Enhancing Vision-Language Pre-training with Rich Supervisions

Yuan Gao[1*†] Kunyu Shi[2*] Pengkai Zhu[2] Edouard Belval[2] Oren Nuriel[2]
Srikar Appalaraju[2] Shabnam Ghadar[2] Zhuowen Tu[2] Vijay Mahadevan[2] Stefano Soatto[2]
[1]Stanford University [2]AWS AI Labs

y1gao@stanford.edu
{kunyus, zhpengka, belvae, onuriel, srikara, shabnam, ztu, vmahad, soattos}@amazon.com

Abstract

We propose Strongly Supervised pre-training with ScreenShots (S4) - a novel pre-training paradigm for Vision-Language Models using data from large-scale web screenshot rendering. Using web screenshots unlocks a treasure trove of visual and textual cues that are not present in using image-text pairs. In S4, we leverage the inherent tree-structured hierarchy of HTML elements and the spatial localization to carefully design 10 pre-training tasks with large scale annotated data. These tasks resemble downstream tasks across different domains and the annotations are cheap to obtain. We demonstrate that, compared to current screenshot pre-training objectives, our innovative pre-training method significantly enhances performance of image-to-text model in nine varied and popular downstream tasks - up to 76.1% improvements on Table Detection, and at least 1% on Widget Captioning.

1. Introduction

In recent years, there has been significant progress in Language Models (LMs) [7, 11, 47, 52] and Vision Language Models (VLMs) [1, 2, 6, 8, 10, 15–18, 21, 23, 25–30, 39–41, 44, 45, 49, 50, 56, 57, 60–63, 65–75, 77–80] exhibiting strong zero-shot generalization and adaptability to a wide range of tasks. Though they may differ in architecture, data and task formulation, such foundational models predominantly rely on large-scale pre-training on giant corpora of web scraped data which serves as the source of generalization capability - C4 [51], The Pile [20], Laion 5B [54].

The pre-training of LMs and VLMs were mostly studied separately. For LMs, the inputs and outputs reside within a homogeneous space, and pre-training tasks that reconstruct inputs such as Masked Language Modeling (MLM) [13, 52] and Casual Language Modeling (CLM) [7, 48] have exhibited the capability of learning knowledge from large corpora of text extracted from web crawls, which translates well to downstream task performance. On the other hand, although the input reconstruction type of pre-training for VLMs have shown performance improvements in certain settings, in general they are less effective compared to what is observed in language domain [10, 34] due to heterogeneity of vision tasks [12, 35, 43]. In addition to self-supervised learning tasks, many VLMs [10, 61] use a mixture of supervised pre-training tasks (e.g. object detection, VQA, captioning, etc), relying on human manually labeled datasets such as COCO [43], Object365 [55], VQA [22], etc as well as datasets generated in automated fashion such as LAION-5B [54], WebLi-10B [10].

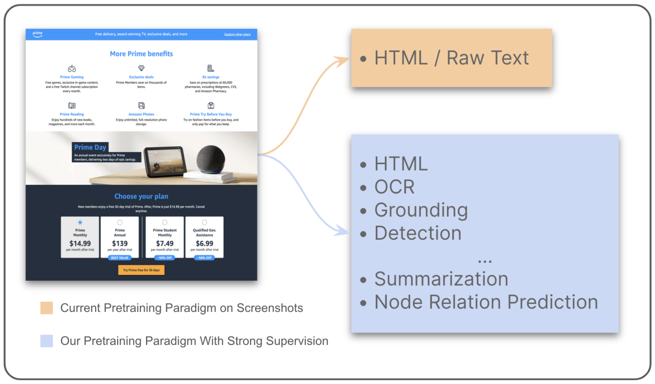

• HTML / Raw Text
• HTML
• OCR
• Grounding
• Detection
...
• Summarization
• Node Relation Prediction
Current Pretraining Paradigm on Screenshots
Our Pretraining Paradigm With Strong Supervision

Figure 1. We propose a novel pre-training paradigm - S4, composed of ten carefully designed tasks on large scale web-screenshots. Compared to image-to-text pretraining objectives on screenshots, which mainly utilized HTML[36] or its subset like raw texts[31, 37], our paradigm utilizes rich and diverse supervisions generated from web rendering that is also cheap to obtain.

Advancements of supervised datasets have powered the advancements of VLMs. Increasing amounts of human annotated datasets were released [32, 33], which benefit the performance of similar or relevant downstream tasks, albeit at a steep cost. Approaches that can automatically generate supervisions at scale have also been explored [36, 54].

[*] Equal contribution
[†] Work conducted during an internship at Amazon.

Notably, the use of massive amount of image-caption pairs, which are automatically generated using images and their associated Hypertext Markup Language (HTML) alt-text has enabled the development of some important VLMs such as CLIP models [50] and diffusion models [53]. Similarly, the use of screenshots and simplified HTML text pairs powered the Pix2Struct models [36]. However, methods capable of producing automatically annotated data beyond basic image-text pairs are currently under explored. Consequently, the effects of employing explicit, automatically generated, fine-grained supervision for pre-training have been understudied.

Therefore, in this work, we extend the use of web crawl corpuses and propose a novel pre-training framework that utilizes rich and diverse supervisions generated from web rendering. Modern websites are built using a combination of technologies such as HTML, CSS, JavaScript, that enable website creators to design dynamic content and interactive elements with diverse layouts.

