NOUGAT: NEURAL OPTICAL UNDERSTANDING FOR ACADEMIC DOCUMENTS

Lukas Blecher* **Guillem Cucurull** **Thomas Scialom** **Robert Stojnic**

Meta AI

ABSTRACT

Scientific knowledge is predominantly stored in books and scientific journals, often in the form of PDFs. However, the PDF format leads to a loss of semantic information, particularly for mathematical expressions. We propose Nougat (**N**eural **O**ptical **U**nderstandin**g** for **A**cademic Documen**t**s), a Visual Transformer model that performs an *Optical Character Recognition* (OCR) task for processing scientific documents into a markup language, and demonstrate the effectiveness of our model on a new dataset of scientific documents. The proposed approach offers a promising solution to enhance the accessibility of scientific knowledge in the digital age, by bridging the gap between human-readable documents and machine-readable text. We release the models and code to accelerate future work on scientific text recognition.

1 INTRODUCTION

The majority of scientific knowledge is stored in books or published in scientific journals, most commonly in the Portable Document Format (PDF). Next to HTML, PDFs are the second most prominent data format on the internet, making up 2.4% of common crawl (Sebastian Spiegler, 2013). However, the information stored in these files is very difficult to extract into any other formats. This is especially true for highly specialized documents, such as scientific research papers, where the semantic information of mathematical expressions is lost.

Existing Optical Character Recognition (OCR) engines, such as Tesseract OCR (Smith, 2007), excel at detecting and classifying individual characters and words in an image, but fail to understand the relationship between them due to their line-by-line approach. This means that they treat superscripts and subscripts in the same way as the surrounding text, which is a significant drawback for mathematical expressions. In mathematical notations like fractions, exponents, and matrices, relative positions of characters are crucial.

Converting academic research papers into machine-readable text also enables accessibility and searchability of science as a whole. The information of millions of academic papers can not be fully accessed because they are locked behind an unreadable format. Existing corpora, such as the S2ORC dataset (Lo et al., 2020), capture the text of 12M[1] papers using GROBID (Lopez, 2023), but are missing meaningful representations of the mathematical equations.

To this end, we introduce Nougat, a transformer based model that can convert images of document pages to formatted markup text.

The primary contributions in this paper are

- Release of a pre-trained model capable of converting a PDF to a lightweight markup language. We release the code and the model on GitHub[2]
- We introduce a pipeline to create dataset for pairing PDFs to source code
- Our method is only dependent on the image of a page, allowing access to scanned papers and books

[*] Correspondence to: lblecher@meta.com

[1] The paper reports 8.1M papers but the authors recently updated the numbers on the GitHub page https://github.com/allenai/s2orc

[2] https://github.com/facebookresearch/nougat

2 Related Work

Optical Character Recognition (OCR) is an extensively researched field in computer vision for a variety applications, such as document digitalization (Moysset et al., 2017; Smith, 2007), handwriting recognition and scene text recognition (Bautista & Atienza, 2022; Li et al., 2022; Diaz et al., 2021). More concretely, recognizing mathematical expressions is a heavily researched subtopic. Grammar based methods (MacLean & Labahn, 2013; Awal et al., 2014; Álvaro et al., 2014) for handwritten mathematical expressions were improved upon by different encoder-decoder models. The fully convolutional model (Yan et al., 2020) was succeeded by various RNN decoder models (Deng et al., 2016; Le & Nakagawa, 2017; Singh, 2018; Zhang et al., 2018; Wang & Liu, 2019), both for handwritten and printed formulas. Recently, the decoder (Zhao et al., 2021; Mahdavi et al., 2019) as well as the encoder (Blecher, 2023) were replaced with the Transformer (Vaswani et al., 2017) architecture.

Visual Document Understanding (VDU) is another related topic of deep learning research and focuses on extracting relevant information of a variety of document types. Previous works depend on pre-trained models that learn to extract information by jointly modeling text and layout information using the Transformer architecture. The LayoutLM model family (Xu et al., 2020; 2022; Huang et al., 2022) uses masked layout prediction task to capture the spatial relationships between different document elements.

Open source solutions with a related goal as ours include GROBID (Lopez, 2023), which parses digital-born scientific documents to XML with a focus on the bibliographic data and `pdf2htmlEX` (Lu Wang & Wanmin Liu, 2013), that converts digital-born PDFs to HTML while preserving the layout and appearance of the document. However, both solutions can not recover the semantic information of mathematical expressions.

Previous VDU methods either rely on OCR text from a third party tool (Xu et al., 2020; 2022; Appalaraju et al., 2021) or focus on document types such as receipts, invoices or form-like documents (Majumder et al., 2020). Recent studies (Kim et al., 2022; Davis et al., 2022) show that an external OCR engine is not necessarily needed to achieve competitive results in VDU.

3 Model

The architecture is a encoder-decoder transformer (Vaswani et al., 2017) architecture, that allows for an end-to-end training procedure. We build on the Donut (Kim et al., 2022) architecture. The model does not require any OCR related inputs or modules. The text is recognized implicitly by the network. See Fig. 1 for an overview of the approach.

Encoder The visual encoder receives a document image $\mathbf{x} \in \mathbb{R}^{3\times H_0\times W_0}$, crops the margins and resizes the image to fit in a fixed rectangle of size $(H,\, W)$. If the image is smaller than the rectangle, additional padding is added to ensure each image has the same dimensionality. We use a Swin Transformer (Liu et al., 2021), a hierarchical vision transformer (Dosovitskiy et al., 2021) that splits the image into non-overlapping windows of fixed size and applies a series of self-attention layers to aggregate information across these windows. The model output a sequence of the embedded patches $\mathbf{z} \in \mathbb{R}^{d\times N}$ where d is the latent dimension and N is the number of patches.

Decoder The encoded image $\mathbf{z}$ is decoded into a sequence of tokens using a transformer decoder architecture with cross-attention. The tokens are generated in an auto-regressive manner, using self-attention and cross-attention to attend to different parts of the input sequence and encoder output respectively. Finally, the output is projected to the size of the vocabulary v, yielding the logits $\boldsymbol{\ell} \in \mathbb{R}^v$. Following Kim et al. (2022), we use the implementation of the mBART (Lewis et al., 2019) decoder. We use the same tokenizer as Taylor et al. (2022) because their model is also specialized in the scientific text domain.

3.1 Setup

We render the document images at a resolution of 96 DPI. Due to the restrictive possible input dimensions of the Swin Transformer we choose the input size $(H,\, W) = (896,\, 672)$. The aspect ratio is in between the US letter and Din A4 format $\frac{22}{17} < \frac{4}{3} < \sqrt{2}$. The document images are resized

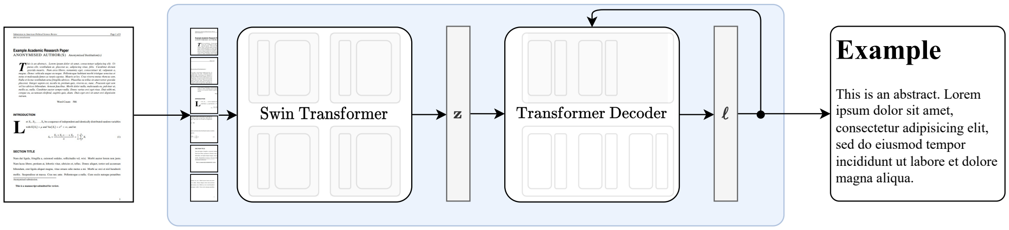

Swin Transformer
z
Transformer Decoder
ℓ
Example
This is an abstract. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Figure 1: Our simple end-to-end architecture following Donut (Kim et al., 2022). The Swin Transformer encoder takes a document image and converts it into latent embeddings, which are subsequently converted to a sequence of tokens in a auto-regressive manner

and then padded to achieve the desired input size. This input size allows us to use the Swin base model architecture (Liu et al., 2021). We initialize the model with the pre-trained weights.
The Transformer decoder has a maximal sequence length of $S = 4096$. This relatively large sizing is due to the fact that the text of academic research papers can be dense and the syntax for tables in particular is token intensive. The BART decoder is a decoder-only transformer with 10 layers. The entire architecture has a total of 350M parameters.
We also test experiment with a smaller model (250M parameters) with a slightly smaller sequence length of $S = 3584$ and only 4 decoder layers, where we start from the pre-trained base model.
During inference the text is generated using greedy decoding.

Training We use an AdamW optimizer (Loshchilov & Hutter, 2019) to train for 3 epochs with an effective batch size of 192. Due to training instabilities, we choose a learning rate of $\mathrm{lr}_{\mathrm{init}} = 5 \cdot 10^{-5}$ which is reduced by a factor of 0.9996 every 15 updates until it reaches $\mathrm{lr}_{\mathrm{end}} = 7.5 \cdot 10^{-6}$.

3.2 Data Augmentation

In image recognition tasks, it is often beneficial to use data augmentation to improve generalization. Since we are only using digital-born academic research papers, we need to employ a number of transformations to simulate the imperfections and variability of scanned documents. These transformations include erosion, dilation, gaussian noise, gaussian blur, bitmap conversion, image compression, grid distortion and elastic transform (Simard et al., 2003). Each has a fixed probability of being applied to a given image. The transformations are implemented in the *Albumentations* (Buslaev et al., 2020) library. For an overview of the effect of each transformation, see Fig. A.1.
During training time, we also add perturbations to the ground truth text by randomly replacing tokens. We found this to reduce the collapse into a repeating loop significantly. For more details, see Section 5.4.

4 Datasets

To the best of our knowledge there is no paired dataset of PDF pages and corresponding source code out there, so we created our own from the open access articles on arXiv.[3] For layout diversity we also include a subset of the *PubMed Central* [4] (PMC) open access non-commercial dataset. During the pretraining, a portion of the *Industry Documents Library* [5] (IDL) is included. See Table A.1 for the dataset composition.

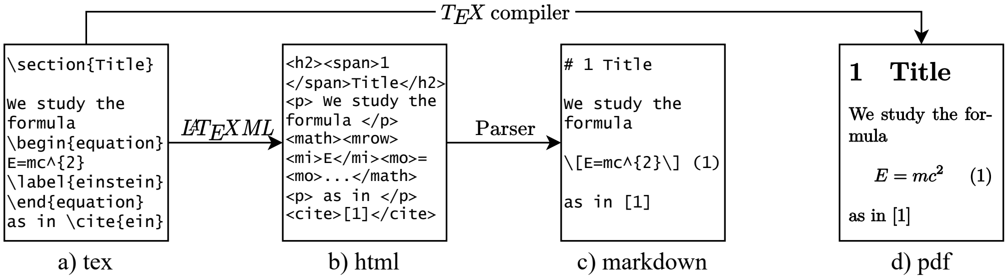
arXiv We collected the source code and compiled PDFs from 1,748,201 articles released on arXiv. To ensure consistent formatting, we first process the source files using *LaTeXML*[6] and convert them into HTML5 files. This step was important as it standardized and removed ambiguity from the LaTeX source code, especially in mathematical expressions. The conversion process included replacing user-defined macros, standardizing whitespace, adding optional brackets, normalizing tables, and replacing references and citations with their correct numbers.

[3] https://arxiv.org/

[4] https://www.ncbi.nlm.nih.gov/pmc/

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

[6] http://dlmf.nist.gov/LaTeXML/



a) tex

```
\section{Title}

We study the
formula
\begin{equation}
E=mc^{2}
\label{einstein}
\end{equation}
as in \cite{ein}
```

b) html

```
<h2><span>1
</span>Title</h2>
<p> We study the
formula </p>
<math><mrow>
<mi>E</mi><mo>=
<mo>...</math>
<p> as in </p>
<cite>[1]</cite>
```

c) markdown

```
# 1 Title

We study the
formula

\[E=mc^{2}\] (1)

as in [1]
```

d) pdf

1 Title

We study the formula

$$E = mc^2 \quad (1)$$

as in [1]

Figure 2: Data processing. The source file is converted into HTML which is then converted to Markdown. a) The LaTeX source provided by the authors. b) The HTML file computed form the LaTeX source using LaTeXML. c) The Markdown file parsed from the HTML file. d) The PDF file provided by the authors

We then parse the HTML files and convert them into a lightweight markup language that supports various elements such as headings, bold and italic text, algorithms, LaTeX inline and display math and LaTeX tables. This way, we ensure that the source code is properly formatted and ready for further processing.
The process is visualized in Fig. 2.

PMC We also processed articles from PMC, where XML files with semantic information are available in addition to the PDF file. We parse these files into the same markup language format as the arXiv articles. We chose to use far fewer articles from PMC because the XML files are not always as rich in semantic information. Often times equations and tables are stored as images and these cases are not trivial to detect, which leads to our decision to limit the use of PMC articles to the pre-training phase.

The XML files are parsed into the same markup language as described above.

IDL The IDL is a collection of documents produced by industries that have an impact on public health and is maintained by the University of California, San Francisco Library. Biten et al. (2022) provide high quality OCR text for PDFs from the IDL dataset. This does not include text formatting and is only used for pre-training to teach the model basic OCR of scanned documents.

