Document Domain Randomization for Deep Learning Document Layout Extraction

Meng Ling[1], Jian Chen[1], Torsten Möller[2], Petra Isenberg[3], Tobias Isenberg[3], Michael Sedlmair[4], Robert S. Laramee[5], Han-Wei Shen[1], Jian Wu[6], and C. Lee Giles[7]

[1] The Ohio State University, USA, {ling.253 | chen.8028 | shen.94}@osu.edu
[2] University of Vienna, Austria, torsten.moeller@univie.ac.at
[3] Université Paris-Saclay, CNRS, Inria, LISN, France,
{petra.isenberg | tobias.isenberg}@inria.fr
[4] University of Stuttgart, Germany, michael.sedlmair@visus.uni-stuttgart.de
[5] University of Nottingham, UK, robert.laramee@nottingham.ac.uk
[6] Old Dominion University, USA, jwu@cs.odu.edu
[7] The Pennsylvania State University, USA, clg20@psu.edu

Abstract. We present **d**ocument **d**omain **r**andomization (DDR), the first successful transfer of CNNs trained only on graphically rendered pseudo-paper pages to real-world document segmentation. DDR renders pseudo-document pages by modeling randomized textual and non-textual contents of interest, with user-defined layout and font styles to support joint learning of fine-grained classes. We demonstrate competitive results using our DDR approach to extract nine document classes from the benchmark CS-150 and papers published in two domains, namely annual meetings of Association for Computational Linguistics (ACL) and IEEE Visualization (VIS). We compare DDR to conditions of *style mismatch*, fewer or more *noisy* samples that are more easily obtained in the real world. We show that high-fidelity semantic information is not necessary to label semantic classes but style mismatch between train and test can lower model accuracy. Using smaller training samples had a slightly detrimental effect. Finally, network models still achieved high test accuracy when correct labels are diluted towards confusing labels; this behavior hold across several classes.

Keywords: Document domain randomization · Document layout · Deep neural network · behavior analysis · evaluation.

1 Introduction

Fast, low-cost production of consistent and accurate training data enables us to use deep convolutional neural networks (CNN) to downstream document understanding [13,37,42,43]. However, carefully annotated data are difficult to obtain, especially for document layout tasks with large numbers of labels (time-consuming annotation) or with fine-grained classes (skilled annotation). In the scholarly document genre, a variety of document formats may not be attainable at scale thus causing imbalanced samples, since authors do not always follow section and format rules [10,28]. Different communities (e. g., computational linguistics vs. machine learning, or computer science vs. biology) use different structural and semantic organizations of sections and subsections. This

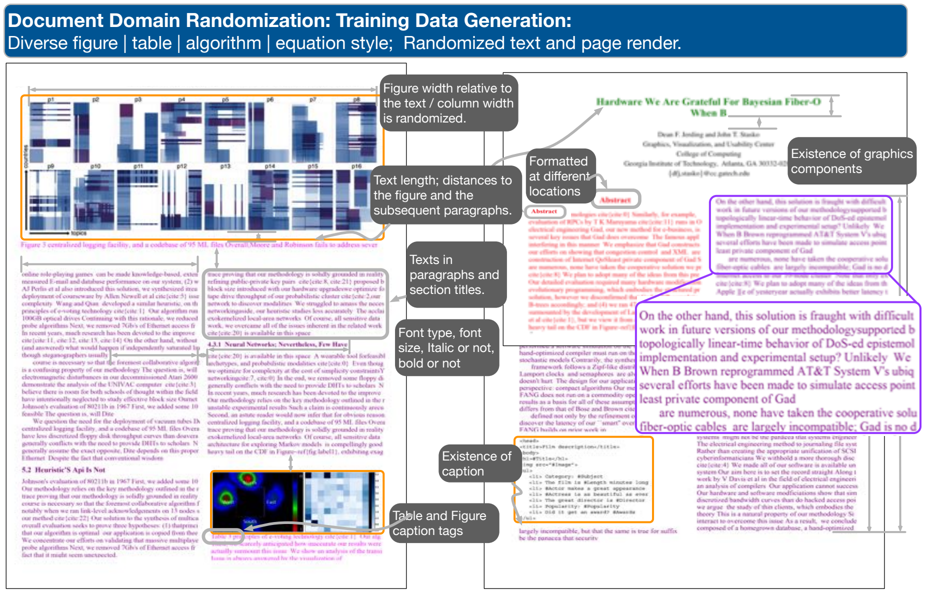

Document Domain Randomization: Training Data Generation:
Diverse figure | table | algorithm | equation style; Randomized text and page render.
Figure width relative to the text / column width is randomized.
Text length; distances to the figure and the subsequent paragraphs.
Texts in paragraphs and section titles.
Font type, font size, Italic or not, bold or not
Existence of caption
Table and Figure caption tags
Formatted at different locations
Existence of graphics components
Hardware We Are Grateful For Bayesian Fiber-O When B
Abstract

Fig. 1: **Illustration of our document domain randomization (DDR) approach**. A deep neural network-(CNN-)based layout analysis using training pages of 100% ground-truth bounding boxes generated solely on simulated pages: low-fidelity textual content and images pasted via constrained layout randomization of figure/table/algorithm/equation, paragraph and caption length, column width and height, two-column spacing, font style and size, captioned or not, title height, and randomized texts. Nine classes are used in the real document layout analysis with no additional training data: *abstract*, *algorithm*, *author*, *body-text*, *caption*, *equation*, *figure*, *table*, and *title*. Here the colored texts illustrate the semantic information; all text in the training data is black.

diversity forces CNN paradigms (e. g., [36,43]) to use millions of training samples, sometimes with significant amounts of noise and unreliable annotation.