To leverage such information, our solution renders crawled web-pages into screenshot images. We also have access to textual content, position, attribute and relationship of HTML elements - all of which can be obtained cheaply and utilized in pre-training. Building on this extracted data, we propose a set of pre-training tasks (see details in 3.2) that are highly synergistic to downstream tasks. Our results demonstrate significant performance improvements compared to the image-to-text pre-training baseline. On average, we observed an improvement of **+2.7%** points across 5 datasets (ChartQA, RefExp, Widget Captioning, Screen Summarization and WebSRC) with language outputs, and a notable average increase of **+25.3%** points on 4 datasets (PubLayNet, PubTables1M, RefExp candidate free and ICDAR 2019 modern) with localization outputs. See more in Tables 1 and 2. Our key contributions:

- We develop an automatic data annotation pipeline that is able to render web crawls and create rich labels. Coupled with our carefully designed data cleaning process, we create a high-quality and large-scale vision language pre-training dataset.
- We propose a novel pre-training paradigm - S4, composed of ten carefully designed tasks on large scale web-screenshots showing the effectiveness on a wide range of benchmarks.
- Comparing to current screenshot pre-training objectives, our innovative pre-training method significantly enhances performance of image-to-text model in nine varied and popular downstream tasks - up to 76.1% improvements on Table Detection, and at least 1% on Widget Captioning.

2. Related Work

Next, we discuss in detail the difference to previous pre-training approaches.

Masked Signal Modeling. Pre-training through self-supervision has revolutionized the field of natural language processing (NLP). Pioneering models like BERT [52] and GPTs [7, 48] demonstrated the profound impact of self-supervised learning in enhancing generalization across a variety of language tasks. The success in NLP spurred analogous research in computer vision, leading to innovations in Masked Image Modeling (MIM) with approaches such as BEiT [5], SimMIM [64], and MAE [24] that recover masked pixels or patches, and improvements on classic vision tasks such as classification, semantic segmentation, etc are observed. In the domain of Vision Language(VL), MLIM [3] and MaskVLM [34] propose to integrate MLM and MIM and conduct VL pretraining in a joint manner.

Supervised Pre-training In supervised pre-training, image-caption pair annotations are generated on a large scale automatically from web crawls. This enables the training of models that generalize well in tasks like classification, retrieval, and captioning, as seen in works like CLIP, OFA, PaLi [10, 50, 61]. Donut [31] proposes a OCR-free model that relies on text reading pre-training of documents. SPOTLIGHT uses [37] region to text pre-training task on website and UI datasets. Pix2Struct [36] leverages screenshot and image pairs with a screen parsing pre-training task that converts webpage screenshots to HTML text. Our work proposes a pre-training paradigm that goes beyond image-text pairing type of tasks. We develop a suite of diverse, heterogeneous tasks specifically crafted to mirror the nature of downstream applications.

3. S4 Pre-training

In this section, we propose a novel pre-training paradigm for Vision-Language Models — Strongly Supervised pre-training with ScreenShots (S4) from large scale website rendering. We will first describe the creation procedure of our dataset for S4 pretraining, which we will call S4 Data, and then go through our proposed pre-training tasks enabled by our novel preprocessing method.

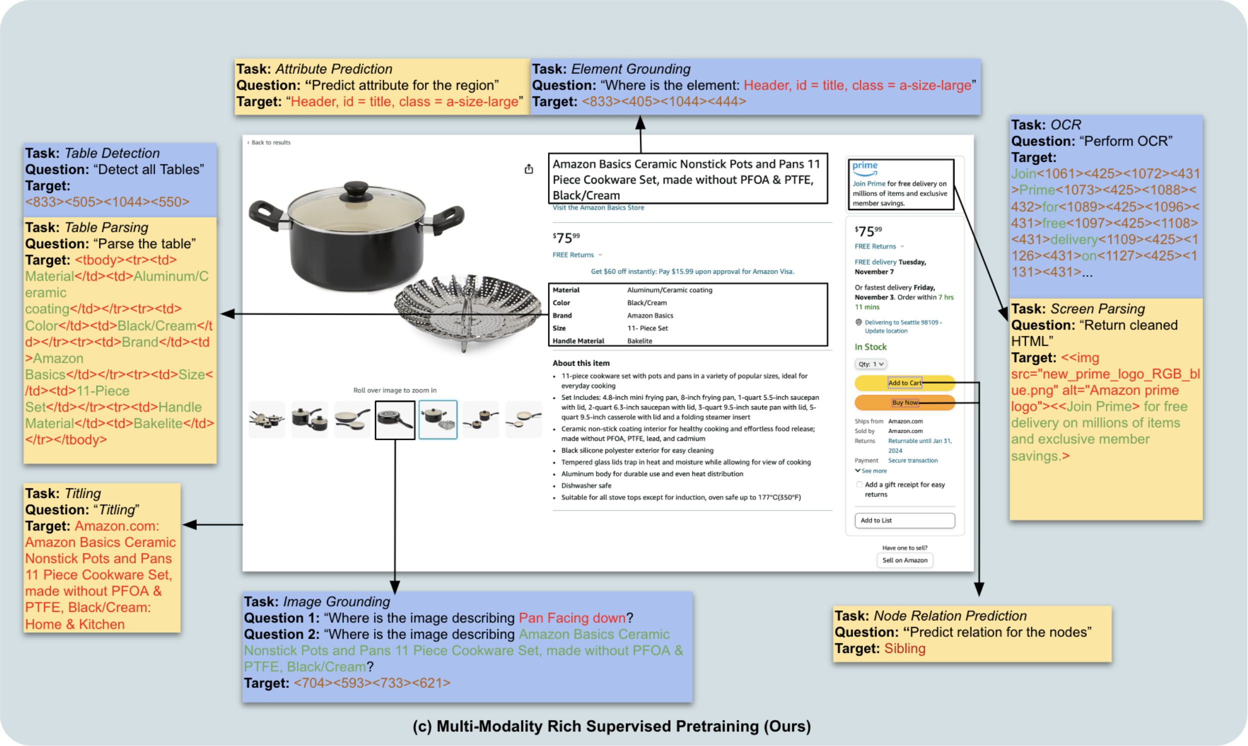
3.1. Dataset

3.1.1 Dataset Description

CommonCrawl[†] provides access to a large-scale web page corpus spanning over a decade. We download the web crawls from the Registry of Open Data on AWS[†] and we filter content with an explicit copyright notice. We execute our rendering and extraction pipeline (described in 3.1.2) and data pre-processing and cleaning procedure (described in 3.1.3) to obtain 15M screenshots enriched with supervisions. We applied deduplication based on urls to make sure our screenshots are unique. Each page is rendered at a