4.1 Splitting the pages

We split the markdown files according to the page breaks in the PDF file and rasterize each page as an image to create the final paired dataset. During the compilation, the LaTeX compiler determines the page breaks of the PDF file automatically. Since we are not recompiling the LaTeX sources for each paper, we must heuristically split the source file into parts, which correspond to different pages. To achieve that we are using the embedded text on the PDF page and match it to source text.
However, figures and tables in the PDF may not correspond to their position in the source code. To address this issue, we remove these elements in a pre-processing step using `pdffigures2` (Clark & Divvala, 2016). The recognized captions are are then compared to the captions in the XML file and matched based on their Levenshtein distance (Levenshtein, 1965). Once the source document has been split into individual pages, the removed figures and tables are reinserted at the end of each page. For a better matching we also replaced unicode characters in the PDF text with corresponding LaTeX commands using the pylatexenc-library[7].

Bag of Words matching First we extract the text lines from the PDF using MuPDF[8] and preprocess them to remove page numbers and potential headers/footers. We then use a *Bag of Words* model (Harris, 1954) with TF-IDF vectorizer and a linear Support Vector Machine classifier. The model is fitted to the PDF lines with the page number as label. Next we split the LaTeX source into paragraphs and predict the page number for each of them.

[7] https://github.com/phfaist/pylatexenc

[8] https://mupdf.com/

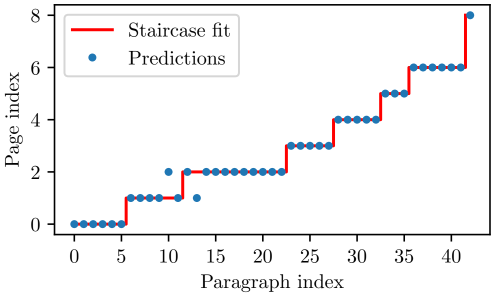


Figure 3: Example for splitting the paragraphs in the source code into different pages. The points in blue denote the page index predicted by the SVM.

Ideally, the predictions will form a stair case function but in practice the signal will be noisy. To find the best boundary points we employ a similar logic as decision trees and minimize a measure based on the *Gini* impurity

$$G_{[a,b]}(i) = (b - a) \cdot \left(1 - p^2_{[a,b]}(i) - p^2_{[a,b]}(i+1)\right), \tag{1}$$

where $p_{[a,b]}(i)$ is the probability of choosing an element with the predicted page number i in the interval of paragraph indices $[a,\ b]$ that describes which paragraphs (elements) were considered for the split.
The best splitting position t in the interval $[a,\ b]$ is then

$$\hat{t}_i = \arg\min_t \left(G_{[a,t]}(i) + G_{[t,b]}(i)\right). \tag{2}$$

The search process starts with all paragraphs and for each subsequent page break, the lower bound of the search interval is set to the previous split position. See Fig. 3 for a visualization of an example page.

Fuzzy matching After this first coarse document splitting we try to find the exact position within the paragraph. This is done by comparing the source text within the neighborhood of the predicted splitting position to the last sentences of the previous page of the embedded PDF text, and the first sentences of the next page using the `fuzzysearch` library[9]. If the two dividing points are at the same location in the source text, the page break is considered "accurate" and receives a score of 1. On the other hand, if the splitting positions differ, the one with the smallest normalized Levenshtein distance is selected and given a score of 1 minus the distance. To be included in the dataset, a PDF page must have an average score of at least 0.9 for both page breaks. This results in an acceptance rate of about 47% of all pages.

4.2 Ground truth artifacts

Because the dataset was pre-processed by LaTeXML, the markup version of the source code can contain artifacts and commands from unsupported packages. The HTML file may contain subsection titles with numbering even though they are not numbered in the PDF. There may also be instances where figures or tables are missing from the ground truth due to processing errors.

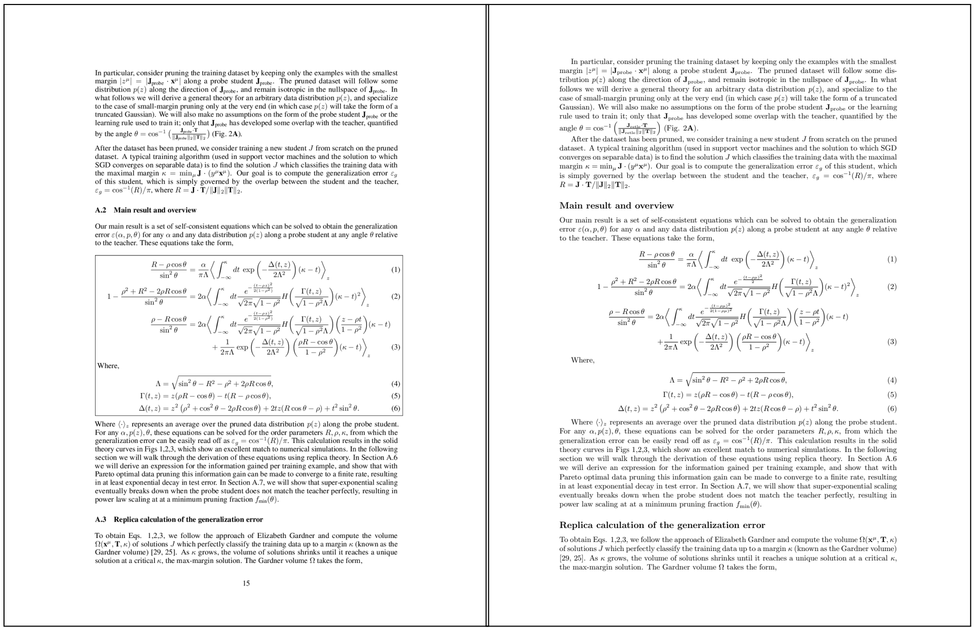
In addition, the splitting algorithm of the source code will in some cases include text from the previous page or cut off words from the end. This is especially true for "invisible" characters used for formatting, like italic, bold text or section header.

For PMC papers the inline math is written as Unicode or italic text, while display math equations or tables are often included in image format and will therefore be ignored.

Each of these issues reduces the overall data quality. However, the large number of training samples compensates for these small errors.

[9] `https://github.com/taleinat/fuzzysearch`

5 Results & Evaluation



In particular, consider pruning the training dataset by keeping only the examples with the smallest margin $|z^\mu| = |\mathbf{J}_{\text{probe}} \cdot \mathbf{x}^\mu|$ along a probe student $\mathbf{J}_{\text{probe}}$. The pruned dataset will follow some distribution $p(z)$ along the direction of $\mathbf{J}_{\text{probe}}$, and remain isotropic in the nullspace of $\mathbf{J}_{\text{probe}}$. In what follows we will derive a general theory for an arbitrary data distribution $p(z)$, and specialize to the case of small-margin pruning only at the very end (in which case $p(z)$ will take the form of a truncated Gaussian). We will also make no assumptions on the form of the probe student $\mathbf{J}_{\text{probe}}$ or the learning rule used to train it; only that $\mathbf{J}_{\text{probe}}$ has developed some overlap with the teacher, quantified by the angle $\theta = \cos^{-1}\left(\frac{\mathbf{J}_{\text{probe}} \cdot \mathbf{T}}{\|\mathbf{J}_{\text{probe}}\|_2 \|\mathbf{T}\|_2}\right)$ (Fig. 2**A**).

After the dataset has been pruned, we consider training a new student J from scratch on the pruned dataset. A typical training algorithm (used in support vector machines and the solution to which SGD converges on separable data) is to find the solution J which classifies the training data with the maximal margin $\kappa = \min_\mu \mathbf{J} \cdot (y^\mu \mathbf{x}^\mu)$. Our goal is to compute the generalization error ε_g of this student, which is simply governed by the overlap between the student and the teacher, $\varepsilon_g = \cos^{-1}(R)/\pi$, where $R = \mathbf{J} \cdot \mathbf{T}/\|\mathbf{J}\|_2 \|\mathbf{T}\|_2$.

A.2 Main result and overview

Our main result is a set of self-consistent equations which can be solved to obtain the generalization error $\varepsilon(\alpha, p, \theta)$ for any α and any data distribution $p(z)$ along a probe student at any angle θ relative to the teacher. These equations take the form,

$$\frac{R - \rho\cos\theta}{\sin^2\theta} = \frac{\alpha}{\pi\Lambda}\left\langle \int_{-\infty}^{\kappa} dt\ \exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)(\kappa - t)\right\rangle_z \tag{1}$$

$$1 - \frac{\rho^2 + R^2 - 2\rho R\cos\theta}{\sin^2\theta} = 2\alpha\left\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)(\kappa - t)^2\right\rangle_z \tag{2}$$

$$\begin{aligned}\frac{\rho - R\cos\theta}{\sin^2\theta} = 2\alpha\Bigg\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho z)^2}{2(1-\rho^2)}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)\left(\frac{z - \rho t}{1-\rho^2}\right)(\kappa - t) \\ + \frac{1}{2\pi\Lambda}\exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)\left(\frac{\rho R - \cos\theta}{1-\rho^2}\right)(\kappa - t)\Bigg\rangle_z\end{aligned} \tag{3}$$

Where,

$$\Lambda = \sqrt{\sin^2\theta - R^2 - \rho^2 + 2\rho R\cos\theta}, \tag{4}$$

$$\Gamma(t,z) = z(\rho R - \cos\theta) - t(R - \rho\cos\theta), \tag{5}$$

$$\Delta(t,z) = z^2\left(\rho^2 + \cos^2\theta - 2\rho R\cos\theta\right) + 2tz(R\cos\theta - \rho) + t^2\sin^2\theta. \tag{6}$$

Where $\langle\cdot\rangle_z$ represents an average over the pruned data distribution $p(z)$ along the probe student. For any $\alpha, p(z), \theta$, these equations can be solved for the order parameters R, ρ, κ, from which the generalization error can be easily read off as $\varepsilon_g = \cos^{-1}(R)/\pi$. This calculation results in the solid theory curves in Figs 1,2,3, which show an excellent match to numerical simulations. In the following section we will walk through the derivation of these equations using replica theory. In Section A.6 we will derive an expression for the information gained per training example, and show that with Pareto optimal data pruning this information gain can be made to converge to a finite rate, resulting in at least exponential decay in test error. In Section A.7, we will show that super-exponential scaling eventually breaks down when the probe student does not match the teacher perfectly, resulting in power law scaling at at a minimum pruning fraction $f_{\text{min}}(\theta)$.

A.3 Replica calculation of the generalization error

To obtain Eqs. 1,2,3, we follow the approach of Elizabeth Gardner and compute the volume $\Omega(\mathbf{x}^\mu, \mathbf{T}, \kappa)$ of solutions J which perfectly classify the training data up to a margin κ (known as the Gardner volume) [29, 25]. As κ grows, the volume of solutions shrinks until it reaches a unique solution at a critical κ, the max-margin solution. The Gardner volume Ω takes the form,

15

In particular, consider pruning the training dataset by keeping only the examples with the smallest margin $|z^\mu| = |\mathbf{J}_{\text{probe}} \cdot \mathbf{x}^\mu|$ along a probe student $\mathbf{J}_{\text{probe}}$. The pruned dataset will follow some distribution $p(z)$ along the direction of $\mathbf{J}_{\text{probe}}$, and remain isotropic in the nullspace of $\mathbf{J}_{\text{probe}}$. In what follows we will derive a general theory for an arbitrary data distribution $p(z)$, and specialize to the case of small-margin pruning only at the very end (in which case $p(z)$ will take the form of a truncated Gaussian). We will also make no assumptions on the form of the probe student $\mathbf{J}_{\text{probe}}$ or the learning rule used to train it; only that $\mathbf{J}_{\text{probe}}$ has developed some overlap with the teacher, quantified by the angle $\theta = \cos^{-1}\left(\frac{\mathbf{J}_{\text{retle}} \cdot \mathbf{T}}{\|\mathbf{J}_{\text{retle}}\|_2 \|\mathbf{T}\|_2}\right)$ (Fig. 2**A**).

After the dataset has been pruned, we consider training a new student J from scratch on the pruned dataset. A typical training algorithm (used in support vector machines and the solution to which SGD converges on separable data) is to find the solution J which classifies the training data with the maximal margin $\kappa = \min_\mu \mathbf{J} \cdot (y^\mu \mathbf{x}^\mu)$. Our goal is to compute the generalization error ε_g of this student, which is simply governed by the overlap between the student and the teacher, $\varepsilon_g = \cos^{-1}(R)/\pi$, where $R = \mathbf{J} \cdot \mathbf{T}/\|\mathbf{J}\|_2 \|\mathbf{T}\|_2$.