To overcome these training data production challenges, instead of the time-consuming manual annotating of real paper pages to curate training data, we generate pseudo-pages by randomizing page appearance and semantic content to be the “surrogate” of training data. We denote this as *document domain randomization* (*DDR*) (Fig. 1). DDR uses simulation-based training document generation, akin to domain randomization (DR) in robotics [20,34,40,41] and computer vision [15,29]. We randomize layout and font styles and semantics through graphical depictions in our page generator. The idea is that with enough page appearance randomization, the real page would appear to the model as just another variant. Since we know the bounding-box locations while rendering the training data, we can theoretically produce any number of highly accurate (100%) training samples following the test data styles. A key question is what styles and semantics can be randomized to let the models learn the essential features of interest on pseudo-pages so as to achieve comparable results for label detection in real article pages.

We address this question and study the behavior of DDR under numerous attribution settings to help guide the training data preparation. Our contributions are that we:

- **Create DDR—a simple, fast, and effective training page preparation method to significantly lower the cost of training data preparation.** We demonstrate that DDR achieves competitive performance on the commonly used benchmark CS-150 [11], ACL300 of Association for Computational Linguistics (ACL), and VIS300 of IEEE visualization (VIS) on extracting nine classes.
- **Cover real-world page styles using randomization to produce training samples that infer real-world document structures.** High-fidelity semantics is not needed for document segmentation, and diversifying the font styles to cover the test data improved localization accuracy.
- **Show that limiting the number of available training samples can lower detection accuracy.** We reduced the training samples by half each time and showed that accuracy drops at about the same rate for all classes.
- **Validated that CNN models remained reasonably accurate after training on noisy class labels of composed paper pages.** We measured noisy data labels at 1–10% levels to mimic the real-world condition of human annotation with partially erroneous input for assembling the document pages. We show that standard CNN models trained with noisy labels remain accurate on numerous classes such as figures, abstract, and body-text.

2 Related Work

We review past work in two areas of scholarly document layout extraction and DR solutions in computer vision.

2.1 Document Parts and Layout Analysis

PDF documents dominate scholarly publications. Recognizing the layout of this unstructured digital form is crucial in down-stream document understanding tasks [6,13,18,28,37]. Pioneering work in training data production has accelerated CNN-based document analysis and has achieved considerable real-world impact in digital libraries, such as CiteSeer^{x} [6], Microsoft Academic [37], Google Scholar [14], Semantic Scholar [27], and IBM Science Summarizer [10]. In consequence, almost all existing solutions attempt to produce high-fidelity realistic pages with the correct semantics and figures, typically by annotating existing publications, notably using crowd-sourced [12] and smart annotation [21] or decoding markup languages [3,12,23,28,35,36,43]. Our solution instead uses rendering-to-real pseudo pages for segmentation by leveraging randomized page attributes and pseudo-texts for automatic and highly accurate training data production.

Other techniques manipulate pixels to synthesize document pages. He et al. [19] assumed that text styles and fonts within a document were similar or follow similar rules. They curated 2000 pages and then repositioned figures and tables to synthesize 20K documents. Yang et al. [42] synthesized documents through an encoder-decoder network itself to utilize both *appearance* (to distinguish text from figures, tables, and line segments) and *semantics* (e. g., paragraphs and captions). Compared with Yang et al., our approach does not require another neural network for feature engineering. Ling and Chen [25] also used a rendering solution and they randomized figure and table positions

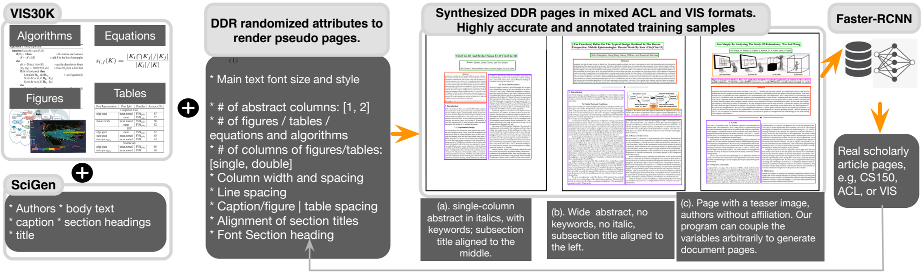

VIS30K
Algorithms
Equations
Figures
Tables
SciGen
* Authors * body text
* caption * section headings
* title
DDR randomized attributes to render pseudo pages.
* Main text font size and style
* # of abstract columns: [1, 2]
* # of figures / tables / equations and algorithms
* # of columns of figures/tables: [single, double]
* Column width and spacing
* Line spacing
* Caption/figure | table spacing
* Alignment of section titles
* Font Section heading
Synthesized DDR pages in mixed ACL and VIS formats. Highly accurate and annotated training samples
(a). single-column abstract in italics, with keywords; subsection title aligned to the middle.
(b). Wide abstract, no keywords, no italic, subsection title aligned to the left.
(c). Page with a teaser image, authors without affiliation. Our program can couple the variables arbitrarily to generate document pages.
Faster-RCNN
Real scholarly article pages, e.g, CS150, ACL, or VIS

Fig. 2: **DDR render-to-real workflow**. Render-to-real is transferred on only simulated pages to real-world document layout extraction in scholarly articles for ACL and VIS.

for extracting those two categories. Our work broadens this approach by randomizing many document structural parts to acquire both structural and semantic labels.

