al., 2021) followed by next token prediction (Chen et al., 2020; Ramesh et al., 2021), and prefix language modeling that predicts future visual patches from bidirectional pixel contexts (El-Nouby et al., 2024).

These models are trained on regular images. Our research diverges by focusing exclusively on visual and rendered texts, thereby extending the capability of autoregressive models to understand and generate language from its visual form.

3 Pre-training on Pixels and Texts

3.1 Rendering Text as Images

Following Rust et al. (2023), we utilize text renderer adept at converting textual data into a visually-rich RGB format. This pivotal component takes input text and transforms it into a detailed RGB image, $x \in \mathbb{R}^{H \times W \times C}$. We define the height



(a) Visual text image pre-training (*PixelGPT*). (b) Model architecture.

Figure 1: Illustration of pixel-based autoregressive pre-training.

(H) at 16 pixels and the width (W) at 16,384 pixels, encapsulating the text within a 24-bit color depth across three channels ($C = 3$), thus forming a visual text image that represents a grid of 1024 patches, each 16x16 pixels in size.

The text renderer supports rendering required for a diverse set of textual representations, including multicolored emojis, bidirectional text systems, and scripts necessitating the use of ligatures. In alignment with models like PIXEL, our text sequences may be single paragraphs or pairs of related segments. We use 16x16 black patches as visual cues for end-of-sequence (EOS) marker. These patches are treated as non-interactive elements by our model, where no attention mechanism is engaged or loss calculated.

When confronted with sequences that surpass the maximum length threshold, our model employs strategies of truncation or segmentation into multiple sequences, ensuring efficient processing while preserving contextual integrity. We refer to Appendix §A for the rendering details.

3.2 Input Representation

The transformer decoder ingests a linear sequence of embeddings, each derived from discrete patches of image data or textual tokens, for visual or text inputs, respectively.

Image Input Inspired by the Vision Transformer (ViT; Dosovitskiy et al., 2020), our method tailors the image patch processing paradigm to the sequential processing needs of autoregressive transformer decoders handling visual text imagery, as shown in Figure 1(a). This process commences by rendering the textual input as RGB images $x \in \mathbb{R}^{H \times W \times C}$ as aforementioned in §3.1, subsequently partitioning these into uniform patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ illustrated as Figure 8, where (H, W) defines the original image's resolution, (P, P) specifies each patch's resolution with $P = H$, and $N = W/P$ denotes the total number of patches. The patches are then flattened, mapped to a D-dimensional space through a learnable linear projection, and finally fed into the transformer's sequential processing stream. Unlike ViT, which caters to two-dimensional inputs, our model processes these patches in the sequence order in which the text appears, emulating the linear progression of reading. This patch-based segmentation aligns with the sequential nature of language, enabling our model to predictively learn from the visual data.

Text Input We leverage the same tokenizer as Llama 2, segmenting input text into discrete tokens with a total vocabulary size of 32k. These tokens are then transformed into dense vector representations through an embedding lookup table.

3.3 Pre-training Objectives

As illustrated in Figure 2, our training architecture features separate heads following the terminal transformer layers for various inputs.

Next Patch Prediction Given a sequence of N visual patches $x_p = (x_p^1, x_p^2, \cdots, x_p^N)$ where each visual patch x_p^t is a flattened patch embedding. We decompose the image patch sequence into the production of N conditional probabilities:

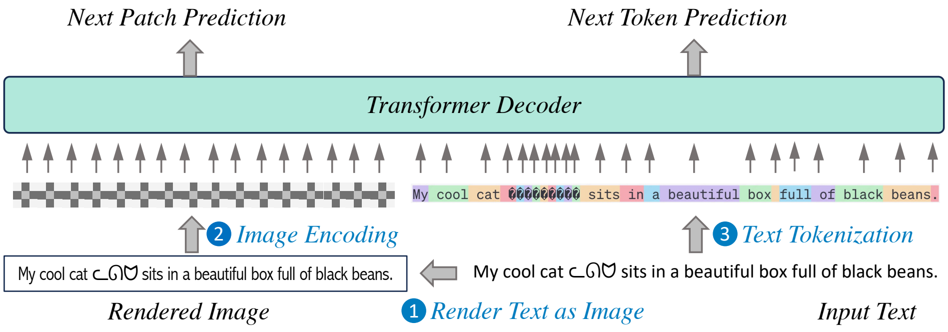


Figure 2: Illustration of *dual-modality* pre-training on paired text-image (`DualGPT`). Autoregressive pre-training on pure text and visual text images, apply next patch prediction and next token prediction, respectively.

$$p(x_p^1, x_p^2, \cdots, x_p^N) = \prod_{t=1}^{N} p(x_p^t | x_p^1, x_p^2, \cdots, x_p^{t-1}) \tag{1}$$

For visual inputs, we employ a *next patch prediction* strategy, where a normalized mean squared error (MSE) loss quantifies the pixel reconstruction accuracy by comparing the normalized target image patches with the reconstructed outputs, excluding the EOS patches.

Next Token Prediction For text inputs, we utilize a conventional *next token prediction* objective, optimizing a cross-entropy loss that evaluates the fidelity of predicted token sequences generated via *teacher-forcing* against the ground truth tokens.

3.4 Model Configuration

To explore previous research questions, our pre-training regimen explores various configurations for ablation analysis: **(1)** **`TextGPT`**: Pre-training solely on text data. **(2)** **`PixelGPT`**: This involves training solely on rendered image data, employing a mean squared error (MSE) loss, as visualized in Figure 1(a). **(3)** **`MonoGPT`**: Trained on separate streams of rendered image and text data without any intermodal pairing. **(4)** **`DualGPT`**: Trained on unpaired image and text input, and on paired image-text data (dual-modality). When handling paired data, we concatenate the image data sequence before the text sequence and feed them simultaneously to the model, as delineated in Figure 2. We refer to Appendix §D for details.

3.5 Pre-training Details

Model Architecture Our architecture, illustrated in Figure 1(b), is built upon a stack of $N = 24$ standard transformer decoder (Vaswani et al., 2017), following Llama 2 (Touvron et al., 2023b). We incorporate RMSNorm for pre-normalization (Zhang and Sennrich, 2019), SwiGLU activation functions (Shazeer, 2020; Chai et al., 2020), rotary position embeddings (Su et al., 2024), and grouped query attention (Ainslie et al., 2023). Comprehensive specifications and additional implementation details of our architecture are in Appendix §B.

Data For visual image data, we use rendered the corpus of peS2o, English Wikipedia and C4 datasets for pre-training; while for text data, we adopt peS2o, English Wikipedia, C4, Common Crawl, and The Stack v1. We refer the readers to Appendix §C for details.

4 Experiments

4.1 Experimental Setup

Fine-tuning Protocols Our evaluation entailed fine-tuning an autoregressive pixel-based pre-trained model for downstream tasks to thoroughly assess its performance. We adapted our pixel-based model to various downstream tasks by substituting the language modeling head with a linear MLP for downstream tasks. Specifically, `PixelGPT`, initially pre-trained on pixel data, undergoes fine-tuning on similarly rendered pixel data. Conversely, `MonoGPT` and `DualGPT`, which benefitted from a joint pre-training regime incorporating both text and pixel data, were fine-tuned across different input modalities: pixel, text, and a combination of both.