Main result and overview

Our main result is a set of self-consistent equations which can be solved to obtain the generalization error $\varepsilon(\alpha, p, \theta)$ for any α and any data distribution $p(z)$ along a probe student at any angle θ relative to the teacher. These equations take the form,

$$\frac{R - \rho\cos\theta}{\sin^2\theta} = \frac{\alpha}{\pi\Lambda}\left\langle \int_{-\infty}^{\kappa} dt\ \exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)(\kappa - t)\right\rangle_z \tag{1}$$

$$1 - \frac{\rho^2 + R^2 - 2\rho R\cos\theta}{\sin^2\theta} = 2\alpha\left\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho\mu)^2}{2}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)(\kappa - t)^2\right\rangle_z \tag{2}$$

$$\begin{aligned}\frac{\rho - R\cos\theta}{\sin^2\theta} = 2\alpha\Bigg\langle \int_{-\infty}^{\kappa} dt \frac{e^{-\frac{(t-\rho\mu)^2}{2(1-\rho\mu)^2}}}{\sqrt{2\pi}\sqrt{1-\rho^2}} H\left(\frac{\Gamma(t,z)}{\sqrt{1-\rho^2}\Lambda}\right)\left(\frac{z - \rho t}{1-\rho^2}\right)(\kappa - t) \\ + \frac{1}{2\pi\Lambda}\exp\left(-\frac{\Delta(t,z)}{2\Lambda^2}\right)\left(\frac{\rho R - \cos\theta}{1-\rho^2}\right)(\kappa - t)\Bigg\rangle_z\end{aligned} \tag{3}$$

Where,

$$\Lambda = \sqrt{\sin^2\theta - R^2 - \rho^2 + 2\rho R\cos\theta}, \tag{4}$$

$$\Gamma(t,z) = z(\rho R - \cos\theta) - t(R - \rho\cos\theta), \tag{5}$$

$$\Delta(t,z) = z^2\left(\rho^2 + \cos^2\theta - 2\rho R\cos\theta\right) + 2tz(R\cos\theta - \rho) + t^2\sin^2\theta. \tag{6}$$

Where $\langle\cdot\rangle_z$ represents an average over the pruned data distribution $p(z)$ along the probe student. For any $\alpha, p(z), \theta$, these equations can be solved for the order parameters R, ρ, κ, from which the generalization error can be easily read off as $\varepsilon_g = \cos^{-1}(R)/\pi$. This calculation results in the solid theory curves in Figs 1,2,3, which show an excellent match to numerical simulations. In the following section we will walk through the derivation of these equations using replica theory. In Section A.6 we will derive an expression for the information gained per training example, and show that with Pareto optimal data pruning this information gain can be made to converge to a finite rate, resulting in at least exponential decay in test error. In Section A.7, we will show that super-exponential scaling eventually breaks down when the probe student does not match the teacher perfectly, resulting in power law scaling at at a minimum pruning fraction $f_{\text{min}}(\theta)$.

Replica calculation of the generalization error

To obtain Eqs. 1,2,3, we follow the approach of Elizabeth Gardner and compute the volume $\Omega(\mathbf{x}^\mu, \mathbf{T}, \kappa)$ of solutions J which perfectly classify the training data up to a margin κ (known as the Gardner volume) [29, 25]. As κ grows, the volume of solutions shrinks until it reaches a unique solution at a critical κ, the max-margin solution. The Gardner volume Ω takes the form,

Figure 4: Example of a page with many mathematical equations taken from (Sorscher et al., 2022). Left: Image of a page in the document, Right: Model output converted to LaTeX and rendered to back into a PDF. Examples of scanned documents can be found in the appendix B.

In this section we discuss the results and performance of the model. For an example see Fig. 4 or go to Sec. B. The model focuses only on the important content relevant features of the page. The box around the equations is skipped.

5.1 Metrics

We report the following metrics on our test set.

Character Error Rate The character error rate (CER), or normalized Levenshtein distance (Levenshtein, 1965), measures the number of character manipulations (insertions, deletions, substitutions) it takes to get from one string to another.

BLEU The BLEU (Papineni et al., 2002) metric was originally introduced for measuring the quality of text that has been machine-translated from one language to another. The metric computes a score based on the number of matching n-grams between the candidate and reference sentence.

METEOR Another machine-translating metric with a focus on recall instead of precision, introduced in (Banerjee & Lavie, 2005).

F-measure We also compute the F1-score and report the precision and recall.

5.2 Text modalities

In a scientific research article, there are three distinct types of text: 1) plain text, which comprises the majority of the document, 2) mathematical expressions, and 3) tables. It is important to separately examine each of these components during the evaluation process. This is necessary because in LaTeX, there are multiple ways to express the same mathematical expression. While some variability has been eliminated during the LaTeXML pre-processing step, there still is a significant amount of ambiguity present, like ordering of subscript and superscript, equivalent commands with different notation (`stackrel`, `atop`, `substack` or `frac`, `over`), situationally interchangeable commands (`bm`, `mathbf`, `boldsymbol`, `bf` or `\left(`, `\big(`, etc.), whitespace commands, additional layers of brackets, and more. As a consequence, there can be a discrepancy between prediction and ground

Method	Modality	CER	BLEU	METEOR	Prec.	Rec.	F1
PDF	All	25.5	65.8	82.1	77.1	81.4	79.2
GROBID	All	31.2	55.6	71.9	74.0	72.1	73.0
	Tables	62.6	25.1	64.5	61.4	80.7	69.7
+ LaTeX OCR (Blecher)	Plain text	36.3	57.4	69.2	82.1	70.5	75.9
	Math	72.7	0.3	5.0	11.0	8.6	9.7
Nougat small (250M*)	All	7.3	88.9	92.8	**93.6**	92.2	92.9
	Tables	22.0	68.5	78.6	75.0	79.8	77.3
	Plain text	5.8	91.0	94.3	96.1	95.3	95.7
	Math	11.7	56.0	74.7	77.1	76.8	76.9
Nougat base (350M*)	All	**7.1**	**89.1**	**93.0**	93.5	**92.8**	**93.1**
	Tables	21.1	69.7	79.1	75.4	80.7	78.0
	Plain text	5.8	91.2	94.6	96.2	95.3	95.7
	Math	12.8	56.9	75.4	76.5	76.6	76.5

Table 1: Results on arXiv test set. PDF is the text embedded in the PDF file. The modality "All" refers to the output text without any splitting. All metrics are reported in percent. *Number of parameters.

truth, even if the rendered formulas appear identical.
In addition, it is not always possible to determine, where an inline math environment ends and text begins, when writing numbers and punctuation (Example: `$\mathrm{H}_{0}$1,` vs. `H$_{0}1,$` → $\mathrm{H}_0$1, vs. $\mathrm{H}_0$1,). This ambiguity reduces both math and plain text scores.
The expected score for mathematical expressions is lower than for plain text.

5.3 Comparison

We present our results in Table 1. As expected, the mathematical expressions have the worst agreement with the ground truth. For the plain text, most discrepancies come from formatting ambiguities and missing text due to inline math, as described above. The output format of GROBID is an XML file, which we convert into a compatible markup language, similar to the PMC or arXiv files. To some extent, GROBID provides support for formulas in its output, but it identifies and stores them as the Unicode representations embedded in the PDF. We replace each Unicode symbol with its corresponding LaTeX command to increase the similarity. Additionally, GROBID mislabels small inline expressions as text. For identified formulas, GROBID stores the bounding box coordinates. We modify the program by sending the snippet to the external formula recognition software LaTeX-OCR (Blecher, 2023). This way we can also get a signal for math modality. The reported results in this section are quite poor, primarily due to the amount of missed formulas by GROBID and the equation prediction accuracy is affected by the quality of the bounding boxes. The performance of the embedded PDF text alone is better than GROBID, which is due to formatting differences for the title page or reference section.
Both Nougat small and base are able to outperform the other approach and achieve high scores in all metrics. We note that the performance of the smaller model is on par with the larger base model.

5.4 Repetitions during inference

We notice that the model degenerates into repeating the same sentence continuously, a state from which it cannot autonomously recover. In its simplest form, the last sentence or paragraph is repeated ad infinitum. We observed this behavior in 1.5% of pages in the test set, but the frequency increases for out-of-domain documents. Our evaluation extends to a diverse selection of academic documents, encompassing a range of sources such as scanned books. See Appendix B for examples.

Getting stuck in a repetitive loop is a known problem with Transformer-based models, when sampled with greedy decoding (Holtzman et al., 2020). We conducted experiments with nucleus sampling,

but encountered a susceptibility to repetitive outputs. Consequently, we opted against utilizing this technique due to the introduced randomness failing to effectively mitigate the issue of unwanted sentence repetitions.
It can also happen that the model alternates between two sentences but sometimes changes some words, so a strict repetition detection will not suffice. Even harder to detect are predictions where the model counts its own repetitions, which sometimes happens in the references section.
In general we notice this kind behavior after a mistake by the model. The model is not able to recover from the collapse.

Anti-repetition augmentation Because of that we introduce a random perturbation during training. This helps the model to learn how to handle a wrongly predicted token. For each training example, there is a fixed probability that a random token will be replaced by any other randomly chosen token. This process continues until the newly sampled number is greater than a specified threshold (in this case, 10%). We did not observe a decrease in performance with this approach, but we did notice a significant reduction in repetitions. Particularly for out-of-domain documents, where we saw a 32% decline in failed page conversions.

Repetition detection Since we are generating a maximum of 4096 tokens, the model will stop at some point. However, it is very inefficient and resource-intensive to wait for an ”end of sentence” token that will not be generated. To detect repetition during inference, we examine the largest logit value $\ell_i = \max \boldsymbol{\ell}_i$ for the ith token. We observed that logits after a collapse can be separated using the following heuristic consisting of two steps. First, calculate the variance of the logits for a sliding window of size $B = 15$

$$\mathrm{VarWin}_B[\boldsymbol{\ell}](x) = \frac{1}{B}\sum_{i=x}^{x+B}\left(\ell_i - \frac{1}{B}\sum_{j=x}^{x+B}\ell_j\right)^2. \tag{3}$$

Here, ℓ is the signal of logits, and x is the index. Using this new signal, compute variances again, but this time from point x to the end of the sequence:

$$\mathrm{VarEnd}_B[\boldsymbol{\ell}](x) = \frac{1}{S-x}\sum_{i=x}^{S}\left(\mathrm{VarWin}_B[\boldsymbol{\ell}](i) - \frac{1}{S-x}\sum_{j=x}^{S}\mathrm{VarWin}_B[\boldsymbol{\ell}](i)\right)^2. \tag{4}$$

If this signal drops below a certain threshold (we choose 6.75) and stays below for the remainder of the sequence, we classify the sequence as having repetitions.

During inference time, it is not possible to compute the entire sequence if our goal is to stop generation earlier. Here, we work with a subset of the last 200 tokens and half the threshold. After the generation is finished, the procedure as described above is repeated for the full sequence.

5.5 Limitations & Future work

Utility The utility of the model is limited by a number of factors. First, the problem with repetitions outlined in section 5.4. The model is trained on research papers, which means it works particularly well on documents with a similar structure. However, it can still accurately convert other types of documents.
Nearly every dataset sample is in English. Initial tests on a small sample suggest that the model's performance with other Latin-based languages is satisfactory, although any special characters from these languages will be replaced with the closest equivalent from the Latin alphabet. Non-Latin script languages result in instant repetitions.
Generation Speed On a machine with a NVIDIA A10G graphics card with 24GB VRAM we can process 6 pages in parallel. The generation speed depends heavily on the amount of text on any given page. With an average number of tokens of ≈ 1400 we get an mean generation time of 19.5s per batch for the base model without any inference optimization. Compared to classical approaches (GROBID 10.6 PDF/s (Lopez, 2023)) this is very slow, but it is not limited to digital-born PDFs and can correctly parse mathematical expressions.
Future work The model is trained on one page at a time without knowledge about other pages in the document. This results in inconsistencies across the document. Most notably in the bibliography where the model was trained on different styles or section titles where sometimes numbers are skipped

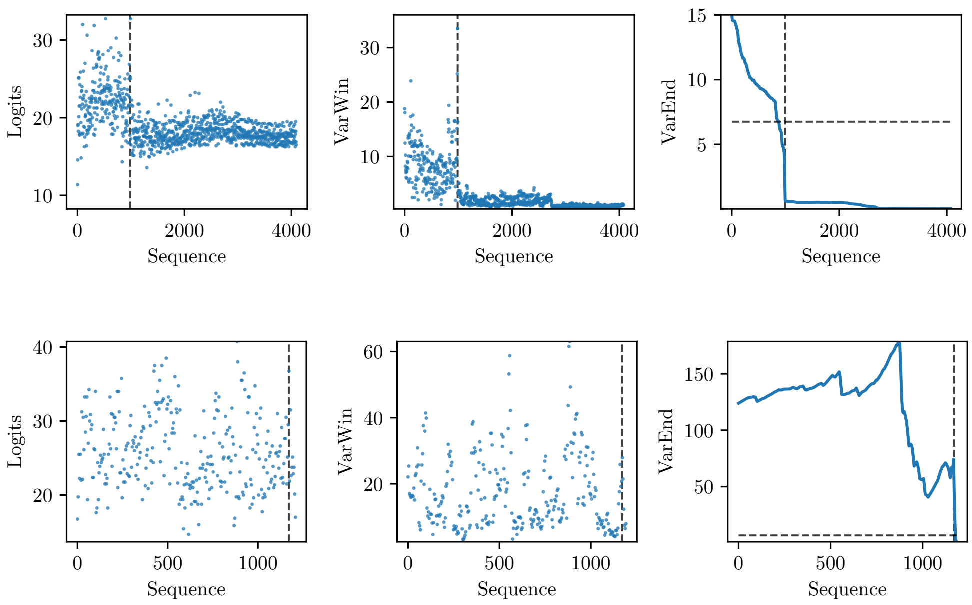


Figure 5: Examples for repetition detection on logits. The vertical index denotes the sequence index, where repetitions start. Top: Sample generation with repetition, Bottom: Sample generation without repetition. Left: Highest logit score for each token in the sequence $\ell(x)$, Center: Sliding window variance of the logits $\mathrm{VarWin}_B[\ell](x)$, Right: Variance of the window variance (3) from the position to the end $\mathrm{VarEnd}_B[\ell](x)$ with threshold as horizontal dashed line.

or hallucinated. Though handling each page separately significantly improves parallelization and scalability, it may diminish the quality of the merged document text.
The primary challenge to solve is the tendency for the model to collapse into a repeating loop, which is left for future work.