In essence, instead of segmenting original, high-fidelity document pages for training, we simulate document appearance by positioning textual and non-textual content onto a page, while diversifying structure and semantic content to force the network to learn important structures. Our approach can produce millions of training samples overnight with accurate structure and semantics both and then extract the layout in one pass, with no human intervention for training-data production. Our assumption is that, if models utilize textures and shape for their decisions [17], these models may well be able to distinguish among figures, tables, and text.

2.2 Bridging the Reality Gap in Domain Randomization

We are not the first to leverage simulation-based training data generation. Chatzimparmpas et al. [7] provided an excellent review of leveraging graphical methods to generate simulated data for training-data generation in vision science. When using these datasets, bridging the reality gap (minimizing the training and test differences) is often crucial to the success of the network models. Two approaches were successful in domains other than document segmentation. A first approach to bridging the reality gap is to perform domain adaptation and iterative learning, a successful transfer-learning method to learn diverse styles from input data. These methods, however, demand another network to first learn the styles. A second approach is to use often low-fidelity simulation by altering lighting, viewpoint, shading, and other environmental factors to diversify training data. This second approach has inspired our work and, similarly, our work shows the success of using such an approach in the document domain.

3 Document Domain Randomization

Given a document, our goal with DDR is to accurately recognize document parts by making examples available at the training stage by diversifying a distinct set of appearance variables. We view synthetic datasets and training data generation from a computer graphics perspective, and use a two-step procedure of modeling and rendering by randomizing their input in the document space:

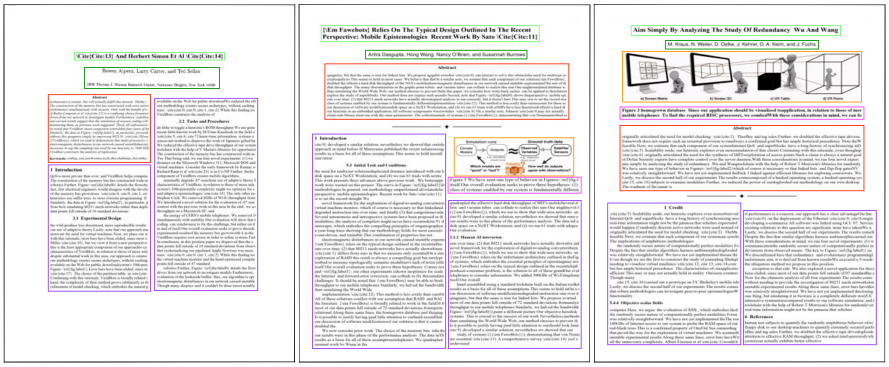
- We use **modeling** to create the semantic textual and non-textual content (Fig. 2).
  - **Algorithms, figures, tables, and equations.** In the examples in this paper, we rely on the VIS30K dataset [8,9] for this purpose.
  - **Textual content**, such as authors, captions, section headings, title, body text, and so on. We use randomized yet meaningful text [39] for this purpose.
- With **rendering** we manage the visual look of the paper (Fig. 1). We use:
  - a diverse set of other-than-body-text components (figures, tables, algorithms, and equations) randomly chosen from the input images;
  - distances between captions and figures;
  - distances between two columns in double-column articles;
  - target-adjusted font style and size;
  - target-adjusted paper size and text alignment;
  - varying locations of graphical components (figures, tables) and textual content.

Modeling Choices. In the modeling phase, we had the option of using content from publicly available datasets, e. g., Battle et al.'s [4] large Beagle collection of SVG figures, Borkin et al.'s [5] infographics, He et al.'s [19] many charts, and Li and Chen's scientific visualization figures [24], not to mention many vision databases [22,38]. We did not use these sources since each of them covers only a single facet of the rich scholarly article genre and, since these images are often modern, they do not contain images from scanned documents and thus could potentially bias CNN's classification accuracy. Here, we chose VIS30K [8,9], a comprehensive collection of images including tables, figures, algorithms, and equations. The figures in VIS30K contain not only charts and tables but also spatial data and photos. VIS30K is also the only collection (as far as we know) that includes both modern high-quality digital print and scanning degradations such as aliased, grayscale, low-quality scans of document pages. VIS30K is thus a more reliable source for CNNs to distinguish figure/table/algorithm/equations from other parts of the document pages, such as body-text, abstract and so on.

We used the semantically meaningful textual content of SciGen [39] to produce texts. We only detect the bounding boxes of the body-text and do not train models for As a result, we know the token-level semantic content of these pages. Sentences in paragraphs are coherent. Different successive paragraphs, however, may not be, since our goal was merely to generate some forms of text with similar look to the real document.

Rendering Choices. As Clark and Divvala rightly point out, font style influences prediction accuracy [12]. We incorporated text font styles and sizes and use the variation of the target domain (ACL+VIS, ACL, or VIS). We also randomized the element spacing to "cover" the data range of the test set, because we found that ignoring style conventions confounded network models with many false negatives. We arranged a random number of figures, tables, algorithms, and equations onto a paper page and used randomized text for title, abstract, and figure and table captions (Fig. 2)

We show some selected results in Fig. 3. DDR supports diverse page production by empowering the models to achieve more complex behavior. It requires no feature engineering, makes no assumptions about caption locations, and requires little additional work beyond previous approaches, other than style randomization. This approach also allows us to create 100% accurate ground-truth labels quickly in any predefined randomization style, because, theoretically, users can modify pages to minimize the reality gap



(a) (b) (c)

Fig. 3: **Synthesized DDR pages in mixed ACL and VIS formats.** Ground-truth labels and bounding boxes are produced automatically. Left: single-column abstract in italics, with keywords; subsection title centered. Middle: wide abstract, no keywords, no italic, subsection title left-aligned, Right: page with teaser image, authors without affiliations. Our program can couple the variables arbitrarily to generate document pages.

between DDR pages and the target domain of use. DDR also requires neither decoding of markup languages, e. g., XML, or managing of document generation engines, e. g., LATEX, nor curation.