Evaluation Tasks Our assessment of the generative pixel pre-training models encompasses tasks in natural language understanding (NLU) and cross-lingual language understanding. For NLU, we utilize the GLUE benchmark, aligning the fine-tuning data rendering approach with the pre-training process outlined in Appendix A. Sentence pairs from GLUE's natural language inference tasks are indi-

Model	#Param	Input Modality		MNLI-m/mm	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	WNLI	Avg.
		Text	Pixel	Acc	F1	Acc	Acc	MCC	Spear.	F1	Acc	Acc	
BERT	110M	✓	✗	84.0/84.2	87.6	91.0	92.6	60.3	88.8	90.2	69.5	51.8	80.0
GPT-2	126M	✓	✗	81.0	89.4	87.7	92.5	77.0	74.9	71.5	52.0	54.9	75.6
DONUT	143M	✗	✓	64.0	77.8	69.7	82.1	13.9	14.4	81.7	54.9	57.7	57.2
CLIPPO	93M	✗	✓	77.7/77.2	85.3	83.1	**90.9**	28.2	**83.4**	84.5	59.2	-	-
PIXEL	86M	✗	✓	78.1/**78.9**	84.5	**87.8**	89.6	38.4	81.1	**88.2**	60.5	53.8	74.1
PixelGPT	317M	✗	✓	**79.0**/78.2	**86.0**	85.6	90.1	35.3	80.3	84.6	**63.9**	**59.2**	**74.2**

Table 1: Comparative evaluation on the GLUE benchmark. Performance metrics for each model across various GLUE tasks are presented, along with the aggregate average performance. `#Param` indicates the model scale. `PixelGPT` stands out as the leading model, surpassing other pixel-based counterparts in terms of overall performance.

vidually rendered and subsequently concatenated, with a black block serving as the end-of-sentence token. The cross-lingual understanding capability is evaluated on the XNLI dataset over fifteen different languages. Following Conneau et al. (2020), our evaluation is performed in two distinct scenarios: (1) *Translate-Train-All*, where the model is fine-tuned on a blend of original English and machine-translated data from other 14 languages, aiming to appraise the model's multilingual understanding; (2) *Cross-lingual Transfer* settings, wherein fine-tuning is conducted solely on English data, with multi-language test sets employed to evaluate the model's transferability across languages. Comprehensive experimental details are provided in the Appendix §E.

Baselines For a thorough evaluation, we benchmark against models specialized in textual and visual representations. In the textual category, BERT and GPT-2 (Radford et al., 2019) are chosen. For pixel-based models, we contrast our approach with DONUT (Kim et al., 2022), CLIPPO (Tschannen et al., 2023), and PIXEL (Rust et al., 2023), which are trained on pixel-based representation. Detailed discussions are provided in Appendix §F.

4.2 Results

RQ1: Autoregressive Pixel-based Pre-training Rivals PIXEL. Our empirical investigation, detailed in Table 1, scrutinizes the feasibility of pure pixel-based autoregressive pre-training on RGB images of visual texts. The proposed `PixelGPT` model, training solely on rich raw visual inputs (24-bit RGB images), demonstrates not merely a competitive edge but, in several tasks, surpasses the performance of models pre-trained on text alone. Specifically, `PixelGPT` exhibits remarkable superiority on GLUE benchmarks – evidenced by its marked performance increases on the STS-B (`+5.4`), MRPC (`+13.1`), RTE (`+11.9`), and WNLI (`+4.3`) assessments compared to GPT-2. This demonstrates the viability of pixel-based pre-training in capturing complex linguistic constructs.

When compared to PIXEL, which leverages a bidirectional encoder architecture, `PixelGPT` exhibits enhanced performance in QQP (`+1.5`), RTE (`+3.4`), and WNLI (`+5.4`). These results collectively affirm the hypothesis that autoregressive pre-training on raw visual images is feasible for language modeling. `PixelGPT` achieves the optimal performance among pixel-based approaches on GLUE, underscoring the transformative impact of integrating rich visual information into pre-training. Refer to §G.5 for detailed discussion.

As shown in Figures 3 and 4, `PixelGPT` demonstrates a scaling trend with increased training data compute, indicating a promising direction for data scaling. This suggests that with more extensive training, `PixelGPT` has the potential to outperform text-based models, such as GPT-2 and BERT. Due to computational constraints, we will explore this in future work.

RQ2: Impact of Autoregressive Pixel Pre-training on Multilingual Tasks. Traditional language models, exemplified by BERT, typically utilize a subword tokenization process such as Word-Piece (Devlin et al., 2019) or BPE (Sennrich et al., 2015) that decomposes sentences into a predefined set of text tokens. While effective within the scope of a single language or similar language families, this approach is constrained by a *vocabulary bottleneck* (Rust et al., 2023) in multilingual scenarios, limiting its efficacy. Pixel-based representations, however, transcend this limitation by representing text in a modality that inherently supports unified processing—the visual domain of images.

In our cross-lingual evaluation, conducted on the XNLI dataset in the *translate-train-all* configuration and detailed in Table 2, `PixelGPT` demonstrates a robust capability for multilingual comprehension. It not only matches the performance of BERT, but also consistently surpasses the PIXEL

Model	#lg	#Param	Input Modality		ENG	ARA	BUL	DEU	ELL	FRA	HIN	RUS	SPA	SWA	THA	TUR	URD	VIE	ZHO	Avg.
			Text	Pixel																
Fine-tune model on all training sets (Translate-train-all)																				
mBERT	104	179M	✓	✗	83.3	73.2	77.9	78.1	75.8	78.5	70.1	76.5	79.7	67.2	67.7	73.3	66.1	77.2	77.7	74.8
XLM-R base	100	270M	✓	✗	85.4	77.3	81.3	80.3	80.4	81.4	76.1	79.7	82.2	73.1	77.9	78.6	73.0	79.7	80.2	79.1
BERT	1	110M	✓	✗	83.7	64.8	69.1	70.4	67.7	72.4	59.2	66.4	72.4	62.2	35.7	66.3	54.5	67.6	46.2	63.9
PIXEL	1	86M	✗	✓	77.2	**58.9**	66.5	68.0	64.9	69.4	57.8	63.4	70.3	60.8	**50.2**	64.0	54.1	64.8	**52.0**	62.8
PixelGPT	1	317M	✗	✓	**77.7**	55.4	**66.7**	**69.0**	**67.4**	**71.2**	**59.1**	**65.6**	**71.4**	**61.7**	47.0	**65.2**	**54.4**	**66.1**	50.5	**63.2**

Table 2: Cross-lingual performance evaluation on the XNLI dataset in *translate-train-all* settings. We report the accuracy achieved by each model across the multiple languages featured in the XNLI dataset, along with their average accuracy scores. The number of languages (#lg) incorporated during pre-training and the model size (#Param) are provided for reference. PixelGPT demonstrates superior performance over PIXEL, showcasing the efficacy of exclusive pixel-based input modality in cross-lingual contexts.

Model	Input Modality		MNLI-m/mm	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	WNLI	Avg.
	Text	Pixel	Acc	F1	Acc	Acc	MCC	Spear.	F1	Acc	Acc	
TextGPT (text only)	✓	✗	79.9/80.0	86.1	86.1	91.5	47.3	**85.8**	86.3	63.5	56.3	76.3
MonoGPT (text+pixel)	✓	✗	80.0/**80.5**	85.9	**87.3**	90.1	40.2	83.8	87.0	62.8	56.3	75.4
	✗	✓	64.7/65.9	78.9	77.3	74.8	11.6	73.2	83.5	59.9	57.7	64.8
DualGPT (text+pixel+pair)	✓	✗	**80.1**/80.4	**86.5**	86.8	**91.6**	**49.0**	85.4	**87.6**	65.7	56.3	**76.9**
	✗	✓	71.5/71.7	82.8	81.6	83.4	17.2	80.2	84.1	**66.4**	**59.2**	69.4

Table 3: Ablation results of model performance on the GLUE benchmark.

model in average accuracy across evaluated languages. Remarkably, PixelGPT exhibits pronounced gains over BERT in languages that diverge significantly from English, such as Thai and Chinese, with improvements of +11.3 and +4.3, respectively. This enhanced performance may be attributed to two primary factors: the absence of PixelGPT's reliance on language-specific tokenization, enabling more effective learning from the visual forms of text, and the limitations of BERT's English-centric pre-training, which exhibits shortcomings when faced with linguistically distant families. Thus, PixelGPT's proficiency in leveraging the visual features of text contributes to its advanced multilingual understanding, signaling a significant stride in overcoming the challenges associated with the *vocabulary bottleneck*.

RQ3: Synergistic Effects of Multimodal Pre-training. In our investigation into the interplay between distinct pre-training data modalities, we contrasted the performances of MonoGPT and DualGPT—models that integrate different input modalities—with that of TextGPT under equivalent conditions of aligned text token pre-training. TextGPT and MonoGPT underwent pre-training on 40 billion text tokens, with MonoGPT additionally exposed to 40 billion image patches. DualGPT, on the other hand, was pre-trained on 38.4 billion text tokens complemented by 48 billion image patches and 9.6 billion tokens of image-text paired data.