[†] http://commoncrawl.org/

[†] https://registry.opendata.aws/commoncrawl/



Task: *Attribute Prediction*
Question: "Predict attribute for the region"
Target: "Header, id = title, class = a-size-large"

Task: *Element Grounding*
Question: "Where is the element: Header, id = title, class = a-size-large"
Target: <833><405><1044><444>

Task: *Table Detection*
Question: "Detect all Tables"
Target: <833><505><1044><550>

Task: *Table Parsing*
Question: "Parse the table"
Target: <tbody><tr><td>Material</td><td>Aluminum/Ceramic coating</td></tr><tr><td>Color</td><td>Black/Cream</td></tr><tr><td>Brand</td><td>Amazon Basics</td></tr><tr><td>Size</td><td>11-Piece Set</td></tr><tr><td>Handle Material</td><td>Bakelite</td></tr></tbody>

Task: *Titling*
Question: "*Titling*"
Target: Amazon.com: Amazon Basics Ceramic Nonstick Pots and Pans 11 Piece Cookware Set, made without PFOA & PTFE, Black/Cream: Home & Kitchen

Task: *Image Grounding*
Question 1: "Where is the image describing Pan Facing down?
Question 2: "Where is the image describing Amazon Basics Ceramic Nonstick Pots and Pans 11 Piece Cookware Set, made without PFOA & PTFE, Black/Cream?
Target: <704><593><733><621>

Task: *OCR*
Question: "Perform OCR"
Target: Join<1061><425><1072><431>Prime<1073><425><1088><432>for<1089><425><1096><431>free<1097><425><1108><431>delivery<1109><425><1126><431>on<1127><425><1131><431>...

Task: *Screen Parsing*
Question: "Return cleaned HTML"
Target: <<img src="new_prime_logo_RGB_blue.png" alt="Amazon prime logo"><<Join Prime> for free delivery on millions of items and exclusive member savings.>

Task: *Node Relation Prediction*
Question: "Predict relation for the nodes"
Target: Sibling

Amazon Basics Ceramic Nonstick Pots and Pans 11 Piece Cookware Set, made without PFOA & PTFE, Black/Cream
$75.99
Material Aluminum/Ceramic coating
Color Black/Cream
Brand Amazon Basics
Size 11- Piece Set
Handle Material Bakelite
About this item
prime
Join Prime for free delivery on millions of items and exclusive member savings.
In Stock
Add to Cart
Buy Now
Add to List
(c) Multi-Modality Rich Supervised Pretraining (Ours)

Figure 2. Compared to traditional pre-training paradigms, our rich supervised pre-training leverages much more information that is also cheap to acquire (i.e via browser). We can then utilize the rich semantic and structural annotations to construct novel pre-training tasks that are naturally and directly aligned with downstream tasks. We use green words to refer to the words contained (visible) in the screenshot. We use red words to refer to the words that are not visible in the screenshot. For instance, "price" is not shown on the screenshot, but is the id of an element (refer to picture). We use brown words in the format of <x><y><x><y> to denote the bounding box.

resolution of 1280x1280 and is paired with matching annotations that enable the proposed pre-training tasks described in 3.2.

3.1.2 Efficient Rendering and Supervision Extraction

We use Playwright [†], which provides a programmatic interface to headless browsers that we use to render raw HTML files into screenshots. For each web page, we retrieve and cache the associated CSS, JavaScript fonts and images needed to render the page accurately. Caching those assets avoids making unnecessary requests to the originating website and quicker rendering, allowing us to create 5M parsed screenshot per day with 500 CPUs.

We build the annotations by traversing through the document object model (DOM) tree and collecting annotations for every leaf node of type Text, Image, Table or Input. More information about the dataset and example annotation can be found in the supplementary material.

3.1.3 Pre-processing and Cleaning

During data rendering, we found that directly traversing through the DOM tree and collecting information on each node would lead to the inclusion of elements that were not visible in the page. We solve this issue by inspecting their CSS property for visibility and verifying the alignment to their corresponding bounding box. Specifically, if the element `elem_b` returned by clicking on the center `elem_a`'s bounding box is not a descendent of `elem_a`, then `elem_a` is pruned. This simple heuristics helps us get rid of most of the annotation that contains invisible elements. Also, we implemented recursive pre-order traversal to filter out overflow words in a textnode where the texts are overflowing outside of it's ancestor's bounding box. Without such a filter, words that are occluded by other elements would be included in the final annotation. Finally, we get rid of all `<iframe>` tags since the Same Origin Policy prohibits direct access of the nodes in `<iframe>`.

3.2. Pre-training Task construction

Using the rich information provided by the HTML document structure, we design ten diverse supervised objec-

[†] https://github.com/microsoft/playwright

tives, which improve upon previous supervision like Screen Parsing. Our tasks include: Screen Parsing, OCR, Image Grounding, Element Grounding, Attribute Prediction, Node Relation Prediction, Table Detection, Table Parsing, Screen Titling, and Layout Analysis. We describe the objctives in the sections below, as well as in Figure 2.

3.2.1 Screen Parsing

Similar to Pix2Struct, our Screen Parsing objective aims to reconstruct both the texts and their underlying structure. As shown in Figure 2, the input is simply a screenshot with a bounding box drawn on a region, with words 50% masked words, and the target is the cleaned HTML. We obtain the cleaned and simplified HTML as described in Pix2Struct by removing invisible nodes, and recursively remove chained nesting.