6 Conclusion

In this work, we present Nougat, an end-to-end trainable encoder-decoder transformer based model for converting document pages to markup. We apply recent advances in visual document understanding to a novel OCR task. Distinct from related approaches, our method does not rely on OCR or embedded text representations, instead relying solely on the rasterized document page. Moreover, we have illustrated an automatic and unsupervised dataset generation process that we used to successfully train the model for scientific document to markup conversion. Overall, our approach has shown great potential for not only extracting text from digital-born PDFs but also for converting scanned papers and textbooks. We hope this work can be a starting point for future research in related domains.
All the code for model evaluation, training and dataset generation can be accessed at `https://github.com/facebookresearch/nougat`.

Acknowledgments

Thanks to Ross Taylor, Marcin Kardas, Iliyan Zarov, Kevin Stone, Jian Xiang Kuan, Andrew Poulton and Hugo Touvron for their valuable discussions and feedback.
Thanks to Faisal Azhar for the support throughout the project.

REFERENCES

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. DocFormer: End-to-End Transformer for Document Understanding, September 2021. URL http://arxiv.org/abs/2106.11539. arXiv:2106.11539 [cs].

Ahmad-Montaser Awal, Harold Mouchre, and Christian Viard-Gaudin. A global learning approach for an online handwritten mathematical expression recognition system. *Pattern Recognition Letters*, 35(C):68–77, January 2014. ISSN 0167-8655.

Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909.

Darwin Bautista and Rowel Atienza. Scene Text Recognition with Permuted Autoregressive Sequence Models, July 2022. URL http://arxiv.org/abs/2207.06966. arXiv:2207.06966 [cs] version: 1.

Ali Furkan Biten, Rubèn Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. OCR-IDL: OCR Annotations for Industry Document Library Dataset, February 2022. URL http://arxiv.org/abs/2202.12985. arXiv:2202.12985 [cs].

Lukas Blecher. pix2tex - LaTeX OCR, February 2023. URL https://github.com/lukas-blecher/LaTeX-OCR. original-date: 2020-12-11T16:35:13Z.

Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and Flexible Image Augmentations. *Information*, 11(2):125, February 2020. ISSN 2078-2489. doi: 10.3390/info11020125. URL https://www.mdpi.com/2078-2489/11/2/125.

Christopher Clark and Santosh Divvala. PDFFigures 2.0: Mining Figures from Research Papers. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, pp. 143–152, Newark New Jersey USA, June 2016. ACM. ISBN 978-1-4503-4229-2. doi: 10.1145/2910896.2910904. URL https://dl.acm.org/doi/10.1145/2910896.2910904.

Brian Davis, Bryan Morse, Bryan Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. End-to-end Document Recognition and Understanding with Dessurt, June 2022. URL http://arxiv.org/abs/2203.16618. arXiv:2203.16618 [cs].

Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M. Rush. Image-to-Markup Generation with Coarse-to-Fine Attention, September 2016. URL http://arxiv.org/abs/1609.04938. arXiv:1609.04938 [cs] version: 1.

Daniel Hernandez Diaz, Siyang Qin, Reeve Ingle, Yasuhisa Fujii, and Alessandro Bissacco. Rethinking Text Line Recognition Models, April 2021. URL http://arxiv.org/abs/2104.07787. arXiv:2104.07787 [cs].

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. URL http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs].

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082.

Sanford Gordon, Sheldon Heimel, Frank Zeleznik, Richard Brokaw, Charles Baker, Roger Svehla, Frank Belles, Theodore Brabbs, Marvin Warshay, and David Bittker. Kinetics and Thermodynamics in High-Temperature Gases, January 1970. URL https://ntrs.nasa.gov/citations/19700022795. NTRS Report/Patent Number: N70-32106-116 NTRS Document ID: 19700022795 NTRS Research Center: Glenn Research Center (GRC).

Zellig S. Harris. Distributional Structure. *WORD*, 10(2-3):146–162, 1954. doi: 10.1080/00437956.1954.11659520. URL https://doi.org/10.1080/00437956.1954.11659520. Publisher: Routledge _eprint: https://doi.org/10.1080/00437956.1954.11659520.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration, February 2020. URL http://arxiv.org/abs/1904.09751. arXiv:1904.09751 [cs].

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking, July 2022. URL http://arxiv.org/abs/2204.08387. arXiv:2204.08387 [cs].

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, October 2022. URL http://arxiv.org/abs/2111.15664. arXiv:2111.15664 [cs].

Anh Duc Le and Masaki Nakagawa. Training an End-to-End System for Handwritten Mathematical Expression Recognition by Generated Patterns. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pp. 1056–1061, November 2017. doi: 10.1109/ICDAR.2017.175. ISSN: 2379-2140.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710, 1965. URL https://api.semanticscholar.org/CorpusID:60827152.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, October 2019. URL http://arxiv.org/abs/1910.13461. arXiv:1910.13461 [cs, stat].

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models, September 2022. URL http://arxiv.org/abs/2109.10282. arXiv:2109.10282 [cs].

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, August 2021. URL http://arxiv.org/abs/2103.14030. arXiv:2103.14030 [cs].

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4969–4983, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.447. URL https://aclanthology.org/2020.acl-main.447.

Patrice Lopez. GROBID, February 2023. URL https://github.com/kermitt2/grobid. original-date: 2012-09-13T15:48:54Z.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs, math] version: 3.

Lu Wang and Wanmin Liu. Online publishing via pdf2htmlEX, 2013. URL https://www.tug.org/TUGboat/tb34-3/tb108wang.pdf.

Scott MacLean and George Labahn. A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(2):139–163, June 2013. ISSN 1433-2825. doi: 10.1007/s10032-012-0184-x. URL https://doi.org/10.1007/s10032-012-0184-x.

Mahshad Mahdavi, Richard Zanibbi, Harold Mouchere, Christian Viard-Gaudin, and Utpal Garain. ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1533–1538, Sydney, Australia, September 2019. IEEE. ISBN 978-1-72813-014-9. doi: 10.1109/ICDAR.2019.00247. URL https://ieeexplore.ieee.org/document/8978036/.

Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. Representation Learning for Information Extraction from Form-like Documents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6495–6504, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.580. URL https://aclanthology.org/2020.acl-main.580.

Herman W. (Herman William) March and Henry C. (Henry Charles) Wolff. *Calculus*. New York : McGraw-Hill, 1917. URL http://archive.org/details/calculus00marciala.

Bastien Moysset, Christopher Kermorvant, and Christian Wolf. Full-Page Text Recognition: Learning Where to Start and When to Stop, April 2017. URL http://arxiv.org/abs/1704.08628. arXiv:1704.08628 [cs].

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Sebastian Spiegler. Statistics of the Common Crawl Corpus 2012, June 2013. URL https://docs.google.com/file/d/1_9698uglerxB9nAglvaHkEgU-iZNm1TvVGuCW7245-WGvZq47teNpb_uL5N9.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-Consistency for Robust Visual Question Answering, February 2019. URL http://arxiv.org/abs/1902.05660. arXiv:1902.05660 [cs].

P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, volume 1, pp. 958–963, Edinburgh, UK, 2003. IEEE Comput. Soc. ISBN 978-0-7695-1960-9. doi: 10.1109/ICDAR.2003.1227801. URL http://ieeexplore.ieee.org/document/1227801/.

Sumeet S. Singh. Teaching Machines to Code: Neural Markup Generation with Visual Attention, June 2018. URL http://arxiv.org/abs/1802.05415. arXiv:1802.05415 [cs].

R. Smith. An Overview of the Tesseract OCR Engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, pp. 629–633, Curitiba, Parana, Brazil, September 2007. IEEE. ISBN 978-0-7695-2822-9. doi: 10.1109/ICDAR.2007.4376991. URL http://ieeexplore.ieee.org/document/4376991/. ISSN: 1520-5363.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning, November 2022. URL http://arxiv.org/abs/2206.14486. arXiv:2206.14486 [cs, stat].

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A Large Language Model for Science, November 2022. URL http://arxiv.org/abs/2211.09085. arXiv:2211.09085 [cs, stat].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs].

Zelun Wang and Jyh-Charn Liu. Translating Math Formula Images to LaTeX Sequences Using Deep Neural Networks with Sequence-level Training, September 2019. URL http://arxiv.org/abs/1908.11415. arXiv:1908.11415 [cs, stat].

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding, January 2022. URL http://arxiv.org/abs/2012.14740. arXiv:2012.14740 [cs].

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1192–1200, August 2020. doi: 10.1145/3394486.3403172. URL http://arxiv.org/abs/1912.13318. arXiv:1912.13318 [cs].

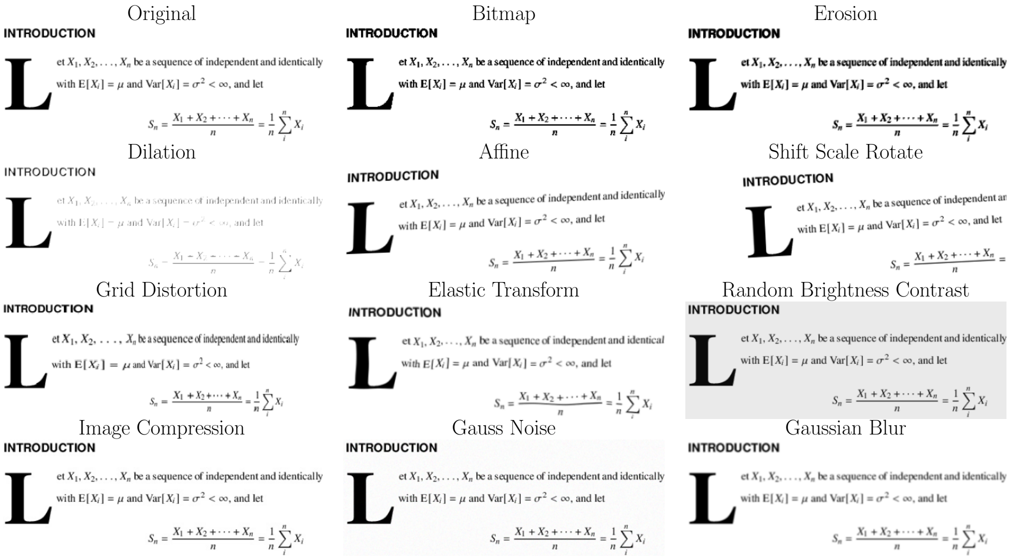
Zuoyu Yan, Xiaode Zhang, Liangcai Gao, Ke Yuan, and Zhi Tang. ConvMath: A Convolutional Sequence Network for Mathematical Expression Recognition, December 2020. URL http://arxiv.org/abs/2012.12619. arXiv:2012.12619 [cs].

Jianshu Zhang, Jun Du, and Lirong Dai. Multi-Scale Attention with Dense Encoder for Handwritten Mathematical Expression Recognition, January 2018. URL http://arxiv.org/abs/1801.03530. arXiv:1801.03530 [cs].

Wenqi Zhao, Liangcai Gao, Zuoyu Yan, Shuai Peng, Lin Du, and Ziyin Zhang. Handwritten Mathematical Expression Recognition with Bidirectionally Trained Transformer, May 2021. URL http://arxiv.org/abs/2105.02412. arXiv:2105.02412 [cs].

Francisco Álvaro, Joan-Andreu Sánchez, and José-Miguel Benedí. Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models. *Pattern Recognition Letters*, 35:58–67, January 2014. ISSN 0167-8655. doi: 10.1016/j.patrec.2012.09.023. URL https://www.sciencedirect.com/science/article/pii/S016786551200308X.

A Dataset


Original

Bitmap

Erosion

Dilation

Affine

Shift Scale Rotate

Grid Distortion

Elastic Transform

Random Brightness Contrast

Image Compression

Gauss Noise

Gaussian Blur

Figure A.1: List of the different image augmentation methods used during training on an example snippet form a sample document.

Name	Number of Pages
arXiv	7,511,745
PMC	536,319
IDL	446,777
Total	**8,494,841**

Table A.1: Dataset composition

The most important data source is arXiv, making up $> 88.4\%$ of the corpus. On arXiv most research documents are paired with the LaTeX source code provided by the authors. The LaTeX source offers more information and is left unprocessed, unlike the XML format from PMC where equations and tables are frequently substituted with images. This allows us to select exactly which information we need to build the dataset.

B Examples

In this section we converted some pages from old text books using the Nougat base model. The text books from the *Internet Archive*[10] and *Project Gutenberg*[11] and are in public domain.