This comparative analysis, spanning both GLUE and XNLI datasets (the latter within the *translate-train-all* settings), is shown in Tables 3 and 4. A pivotal finding is that the incorporation of dual-modality data during pre-training markedly enhances average performance across language understanding tasks: DualGPT (76.9) surpasses both TextGPT (76.3) and MonoGPT (75.4). This suggests that potential conflicts arising from unimodal training can be significantly alleviated through a multimodal pre-training approach. This inference is corroborated by XNLI outcomes, wherein the addition of pixel-text paired data improved the model's multilingual interpretative proficiency.

Further, with pixel modality input, DualGPT surpasses TextGPT across various downstream tasks. This result reinforces the proposition that pre-training modality conflicts can be effectively resolved via the integration of paired dual-modality data, fostering more robust multimodal learning.

4.3 Analysis

Scaling Training Tokens vs. GLUE Performance In Figure 3, we delineate the correlation between the scale of training data and the ensuing performance on the GLUE benchmark. Our analysis encompasses a spectrum of total training tokens/patches from 10 billion (B) to 240B, juxtaposing the trajectories of TextGPT, PixelGPT, MonoGPT, and DualGPT, with BERT and PIXEL serving as benchmarks. The MonoGPT and DualGPT models are evaluated under two different input modalities: text and pixel. From our findings, two primary insights emerge: **(1) Pixel-based autoregressive pretraining models exhibit an increased data demand**. With minimal training (e.g., at 10B),

Model	Input Modality		ENG	ARA	BUL	DEU	ELL	FRA	HIN	RUS	SPA	SWA	THA	TUR	URD	VIE	ZHO	Avg.
	Text	Pixel																
Fine-tune model on all training sets (Translate-train-all)																		
TextGPT (text only)	✓	✗	72.4	60.4	62.8	64.8	63.3	65.0	58.5	61.5	65.2	57.7	**59.9**	61.2	54.9	63.6	**63.1**	62.3
MonoGPT (text+pixel)	✓	✗	**72.9**	60.8	63.2	63.5	63.5	63.6	57.9	60.7	64.4	58.8	59.4	60.6	55.2	63.2	60.7	61.9
	✗	✓	66.8	47.1	61.2	61.8	63.4	64.5	56.7	59.2	64.9	56.8	48.7	61.8	52.1	61.0	50.7	58.4
DualGPT (text+pixel+pair)	✓	✗	72.7	**61.6**	63.8	64.7	63.9	65.1	58.8	61.6	65.4	59.0	59.8	62.2	**55.8**	63.4	62.1	**62.7**
	✗	✓	71.7	55.0	**67.6**	**66.5**	**66.8**	**68.4**	**59.0**	**64.4**	**68.9**	**61.3**	48.7	**64.3**	54.7	**65.8**	54.4	62.5

Table 4: Ablation results of model performance on XNLI under *Translate-Train-All* settings.

pixel-based models initiate at a lower performance threshold in pixel modality (all under 55%), compared to their text modality counterparts, which approximate a performance level of 70%. Nevertheless, with the increase of training data, a critical volume threshold catalyzes a substantial rise in performance for `PixelGPT`, `MonoGPT`, and `DualGPT` in pixel modality. This trajectory reveals a progressive convergence of `PixelGPT` towards the text-based baseline, culminating in its overtaking of PIXEL at around 200B tokens/patches and nearing `TextGPT` with a less than 5-point performance differential, while still on an upward trend. **(2) The integration of paired dual-modality data during pretraining appears to confer significant benefits on multimodal learning, particularly for pixel-based input**. When matched for training data volume, `DualGPT` consistently eclipses `MonoGPT` across comparable benchmarks, with the former maintaining a pronounced lead in pixel modality. This trend underscores the value of incorporating paired text-image data in pretraining to enhance the efficacy of multimodal learning.

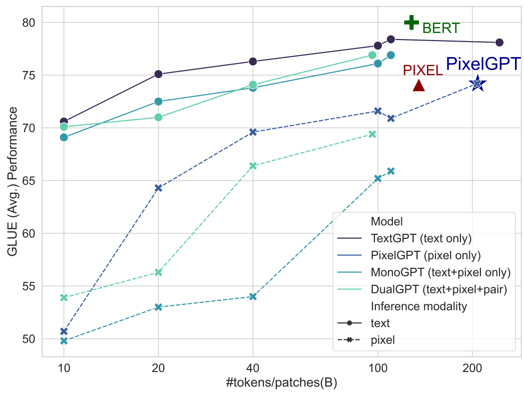


Figure 3: Training tokens/patches versus overall performance on GLUE benchmark.

Scaling Training Tokens vs. XNLI (*Translate-Train-All*) Performance We further explored the progression of model performance in multilingual capability across varying volumes of pre-trained tokens/patches. This comparison, delineated in Figure 4, focused on the *Translate-Train-All* setting of the XNLI benchmark. **(1) Pixel-based autoregressive models display a heightened requirement for training data in multilingual tasks**, corroborating the trend observed on the GLUE benchmark. Initially, there is a notable performance disparity between pixel and text modalities, with pixel-based models lagging behind when training on a lesser volume of tokens/patches. However, this gap diminishes substantially with the increase in training volume. Remarkably, upon reaching the 200B, `PixelGPT` not only surpasses PIXEL but also matches the performance of BERT, indicating a continued potential for further enhancement in its multilingual proficiency with additional training data. **(2) The injection of dual-modality data at the early stages of training appears to be particularly beneficial for models learning from pixel data**. When comparing `DualGPT` and `MonoGPT` under the pixel modality, `DualGPT` demonstrates a notable performance advantage at the outset of training (55% vs. 45.8% at the 10B token/patch mark). Although this edge tapers as the training volume expands, it suggests that early-stage multimodal alignment aids the pixel-based models in leveraging the textual data for enhanced multilingual understanding. **(3) Our text-based pretraining approach, `TextGPT`, demonstrates superior results over BERT**. This is evident when training reaches approximately 100B tokens, where `TextGPT` outperforms BERT. This improvement may be attributed, in part, to our *byte-level* BPE tokenization as utilized in Llama 2, which effec-

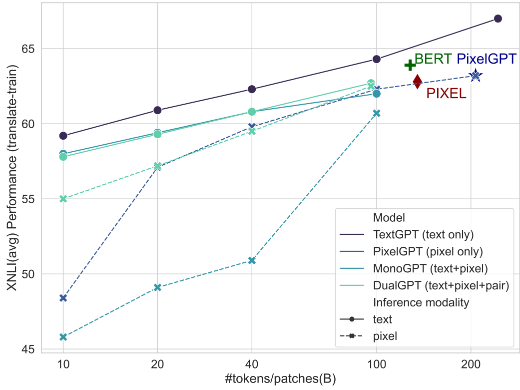


Figure 4: Training tokens/patches versus overall performance on XNLI benchmark.

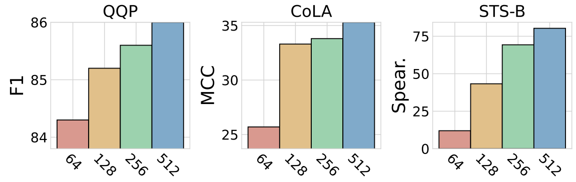


Figure 5: Analysis of escalating the global batch size.

tively deconstructs unseen languages into their constituent raw bytes—a capability not afforded by BERT. Additionally, the enrichment of our text pretraining corpus from diverse sources contributes to this. For a detailed breakdown of the text pretraining data, we refer readers to Appendix §C.2.

A Large Batch Size Improves Stable Training We observe a distinct preference for larger batch sizes when fine-tuning pixel-based modalities across certain datasets. As in Figure 5, we evaluate how different batch sizes—64, 128, 256, and 512—affect model performance on selected GLUE benchmark tasks, namely QQP, CoLA, and STS-B. A clear trend emerges from the data: increasing the batch size correlates with improved model performance. Our analysis suggests that pixel modality fine-tuning exhibits greater variance than text modality and benefits from the use of larger batch sizes. This appears to mitigate the variability inherent in different training batches, thus enhancing training stability. It prevents premature convergence to suboptimal local minima and fosters higher model accuracy.