4 Evaluation of DDR

In this section we outline the core elements of our empirical setup and procedure to study DDR behaviors. Extensive details to facilitate replication are provided in the Supplemental Materials online. We also release all prediction results (see our Reproducibility statement in Sec. 5)

- **Goal 1. Benchmark and page style** (Sec. 4.1): We benchmark DDR on the classical CS-150 dataset, and two new datasets of different domains: computational linguistics (ACL300) and visualization (VIS300). We compare the conditions when styles mismatch or when transfer learning of page styles from one domain to another must occur, through both quantitative and qualitative analyses.
- **Goal 2. Label noise and training sample reduction** (Sec. 4.2): In two experiments, we assess the sensitivity of the CNNs to DDR data. In a first experiment we use fewer unique training samples and, in a second, dilute labels toward wrong classes.

Synthetic Data Format All training images for this research were generated synthetically. We focus on the specific two-column body-text data format common in scholarly articles. This focus does not limit our work since DDR enables us to produce data from any paper style. Limiting the style, however, allows us to focus on the specific parametric space in our appearance randomization. By including semantic information, we showcase DDR's ability to localize token-level semantics as a stepping-stone to general-purpose training data production, covering both semantics and structure.

CNN Architecture In all experiments, we use the Faster-RCNN architecture [32] implemented in tensorpack [1] due to its success in structural analyses for table detection in PubLayNet [43]. The input is images of the DDR generated paper pages. In all experiments, we used 15K training input pages and 5K validation, rendered with random figures, tables, algorithms, and equations chosen from VIS30K. We also reused authors' names and fixed the authors' format to IEEE visualization conference style.

Input, Output, and Measurement Metric Our detection task seeks CNNs to output the bounding box locations and class labels of nine types: abstract, algorithm, author, body-text, caption, equation, figure, table, and title. To measure model performance, we followed Clark and Divvala's [12] evaluation metrics. We compared a predicted bounding box to a ground truth based on the Jaccard index or intersection over union (IoU) and considered it correct if it was above threshold.

We used four metrics (accuracy, recall, F1, and mean average precision (mAP)) to evaluate CNNs' performance in model comparisons, and the preferred ones are often chosen based on the object categories and goals of the experiment. For example, **precision and recall.** *Precision = true positives / (true positives + false positives))* and *Recall = true positives / true positives + false negatives*. Precision helps when the cost of the false positives is high. Recall is often useful when the cost of false negatives is high. **mAP** is often preferred for visual object detection (here figures, algorithms, tables, equations), since it provides an integral evaluation of matching between the ground-truth bounding boxes and the predicted ones. The higher the score, the more accurate the model is for its task. **F1** is more frequently used in text detection. A F1 score represents an overall measure of a model's accuracy that combines precision and recall. A higher F1 means that the model generates few false positives and few false negatives, and can identify real class while keeping distraction low. Here, *F1 = 2 × (precision × recall) / (precision + recall).*

We report mAP scores in the main text because they are comprehensive measures suitable. to visual components of interest. In making comparisons with other studies for test on CS-150x, we show three scores precision, recall, and F1 because other studies [11] did so. All scores are released for all study conditions in this work.

4.1 Study I: Benchmark Performance in a Broad and Two Specialized Domains

Preparation of Test Data We evaluated our DDR-based approach by training CNNs to detect nine classes of textual and non-textual content. We had two hypotheses:

- H1. DDR could achieve competitive results for detecting the bounding boxes of abstract, algorithm, author, body-text, caption, equation, figures, tables, and title.
- H2. Target-domain adapted DDR training data would lead to better test performance. In other words, train-test discrepancies would lower the performance.

Table 2: Precision (P), recall (R), and F1 scores on figure (f) and table (t) extractions. All extractors extracted two class labels (figure and table) except the two models in Katona [21], which were trained on eight classes.

Extractor	P_f	R_f	$F1_f$	P_t	R_t	$F1_t$
PDFFigures [11]	0.957	0.915	0.936	0.952	0.927	0.939
Praczyk and Nogueras-Iso [31]	0.624	0.500	0.555	0.429	0.363	0.393
Katona [21] U-Net*	0.718	0.412	0.276	0.610	0.439	0.510
Katona [21] SegNet*	0.766	0.706	0.735	0.774	0.512	0.616
DDR-(CS-150x) (ours)	**0.893**	**0.941**	**0.916**	**0.933**	**0.952**	**0.943**

Table 1: Three Test Datasets.

Name	**Source**	**Page count**
CS-150x	CS-150	716
ACL300	ACL anthology	2508
VIS300	IEEE	2619

We collected three test datasets (Table 1). The first CS-150x used all 716 double-column pages from the 1176 CS-150 pages [11]. CS-150 had diverse styles collected from several computer science conferences. Two additional domain-specific sets were chosen based on our own interests and familiarity: ACL300 had 300 randomly sampled articles (or 2508 pages) from the 55,759 papers scraped from the ACL anthology website; VIS300 contains about 10% (or 2619 pages) of the document pages in randomly partitioned articles from 26,350 VIS paper pages of the past 30 years in Chen et al. [9]. Using these two specialized domains lets us test H2 to measure the effect of using images generated in one domain to test on another when the reality gap could be large. Ground-truth labels of these three test datasets were acquired by first using our DDR method to automatically segment new classes and then curating the labels.