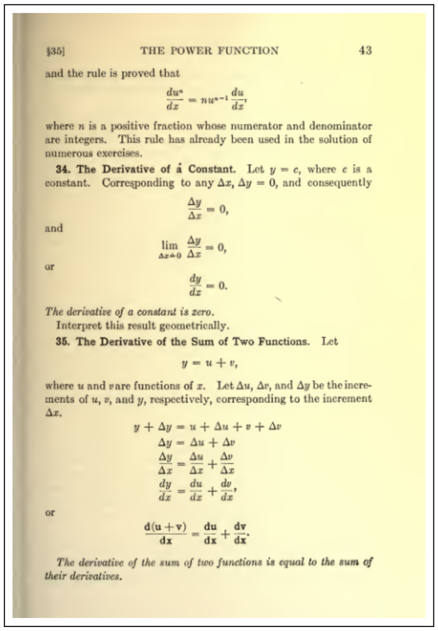
The performance for these scanned pages is noticeable worse than for digital-born documents. However, the model does generate sensible text for each page with few errors. For example see the first row of Fig. B.1. Here the model mistakes the almost illegible exponent n for $*$. In the second row of the same figure the model falls into a repetitive loop after predicting another comma instead of a dot. Similar problems can be seen in Fig. B.2.

In Fig. B.3 we present pages, scanned with a mobile device, from a printed master thesis and the Nougat output. The model is robust to the artifacts that arise when hand-scanning a document.

Explore the examples in this section on the project page: `https://facebookresearch.github.io/nougat/`.

[10] `https://archive.org/`

[11] `https://www.gutenberg.org/`



§35] THE POWER FUNCTION 43

and the rule is proved that

$$\frac{du^n}{dx} = nu^{n-1}\frac{du}{dx},$$

where n is a positive fraction whose numerator and denominator are integers. This rule has already been used in the solution of numerous exercises.

34. The Derivative of ȧ Constant. Let $y = c$, where c is a constant. Corresponding to any Δx, $\Delta y = 0$, and consequently

$$\frac{\Delta y}{\Delta x} = 0,$$

and

$$\lim_{\Delta x \doteq 0} \frac{\Delta y}{\Delta x} = 0,$$

or

$$\frac{dy}{dx} = 0.$$

The derivative of a constant is zero.

Interpret this result geometrically.

35. The Derivative of the Sum of Two Functions. Let

$$y = u + v,$$

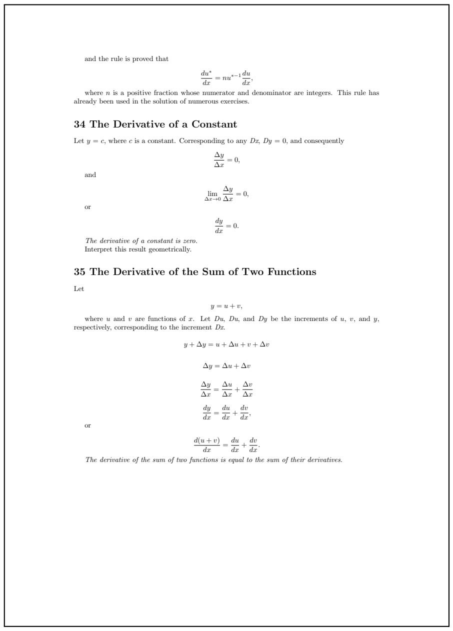
where u and $v$are functions of x. Let Δu, Δv, and Δy be the increments of u, v, and y, respectively, corresponding to the increment Δx.

$$y + \Delta y = u + \Delta u + v + \Delta v$$
$$\Delta y = \Delta u + \Delta v$$
$$\frac{\Delta y}{\Delta x} = \frac{\Delta u}{\Delta x} + \frac{\Delta v}{\Delta x}$$
$$\frac{dy}{dx} = \frac{du}{dx} + \frac{dv}{dx},$$

or

$$\frac{\mathbf{d(u+v)}}{\mathbf{dx}} = \frac{\mathbf{du}}{\mathbf{dx}} + \frac{\mathbf{dv}}{\mathbf{dx}}.$$

The derivative of the sum of two functions is equal to the sum of their derivatives.



and the rule is proved that

$$\frac{du^*}{dx} = nu^{*-1}\frac{du}{dx},$$

where n is a positive fraction whose numerator and denominator are integers. This rule has already been used in the solution of numerous exercises.

34 The Derivative of a Constant

Let $y = c$, where c is a constant. Corresponding to any Dx, $Dy = 0$, and consequently

$$\frac{\Delta y}{\Delta x} = 0,$$

and

$$\lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = 0,$$

or

$$\frac{dy}{dx} = 0.$$

The derivative of a constant is zero.

Interpret this result geometrically.

35 The Derivative of the Sum of Two Functions

Let

$$y = u + v,$$

where u and v are functions of x. Let Du, Du, and Dy be the increments of u, v, and y, respectively, corresponding to the increment Dx.

$$y + \Delta y = u + \Delta u + v + \Delta v$$

$$\Delta y = \Delta u + \Delta v$$

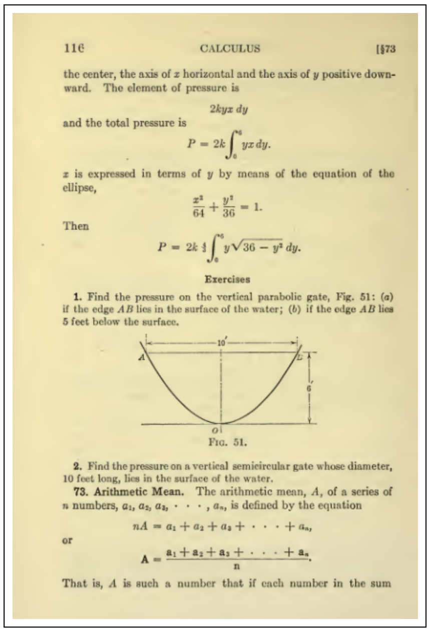
$$\frac{\Delta y}{\Delta x} = \frac{\Delta u}{\Delta x} + \frac{\Delta v}{\Delta x}$$

$$\frac{dy}{dx} = \frac{du}{dx} + \frac{dv}{dx},$$

or

$$\frac{d(u+v)}{dx} = \frac{du}{dx} + \frac{dv}{dx}.$$

The derivative of the sum of two functions is equal to the sum of their derivatives.



116 CALCULUS [§73

the center, the axis of x horizontal and the axis of y positive downward. The element of pressure is

$$2kyx\,dy$$

and the total pressure is

$$P = 2k\int_0^6 yx\,dy.$$

x is expressed in terms of y by means of the equation of the ellipse,

$$\frac{x^2}{64} + \frac{y^2}{36} = 1.$$

Then

$$P = 2k\,\tfrac{4}{3}\int_0^6 y\sqrt{36 - y^2}\,dy.$$

Exercises

1. Find the pressure on the vertical parabolic gate, Fig. 51: (*a*) if the edge AB lies in the surface of the water; (*b*) if the edge AB lies 5 feet below the surface.

Fig. 51.

2. Find the pressure on a vertical semicircular gate whose diameter, 10 feet long, lies in the surface of the water.

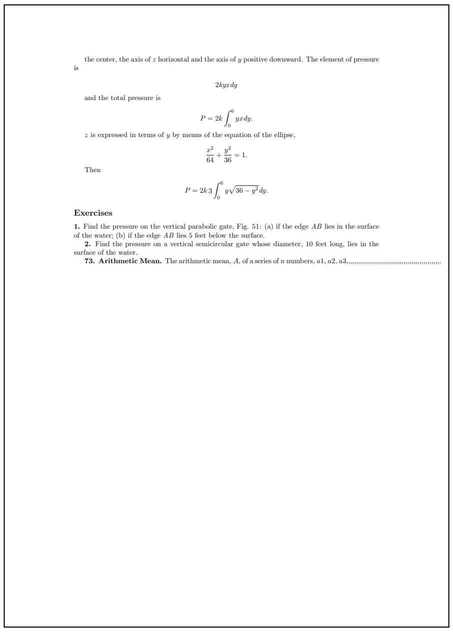
73. Arithmetic Mean. The arithmetic mean, A, of a series of n numbers, $a_1, a_2, a_3, \cdot\cdot\cdot, a_n$, is defined by the equation

$$nA = a_1 + a_2 + a_3 + \cdot\cdot\cdot + a_n,$$

or

$$\mathbf{A} = \frac{\mathbf{a_1 + a_2 + a_3 + \cdot\cdot\cdot + a_n}}{\mathbf{n}}.$$

That is, A is such a number that if each number in the sum



the center, the axis of z horizontal and the axis of y positive downward. The element of pressure is

$$2kyx\,dy$$

and the total pressure is

$$P = 2k\int_0^6 yx\,dy.$$

z is expressed in terms of y by means of the equation of the ellipse,

$$\frac{x^2}{64} + \frac{y^2}{36} = 1.$$

Then

$$P = 2k\,3\int_0^6 y\sqrt{36 - y^2}\,dy.$$

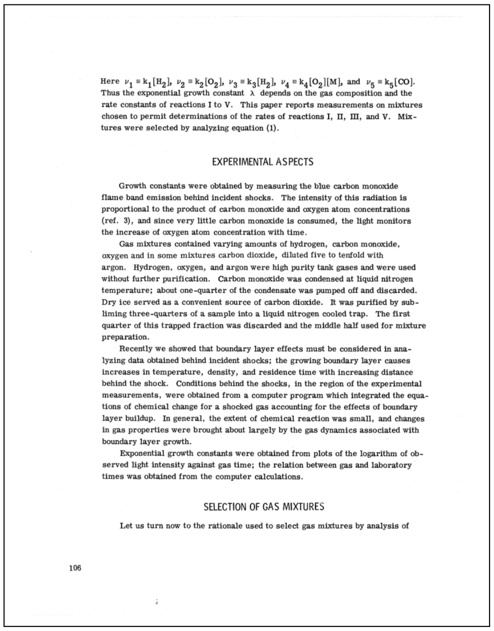
Exercises

1. Find the pressure on the vertical parabolic gate, Fig. 51: (a) if the edge AB lies in the surface of the water; (b) if the edge AB lies 5 feet below the surface.

2. Find the pressure on a vertical semicircular gate whose diameter, 10 feet long, lies in the surface of the water.

73. Arithmetic Mean. The arithmetic mean, A, of a series of n numbers, $a1$, $a2$, $a3$,,

Figure B.1: Example of an old calculus text book (March & Wolff, 1917).



Here $\nu_1 \equiv k_1[H_2]$, $\nu_2 \equiv k_2[O_2]$, $\nu_3 \equiv k_3[H_2]$, $\nu_4 \equiv k_4[O_2][M]$, and $\nu_5 \equiv k_5[CO]$. Thus the exponential growth constant λ depends on the gas composition and the rate constants of reactions I to V. This paper reports measurements on mixtures chosen to permit determinations of the rates of reactions I, II, III, and V. Mixtures were selected by analyzing equation (1).

EXPERIMENTAL ASPECTS

Growth constants were obtained by measuring the blue carbon monoxide flame band emission behind incident shocks. The intensity of this radiation is proportional to the product of carbon monoxide and oxygen atom concentrations (ref. 3), and since very little carbon monoxide is consumed, the light monitors the increase of oxygen atom concentration with time.

Gas mixtures contained varying amounts of hydrogen, carbon monoxide, oxygen and in some mixtures carbon dioxide, diluted five to tenfold with argon. Hydrogen, oxygen, and argon were high purity tank gases and were used without further purification. Carbon monoxide was condensed at liquid nitrogen temperature; about one-quarter of the condensate was pumped off and discarded. Dry ice served as a convenient source of carbon dioxide. It was purified by subliming three-quarters of a sample into a liquid nitrogen cooled trap. The first quarter of this trapped fraction was discarded and the middle half used for mixture preparation.

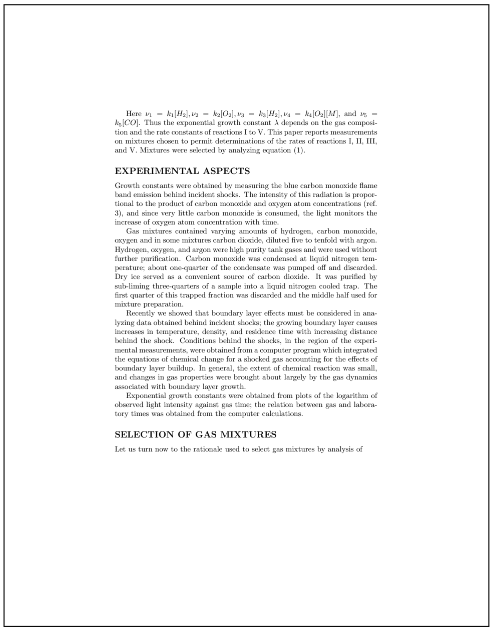
Recently we showed that boundary layer effects must be considered in analyzing data obtained behind incident shocks; the growing boundary layer causes increases in temperature, density, and residence time with increasing distance behind the shock. Conditions behind the shocks, in the region of the experimental measurements, were obtained from a computer program which integrated the equations of chemical change for a shocked gas accounting for the effects of boundary layer buildup. In general, the extent of chemical reaction was small, and changes in gas properties were brought about largely by the gas dynamics associated with boundary layer growth.

Exponential growth constants were obtained from plots of the logarithm of observed light intensity against gas time; the relation between gas and laboratory times was obtained from the computer calculations.

SELECTION OF GAS MIXTURES

Let us turn now to the rationale used to select gas mixtures by analysis of

106



Here $\nu_1 = k_1[H_2], \nu_2 = k_2[O_2], \nu_3 = k_3[H_2], \nu_4 = k_4[O_2][M]$, and $\nu_5 = k_5[CO]$. Thus the exponential growth constant λ depends on the gas composition and the rate constants of reactions I to V. This paper reports measurements on mixtures chosen to permit determinations of the rates of reactions I, II, III, and V. Mixtures were selected by analyzing equation (1).