Font Transfer Analysis We extend to examining the adaptability of `PixelGPT` to diverse font styles during fine-tuning. We employed three distinct fonts for rendering the data: `GoNotoCurrent`, which was utilized during pretraining; `NotoSerif-Regular`, a font stylistically akin to GoNotoCurrent; and `JournalDingbats1`, a font that renders text as distinct image-based symbols, markedly divergent from the others. The adaptability was tested across five datasets from the GLUE benchmark—CoLA, STS-B, MRPC, RTE, and WNLI. As depicted in Figure 6, the performance of `PixelGPT` remained stable across different fonts for all selected datasets barring CoLA. Notably, even when fine-tuned with data rendered in `JournalDingbats1`, which bears little resemblance to the pre-training font, the results demonstrated a commendable degree of resilience, indicating that the pixel pre-training is robust to generalize across significantly varied visual representations.

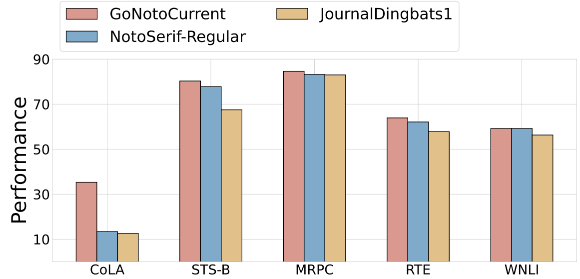


Figure 6: Analysis of fine-tuning on different fonts.

Render Mode	Font	Acc	Δ
Grayscale	Apple Emoji	58.7	-
RGB		61.4	+2.7

Table 5: Comparison performance on HatemojiBuild dataset with grayscale and RGB rendering.

Figure 7: Example cases of **HatemojiBuild** predictions. ✓ and ✗ indicate the correct and incorrect predictions.

Impact Analysis of Color Retention Unlike previous that renders text as grayscale or binary images, `PixelGPT` employs *RGB*-rendered data, retaining richer informational content. We evaluated the performance of these rendering approaches on HatemojiBuild dataset (Kirk et al., 2022), designed for detecting online hate speech conveyed through emojis. Table 5 presents our findings, where the RGB-rendered data fine-tuning significantly outperforms its grayscale counterpart. This performance enhancement can be attributed to the model's capacity to utilize color cues within emojis, which are critical for inferring the emotional context of sentences. For a more detailed illustration, Figure 7 provides specific examples where color retention has improved model interpretability.

5 Conclusion and Future Work

In this paper, we have investigated the potential of pixel-based autoregressive pre-training using visual text images. Our results demonstrate that incorporating visual orthographic features significantly enhances language understanding and multilingual capabilities. Additionally, our empirical findings suggest that using pixel-text paired data effectively reduces modality competition during training, thereby improving model performance. Looking forward, scaling this approach to larger model sizes holds considerable promise for advancing the field of multimodal language processing.

Limitations

Model Scale The current implementation of our model utilizes 24 layers of transformer decoders, which has been effective for the scope of our experimental framework. However, the exploration of scaling our model to much larger configurations, such as 7B, 13B, 70B, or over 100B parameters, remains untested. Expanding the language model's capacity could significantly improve its ability of scaling, potentially enhancing both performance and generalizability.

Training Compute Our training was restricted by computational resources, limiting us to pre-training on only 100 to 200 billion tokens or patches. This constraint curtails our capacity to exploit the full benefits of extensive data scale training. Future work can extend the pre-training to more than 1,000 billion tokens or patches could yield promising insights into the scalability.

Extended Evaluation on Text Generation One limitation of our approach is related to generation tasks. Since the model's input and output are image patches, directly obtaining text outputs requires an additional OCR postprocessing step. This introduces an additional layer of complexity and potential error. We plan to address this in future work, exploring more integrated solutions for text generation tasks.

Preliminary Nature of Study It is crucial to acknowledge that this research constitutes a preliminary foray into the realm of pixel-based autoregressive models for multilingual and multimodal language processing. As such, while the results are encouraging, they should be viewed as exploratory. We invite further research to build upon our initial findings, addressing these limitations and further testing the robustness and applicability of the model in a wider array of settings.

Ethical Considerations

This research into pixel-based autoregressive pre-training for visual text images raises several ethical considerations that warrant careful attention:

Data Privacy and Security The utilization of visual text images, especially from diverse sources such as multilingual datasets, necessitates stringent adherence to data privacy and security guidelines. It is vital to ensure that all data used for training and testing respects the privacy rights of individuals and complies with applicable legal frameworks.

Bias and Fairness Machine learning models, particularly those involved in language processing, are susceptible to biases that may be present in the training data. It is imperative to conduct thorough bias audits and fairness assessments to identify and mitigate any discriminatory patterns in model predictions, ensuring that the technology is equitable across different languages and cultural contexts.

Environmental Impact The training of large-scale models is resource-intensive and has a significant environmental footprint. We must consider sustainable practices in model training, including optimizing computational efficiency and exploring energy-efficient hardware to reduce the overall carbon emissions associated with our research.

Misuse Potential While our study focuses on the positive applications of enhancing multilingual capabilities and understanding, there is a potential for misuse in various contexts. We advocate for responsible use guidelines and transparency in model deployment to prevent malicious applications of the technology.

Continual Monitoring and Evaluation Post-deployment monitoring and ongoing evaluation of the model's performance and societal impact are crucial. This process helps ensure the model adapts to changes over time and continues to operate within the ethical boundaries set forth by evolving standards and expectations.

By addressing these ethical considerations, we aim to promote responsible research and application of advanced machine learning techniques in language processing, contributing positively to the field and society at large.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Yekun Chai, Shuo Jin, and Xinwen Hou. 2020. Highway transformer: Self-gating enhanced self-attentive networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6887–6900, Online. Association for Computational Linguistics.

Yekun Chai, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, and Hua Wu. 2023. ERNIE-code: Beyond English-centric cross-lingual pretraining for programming languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10628–10650, Toronto, Canada. Association for Computational Linguistics.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. 2024. Scalable pre-training of large autoregressive image models. *arXiv preprint arXiv:2401.08541*.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. 2022. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE.

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*, pages 498–517. Springer.

Hannah Kirk, Bertie Vidgen, Paul Rottger, Tristan Thrush, and Scott Hale. 2022. Hatemoji: A test suite and adversarially-generated dataset for benchmarking and detecting emoji-based hate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1352–1368, Seattle, United States. Association for Computational Linguistics.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, et al. 2022. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

Jonas F. Lotz, Elizabeth Salesky, Phillip Rust, and Desmond Elliott. 2023. Text rendering strategies for pixel language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10155–10172. Association for Computational Linguistics.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue

Zhuo, Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian J. McAuley, Han Hu, Torsten Scholak, Sébastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, and et al. 2024. Starcoder 2 and the stack v2: The next generation. *CoRR*, abs/2402.19173.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13845–13861. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

Luca Soldaini and Kyle Lo. 2023. peS2o (Pretraining Efficiently on S2ORC) Dataset. Technical report, Allen Institute for AI. ODC-By, `https://github.com/allenai/pes2o`.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Yintao Tai, Xiyang Liao, Alessandro Suglia, and Antonio Vergari. 2024. Pixar: Auto-regressive language modeling in pixel space. *arXiv preprint arXiv:2401.03321*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Michael Tschannen, Basil Mustafa, and Neil Houlsby. 2023. CLIPPO: image-and-language understanding from pixels only. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 11006–11017. IEEE.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 353–355. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

Contents

1 Introduction **1**

2 Related Work **2**

3 Pre-training on Pixels and Texts **2**
- 3.1 Rendering Text as Images 2
- 3.2 Input Representation 3
- 3.3 Pre-training Objectives 3
- 3.4 Model Configuration 4
- 3.5 Pre-training Details 4

4 Experiments **4**
- 4.1 Experimental Setup 4
- 4.2 Results 5
- 4.3 Analysis 6

5 Conclusion and Future Work **8**

A Text Renderer Details **14**

B Model Architecture **14**

C Pre-training Data **14**
- C.1 Pre-training Data for Visual Images 14
- C.2 Pre-training Data for Text 14

D Pre-training Details **16**

E Fine-tuning Details **16**
- E.1 Fine-tuning Dataset 16
- E.2 Fine-tuning Setting 17
- E.3 Implementation for Different Render Modes 17

F Baselines **17**
- F.1 Text-based Baselines 17
- F.2 Image-based Baselines 18
- F.3 Comparison with Previous Work 19

G Detailed Results & Analysis **19**
- G.1 Performance on Cross-lingual Transfer 19
- G.2 Probing Dual-Modality Fine-Tuning 19
- G.3 RGB vs. Grayscale vs. Binary Rendering 19
- G.4 Comparison on XNLI under *Translate-Train-All* Settings 21
- G.5 Benefits of Pixel-based Models 21

A Text Renderer Details

The renderer transposes one or more segments of text onto a virgin RGB canvas structured into 1024 distinct patches, each delineated into a 16x16 pixel matrix. This configuration is shown in Table 6.