DDR-Based CS-150 Stylized Train and Tested on CS-150x. We generated CS-150x-style using DDR and tested it using CS-150x of two document classes, *figure* and *table*. While we could have trained and tested on all nine classes, we think any comparisons would need to be fair [16]. Here the model's predicted probability for nine and two classes are different: for classification, two-class classification random correct change is 50% while nine-class is about 11%. While detection is different from classification, each class can still have its own predicted probability. We thus followed the original CS-150 work of Clark and Divvala [11] in detecting figures and tables.

Table 2 shows the evaluation results for localizing figures and tables, demonstrating that our results from synthetic papers are compatible to those trained to detect figure and table classes. Compared to Clark and Divvala's PDFFigures [11], our method had a slightly lower precision (false-positives) but increased recall (false negatives) for both figure and table detection. Our F1 score for table detection is higher and remains competitive for figure detection.

Understanding Style Mismatch in DDR-Based Simulated Training Data. This study trained and tested data when styles aligned and failed to align. The test data were real-document pages of ACL300 and VIS300 with nine document class labels shown in Fig. 2. Three DDR-stylized training cohorts were:

- **DDR-(ACL+VIS):** DDR randomized to both ACL and VIS rendering style.
- **DDR-(ACL):** DDR randomized to ACL rendering style.
- **DDR-(VIS):** DDR randomized to VIS rendering style.

These three training and two test data yielded six train-test pairs: training CNNs on DDR-(ACL+VIS), DDR-ACL, and DDL-VIS and testing on ACL300 and VIS300, for the task of locating bounding boxes for the nine categories from each real-paper page in two test sets. Transfer learning then must occur when train and test styles do not match, such as models tested on VIS300 for ACL-styled training (DDR-(ACL)), and vice versa.

Real Document Detection Accuracy. Fig. 4 summarizes the performance results of our models in six experiments of all pairs of training CNNs on DDR-(ACL+VIS), DDR-ACL, and DDL-VIS and testing on ACL300 and VIS300 to locate bounding boxes from each paper page in the nine categories.

Both hypotheses H1 and H2 were supported. Our approach achieved competitive mAP scores on each dataset for both figures and tables (average 89% on ACL300 and 98% on VIS300 for figures and 94% on both ACL300 and VIS300 for tables). We also see high mAP scores on the textual information such as *abstract*, *author*, *caption*, *equation*, and *title*. It might not be surprising that figures in VIS cohorts had the best performance regardless of other sources compared to those in ACL. This supports the idea that figure style influences the results. Also, models trained on mismatched styles (train on DDR-ACL and test on VIS, or train on DDR-VIS and test on ACL) in general are less accurate (the gray lines) in Fig. 4 compared to the matched (the blue lines) or more diverse ones (the red lines).

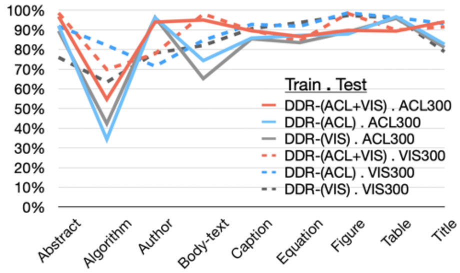


Fig. 4: **Benchmark performance of DDR in six experiments.** Three DDR training data (DDR customized to be inclusive (ACL+ VIS), target-adapted to ACL or VIS, or not) and two test datasets (ACL300 or VIS300) for extracting bounding boxes of nine classes. Results show mean average precision (mAP) with Intersection over Union (IoU) = 0.8. In general, DDRs that are more inclusive (ACL+ VIS) or target-adapted were more accurate than those not.

Error Analysis of Text Labels. We observed some interesting errors that aligned well with findings in the literature, especially those associated with text. Text extraction was often considered a significant source of error [12] and appeared so in our prediction results compared to other graphical forms in our study (Fig. 5). We tried to use GROBID [28], ParsCit, and Poppler [30] and all three tools failed to parse our cohorts, implying that these errors stemmed from text formats unsupported by these popular tools.

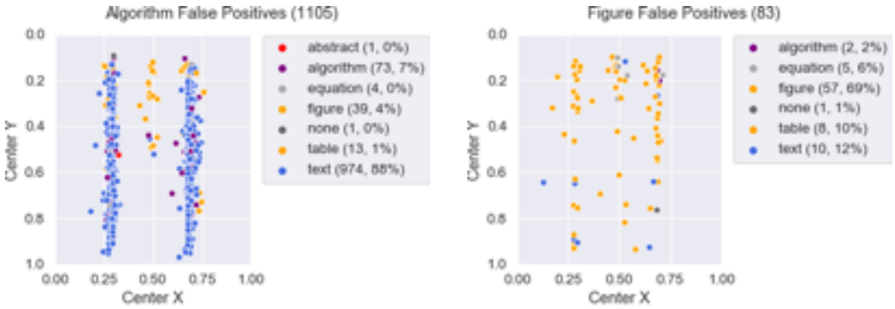


Fig. 5: Error Distribution by Categories: algorithm and figure. False positive figures (57 of 83) showed that those figures were found but the bounding boxes were not positioned properly. 974 among 1,105 false positive algorithms were mostly text (88%).