3.2.2 Optical Character Recognition - OCR

The OCR objective aims to train the model with the ability for spatial understanding. It takes in a screenshot with drawn bounding box specifying a region, and outputs a “word_0<x_0><y_0><x_0><y_0>word_1<x_1><y_1><x_1><y_1>...” sequence. To limit the sequence length, we choose bounding box for a random region that contains at most 50 words. The OCR objective empowers the model with the ability to spatially understand the screenshot, bringing benefits for general detection and grounding tasks.

3.2.3 Image Grounding

Image grounding is an important aspect to enhance image-text alignment, and we want to empower our model with the ability to understand the semantics of the images in the context of a screenshot. We formulate the Image grounding objective as follow: First, we randomly pick an `<img>` element from the screenshot. Then, we obtain two captions for the image: one is from the alt attribute, which we call `alt_caption`, and second is from the text node that is closest to the `<img>` element in the HTML Tree, which we call `neighbor_caption`. We then randomly choose from {`alt_caption`, `neighbor_caption`} and ask the model to predict the bounding box of the image described. Note that since the `neighbor_caption` appears in the screenshot, the model can, instead of learning the image-text relation, simply cheat to locate the `neighbor_caption` first and then predict the bounding box for the image closest to it. Therefore, to avoid leaking spatial information from texts, we mask out 90% of the texts for the input screenshot to the model.

3.2.4 Element Grounding

Element grounding is a generalization of image grounding to other elements in the HTML DOM tree. To build a better representation of the meaning and functionality of each elements shown in the screenshot, we ask the model to localize their position based on a text description. We obtain the text description by concatenating the element tag and attributes from {class, id, label, for, alt, title, type}. However, values of the attributes are often noisy as the id and class label of an element can be randomized (i.e, in web frontend frameworks such as React.js). We address this issue by adding a post-processing step that filters out words that are numerical, single characters or that combines letters and numbers, as they are unlikely to useful labels. As a final step we use the T5 tokenizer to get rid of strings that map to $< unk >$ tokens.

3.2.5 Attribute Prediction

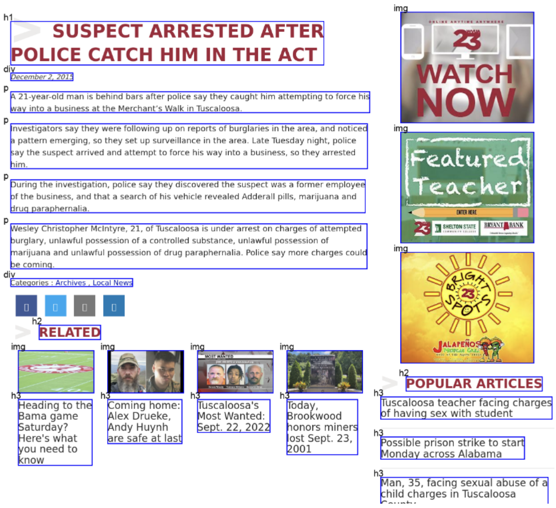
Beyond elements grounding from descriptions, we also ask the model to predict a matching description for a region in HTML. We group the visible elements into groups where they contain the same tag and attributes within {class, id, label, for, alt, title, type}, and randomly specify a group by rendering its bounding box to the input screenshot. The model is then asked to predict the tag and attributes in the following format: “{tag} {tag.class} {tag.id} {tag.label} {tag.for} {tag.alt}”. We apply the same post-processing described in 3.2.4 to filter out noise in the attribute values. The Attribute Prediction task forces the model to reason about the semantic meaning of each element, which could bring benefits downstreams tasks that involves element-level understanding.

3.2.6 Node Relation Prediction (NRP)

This task is a pixel-only adaptation of the Node Relation Prediction objective introduced by MarkupLM[38], which takes the tree-structure of HTML and labels the relationship as either {self, parent, child, sibling, ancestor, descendent, others}. Given two elements outlined with bounding boxes in the input image, the model has to predict their node-level relationship. This task is expected to force the model to learn the relationships between the various layout components and how they interact.

3.2.7 Table Detection

To closely mimic the downstream task for table detection, we construct table detection on our screenshot data. The construction is as simple as merging the bounding box for the elements with `<table[id]>` contained in their Xpaths, which results in the ground truth bounding box for each table. We then ask the model to predict the following sequence:<x_{table0}><y_{table0}><x_{table0}><y_{table0}><x_{table1}><y_{table1}><x_{table1}><y_{table1}>...



SUSPECT ARRESTED AFTER POLICE CATCH HIM IN THE ACT

December 2, 2015

A 21-year-old man is behind bars after police say they caught him attempting to force his way into a business at the Merchant’s Walk in Tuscaloosa.

Investigators say they were following up on reports of burglaries in the area, and noticed a pattern emerging, so they set up surveillance in the area. Late Tuesday night, police say the suspect arrived and attempt to force his way into a business, so they arrested him.

During the investigation, police say they discovered the suspect was a former employee of the business, and that a search of his vehicle revealed Adderall pills, marijuana and drug paraphernalia.

Wesley Christopher McIntyre, 21, of Tuscaloosa is under arrest on charges of attempted burglary, unlawful possession of a controlled substance, unlawful possession of marijuana and unlawful possession of drug paraphernalia. Police say more charges could be coming.

Categories : Archives , Local News

RELATED

Heading to the Bama game Saturday? Here's what you need to know

Coming home: Alex Drueke, Andy Huynh are safe at last

Tuscaloosa's Most Wanted: Sept. 22, 2022

Today, Brookwood honors miners lost Sept. 23, 2001

POPULAR ARTICLES

Tuscaloosa teacher facing charges of having sex with student

Possible prison strike to start Monday across Alabama

Man, 35, facing sexual abuse of a child charges in Tuscaloosa

Figure 3. Visualization of layout parsed from a screenshot. Corresponding HTML tags like <h1> are visualize on top-left corner of the bounding box.