A visual syntax is adopted to distinguish text boundaries: a solitary black patch of 16x16 pixels operates as both a delimiter and an indicator of the sequence's conclusion (End of Sequence, EOS). Subsequent white patches post-EOS are deemed padding—they remain inert in the attention mechanism, thus excluding them from the computation of attention scores.

For the rendition of text documents, the renderer tackles content on a line-by-line basis. It incorporates a binary search algorithm to intelligently gauge the maximum quota of words renderable in a single pass, ensuring the text's width remains within the permissible pixel threshold. This dynamic segmentation capability circumvents potential truncation issues inherent in rendering extensive lines of text, allowing for a seamless integration of longer passages without compromise to visual fidelity or contextual integrity.

Parameter	Value
Background Color	White
DPI	120
Font Color	black
Font type	GoNotoCurrent
Font size	8
Max sequence length	1024
Padding size	3
Pixels per patch	16x16

Table 6: Configuration of text rendering.

B Model Architecture

Table 8 specifies the comprehensive configuration of our model's architecture, based on similar transformer decoder architecture to Llama 2 (Touvron et al., 2023b) with specific adaptations. We employ SwiGLU as the hidden activation function (Shazeer, 2020; Chai et al., 2020), noted for its effective nonlinear processing capabilities. The initializer range is set to 0.02 to promote optimal weight initialization. An intermediate size of 2816 is specified, offering a balance between the model's representational capacity and computational demands. The hidden size and the maximum number of position embeddings are both set at 1024, facilitating detailed representation of inputs and accommodating sequences up to 1024 tokens.

The model's attention architecture utilizes grouped query attention (Ainslie et al., 2023) with 16 attention heads and 8 key-value heads. We use a stack of 24 transformer layers, endowing the model with substantial depth for complex pattern recognition. Also, we use RMSNorm (Zhang and Sennrich, 2019) with epsilon of 1e-05 and rotary embeddings (Su et al., 2024).

C Pre-training Data

For the text-based pre-training, we utilized the expansive Dolma dataset (Soldaini et al., 2024), which comprises an extensive collection of 3 trillion tokens. This dataset is sourced from a heterogenous compilation of materials, including an array of web-based content, scholarly articles, programming code, literary works, and comprehensive encyclopedic entries. For the image-based pre-training, we transformed the textual content from the peS2o corpus, English Wikipedia, and the C4 dataset into visual representations, amounting to a total of over 400 million document images.

C.1 Pre-training Data for Visual Images

We pretrained on a rendered version of the peS2o, English Wikipedia and C4.The peS2o dataset, a curated collection of approximately 40 million creative open-access academic papers, has been meticulously cleaned, filtered, and formatted to facilitate the pretraining of language models. Meanwhile, The C4 dataset represents a substantial refinement of the Common Crawl corpus. This dataset, derived from the extensive Common Crawl web scrape, undergoes rigorous cleaning and preprocessing to ensure the quality and relevance of the text data. The C4 dataset is exclusively composed of English language texts, with a stringent criterion that each page must have at least a 99% probability of being in English, as determined by the langdetect tool, to be included. This selection process ensures that the dataset primarily contains natural language text, free from boilerplate or nonsensical content, and is extensively deduplicated to avoid redundancy.

C.2 Pre-training Data for Text

Common Crawl Common Crawl is a comprehensive web corpus that collects data from a variety of web pages. This dataset uses the URL

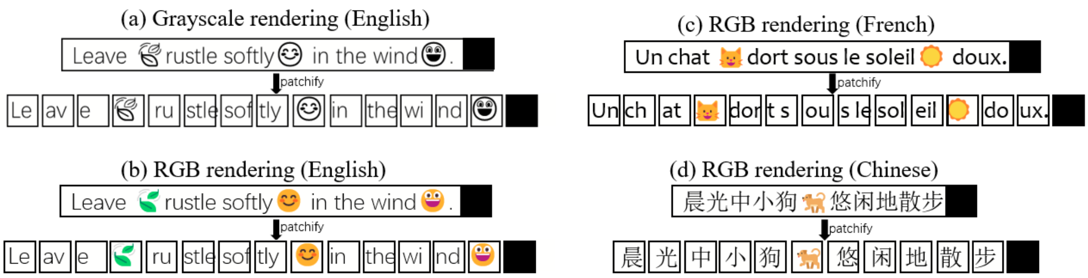


Figure 8: Illustration of patchifying rendered visual images into a sequence of patches, with a black patch as end-of-sequence marker.

Source	Type	Gzip files (GB)	Documents (M)	Tokens (B)
CommonCrawl	web	4,197	4,600	2,415
C4	web	302	364	175
peS2o	academic	150	38.8	57
The Stack	code	319	236	430
Project Gutenberg	books	6.6	0.052	4.8
Wikipedia	encyclopedic	5.8	6.1	3.6
Total		4980.4	5,245	3,084

Table 7: Statistics of pre-training corpus.

Parameter	Value
hidden activation	SwiGLU
initializer_range	0.02
intermediate_size	2816
hidden_size	1024
max_position_embeddings	1024
num_attention_heads	16
num_hidden_layers	24
num_key_value_heads	8
rms_norm_eps	1e-05
rope_scaling	null
rope_theta	10000
tie_word_embeddings	false
vocab_size	32,000

Table 8: Model configuration parameters.

of each web page as its identifier, facilitating the exploration of relationships between different documents. Covering data from May 2020 to June 2023 across 24 shards, Common Crawl includes about 4,600 million documents and 2,415 billion tokens. It is hosted on Amazon S3 as part of the Amazon Web Services' Open Data Sponsorship program and can be accessed freely, adhering to the Common Crawl terms of use.

C4 (Raffel et al., 2020) The C4 dataset is a cleaned and annotated subset of Common Crawl, specifically extracted from a shard dated April 2019. It includes URLs as metadata, which can be used to restore the original HTML files and understand document linkages. The dataset contains 364 million documents, totaling 175 billion tokens, and is available on the HuggingFace Hub under the ODC-By 1.0 license, allowing for broad academic and research usage.

peS2o (Soldaini and Lo, 2023) Derived from the Semantic Scholar Open Research Corpus (S2ORC), peS2o uses the Semantic Scholar Corpus ID to link documents to their corresponding manuscripts, enabling the recovery of original PDFs through associated metadata. The dataset encompasses 38.8 million documents and 57 billion tokens, and is accessible through the Semantic Scholar Public API under the ODC-By 1.0 license.

The Stack (Kocetkov et al., 2022) This dataset comprises a variety of computer code sourced from different GitHub repositories, with metadata that includes filenames and repository names to facilitate the retrieval of original content. The Stack contains 236 million documents and 430 billion tokens and is hosted on the HuggingFace Hub. It features code released under various permissive licenses, supporting diverse software development and research projects.

Project Gutenberg Project Gutenberg offers a collection of public domain books in the U.S., with each document beginning with the book's title to ease identification. This dataset provides access to about 52,000 documents and 4.8 billion tokens, and is freely available at gutenberg.org without any copyright restrictions, making it a valuable resource for literary and historical research.