EXPERIMENTAL ASPECTS

Growth constants were obtained by measuring the blue carbon monoxide flame band emission behind incident shocks. The intensity of this radiation is proportional to the product of carbon monoxide and oxygen atom concentrations (ref. 3), and since very little carbon monoxide is consumed, the light monitors the increase of oxygen atom concentration with time.

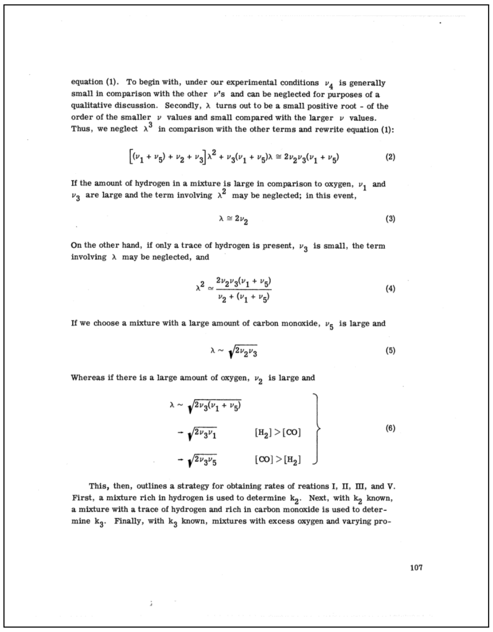
Gas mixtures contained varying amounts of hydrogen, carbon monoxide, oxygen and in some mixtures carbon dioxide, diluted five to tenfold with argon. Hydrogen, oxygen, and argon were high purity tank gases and were used without further purification. Carbon monoxide was condensed at liquid nitrogen temperature; about one-quarter of the condensate was pumped off and discarded. Dry ice served as a convenient source of carbon dioxide. It was purified by sub-liming three-quarters of a sample into a liquid nitrogen cooled trap. The first quarter of this trapped fraction was discarded and the middle half used for mixture preparation.

Recently we showed that boundary layer effects must be considered in analyzing data obtained behind incident shocks; the growing boundary layer causes increases in temperature, density, and residence time with increasing distance behind the shock. Conditions behind the shocks, in the region of the experimental measurements, were obtained from a computer program which integrated the equations of chemical change for a shocked gas accounting for the effects of boundary layer buildup. In general, the extent of chemical reaction was small, and changes in gas properties were brought about largely by the gas dynamics associated with boundary layer growth.

Exponential growth constants were obtained from plots of the logarithm of observed light intensity against gas time; the relation between gas and laboratory times was obtained from the computer calculations.

SELECTION OF GAS MIXTURES

Let us turn now to the rationale used to select gas mixtures by analysis of



equation (1). To begin with, under our experimental conditions ν_4 is generally small in comparison with the other ν's and can be neglected for purposes of a qualitative discussion. Secondly, λ turns out to be a small positive root - of the order of the smaller ν values and small compared with the larger ν values. Thus, we neglect λ^3 in comparison with the other terms and rewrite equation (1):

$$\left[(\nu_1 + \nu_5) + \nu_2 + \nu_3\right]\lambda^2 + \nu_3(\nu_1 + \nu_5)\lambda \cong 2\nu_2\nu_3(\nu_1 + \nu_5) \qquad (2)$$

If the amount of hydrogen in a mixture is large in comparison to oxygen, ν_1 and ν_3 are large and the term involving λ^2 may be neglected; in this event,

$$\lambda \cong 2\nu_2 \qquad (3)$$

On the other hand, if only a trace of hydrogen is present, ν_3 is small, the term involving λ may be neglected, and

$$\lambda^2 \simeq \frac{2\nu_2\nu_3(\nu_1 + \nu_5)}{\nu_2 + (\nu_1 + \nu_5)} \qquad (4)$$

If we choose a mixture with a large amount of carbon monoxide, ν_5 is large and

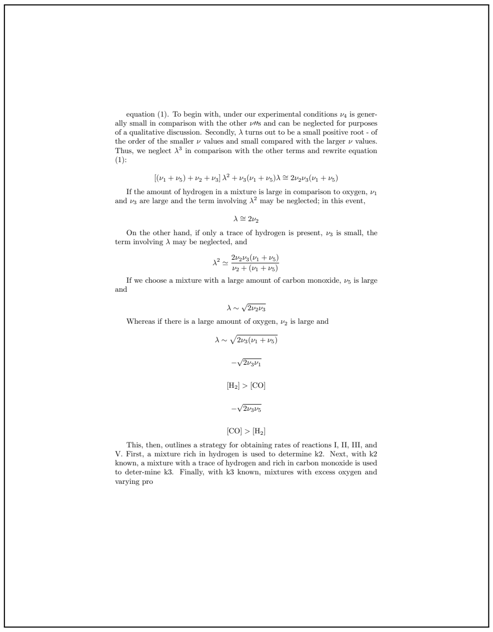
$$\lambda \sim \sqrt{2\nu_2\nu_3} \qquad (5)$$

Whereas if there is a large amount of oxygen, ν_2 is large and

$$\left.\begin{aligned} \lambda &\sim \sqrt{2\nu_3(\nu_1 + \nu_5)} & & \\ &\rightarrow \sqrt{2\nu_3\nu_1} & & [H_2] > [CO] \\ &\rightarrow \sqrt{2\nu_3\nu_5} & & [CO] > [H_2] \end{aligned}\right\} \qquad (6)$$

This, then, outlines a strategy for obtaining rates of reations I, II, III, and V. First, a mixture rich in hydrogen is used to determine k_2. Next, with k_2 known, a mixture with a trace of hydrogen and rich in carbon monoxide is used to determine k_3. Finally, with k_3 known, mixtures with excess oxygen and varying pro-

107



equation (1). To begin with, under our experimental conditions ν_4 is generally small in comparison with the other ν''s and can be neglected for purposes of a qualitative discussion. Secondly, λ turns out to be a small positive root - of the order of the smaller ν values and small compared with the larger ν values. Thus, we neglect λ^3 in comparison with the other terms and rewrite equation (1):

$$[(\nu_1 + \nu_5) + \nu_2 + \nu_3]\,\lambda^2 + \nu_3(\nu_1 + \nu_5)\lambda \cong 2\nu_2\nu_3(\nu_1 + \nu_5)$$

If the amount of hydrogen in a mixture is large in comparison to oxygen, ν_1 and ν_3 are large and the term involving λ^2 may be neglected; in this event,

$$\lambda \cong 2\nu_2$$

On the other hand, if only a trace of hydrogen is present, ν_3 is small, the term involving λ may be neglected, and

$$\lambda^2 \simeq \frac{2\nu_2\nu_3(\nu_1 + \nu_5)}{\nu_2 + (\nu_1 + \nu_5)}$$

If we choose a mixture with a large amount of carbon monoxide, ν_5 is large and

$$\lambda \sim \sqrt{2\nu_2\nu_3}$$

Whereas if there is a large amount of oxygen, ν_2 is large and

$$\lambda \sim \sqrt{2\nu_3(\nu_1 + \nu_5)}$$

$$-\sqrt{2\nu_3\nu_1}$$

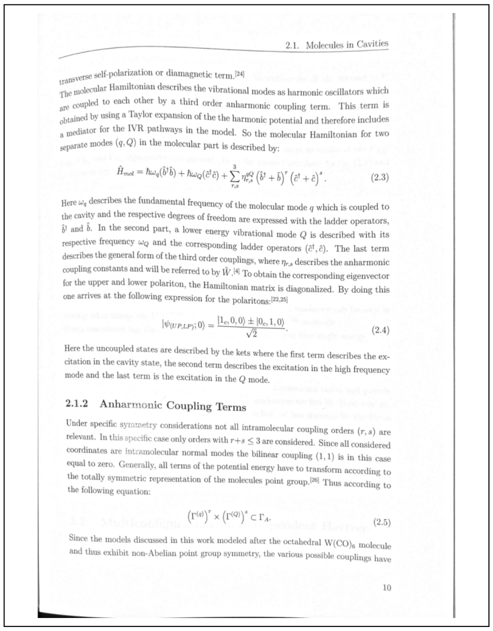
$$[\mathrm{H}_2] > [\mathrm{CO}]$$

$$-\sqrt{2\nu_3\nu_5}$$

$$[\mathrm{CO}] > [\mathrm{H}_2]$$

This, then, outlines a strategy for obtaining rates of reactions I, II, III, and V. First, a mixture rich in hydrogen is used to determine k2. Next, with k2 known, a mixture with a trace of hydrogen and rich in carbon monoxide is used to deter-mine k3. Finally, with k3 known, mixtures with excess oxygen and varying pro

Figure B.2: A selection of pages from a NASA conference from 1970 (Gordon et al., 1970).



transverse self-polarization or diamagnetic term.[24]
The molecular Hamiltonian describes the vibrational modes as harmonic oscillators which are coupled to each other by a third order anharmonic coupling term. This term is obtained by using a Taylor expansion of the the harmonic potential and therefore includes a mediator for the IVR pathways in the model. So the molecular Hamiltonian for two separate modes (q, Q) in the molecular part is described by:

$$\hat{H}_{mol} = \hbar\omega_q(\hat{b}^\dagger\hat{b}) + \hbar\omega_Q(\hat{c}^\dagger\hat{c}) + \sum_{r,s}^{3} \eta_{r,s}^{qQ} \left(\hat{b}^\dagger + \hat{b}\right)^r \left(\hat{c}^\dagger + \hat{c}\right)^s. \tag{2.3}$$

Here ω_q describes the fundamental frequency of the molecular mode q which is coupled to the cavity and the respective degrees of freedom are expressed with the ladder operators, $\hat{b}^\dagger$ and $\hat{b}$. In the second part, a lower energy vibrational mode Q is described with its respective frequency ω_Q and the corresponding ladder operators $(\hat{c}^\dagger, \hat{c})$. The last term describes the general form of the third order couplings, where $\eta_{r,s}$ describes the anharmonic coupling constants and will be referred to by $\hat{W}$.[4] To obtain the corresponding eigenvector for the upper and lower polariton, the Hamiltonian matrix is diagonalized. By doing this one arrives at the following expression for the polaritons:[22,25]

$$|\psi_{(UP,LP)}; 0\rangle = \frac{|1_c, 0, 0\rangle \pm |0_c, 1, 0\rangle}{\sqrt{2}}. \tag{2.4}$$

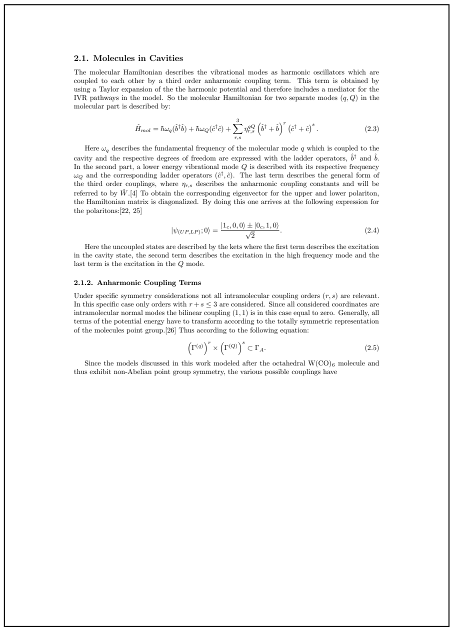
Here the uncoupled states are described by the kets where the first term describes the excitation in the cavity state, the second term describes the excitation in the high frequency mode and the last term is the excitation in the Q mode.

2.1.2 Anharmonic Coupling Terms

Under specific symmetry considerations not all intramolecular coupling orders (r, s) are relevant. In this specific case only orders with $r+s \leq 3$ are considered. Since all considered coordinates are intramolecular normal modes the bilinear coupling $(1, 1)$ is in this case equal to zero. Generally, all terms of the potential energy have to transform according to the totally symmetric representation of the molecules point group.[26] Thus according to the following equation:

$$\left(\Gamma^{(q)}\right)^r \times \left(\Gamma^{(Q)}\right)^s \subset \Gamma_A. \tag{2.5}$$

Since the models discussed in this work modeled after the octahedral W(CO)$_6$ molecule and thus exhibit non-Abelian point group symmetry, the various possible couplings have



2.1. Molecules in Cavities

The molecular Hamiltonian describes the vibrational modes as harmonic oscillators which are coupled to each other by a third order anharmonic coupling term. This term is obtained by using a Taylor expansion of the the harmonic potential and therefore includes a mediator for the IVR pathways in the model. So the molecular Hamiltonian for two separate modes (q, Q) in the molecular part is described by:

$$\hat{H}_{mol} = \hbar\omega_q(\hat{b}^\dagger\hat{b}) + \hbar\omega_Q(\hat{c}^\dagger\hat{c}) + \sum_{r,s}^{3} \eta_{r,s}^{qQ} \left(\hat{b}^\dagger + \hat{b}\right)^r \left(\hat{c}^\dagger + \hat{c}\right)^s. \tag{2.3}$$

Here ω_q describes the fundamental frequency of the molecular mode q which is coupled to the cavity and the respective degrees of freedom are expressed with the ladder operators, $\hat{b}^\dagger$ and $\hat{b}$. In the second part, a lower energy vibrational mode Q is described with its respective frequency ω_Q and the corresponding ladder operators $(\hat{c}^\dagger, \hat{c})$. The last term describes the general form of the third order couplings, where $\eta_{r,s}$ describes the anharmonic coupling constants and will be referred to by $\hat{W}$.[4] To obtain the corresponding eigenvector for the upper and lower polariton, the Hamiltonian matrix is diagonalized. By doing this one arrives at the following expression for the polaritons:[22, 25]

$$|\psi_{(UP,LP)}; 0\rangle = \frac{|1_c, 0, 0\rangle \pm |0_c, 1, 0\rangle}{\sqrt{2}}. \tag{2.4}$$

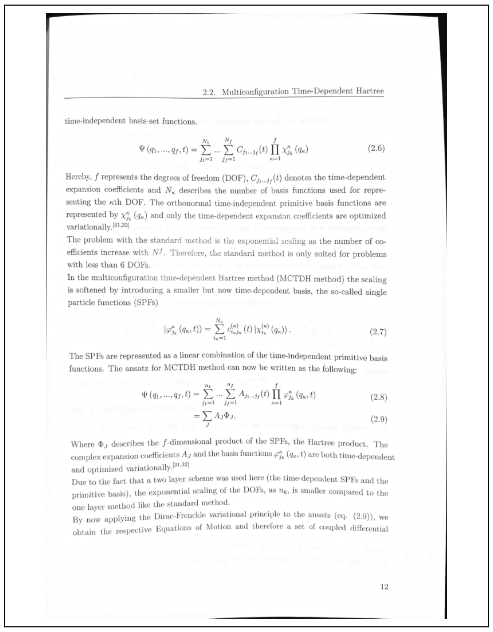
Here the uncoupled states are described by the kets where the first term describes the excitation in the cavity state, the second term describes the excitation in the high frequency mode and the last term is the excitation in the Q mode.