3.2.8 Table Parsing

The original Screen Parsing objective, although encouraging structure-level understanding, does not emphasize the semantics of those structures, as the pre-processing replaces tags with empty brackets. We argue that the information contained in the tags is also useful signal for pre-training, especially for well-structured elements like `<table>`. Therefore, we design a table parsing objective which contains the original tag name as well as the text contents for tables inside a page, as shown in Figure 2.

3.2.9 Screenshot Titling

To encourage the model to summarize the content in the screenshot and improve its ability on image captioning, we propose a screen titling task. Specifically, the main title in the screenshot is masked and the model is asked to generate the title text by only looking at the rest of the web page. The ground truth title is obtained from the $< title >$ node of the HTML DOM tree. The Screenshot Titling task closely resembles the screen summarization task for UI understanding.

3.2.10 Layout Analysis

Obtaining layout from a screenshot is realized by grouping elements under the same sub-tree in the HTML. Specifically, for each element we obtain its *cleaned Xpath* by only keeping tags in `[<p>,<table>,<form>,<dl>,<button>,<ol>, <ul>,<nav>,<img>,<object>]` as they represent the semantic abstraction of the element. Then, we group each elements according to the value of their *cleaned Xpath* to form layout of the screenshot. A visualization of the layout from a screenshot is shown in Figure 3.

3.3. Architecture

We adopt a simple architecture with an image encoder followed by a text decoder, same as Pix2Struct [36] and similar to Donut [31]. The Image encoder is ViT [14] and text decoder is transformer decoder, where the vocabulary is extended with 1000 coordinate tokens (representing discrete positions in images, normalized between 0-1000) to support localization tasks such as object detection and visual grounding. Such image-encoder-text-decoder models don't need text input and have the advantage of being OCR-free, which leads to reduced latency [31]. On the other hand, in order to read textual content that are typically small, input image resolution has to be high for good performance, which leads to increased memory usage. Our proposed S4 pre-training paradigm is not limited to this architecture and can be applied to other approaches as well.

4. Experiments

We validate the effectiveness of our ten proposed pre-training tasks by fine-tuning the model on nine downstream tasks and compare its performance to a Pix2Struct baseline model that was only pre-trained with screen parsing. Based on the output format, we also divide the downstream tasks into two groups.

4.1. Implementation Details

Pre-training schema. We propose 2 pre-training schemes, S4$_{NL}$ for natural language generation and S4$_{Loc}$ for localization, targeting on different downstream tasks. Specifically, S4$_{NL}$ includes the baseline screen parsing task and all the tasks on natural language generation, including Attribute Prediction, Table Parsing, Title Generation, and Node Relation Prediction. S4$_{Loc}$ comprises of tasks with bounding box generations, including OCR, Image Grounding, Element Grounding, Table Detection and Layout Analysis, in addition to the screen parsing task. During pre-training, we randomly sample one task for each image with uniform distribution.

4.1.1 Pretraining Settings

We conducted pretraining on both 2 million and 15 million subsets of our S4 dataset, during which we set the screenshot's viewport to 1280*1280. We initialize our model with weights from Pix2struct-base and set our batch size to 32 for each node and use 4 A100 nodes during pretraining. The maximum sequence length for the pretraining targets is 128 and the patch size for the input image is 2048. Our optimizer is AdamW with learning rate set to 1e-4 with cosine decay, and for both 2M and 15M subsets we pretrain with

Methods	Pre-training Dataset	Pre-training Objectives	Finetune Batchsize	ChartQA ↑	RefExp$_{\text{cls}}$↑	Widget ↑ Cap.	Screen ↑ Sum.	WebSRC ↑
Pix2Struct [36]	Google Priv. Data - 80M	Screen Parsing	32 to 256	56.0	92.2	133.1	107.0	-
Pix2Struct†	Google Priv. Data - 80M	Screen Parsing	8	54.3	91.7	131.1	105.5	60.4
Donut [31]	SynthDoG - 37M	OCR	64	41.8	-	127.4	56.4	-
Pix2Struct*	S4 Data - 2M	Screen Parsing	8	47.4	87.9	129.5	101.3	58.7
Pix2Struct*	S4 Data - 15M	Screen Parsing	8	52.1	88.1	129.2	104.5	60.1
S4* **(Ours)**	S4 Data - 2M	S4$_{NL}$	8	50.5	92.4	130.5	103.2	60.5
S4* **(Ours)**	S4 Data - 15M	S4$_{NL}$	8	**55.0**	**94.9**	**130.6**	**105.7**	**61.1**

Table 1. Results for Chart, Web, and UI Understanding datasets. * denotes that we load Pix2Struct's pre-trained weight and further pre-train on our S4 dataset with corresponding objectives. † denotes the reproduced results on downstream tasks with Pix2Struct-base's pre-trained weight and smaller batch size. Results from gray rows are not directly comparable to our S4 model since we don't have access to their non-released pre-training datasets. The results from last 4 rows show that in addition to the Pix2Struct's pre-training objective, our supervised pre-training extracted from HTML DOM tree brings consistent improvement on various of downstream tasks. Note that the OCR objective for donut doesn't include bounding box prediction.

1 epoch per pretraining task. For instance, for S4$_{NL}$ pretraining with the 2M subset, there are 5 tasks so the total training sample is 5*2M = 10M. Note that since for each screenshots we can obtain multiple tasks, the models sees the same subset of screenshots regardless of the number of pretraining tasks.

4.2. Chart, Web, and UI Understanding

In this section we evaluate on tasks that require generating natural language responses from image inputs. We focus on Chart & Web VQA, UI summarization and UI widget captioning.

4.2.1 Datasets

ChartQA: ChartQA[46] is a VQA dataset for different types of charts (bar charts, line graphs, etc.). It includes both extractive and reasoning questions, which requires analyzing the visual data in charts to extract the relevant information. We follow the convention and report the Relaxed Match metric on ChartQA.