Wikipedia and Wikibooks These datasets consist of encyclopedic content from Wikipedia and educational materials from Wikibooks, featuring metadata that includes URLs from which content is extracted. This allows users to reconstruct the structure and connections between documents. Together, they contain 6.1 million documents and 3.6 billion tokens. The data is freely available via Wikimedia data dumps and is released under the CC BY-SA 4.0 license, promoting widespread educational and informational use.

D Pre-training Details

We list the pre-training hyperparameters in Table 9. Pre-training was executed across a suite of 32 NVIDIA A100 GPUs. For `TextGPT` and `PixelGPT`, we adopted a global batch size of 4 million tokens or patches, respectively. In the case of `MonoGPT`, the global batch size was set at 8 million, maintaining an equal distribution between text and image data. For `DualGPT`, the global batch size was increased to 10 million, with a ratio of text/image/pair data with 4:4:2.

Hyper-parameter	Value
patch size P	16
maximum learning rate	5e-4
max seq length	1024
learning rate scheduler	linear
warmup steps	200
mixed precision	bfloat16
optimizer	AdamW
(β_1, β_2)	(0.9, 0.999)

Table 9: Hyperparameters of pre-training settings.

For clarification, we summarize the training tasks in Table 10 for various training configurations. `TextGPT` was trained exclusively on text data. In contrast, `PixelGPT` was pre-trained solely with image data. `MonoGPT` represents a hybrid approach, utilizing both text and image data independently but not in paired form. `DualGPT` stands as the most integrative model, incorporating text data, image data, and their conjunction in image-text pairs, underscoring the comprehensive nature of its pre-training regimen.

	Text data	Image data	Image-text pair
TextGPT	✓	✗	✗
PixelGPT	✗	✓	✗
MonoGPT	✓	✓	✗
DualGPT	✓	✓	✓

Table 10: Breakdowns of pre-training tasks for various model configurations.

E Fine-tuning Details

In this section, we present the details of the fine-tuning experiments, including (1) the dataset for the experiments, (2) the fine-tuning setting of the different pre-trained models (including `PixelGPT`, `MonoGPT`, `DualGPT` and `TextGPT`), and (3) how the different rendering modes were implemented.

E.1 Fine-tuning Dataset

The main experiments of our fine-tuning phase were conducted on GLUE and XNLI to evaluate the model's language and multilingual understanding ability, respectively. HatemojiBuild was used to analyze the effect of color retention. The details of the dataset are described below:

GLUE (Wang et al., 2018) A benchmark of nine sentence- or sentence-pair language understanding tasks, including MNLI(392k), QQP(363k), QNLI(108k), SST-2(67k), CoLA(8.5k), STS-B(5.7k), MRPC(3.5k), RTE(2.5k), WNLI(635), built on established existing datasets and selected to cover a set of three tasks. In this paper, for MNLI, QNLI, SST-2, RTE, and WNLI tasks, we report the Accuracy (Acc); for QQP and MRPC, we report the F1 score; for CoLA, we report the Matthews correlation coefficient (MCC); for STS-B we report Spearman correlation (Spear.). The MNLI dataset has matched development/test sets with the same sources as those in the training set, and unmatched sets that do not closely resemble any of the sets we saw during training are denoted as MNLI-m/mm. We conduct experiments on both settings. In addition, some previous works ignored WNLI because of its different training and validation/testing set distribution. We still performed on it and found that Pixel pre-training leads to a boost at WNLI.

XNLI (Conneau et al., 2018) The Cross-lingual Natural Language Inference (XNLI) corpus is an extension of the Multi-Genre NLI (MultiNLI) (Williams et al., 2018) corpus, designed for cross-lingual natural language inference, containing data in 15 languages. The dataset was created by manually translating the validation and test sets of MultiNLI into each of these 15 languages. For all languages, the English training set was machine-translated. The task is to predict textual entailment, a classification task determining whether sentence A implies, contradicts, or is neutral to sentence B, given two sentences.

HatemojiBuild (Kirk et al., 2022) HatemojiBuild is a benchmark for online hate detection involving emojis. The dataset includes 5,912 challenging examples of adversarial perturbations generated through a human-and-model-in-the-loop approach on Dynabench. This allows us to predict hateful emotions expressed with emojis.

E.2 Fine-tuning Setting

We fine-tune `PixelGPT`, `MonoGPT`, `DualGPT` and `TextGPT` on downstream tasks. we use NVIDIA Tesla V100 GPUs to fine-tune `TextGPT` and the NVIDIA A100 GPUs to fine-tune pixel-based pre-training models. The same rendering settings as in pre-training are used to render pixel data for fine-tuning `PixelGPT`, `MonoGPT`, and `DualGPT`, unless specified. We use the last patch to predict the label when fine-tuning the generative pixel-based pre-training models. In our analysis experiments, `MonoGPT` and `DualGPT` are also fine-tuned on dual-modality data obtained by concatenating rendered images with the original text. Specifically, we right-fill the image with white padding blocks for alignment. To avoid the impact of padding patches between the image and the text, we then set the attention mask to mask the padding blocks during fine-tuning.

We searched fine-tuning hyperparameters for each dataset in GLUE and two XNLI settings for `PixelGPT`, `MonoGPT`, `DualGPT` and `TextGPT`, respectively. Table 11 shows the searched hyperparameters and values. We present the best searched results for GLUE in Table 12 and Table 13 and for translate-train-all and cross-lingual transfer settings on XNLI in Table 14. During the hyperparameter searching, we found that using a larger batch size to fine-tune the generative pixel-based pre-training model improves training stability and achieves better results on some datasets. For a detailed analysis, see § 4.3.

Fine-Tuning Hyperparameters	Value
Optimizer	AdamW
Adam's betas	(0.9, 0.999)
Adam's epsilon	1e-8
Weight decay	0
Learning rate	{1e-5, 3e-5, 5e-5, 1e-4}
Learning rate schedule	{Cosine Annealing, Linear Decay}
Warmup steps	{10, 100}
Batch size	{32, 64, 128, 256, 512}
Max sequence length	{256, 768}
Training steps	{250, 500, 2000, 8000, 15000, 30000}
Dropout Probability	{0.1, 0}
Early Stopping	True
Seed	42

Table 11: Fine-tuning hyperparameters for grid search.

E.3 Implementation for Different Render Modes

We use RGB render mode for fine-tuning data rendering by default, as described in Appendix A. To obtain and adapt to grayscale and binary rendered data, we modify (1) the data preprocessing process and (2) the model's linear projection in the patch embedding layer. Specifically, we first render the data uniformly using RGB mode and get three-channel RGB images. After that, in the preprocessing stage, to get the grayscale version of the rendered image, we converted the RGB image to grayscale (with pixel values ranging from 0 to 255) using the convert function of the Image class in the PIL library and setting the function parameter model to 'L' to get the rendered binary image, we set the pixel threshold (set to 128 in our experiments) based on the converted grayscale image and set the pixels below the threshold in the grayscale image to 0 and the pixels above the threshold to 255. This way, we transformed the three-channel RGB-rendered image into a single-channel grayscale and binary image. Next, since the patch embeeding layer of the pre-trained model takes the three-channel image as input by default, we need to modify the linear projection layer in it to adapt to the single-channel image. Therefore, we average the linear layer weights by channel and use them as initial weights before fine-tuning so that the model supports the processing of single-channel images.

F Baselines

F.1 Text-based Baselines

GPT-2 GPT-2 (Radford et al., 2019) is an extension of the original GPT model, substantially

Hyperparameters	MNLI-m/mm	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	WNLI
Max Sequence Length					768				
Batch Size	64	64	64	64	32	64	32	64	32
Learning Rate	3e-5	3e-5	5e-5	3e-5	1e-5	5e-5	5e-5	1e-5	3e-5
Learning Rate Schedule					Linear Decay				
Warmup steps	100	100	100	100	10	10	10	10	10
Dropout Probability					0.0				

Table 12: Settings for fine-tuning `TextGPT` on GLUE.