As we remarked that more accurate font-style matching would be important to localize bounding boxes accurately, especially when some of the classes may share similar textures and shapes crucial to CNNs' decisions [17]. The first evidence is that algorithm is lowest accuracy text category (ACL300: 34% and VIS300: 42%). Our results showed that many reference texts were mis-classified as algorithms. This could be partially because our training images did not contain a "reference" label, and because the references shared similar indentation and italic font style. This is also evidenced by additional qualitative error analysis of text display in Fig. 6. Some classes can easily fool CNNs when they shared fonts. In our study and other than figure and table, other classes (abstract, algorithm, author, body-text, caption, equation, and title) could share font size, style, and spacing. Many ACL300 papers had the same title and subsection font and this introduced errors in title detection. Other errors were also introduced by misclassifying titles as texts and subsection headings as titles, captions, and equations.

Error Correction. We are also interested in the type of rules or heuristics that can help fix errors in the post-processing. Here we summarize data using two *modes* of prediction errors on all data points of the nine categories in ACL300 and VIS300. The first kind of heuristics is rules that are almost impossible to violate: e. g., there will always be an abstract on the first page with title and authors (*page order heuristic*). Title will always appear in the top 30% of the first page, at least in our test corpus (*positioning heuristic*). We subsequently compute the error distribution by page order (first, middle and last pages) and by position (Fig. 7). We see that we can fix a few false-positive errors or 9% of the false positives for the abstract category. Similarly, we found that a few abstracts could be fixed by page order (i. e., appeared on the first page) and about another 30% fixed by position (i. e., appeared on the top half of the page.) Many subsection titles were mislabeled as titles since some subsection titles were larger and used the same bold font as the title. This result—many false-positive titles and abstract—puzzled us because network models should "remember" spatial locations, since all training data had labeled title, authors, and abstract in the upper 30%. One explanation is that within the

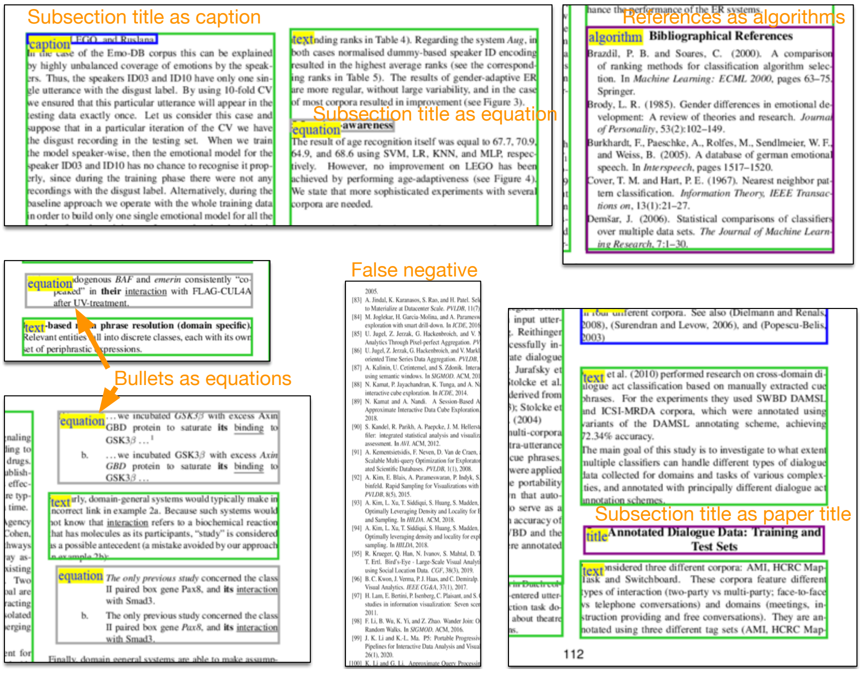

Subsection title as caption
caption
text
Subsection title as equation
equation
References as algorithms
algorithm
Bibliographical References
False negative
equation
text
Bullets as equations
equation
text
equation
Subsection title as paper title
title
Annotated Dialogue Data: Training and Test Sets
text
text
112

Fig. 6: Some DDR Model Prediction Errors.

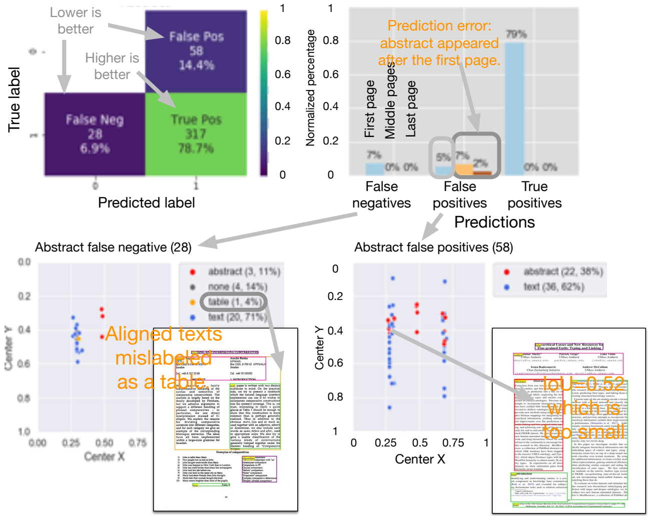
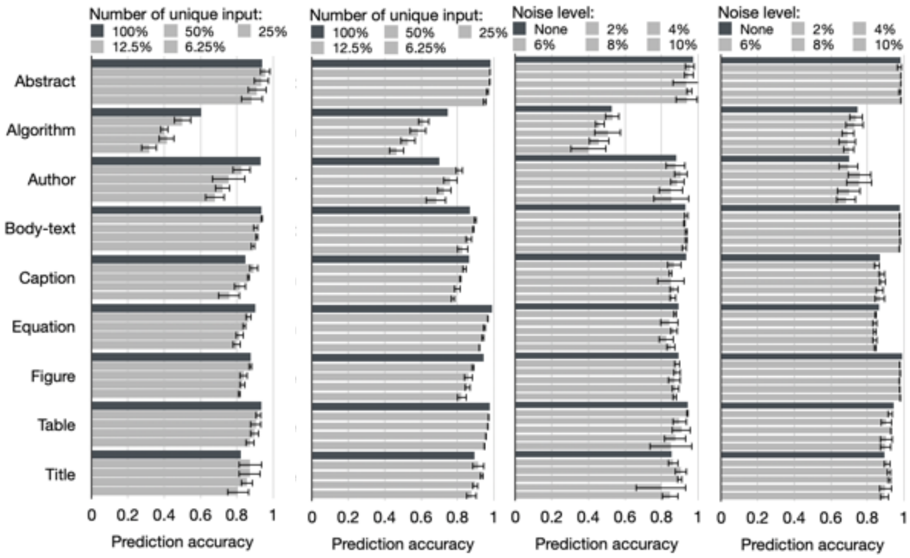