WebSRC: WebSRC[9] is a web-based VQA dataset. It contains both cleaned HTML and the screenshot of web pages, and the task is to answer the question about the content in the web page. Prior arts mostly tackle this problem by taking the ground truth cleaned HTML code as inputs, which is an unrealistic setting as real-word applications often have much more complex HTML codes than the cleaned data. Instead, our model only takes the screenshot as inputs and predicts the answer from pure-vision information. On WebSRC the Exact Match metric is reported.

Screen2words: Screen2words[59] is a dataset for extracting summarization from screenshots of mobile app screens. We use Bleu and Cider scores as the evaluation metrics.

Widget Captioning: Widget Captioning[42] is a dataset for generating descriptive captions for UI widgets. The task is to generate captions that accurate describe the purpose or function of a widget in a bounding box, such as a button, slider, or a checkbox. In the input screenshot, the target widget is specified through the rendered bounding box. Bleu and Cider scores are used as evaluation metrics.

UI RefExp$_{\text{cls}}$ UI Referential Expression (RefExp) [4] is a dataset specifically designed for grounding referring expressions for UI elements in screenshots. Current SOTA usually approach this problem in a simplified classification formulation: given a question and a candidate widget from annotation, the model is asked to predict whether the widget and question are related. This setting requires little localization ability from the model as the candidates widget bounding boxes are provided as inputs. We call this classification setting RefExp$_{\text{cls}}$and report the classification accuracy as the metric.

4.2.2 Settings

Following the same training and evaluation protocol [36], our model was pre-trained with S4$_{NL}$ objectives and fine-tuned on the Chart, Web, and UI Understanding tasks. Since there's no access to Google's 80M private data, we compare to two PixStruct variations. The first one is with the weights pre-trained on its private data using screen parsing released by the original author. The second one is initialized with the former's weights, and is further pre-trained on our 2M and 15M S4 data with only screen parsing objective. To have a fair comparison, our model is initialized with the same weights, and pre-trained on the same amount of data (2M and 15M S4 data) but with extra tasks. We also compare to Donut[31], which is another model uses pure-vision inputs and produces text predictions.

4.2.3 Results

We tabulate the results in Tab. 1. Our method consistently outperforms Pix2Struct on all downstream tasks with significant margins when pre-trained with the same

Methods	Pre-training Dataset	Pre-training Objectives	RefExp ↑	PublayNet ↑	PubTables1M ↑	ICDAR 2019 ↑
			cand_free		(Table Det.)	(Modern Subset)
			30k samples	1M samples	400k samples	600 samples
DETR	-	-	-	-	99.5	-
DiT-B (Cascade RCNN)	IIT-CDIP - 42M	MIM	-	95.4	-	97.2
Pix2Struct [36]	Google Priv. Data - 80M	Screen Parsing	55.1	91.1	97.0	3.6
Pix2Struct*	S4 Data - 2M	Screen Parsing	52.7	91.0	97.1	3.3
S4* (**Ours**)	S4 Data - 2M	S4$_{Loc}$	83.6	92.5	98.4	70.7
S4* (**Ours**)	S4 Data - 15M	S4$_{Loc}$	**84.3**	**93.1**	**99.0**	**79.4**

Table 2. Results on Detection and Grounding datasets. We train all of the models with batch size = 32 and patch size = 2048.* denotes that we load Pix2Struct-base's pre-trained weight and further pre-train on our S4 dataset with corresponding objectives. Models denoted gray are specialist detection models that cannot parse language input (i.e cannot do grounding tasks). With our pre-training objectives, autogressive models can get significant boosts on various detection & grounding tasks. MIM refers to Masked Image Modeling.

data. Specifically, when pre-trained with 15M image-text pairs, our method achieves 2.9, 6.8, 1.4, 1.2, and 1.0 improvement over Pix2Struct on ChartQA, RefExp$_{\text{cls}}$, Widget Captioning, Screen Summarization, and WebSRC, respectively. Notice that the largest improvement is obtained on RefExp$_{\text{cls}}$, because the pre-training tasks we proposed, such the attribute prediction and node relation prediction, help the model build robust connections between the UI elements and their referring expressions. In addition, when more data (15M) is available, our pre-training scheme S4$_{NL}$ gets improved accuracy compared to less training data (2M). The outstanding results comparing to the baseline and Donut demonstrate the efficacy of the proposed S4 pre-training scheme for downstream tasks that involve chart, web, and UI understanding.

We also noticed that, comparing to the original Pix2struct, further pre-training it on our 2M data with screen parsing objective harms the performance across different datasets. This is expected as our data collected from Common Crawl has different distribution against the original Pix2struct data due to the data size (2M vs. 80M) and potentially different website filtering strategies. Specifically, we filter out website whose CSS and JS files failed to download, while the filtering process remain unclear for Pix2struct. In addition, we also used a much smaller batch size (128 vs. 2048) due to computational constraints. Therefore, the pre-train on 2M data might drive the model weight to a less optimal state. With 15M pre-training data, we observed performance improvements over 2M data, implying that more data might compensate this distribution shift.

4.3. Detection and Grounding

We further investigate the effect of S4 pre-training on tasks that require spatial information understanding, such as image grounding and localization. While current SOTA models adopt specific architectures for detection, we show that with sufficient pre-training tasks on localization, an autogressive model can close the gap towards detection-specific architectures.

4.3.1 Datasets

PubLayNet: PubLayNet [76] is a large-scale dataset for document layout analysis, containing more than 360,000 pages of scientific articles. We evaluate on the bounding box prediction task and report AP 50 as the metric.
ICDAR2019: ICDAR2019 [19] is a table detection dataset that contains 1200 modern and archived documents. We only evaluate on the modern document split as the archived documents do not have bounding boxes. We use AP 50 as our evaluation metric.
PubTables-1M PubTables-1M [58] has around 400k images with 947,642 tables from PMCOA scientific articles and we use it for table detection experiments. AP 50 is used as the evaluation metric.
UI RefExp$_{\text{cand\_free}}$ As mentioned earlier, current works mostly treat the UI RefExp task as a binary classification problem using the ground truth bounding boxes as candidates, making it less challenging as it does not measure whether the model can localize the UI elements. In this work, we propose a new task, **UI RefExp**$_{\text{cand\_free}}$, to include the element grounding into the challenges. In this task, the input is only the screenshot with the text description, and the model is asked to predict the bounding box of the related UI element directly, thus ”candidate free”. For evaluation, the predicted bounding box will be matched to the closest ground truth box to compute the accuracy.