Hyperparameters	MNLI-m/mm	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	WNLI
Max Sequence Length					768				
Batch Size	64	512	64	64	512	512	32	32	32
Learning Rate	5e-5	1e-4	5e-5	5e-5	5e-6	3e-5	5e-5	3e-5	3e-5
Learning Rate Schedule	Linear Decay	Cosine Annealing	Linear Decay	Cosine Annealing	Cosine Annealing	Cosine Annealing	Linear Decay	Linear Decay	Linear Decay
Warmup steps	100	100	100	100	10	10	10	10	10
Dropout Probability	0.0	0.1	0.0	0.1	0.1	0.1	0.0	0.0	0.0
Max Training Steps	15000	1500	8000	8000	2000	2000	2000	2000	250

Table 13: Settings for fine-tuning `PixelGPT` on the GLUE benchmark.

Hyperpameters	TextGPT	PixelGPT	MonoGPT(pixel)	MonoGPT(text)	MonoGPT(pair)	DualGPT(pixel)	DualGPT(text)	DualGPT(pair)
				Fine-tune model on all training sets (Translate-Train-All)				
Max Sequence Length	768	256	256	256	256	256	256	256
Batch Size	64	512	512	64	256	512	64	512
Learning Rate	5e-5	1e-4	1e-4	5e-5	5e-5	1e-4	5e-5	5e-5
Max Training Steps	15000	30000	30000	15000	30000	30000	15000	30000
Learning Rate Schedule					Linear Decay			
Warmup steps					100			
Dropout Probability					0			
				Fine-tune model on English training set (Cross-lingual Transfer)				
Max Sequence Length	768	256	256	768	256	256	768	256
Batch Size	64	256	256	64	256	512	64	512
Learning Rate	5e-5	1e-4	5e-5	5e-5	5e-5	1e-4	5e-5	3e-5
Max Training Steps	15000	15000	30000	15000	30000	15000	15000	30000
Learning Rate Schedule					Linear Decay			
Warmup steps					100			
Dropout Probability					0			

Table 14: Fine-tuning settings for XNLI. We report the best hyperparameters for all models on *Translate-Train-All* and *Cross-lingual Transfer*, respectively.

increases the parameter count to 1.5 billion, which enhances its ability to generate more coherent and contextually relevant text across a wide array of domains without task-specific training. With a transformer-based architecture, GPT-2 operates on unsupervised learning, using only a large corpus of text data scraped from the internet (WebText) to learn various language patterns and tasks. This model exemplifies a significant shift towards more robust and generalized language models, thereby supporting the development of AI systems capable of understanding and generating human-like text with minimal task-specific data.

BERT BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking model in natural language processing introduced by Devlin et al. (2019) at Google AI Language. It utilizes the bidirectional Transformer, an attention mechanism that learns contextual relations between words in a text. Unlike previous models that only consider text in a single direction (left-to-right or right-to-left), BERT processes words simultaneously in both directions. This bi-directionality allows the model to capture a richer understanding of context. Pre-trained on a large corpus of unlabeled text, BERT is fine-tuned with additional output layers to perform a wide array of language processing tasks.

F.2 Image-based Baselines

DONUT This OCR-free visual document understanding model (Kim et al., 2022) is fundamentally designed to interpret and extract structured information directly from document images, bypassing traditional optical character recognition (OCR) techniques. DONUT leverages a transformer architecture to encode document images into embeddings and decode these embeddings into structured outputs like JSON formats without preliminary text detection and recognition stages. Pre-trained using a combination of real and synthetically generated document images, DONUT achieves impressive benchmarks on several visual document understanding tasks, outperforming state-of-the-art OCR-dependent models in terms of both accuracy and processing speed. A synthetic data generator further enhances The model's pre-training, enabling it to readily adapt to different languages and document formats, thereby extending its applicability to global and diverse application scenarios.

CLIPPO CLIPPO (Tschannen et al., 2023) integrates a single vision transformer that processes all input types—images and text—equally, using the same model parameters. By adopting a contrastive learning framework, this unified model learns to align the representations of text and images into a cohesive latent space. This approach simplifies the architecture by removing the necessity for separate text and image towers and enhances efficiency by halving the parameter count compared to dual-tower systems. The key innovation of CLIPPO lies in its ability to perform complex multimodal tasks, including zero-shot classification and natural language understanding, with competitive performance while relying solely on pixel data.

PIXEL The PIXEL (Rust et al., 2023) (Pixel-based Encoder of Language) model reimagines language modeling by rendering text as images, effectively bypassing the vocabulary bottleneck of language models. This pre-trained model converts text into fixed-sized image patches, which are then processed by a Vision Transformer (ViT) encoder. Unlike conventional models that predict a distribution over a vocabulary of tokens, PIXEL focuses on reconstructing the pixels of masked image patches. This approach allows PIXEL to support many languages and scripts, leveraging orthographic similarities. The model performs better in handling scripts not present in its training data and is robust against orthographic attacks and linguistic code-switching.

F.3 Comparison with Previous Work

We summarize the comparison of our `PixelGPT` with pixel-based baselines, including PIXEL, PIXAR (Tai et al., 2024), in Table 15. *Please note that our work is different from PIXAR, which uses different training strategies and data rendering approaches from PIXEL and ours.* Instead, our model can be seen as an autoregressive GPT version of the PIXEL models.

Models	PIXEL	PIXAR	`PixelGPT` (Ours)
Image format	Grayscale (0-1)	Binary (0/1)	RGB (0-255)
Modeling	Bidirectional	Autoregressive	Autoregressive
Training Objective	Regression	**Classification**	**Regression**
Modeling Space	Continuous	**Discrete**	**Continuous**
Loss function	Mean Squared Error	**Binary Cross Entropy**	**Mean Squared Error**

Table 15: Detailed comparison with pixel-based baselines.

G Detailed Results & Analysis

G.1 Performance on Cross-lingual Transfer

In this section, We analyze the cross-lingual transfer ability of pixel-based autoregressive models on XNLI under the *Cross-lingual Transfer* setting. As shown in Table 16, we compared three different models: `PixelGPT`, `MonoGPT`, and `DualGPT`. Our findings indicate that incorporating additional text modality data in the pre-training phase enhances the cross-lingual transfer capabilities of these models. Nevertheless, a notable performance disparity remains when benchmarked against the multilingual prowess of the XLM-R base, a model pre-trained extensively across 100 languages.

G.2 Probing Dual-Modality Fine-Tuning

We delved into the synergistic potential between text and pixel modalities during the fine-tuning phase. A comparative experimental design was implemented to fine-tune pixel pre-trained models in two distinct manners: (1) exclusively on text data, and (2) on an amalgamation of rendered image data and original text. We assessed the performance impact of these fine-tuning approaches with `MonoGPT` and `DualGPT` models on XNLI. As delineated in Table 17, the models fine-tuned with dual-modality data consistently outperformed those fine-tuned on text data alone, with clear gains in multilingual understanding tasks. This evidence suggests that the inherent strengths of pixel-based representations in capturing multilingual nuances are amplified when combined with textual information during fine-tuning.

G.3 RGB vs. Grayscale vs. Binary Rendering

Rendering modes offer trade-offs between the richness of information and processing efficiency, with RGB providing a three-channel image dense with information, whereas grayscale and binary modes are optimized for speed. To assess the impact of these rendering choices, we explored the robustness of our model, pre-trained using RGB visual text, across different rendering modes within the down-

Model	#lg	#Param	Input Modality		ENG	ARA	BUL	DEU	ELL	FRA	HIN	RUS	SPA	SWA	THA	TUR	URD	VIE	ZHO	Avg.
			Text	Pixel																
Fine-tune model on English training set (*Cross-lingual Transfer*)																				
XLM-R base	100	270M	✓	✗	85.8	73.8	79.6	78.7	77.5	79.7	72.4	78.1	80.7	66.5	74.6	74.2	68.3	76.2	76.7	76.2
PixelGPT (pixel only)	1		✗	✓	**75.1**	35.1	36.9	37.3	37.0	42.2	35.6	34.9	43.1	37.4	35.9	38.1	33.8	38.4	35.5	39.8
MonoGPT (text+pixel)	1	317M	✗	✓	67.1	34.6	**40.6**	**41.7**	**44.2**	**47.5**	**36.4**	**40.8**	**51.4**	**41.7**	37.0	**41.1**	34.4	38.8	34.1	**42.1**
DualGPT (text+pixel+pair)	1		✗	✓	71.0	**36.9**	40.3	39.7	39.6	47.2	36.3	38.9	48.2	38.7	**38.0**	40.1	**37.0**	**41.3**	**36.8**	42.0

Table 16: Comparison of pixel-based pre-training models on XNLI dataset in *Cross-lingual Transfer* setting.