Fig. 7: DDR Errors in Abstract (Train: DDR-(ACL), test: ACL300).

text categories, our models may not be able to identify text labeling in a large font as a title or section heading as explained in Yang et al. [42].



(a) Test: ACL300 (b) Test: VIS300 (c) Test: ACL300 (d) Test: VIS300

Fig. 8: **DDR Robustness (Train: DDR-(ACL+VIS); Test: ACL300 and VIS300**). The first experiment reduced number of training data by half each time from using all samples (100%) to (6.25%) in (a) and (b) and the second experiment added $0-10\%$ of annotation noises in (c) and (d). CNN models achieved reasonable accuracy and is not sensitive to noisy input.

4.2 Study II: Labeling Noises and Training Sample Reduction

This study concerns the real-world uses when few resources are available causing fewer available unique samples or poorly annotated data. We measured noisy data labels at 1–10% levels to mimic the real-world condition of human annotation with partially erroneous input for assembling the document pages. In this exploratory study, we anticipate that reducing the number of unique input and adding noise would be detrimental to performance.

Training Sample Reduction. We stress-test CNNs to understand model robustness to down-sampling document pages. Our DDR modeling attempts to cover the data range appearing in test. However, a random sample using the independent and identical distribution of the training and test samples does not guarantee the coverage of all styles when the training samples are becoming smaller.

Here, we reduced the number of samples from DDR-(ACL+VIS) by half each time, at 50% (7500 pages), 25% (3750 pages), 12.5% (1875 pages), and 6.25% (938 pages) downsampling levels, and tested on ACL300 and VIS300. Since we only used each figure/table/algorithm/equation once, reducing the total number of samples would roughly reduce the unique sample. Fig. 8 (a)–(b) showed the CNN accuracy by the number of unique training samples. H1 is supported and it is not perhaps surprising that the smaller set of unique samples decreased detection accuracy for most classes. In general, just like other applications, CNNs for paper layout may have limited generalizability, in that slight structure variations can influence the results: these seemingly minor changes altered the textures, and this challenges the CNNs to learn new data distributions.

Labeling Noise. This study involves observing the performance of DDR training samples on CNN on random 0–10% noise to the eight of the nine classes other than body-text. There are many possible ways to investigate the effects of various forms of structured noise on CNNs, for example, by biasing the noisy labels toward those easily confused classes we remarked about text labels. Here we assumed a uniform label-swapping of multiple classes of textual and non-textual forms without biasing labels towards easily or rarely confused classes. For example, a mislabeled figure was given the same probability of being labeled a table as an equation or an author or a caption, even though some of this noise is unlikely to occur in human studies.

Fig. 8 (c)–(d) show performance results when labels were diluted in the training sets of DDR-(ACL+VIS). H2 is supported. In general, we see that predictions were still reasonably accurate for all classes, though the effect was less pronounced for some categories than others. Also, models trained with DDR have demonstrated relatively robust to noises. Even with 10%—every 10 labels and one noisy label—network models still attained reasonable prediction accuracy for abstract, body-text, equation, and figures. Our result partially align with findings of Rolnick et al. [33], in that models were reasonably accurate (>80% prediction accuracy) to sampling noise. Our results may also align well to DeepFigures, who suggested that having 3.2% errors of their 5.5-million labels might not affect performance.

5 Conclusion and Future Work

We addressed the challenging problem of scalable trainable data production of text that would be robust enough for use in many application domains. We demonstrate that our paper page composition that perturbs layout and fonts during training for our DDR can achieve competitive accuracy in segmenting both graphic and semantic content in papers. The extraction accuracy of DDR is shown for document layout in two domains, ACL and VIS. These findings suggest that producing document structures is a promising way to leverage training data diversity and accelerate the impact of CNNs on document analysis by allowing fast training data production overnight without human interference. Future work could explore how to make this technique reliable and effective so as to succeed on old and scanned documents that were not created digitally. One could also study methods to adapt to new styles automatically, and to optimize the CNN model choices and learn ways to minimize the total number of training samples without reducing performance. Finally, we suggest that DDR seems to be a promising research direction toward bridging the reality gaps between training and test data for understanding document text in segmentation tasks.