4.3.2 Settings

Our model was pre-trained with S4$_{Loc}$ for the benefits on localization related tasks. The model is then fine-tuned and evaluated on each downstream task dataset. We compare to

Titling	Attibute Pred.	NRP	Table Parsing.	Screen Parsing	ChartQA	Widget Cap.	Screen Sum.	RefExp$_{\text{cls}}$
				✓	47.4	129.5	101.3	87.9
			✓	✓	48.7	128.3	101.4	87.8
		✓	✓	✓	**50.7**	128.3	101.3	89.4
	✓	✓	✓	✓	50.1	**130.7**	101.1	**92.7**
✓	✓	✓	✓	✓	50.5	130.5	**103.2**	92.4

Table Detection	Layout Analysis	Image & Element Grounding	OCR	Screen Parsing	ICDAR	RefExp$_{\text{cand\_free}}$
				✓	3.3	52.7
			✓	✓	50.1	68.6
		✓	✓	✓	52.9	66.2
✓	✓	✓	✓	✓	**70.7**	**83.6**

Table 3. Ablation study on adding different pre-training objectives using 2M S4 data.

two PixStruct variations. The first one is the weights pre-trained on its private data using screen parsing released by the original author. The second one is initialized with the former weights and is further pre-trained on our S4 data 2M with only screen parsing, to be compared to our model pre-trained on the same amount of data.

4.3.3 Results

The evaluation results on all detection and grounding related tasks are tabulated in Tab. 2. Our method shows clear advantages over the baseline Pix2Struct model that was only pre-trained with the screen parsing task. Specifically, on RefExp$_{\text{cand\_free}}$, when pre-trained with 2M data, our method outperforms Pix2Struct by a significant margin of 30.9 (83.6 vs. 52.7). This is because our pre-training tasks like OCR prediction and element grounding, help the model learn to localize items in the image given their semantic descriptions. Similarly, on ICDAR, which only has 600 training samples, our method achieves 70.7 AP with 2M pre-training data, while the baseline Pix2Struct only obtains 3.3, due to its lack of localization ability. When there are more fine-tuning data for the downstream tasks (PublayNet 1M & PubTables 400K), Pix2Struct can learn to localize the objects and obtain decent performance, but our training scheme S4$_{Loc}$ still benefits the model and improves the performance by 1.5 on PublayNet and 1.3 on PubTables.

The benefits of S4$_{Loc}$ pre-training becomes more prominent when more pre-train data is available. Our model pre-trained with 15M data consistently improves the accuracy on all four downstream tasks compared to 2M pre-train data. In particular, on ICDAR the 15M pre-train data improves the accuracy from 70.7 to 79.4, showing that the pre-training task benefits more when the downstream task has less data. It is worth noting that, as a generic auto-regressive text generation model, our method with 15M pre-training data achieves comparable performance on PublayNet and PubTables to detection specific models like DeTR and Dit-B, showing that sufficient pre-training with proper tasks helps close the gap between auto-regressive models and detection-specific architectures.

4.4. Contribution of each task

We conducted ablative studies to show the effectiveness of each individual pre-training tasks besides screen parsing. For S4$_{NL}$, we evaluate on ChartQA, Widget Captioning, Screen Summarization, and RefExp$_{\text{cls}}$, by adding the natural language related tasks gradually. For S4$_{Loc}$, we also add the localization related tasks incrementally and evaluate on RefExp$_{\text{cand\_free}}$ and ICDAR. The results are shown in Tab. 3. Observe that the downstream task usually benefits from the addition of the most related pre-training task. For example, Screen Summarization gets 2.1 performance improvement when the screen tilting pre-training task is added, while the other tasks have little effect on the performance. The attribute prediction task encourages the model to associate website elements to their text description. Therefore, adding it to the pre-training scheme significantly improves the performance on both Widget Captioning and RefExp$_{\text{cls}}$, which requires the model to associate UI elements to texts. Similarly, adding all the localization related pre-train task substantially improves the model's ability on grounding elements, resulting in higher accuracy on both ICDAR and RefExp$_{\text{cand\_free}}$.

5. Conclusions

We introduced a novel pre-training framework for vision-language models, with which models are exposed with a variety of supervised tasks on diverse and massive amount website screenshots. This innovation is enabled by our proposed data pipeline, in which the web pages are rendered, extracted, and cleaned automatically to generate screenshots and corresponding annotations. The tasks in our pre-training scheme are designed to maximize the utilization of annotations in our data as well as the similarities between the downstream tasks. Through extensive experiments, we demonstrated the efficacy of our method on boosting the downstream tasks performance on 9 different datasets.

References

[1] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 993–1003, 2021. 1

[2] Srikar Appalaraju, Peng Tang, Qi Dong, Nishant Sankaran, Yichu Zhou, and R. Manmatha. Docformerv2: Local features for document understanding. *AAAI*, abs/2306.01733, 2024. 1

[3] Tarik Arici, Mehmet Saygin Seyfioglu, Tal Neiman, Yi Xu, Son Train, Trishul Chilimbi, Belinda Zeng, and Ismail Tutar. Mlim: Vision-and-language model pre-training with masked language and image modeling. *arXiv preprint arXiv:2109.12178*, 2021. 2

[4] Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, et al. Uibert: Learning generic multimodal representations for ui understanding. *arXiv preprint arXiv:2107.13731*, 2021. 6

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2

[6] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R. Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16548–16558, 2022. 1

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2

[8] Feilong Chen, Duzhen Zhang, Minglun Han, Xiuyi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20: 38–56, 2022. 1

[9] Xingyu Chen, Zihan Zhao, Lu Chen, Danyang Zhang, Jiabao Ji, Ao Luo, Yuxuan Xiong, and Kai Yu. Websrc: A dataset for web-based structural reading comprehension, 2021. 6

[10] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 1, 2

[11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 1

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5

[15] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, Jianfeng Gao, and Lijuan Wang. Coarse-to-fine vision-language pre-training with fusion in the backbone. *ArXiv*, abs/2206.07643, 2022. 1

[16] Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. In *International Joint Conference on Artificial Intelligence*, 2022.