Model	**Input Modality**		ENG	ARA	BUL	DEU	ELL	FRA	HIN	RUS	SPA	SWA	THA	TUR	URD	VIE	ZHO	**Avg.**
	Text	Pixel																
Fine-tune model on all training sets (Translate-train-all)																		
MonoGPT (text+pixel)	✓	✗	74.0	60.9	62.7	63.4	63.4	64.2	58.2	59.9	64.3	58.6	59.3	61.0	55.0	63.6	61.3	62.0
	✓	✓	75.4	61.9	65.0	65.2	66.8	66.7	59.3	63.3	67.7	**61.1**	59.9	63.6	54.9	66.2	62.9	64.0
DualGPT (text+pixel+pair)	✓	✗	72.7	61.6	63.8	64.7	63.9	65.1	58.8	61.6	65.4	59.0	59.8	62.2	55.8	63.4	62.1	62.7
	✓	✓	**75.8**	**64.4**	**66.5**	**66.3**	**67.7**	**68.0**	**61.4**	**65.1**	**69.0**	**61.1**	**60.4**	**64.4**	**57.5**	**67.7**	**64.0**	**65.3**
Fine-tune model on English training set (Cross-lingual Transfer)																		
MonoGPT (text+pixel)	✓	✗	**79.9**	34.4	35.3	37.6	34.3	38.9	34.4	35.4	44.4	39.3	34.2	39.2	33.3	35.0	**37.4**	39.5
	✓	✓	77.5	35.6	**37.7**	40.4	**37.0**	**43.7**	34.9	**38.1**	**46.6**	**41.0**	35.0	41.0	33.8	37.1	**37.4**	**41.1**
DualGPT (text+pixel+pair)	✓	✗	79.1	35.5	36.0	40.8	35.1	41.3	**35.4**	36.6	44.6	38.2	**35.2**	38.2	34.6	36.4	**37.4**	40.3
	✓	✓	75.2	**38.5**	36.0	**42.3**	36.9	40.3	34.9	36.9	45.4	39.2	34.8	**42.8**	**36.3**	**37.8**	35.8	40.9

Table 17: Comparison of using dual-modalitiy and text-only modality for fine-tuning on XNLI. Adding pixel data for fine-tuning boosts the model's multilingual ability in the settings of *Translate-Train-All* and *Cross-lingual Transfer*.

Render Mode	**ENG**	**ARA**	**BUL**	**DEU**	**ELL**	**FRA**	**HIN**	**RUS**	**SPA**	**SWA**	**THA**	**TUR**	**URD**	**VIE**	**ZHO**	**Avg.**
Fine-tune model on all training sets (Translate-train-all)																
RGB	77.7	55.4	66.7	**69.0**	**67.4**	**71.2**	**59.1**	**65.6**	**71.4**	**61.7**	47.0	**65.2**	**54.4**	**66.1**	50.5	**63.2**
Binary	**78.2**	**55.8**	**67.0**	68.4	66.8	70.6	58.1	63.9	70.7	**61.7**	**47.5**	64.1	53.3	65.9	**52.9**	63.0
Grayscale	77.0	55.0	65.2	67.6	66.3	69.8	57.1	62.4	70.8	61.2	46.3	63.9	52.1	63.7	51.9	62.0
Fine-tune model on English training set (Cross-lingual Transfer)																
RGB	**77.3**	35.9	**38.0**	39.7	38.0	44.7	36.3	37.5	**46.4**	**39.6**	35.8	40.9	35.3	**41.8**	35.0	**41.5**
Binary	76.3	**37.8**	37.9	37.2	**38.9**	42.1	**37.8**	**39.0**	43.2	37.8	**37.9**	38.8	**36.9**	40.7	**36.7**	41.3
Grayscale	**77.3**	34.2	37.3	40.7	36.6	**46.0**	35.6	38.4	**46.4**	**39.6**	36.3	**41.4**	33.7	40.6	34.3	41.2

Table 18: Comparison of using three different render modes to fine-tune PixelGPT on XNLI. *RGB* rendering yields the best results.

stream context of the XNLI task. As shown in Figure 9, our experiments reveal that the performance when fine-tuning in grayscale and binary modes closely parallels that of RGB. This equivalence underscores the robustness of the pixel-based pre-training, indicating that its cross-linguistic transfer capability transcends the specific rendering mode employed in downstream tasks. Detailed experimental results are in the Table 18.

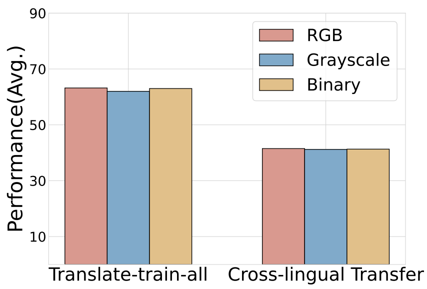


Figure 9: Performance of using three render modes to fine-tune PixelGPT on XNLI. PixelGPT shows strong robustness to fine-tuning render mode

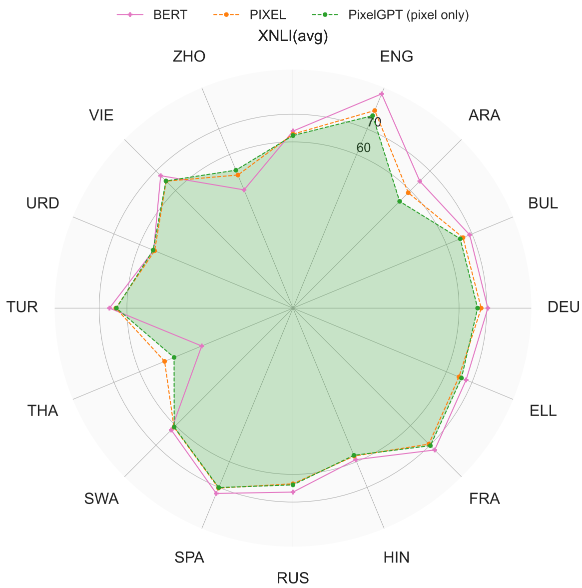


Figure 10: Comparison of our PixelGPT to PIXEL and BERT baselines in the *translate-train-all* settings.

G.4 Comparison on XNLI under *Translate-Train-All* Settings

We evaluate the efficacy of `PixelGPT` against the PIXEL and BERT baselines across fifteen diverse languages within the XNLI dataset's *Translate-Train-All* configuration. The comparative performance, visualized in Figure 10, demonstrates that `PixelGPT` outstrips PIXEL in twelve of the fifteen assessed languages. Notably, `PixelGPT` achieves performance parity with BERT in all but English and Arabic. Particularly, `PixelGPT` registers marked improvements over BERT in Thai and Chinese languages. These results suggest that the tokenizer-independent, pixel-based autoregressive design of `PixelGPT` offers a potent solution to the *vocabulary bottleneck* issue commonly encountered in language models, thus enhancing its applicability to multilingual tasks.

G.5 Benefits of Pixel-based Models

Our pixel-based method offers significant advantages:

1. **Tokenization-Free**: Eliminates the need for tokenization, thereby removing the vocabulary bottleneck problem, which is critical for handling diverse linguistic constructs and scaling effectively to multilingual contexts.

2. **Rich Visual Representation**: Leverages the rich information content of real-valued RGB images, capturing nuances that text-based tokenization may miss.

3. **Modality Interplay**: Demonstrates the potential for effective integration of visual and textual data, enhancing the overall model performance in language understanding tasks.

While all language models with pixel-based modalities currently match or slightly underperform compared to text modality models, the potential for scaling and the removal of tokenization challenges present a compelling case for further development and research in this area.