2.1.2. Anharmonic Coupling Terms

Under specific symmetry considerations not all intramolecular coupling orders (r, s) are relevant. In this specific case only orders with $r+s \leq 3$ are considered. Since all considered coordinates are intramolecular normal modes the bilinear coupling $(1, 1)$ is in this case equal to zero. Generally, all terms of the potential energy have to transform according to the totally symmetric representation of the molecules point group.[26] Thus according to the following equation:

$$\left(\Gamma^{(q)}\right)^r \times \left(\Gamma^{(Q)}\right)^s \subset \Gamma_A. \tag{2.5}$$

Since the models discussed in this work modeled after the octahedral W(CO)$_6$ molecule and thus exhibit non-Abelian point group symmetry, the various possible couplings have



time-independent basis-set functions.

$$\Psi(q_1, ..., q_f, t) = \sum_{j_1=1}^{N_1} ... \sum_{j_f=1}^{N_f} C_{j_1...j_f}(t) \prod_{\kappa=1}^{f} \chi_{j_k}^{\kappa}(q_\kappa) \tag{2.6}$$

Hereby, f represents the degrees of freedom (DOF), $C_{j_1...j_f}(t)$ denotes the time-dependent expansion coefficients and N_κ describes the number of basis functions used for representing the κth DOF. The orthonormal time-independent primitive basis functions are represented by $\chi_{j_k}^{\kappa}(q_\kappa)$ and only the time-dependent expansion coefficients are optimized variationally.[31,32]
The problem with the standard method is the exponential scaling as the number of coefficients increase with N^f. Therefore, the standard method is only suited for problems with less than 6 DOFs.
In the multiconfiguration time-dependent Hartree method (MCTDH method) the scaling is softened by introducing a smaller but now time-dependent basis, the so-called single particle functions (SPFs)

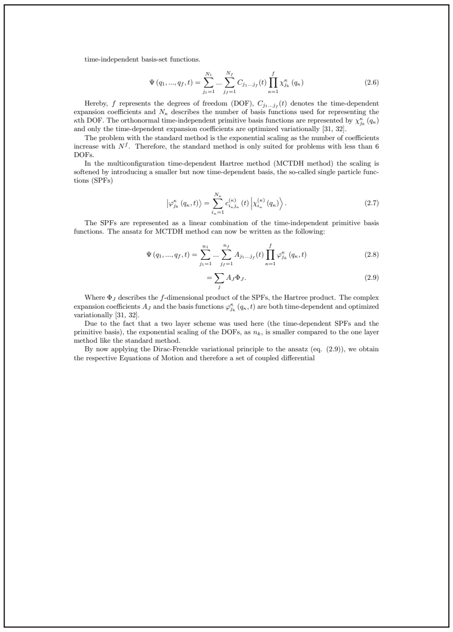
$$|\varphi_{j_k}^{\kappa}(q_\kappa, t)\rangle = \sum_{i_\kappa=1}^{N_\kappa} c_{i_\kappa j_\kappa}^{(\kappa)}(t) |\chi_{i_\kappa}^{(\kappa)}(q_\kappa)\rangle. \tag{2.7}$$

The SPFs are represented as a linear combination of the time-independent primitive basis functions. The ansatz for MCTDH method can now be written as the following:

$$\Psi(q_1, ..., q_f, t) = \sum_{j_1=1}^{n_1} ... \sum_{j_f=1}^{n_f} A_{j_1...j_f}(t) \prod_{\kappa=1}^{f} \varphi_{j_k}^{\kappa}(q_\kappa, t) \tag{2.8}$$

$$= \sum_{J} A_J \Phi_J. \tag{2.9}$$

Where Φ_J describes the f-dimensional product of the SPFs, the Hartree product. The complex expansion coefficients A_J and the basis functions $\varphi_{j_k}^{\kappa}(q_\kappa, t)$ are both time-dependent and optimized variationally.[31,32]
Due to the fact that a two layer scheme was used here (the time-dependent SPFs and the primitive basis), the exponential scaling of the DOFs, as n_k, is smaller compared to the one layer method like the standard method.
By now applying the Dirac-Frenckle variational principle to the ansatz (eq. (2.9)), we obtain the respective Equations of Motion and therefore a set of coupled differential



time-independent basis-set functions.

$$\Psi(q_1, ..., q_f, t) = \sum_{j_1=1}^{N_1} ... \sum_{j_f=1}^{N_f} C_{j_1...j_f}(t) \prod_{\kappa=1}^{f} \chi_{j_k}^{\kappa}(q_\kappa) \tag{2.6}$$

Hereby, f represents the degrees of freedom (DOF), $C_{j_1...j_f}(t)$ denotes the time-dependent expansion coefficients and N_κ describes the number of basis functions used for representing the κth DOF. The orthonormal time-independent primitive basis functions are represented by $\chi_{j_k}^{\kappa}(q_\kappa)$ and only the time-dependent expansion coefficients are optimized variationally [31, 32].

The problem with the standard method is the exponential scaling as the number of coefficients increase with N^f. Therefore, the standard method is only suited for problems with less than 6 DOFs.

In the multiconfiguration time-dependent Hartree method (MCTDH method) the scaling is softened by introducing a smaller but now time-dependent basis, the so-called single particle functions (SPFs)

$$\left|\varphi_{j_k}^{\kappa}(q_\kappa, t)\right\rangle = \sum_{i_\kappa=1}^{N_\kappa} c_{i_\kappa j_\kappa}^{(\kappa)}(t) \left|\chi_{i_\kappa}^{(\kappa)}(q_\kappa)\right\rangle. \tag{2.7}$$

The SPFs are represented as a linear combination of the time-independent primitive basis functions. The ansatz for MCTDH method can now be written as the following:

$$\Psi(q_1, ..., q_f, t) = \sum_{j_1=1}^{n_1} ... \sum_{j_f=1}^{n_f} A_{j_1...j_f}(t) \prod_{\kappa=1}^{f} \varphi_{j_k}^{\kappa}(q_\kappa, t) \tag{2.8}$$

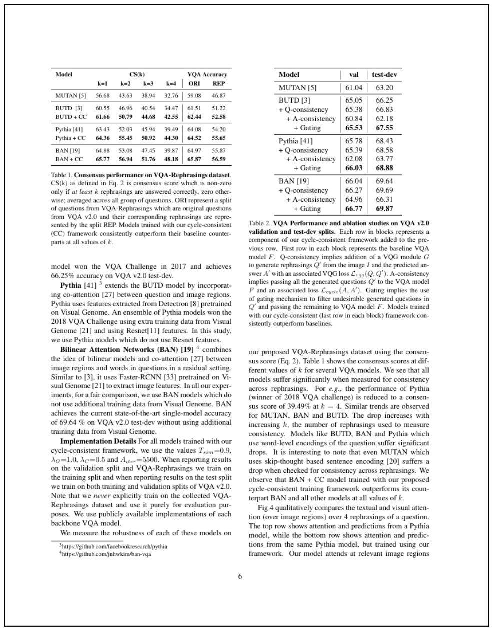
$$= \sum_{j} A_J \Phi_J. \tag{2.9}$$

Where Φ_J describes the f-dimensional product of the SPFs, the Hartree product. The complex expansion coefficients A_J and the basis functions $\varphi_{j_k}^{\kappa}(q_\kappa, t)$ are both time-dependent and optimized variationally [31, 32].

Due to the fact that a two layer scheme was used here (the time-dependent SPFs and the primitive basis), the exponential scaling of the DOFs, as n_k, is smaller compared to the one layer method like the standard method.

By now applying the Dirac-Frenckle variational principle to the ansatz (eq. (2.9)), we obtain the respective Equations of Motion and therefore a set of coupled differential

Figure B.3: Scan of a modern thesis with a mobile device camera, with permission from the author.



Model	CS(k)				VQA Accuracy	
	k=1	k=2	k=3	k=4	ORI	REP
MUTAN [5]	56.68	43.63	38.94	32.76	59.08	46.87
BUTD [3]	60.55	46.96	40.54	34.47	61.51	51.22
BUTD + CC	**61.66**	**50.79**	**44.68**	**42.55**	**62.44**	**52.58**
Pythia [41]	63.43	52.03	45.94	39.49	64.08	54.20
Pythia + CC	**64.36**	**55.45**	**50.92**	**44.30**	**64.52**	**55.65**
BAN [19]	64.88	53.08	47.45	39.87	64.97	55.87
BAN + CC	**65.77**	**56.94**	**51.76**	**48.18**	**65.87**	**56.59**

Table 1. **Consensus performance on VQA-Rephrasings dataset**. CS(k) as defined in Eq. 2 is consensus score which is non-zero only if *at least* k rephrasings are answered correctly, zero otherwise; averaged across all group of questions. ORI represent a split of questions from VQA-Rephrasings which are original questions from VQA v2.0 and their corresponding rephrasings are represented by the split REP. Models trained with our cycle-consistent (CC) framework consistently outperform their baseline counterparts at all values of k.

Model	val	test-dev
MUTAN [5]	61.04	63.20
BUTD [3]	65.05	66.25
+ Q-consistency	65.38	66.83
+ A-consistency	60.84	62.18
+ Gating	**65.53**	**67.55**
Pythia [41]	65.78	68.43
+ Q-consistency	65.39	68.58
+ A-consistency	62.08	63.77
+ Gating	**66.03**	**68.88**
BAN [19]	66.04	69.64
+ Q-consistency	66.27	69.69
+ A-consistency	64.96	66.31
+ Gating	**66.77**	**69.87**

Table 2. **VQA Performance and ablation studies on VQA v2.0 validation and test-dev splits**. Each row in blocks represents a component of our cycle-consistent framework added to the previous row. First row in each block represents the baseline VQA model F. Q-consistency implies addition of a VQG module G to generate rephrasings Q' from the image I and the predicted answer A' with an associated VQG loss $\mathcal{L}_{vqg}(Q, Q')$. A-consistency implies passing all the generated questions Q' to the VQA model F and an associated loss $\mathcal{L}_{cycle}(A, A')$. Gating implies the use of gating mechanism to filter undesirable generated questions in Q' and passing the remaining to VQA model F. Models trained with our cycle-consistent (last row in each block) framework consistently outperform baselines.

model won the VQA Challenge in 2017 and achieves 66.25% accuracy on VQA v2.0 test-dev.

Pythia [41] [3] extends the BUTD model by incorporating co-attention [27] between question and image regions. Pythia uses features extracted from Detectron [8] pretrained on Visual Genome. An ensemble of Pythia models won the 2018 VQA Challenge using extra training data from Visual Genome [21] and using Resnet[11] features. In this study, we use Pythia models which do not use Resnet features.

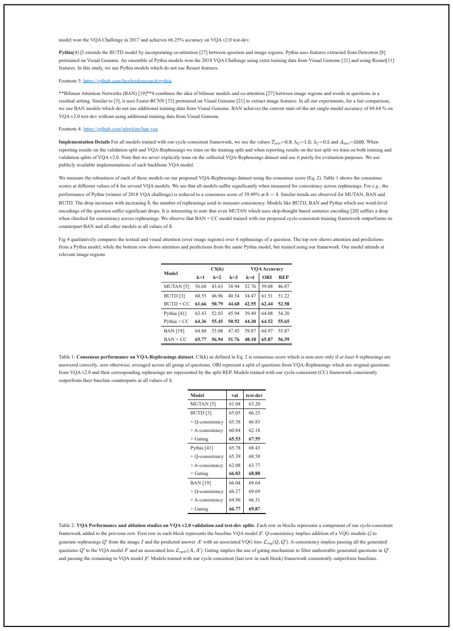
Bilinear Attention Networks (BAN) [19] [4] combines the idea of bilinear models and co-attention [27] between image regions and words in questions in a residual setting. Similar to [3], it uses Faster-RCNN [33] pretrained on Visual Genome [21] to extract image features. In all our experiments, for a fair comparison, we use BAN models which do not use additional training data from Visual Genome. BAN achieves the current state-of-the-art single-model accuracy of 69.64 % on VQA v2.0 test-dev without using additional training data from Visual Genome.