[17] Roy Ganz, Oren Nuriel, Aviad Aberdam, Yair Kittenplon, Shai Mazor, and Ron Litman. Towards models that can see and read. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 21718–21728, 2023.

[18] Roy Ganz, Yair Kittenplon, Aviad Aberdam, Elad Ben Avraham, Oren Nuriel, Shai Mazor, and Ron Litman. Question aware vision transformer for multimodal reasoning. *arXiv preprint arXiv:2402.05472*, 2024. 1

[19] Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. Icdar 2019 competition on table detection and recognition (ctdar). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1510–1515, 2019. 7

[20] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. 1

[21] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Jiao Qiao. Clip-adapter: Better vision-language models with feature adapters. *ArXiv*, abs/2110.04544, 2021. 1

[22] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 1

[23] Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. Mixgen: A new multi-modal data augmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 379–389, 2023. 1

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2

[25] Chih-Hui Ho, Srikar Appalaraju, Bhavan Jasani, R. Manmatha, and Nuno Vasconcelos. Yoro - lightweight end to end visual grounding. In *ECCV Workshops*, 2022. 1

[26] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pretraining for image captioning. *2022*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 17959–17968, 2021.

[27] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing out of the box: End-to-end pre-training for vision-language representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12971–12980, 2021.

[28] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021.

[29] Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. In *Annual Meeting of the Association for Computational Linguistics*, 2021.

[30] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr - modulated detection for end-to-end multi-modal understanding. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1760–1770, 2021. 1

[31] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer, 2022. 1, 2, 5, 6

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[33] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 1

[34] Gukyeong Kwon, Zhaowei Cai, Avinash Ravichandran, Erhan Bas, Rahul Bhotika, and Stefano Soatto. Masked vision and language modeling for multi-modal representation learning. *arXiv preprint arXiv:2208.02131*, 2022. 1, 2

[35] Justin Lazarow, Kwonjoon Lee, Kunyu Shi, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10720–10729, 2020. 1

[36] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding, 2023. 1, 2, 5, 6, 7

[37] Gang Li and Yang Li. Spotlight: Mobile ui understanding using vision-language models with a focus. 2023. 1, 2

[38] Junlong Li, Yiheng Xu, Lei Cui, and Furu Wei. Markuplm: Pre-training of text and markup language for visually-rich document understanding. 2021. 4

[39] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 2022. 1

[40] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10955–10965, 2021.

[41] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020. 1

[42] Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: Generating natural language description for mobile user interface elements, 2020. 6

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[44] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Neural Information Processing Systems*, 2019. 1

[45] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *ArXiv*, abs/2206.08916, 2022. 1

[46] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, 2022. 6

[47] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022. 1

[48] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1, 2

[49] 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. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 1

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[51] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and

Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019. 1

[52] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 1, 2

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[54] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 1

[55] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 1

[56] Kunyu Shi, Qi Dong, Luis Goncalves, Zhuowen Tu, and Stefano Soatto. Non-autoregressive sequence-to-sequence vision-language models, 2024. 1

[57] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *ArXiv*, abs/2209.07511, 2022. 1

[58] Brandon Smock, Rohith Pesala, and Robin Abraham. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4634–4642, 2022. 7

[59] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning, 2021. 6

[60] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, 2022. 1

[61] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 1, 2

[62] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *ArXiv*, abs/2111.02358, 2021.

[63] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *ArXiv*, abs/2208.10442, 2022. 1

[64] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 2

[65] Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao, and Fei Huang. E2e-vlp: End-to-end vision-language pre-training enhanced by visual learning. *ArXiv*, abs/2106.01804, 2021. 1

[66] Jinyu Yang, Jiali Duan, S. Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul M. Chilimbi, and Junzhou Huang. Vision-language pre-training with triple contrastive learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15650–15659, 2022.

[67] Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. Causal attention for vision-language tasks. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9842–9852, 2021.

[68] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *ArXiv*, abs/2109.11797, 2021.

[69] Andy Zeng, Adrian S. Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael S. Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Peter R. Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *ArXiv*, abs/2204.00598, 2022.

[70] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *ArXiv*, abs/2111.08276, 2021.

[71] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun-Juan Zhu, Lionel Ming shuan Ni, and Heung yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *ArXiv*, abs/2203.03605, 2022.

[72] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *ArXiv*, abs/2206.05836, 2022.

[73] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *ArXiv*, abs/2101.00529, 2021.

[74] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *CVPR 2021*, 2021.

[75] Zhaoyang Zhang, Yantao Shen, Kunyu Shi, Zhaowei Cai, Jun Fang, Siqi Deng, Hao Yang, Davide Modolo, Zhuowen Tu, and Stefano Soatto. Musketeer (all for one, and one for all): A generalist vision-language model with task explanation prompts, 2023. 1

[76] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, 2019. 7

[77] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130:2337 – 2348, 2021. 1

[78] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16795–16804, 2022.

[79] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. *ArXiv*, abs/1909.11059, 2019.

[80] Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Linbo Jin, Ben Chen, Hao Zhou, Minghui Qiu, and Ling Shao. Kaleido-bert: Vision-language pre-training on fashion domain. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12642–12652, 2021. 1