Implementation Details For all models trained with our cycle-consistent framework, we use the values T_{sim}=0.9, λ_G=1.0, λ_C=0.5 and A_{iter}=5500. When reporting results on the validation split and VQA-Rephrasings we train on the training split and when reporting results on the test split we train on both training and validation splits of VQA v2.0. Note that we *never* explicitly train on the collected VQA-Rephrasings dataset and use it purely for evaluation purposes. We use publicly available implementations of each backbone VQA model.

We measure the robustness of each of these models on our proposed VQA-Rephrasings dataset using the consensus score (Eq. 2). Table 1 shows the consensus scores at different values of k for several VQA models. We see that all models suffer significantly when measured for consistency across rephrasings. For *e.g.*, the performance of Pythia (winner of 2018 VQA challenge) is reduced to a consensus score of 39.49% at $k = 4$. Similar trends are observed for MUTAN, BAN and BUTD. The drop increases with increasing k, the number of rephrasings used to measure consistency. Models like BUTD, BAN and Pythia which use word-level encodings of the question suffer significant drops. It is interesting to note that even MUTAN which uses skip-thought based sentence encoding [20] suffers a drop when checked for consistency across rephrasings. We observe that BAN + CC model trained with our proposed cycle-consistent training framework outperforms its counterpart BAN and all other models at all values of k.

Fig 4 qualitatively compares the textual and visual attention (over image regions) over 4 rephrasings of a question. The top row shows attention and predictions from a Pythia model, while the bottom row shows attention and predictions from the same Pythia model, but trained using our framework. Our model attends at relevant image regions

[3] https://github.com/facebookresearch/pythia
[4] https://github.com/jnhwkim/ban-vqa



model won the VQA Challenge in 2017 and achieves 66.25% accuracy on VQA v2.0 test-dev.

Pythia[41]3 extends the BUTD model by incorporating co-attention [27] between question and image regions. Pythia uses features extracted from Detectron [8] pretrained on Visual Genome. An ensemble of Pythia models won the 2018 VQA Challenge using extra training data from Visual Genome [21] and using Resnet[11] features. In this study, we use Pythia models which do not use Resnet features.

Footnote 3: https://github.com/facebookresearch/pythia

Bilinear Attention Networks (BAN) [19]4 combines the idea of bilinear models and co-attention [27] between image regions and words in questions in a residual setting. Similar to [3], it uses Faster-RCNN [33] pretrained on Visual Genome [21] to extract image features. In all our experiments, for a fair comparison, we use BAN models which do not use additional training data from Visual Genome. BAN achieves the current state-of-the-art single-model accuracy of 69.64 % on VQA v2.0 test-dev without using additional training data from Visual Genome.

Footnote 4: https://github.com/jnhwkim/ban-vqa

Implementation Details For all models trained with our cycle-consistent framework, we use the values T_{sim}=0.9, λ_G=1.0, λ_C=0.5 and A_{iter}=5500. When reporting results on the validation split and VQA-Rephrasings we train on the training split and when reporting results on the test split we train on both training and validation splits of VQA v2.0. Note that we *never* explicitly train on the collected VQA-Rephrasings dataset and use it purely for evaluation purposes. We use publicly available implementations of each backbone VQA model.

We measure the robustness of each of these models on our proposed VQA-Rephrasings dataset using the consensus score (Eq. 2). Table 1 shows the consensus scores at different values of k for several VQA models. We see that all models suffer significantly when measured for consistency across rephrasings. For *e.g.*, the performance of Pythia (winner of 2018 VQA challenge) is reduced to a consensus score of 39.49% at $k = 4$. Similar trends are observed for MUTAN, BAN and BUTD. The drop increases with increasing k, the number of rephrasings used to measure consistency. Models like BUTD, BAN and Pythia which use word-level encodings of the question suffer significant drops. It is interesting to note that even MUTAN which uses skip-thought based sentence encoding [20] suffers a drop when checked for consistency across rephrasings. We observe that BAN + CC model trained with our proposed cycle-consistent training framework outperforms its counterpart BAN and all other models at all values of k.

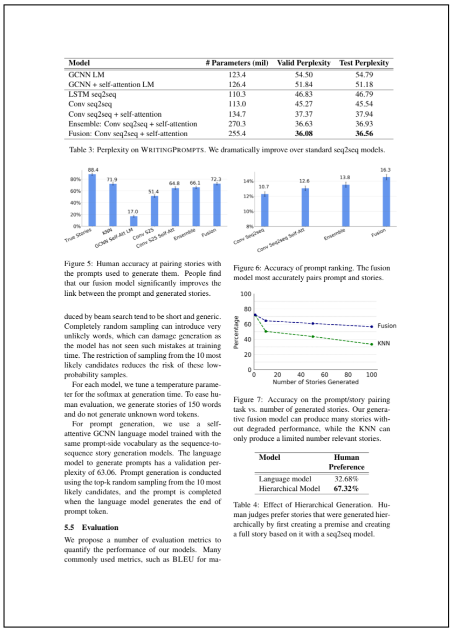
Fig 4 qualitatively compares the textual and visual attention (over image regions) over 4 rephrasings of a question. The top row shows attention and predictions from a Pythia model, while the bottom row shows attention and predictions from the same Pythia model, but trained using our framework. Our model attends at relevant image regions

Model	CS(k)				VQA Accuracy	
	k=1	k=2	k=3	k=4	ORI	REP
MUTAN [5]	56.68	43.63	38.94	32.76	59.08	46.87
BUTD [3]	60.55	46.96	40.54	34.47	61.51	51.22
BUTD + CC	**61.66**	**50.79**	**44.68**	**42.55**	**62.44**	**52.58**
Pythia [41]	63.43	52.03	45.94	39.49	64.08	54.20
Pythia + CC	**64.36**	**55.45**	**50.92**	**44.30**	**64.52**	**55.65**
BAN [19]	64.88	53.08	47.45	39.87	64.97	55.87
BAN + CC	**65.77**	**56.94**	**51.76**	**48.18**	**65.87**	**56.59**

Table 1: **Consensus performance on VQA-Rephrasings dataset**. CS(k) as defined in Eq. 2 is consensus score which is non-zero only if *at least* k rephrasings are answered correctly, zero otherwise; averaged across all group of questions. ORI represent a split of questions from VQA-Rephrasings which are original questions from VQA v2.0 and their corresponding rephrasings are represented by the split REP. Models trained with our cycle-consistent (CC) framework consistently outperform their baseline counterparts at all values of k.

Model	val	test-dev
MUTAN [5]	61.04	63.20
BUTD [3]	65.05	66.25
+ Q-consistency	65.38	66.83
+ A-consistency	60.84	62.18
+ Gating	**65.53**	**67.55**
Pythia [41]	65.78	68.43
+ Q-consistency	65.39	68.58
+ A-consistency	62.08	63.77
+ Gating	**66.03**	**68.88**
BAN [19]	66.04	69.64
+ Q-consistency	66.27	69.69
+ A-consistency	64.96	66.31
+ Gating	**66.77**	**69.87**

Table 2: **VQA Performance and ablation studies on VQA v2.0 validation and test-dev splits**. Each row in blocks represents a component of our cycle-consistent framework added to the previous row. First row in each block represents the baseline VQA model F. Q-consistency implies addition of a VQG module G to generate rephrasings Q' from the image I and the predicted answer A' with an associated VQG loss $\mathcal{L}_{vqg}(Q, Q')$. A-consistency implies passing all the generated questions Q' to the VQA model F and an associated loss $\mathcal{L}_{cycle}(A, A')$. Gating implies the use of gating mechanism to filter undesirable generated questions in Q' and passing the remaining to VQA model F. Models trained with our cycle-consistent (last row in each block) framework consistently outperform baselines.



Model	# Parameters (mil)	Valid Perplexity	Test Perplexity
GCNN LM	123.4	54.50	54.79
GCNN + self-attention LM	126.4	51.84	51.18
LSTM seq2seq	110.3	46.83	46.79
Conv seq2seq	113.0	45.27	45.54
Conv seq2seq + self-attention	134.7	37.37	37.94
Ensemble: Conv seq2seq + self-attention	270.3	36.63	36.93
Fusion: Conv seq2seq + self-attention	255.4	**36.08**	**36.56**

Table 3: Perplexity on WritingPrompts. We dramatically improve over standard seq2seq models.

Figure 5: Human accuracy at pairing stories with the prompts used to generate them. People find that our fusion model significantly improves the link between the prompt and generated stories.

Figure 6: Accuracy of prompt ranking. The fusion model most accurately pairs prompt and stories.

Figure 7: Accuracy on the prompt/story pairing task vs. number of generated stories. Our generative fusion model can produce many stories without degraded performance, while the KNN can only produce a limited number relevant stories.

duced by beam search tend to be short and generic. Completely random sampling can introduce very unlikely words, which can damage generation as the model has not seen such mistakes at training time. The restriction of sampling from the 10 most likely candidates reduces the risk of these low-probability samples.

For each model, we tune a temperature parameter for the softmax at generation time. To ease human evaluation, we generate stories of 150 words and do not generate unknown word tokens.

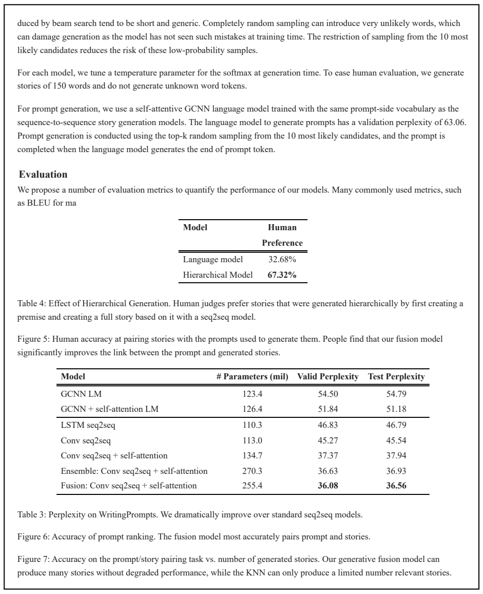
For prompt generation, we use a self-attentive GCNN language model trained with the same prompt-side vocabulary as the sequence-to-sequence story generation models. The language model to generate prompts has a validation perplexity of 63.06. Prompt generation is conducted using the top-k random sampling from the 10 most likely candidates, and the prompt is completed when the language model generates the end of prompt token.

5.5 Evaluation

We propose a number of evaluation metrics to quantify the performance of our models. Many commonly used metrics, such as BLEU for ma-

Model	Human Preference
Language model	32.68%
Hierarchical Model	**67.32%**

Table 4: Effect of Hierarchical Generation. Human judges prefer stories that were generated hierarchically by first creating a premise and creating a full story based on it with a seq2seq model.



duced by beam search tend to be short and generic. Completely random sampling can introduce very unlikely words, which can damage generation as the model has not seen such mistakes at training time. The restriction of sampling from the 10 most likely candidates reduces the risk of these low-probability samples.

For each model, we tune a temperature parameter for the softmax at generation time. To ease human evaluation, we generate stories of 150 words and do not generate unknown word tokens.

For prompt generation, we use a self-attentive GCNN language model trained with the same prompt-side vocabulary as the sequence-to-sequence story generation models. The language model to generate prompts has a validation perplexity of 63.06. Prompt generation is conducted using the top-k random sampling from the 10 most likely candidates, and the prompt is completed when the language model generates the end of prompt token.

Evaluation

We propose a number of evaluation metrics to quantify the performance of our models. Many commonly used metrics, such as BLEU for ma

Model	Human Preference
Language model	32.68%
Hierarchical Model	**67.32%**

Table 4: Effect of Hierarchical Generation. Human judges prefer stories that were generated hierarchically by first creating a premise and creating a full story based on it with a seq2seq model.

Figure 5: Human accuracy at pairing stories with the prompts used to generate them. People find that our fusion model significantly improves the link between the prompt and generated stories.

Model	# Parameters (mil)	Valid Perplexity	Test Perplexity
GCNN LM	123.4	54.50	54.79
GCNN + self-attention LM	126.4	51.84	51.18
LSTM seq2seq	110.3	46.83	46.79
Conv seq2seq	113.0	45.27	45.54
Conv seq2seq + self-attention	134.7	37.37	37.94
Ensemble: Conv seq2seq + self-attention	270.3	36.63	36.93
Fusion: Conv seq2seq + self-attention	255.4	**36.08**	**36.56**

Table 3: Perplexity on WritingPrompts. We dramatically improve over standard seq2seq models.

Figure 6: Accuracy of prompt ranking. The fusion model most accurately pairs prompt and stories.

Figure 7: Accuracy on the prompt/story pairing task vs. number of generated stories. Our generative fusion model can produce many stories without degraded performance, while the KNN can only produce a limited number relevant stories.

Figure B.4: Pages with tables. Upper: Fan et al. (2018) page 6, Lower: Shah et al. (2019) page 6