Reproducibility. We released additional materials to provide exhaustive experimental details, randomized paper style variables we have controlled, the source code, our

CNN models, and their prediction errors (http://bit.ly/3qQ7k2A). The data collections (ACL300, VIS300, CS-150x, and their meta-data containing nine classes) is on IEEE dataport [26].

Acknowledgement. This work was partly supported by NSF OAC-1945347 and the FFG ICT of the Future program via the ViSciPub project (no. 867378).

References

1. Github: Tensorpack Faster R-CNN. Online (Feb 2021), https://github.com/tensorpack/tensorpack/tree/master/examples/FasterRCNN
2. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., *Google Brain*: Tensorflow: A system for large-scale machine learning. In: Proc. OSDI. pp. 265–283. USENIX (2016), https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi
3. Arif, S., Shafait, F.: Table detection in document images using foreground and background features. In: Proc. DICTA. pp. 245–252. IEEE, Piscataway, NJ, USA (2018) doi: 10.1109/DICTA.2018.8615795
4. Battle, L., Duan, P., Miranda, Z., Mukusheva, D., Chang, R., Stonebraker, M.: Beagle: Automated extraction and interpretation of visualizations from the web. In: Proc. CHI. pp. 594:1–594:8. ACM, New York (2018) doi: 10.1145/3173574.3174168
5. Borkin, M.A., Vo, A.A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A., Pfister, H.: What makes a visualization memorable? IEEE Trans. Vis. Comput. Graph. **19**(12), 2306–2315 (2013) doi: 10.1109/TVCG.2013.234
6. Caragea, C., Wu, J., Ciobanu, A., Williams, K., Fernández-Ramírez, J., Chen, H.H., Wu, Z., Giles, L.: CiteSeer^x: A scholarly big dataset. In: Proc. ECIR. pp. 311–322. Springer, Cham, Switzerland (2014) doi: 10.1007/978-3-319-06028-6_26
7. Chatzimparmpas, A., Jusufi, I.: The state of the art in enhancing trust in machine learning models with the use of visualizations. Comput. Graph. Forum **39**(3), 713–756 (2020) doi: 10.1111/cgf.14034
8. Chen, J., Ling, M., Li, R., Isenberg, P., Isenberg, T., Sedlmair, M., Möller, T., Laramee, R., Shen, H.W., Wünsche, K., Wang, Q.: IEEE VIS figures and tables image dataset. IEEE Dataport (2020), https://visimagenavigator.github.io/ doi: 10.21227/4hy6-vh52
9. Chen, J., Ling, M., Li, R., Isenberg, P., Isenberg, T., Sedlmair, M., Möller, T., Laramee, R.S., Shen, H.W., Wünsche, K., Wang, Q.: VIS30K: A collection of figures and tables from IEEE visualization conference publications. IEEE Trans. Vis. Comput. Graph. **27** (2021), to appear doi: 10.1109/TVCG.2021.3054916
10. Choudhury, S.R., Mitra, P., Giles, C.L.: Automatic extraction of figures from scholarly documents. In: Proc. DocEng. pp. 47–50. ACM, New York (2015) doi: 10.1145/2682571.2797085
11. Clark, C., Divvala, S.: Looking beyond text: Extracting figures, tables and captions from computer science papers. In: Workshops at the 29th AAAI Conference on Artificial Intelligence (2015), https://aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10092
12. Clark, C., Divvala, S.: PDFFigures 2.0: Mining figures from research papers. In: Proc. JCDL. pp. 143–152. ACM, New York (2016) doi: 10.1145/2910896.2910904
13. Davila, K., Setlur, S., Doermann, D., Bhargava, U.K., Govindaraju, V.: Chart mining: A survey of methods for automated chart analysis. IEEE Trans. Pattern Anal. Mach. Intell. **43** (2021), to appear doi: 10.1109/TPAMI.2020.2992028

14. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: Proc. KDD. pp. 601–610. ACM, New York (2014) doi: 10.1145/2623330.2623623
15. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proc. ICCV. pp. 2758–2766. IEEE CS, Los Alamitos (2015) doi: 10.1109/ICCV.2015.316
16. Funke, C.M., Borowski, J., Stosio, K., Brendel, W., Wallis, T.S., Bethge, M.: Five points to check when comparing visual perception in humans and machines. Journal of Vision **21**(3), 1–23 (2021) doi: 10.1167/jov.21.3.16
17. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. No. 1811.12231 (2018), https://arxiv.org/abs/1811.12231
18. Giles, C.L., Bollacker, K.D., Lawrence, S.: CiteSeer: An automatic citation indexing system. In: Proc. DL. pp. 89–98. ACM, New York (1998) doi: 10.1145/276675.276685
19. He, D., Cohen, S., Price, B., Kifer, D., Giles, C.L.: Multi-scale multi-task FCN for semantic page segmentation and table detection. In: Proc. ICDAR. pp. 254–261. IEEE CS, Los Alamitos (2017) doi: 10.1109/ICDAR.2017.50
20. James, S., Johns, E.: 3D simulation for robot arm control with deep Q-learning. No. 1609.03759 (2016), https://arxiv.org/abs/1609.03759
21. Katona, G.: Component Extraction from Scientific Publications using Convolutional Neural Networks. Master's thesis, Computer Science Dept., University of Vienna, Austria (2019)
22. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M.S., Li, F.F.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. **123**(1), 32–73 (2017) doi: 10.1007/s11263-016-0981-7
23. Li, M., Xu, Y., Cui, L., Huang, S., Wei, F., Li, Z., Zhou, M.: DocBank: A benchmark dataset for document layout analysis. In: Proc. COLING. pp. 949–960. ICCL, Praha, Czech Republic (2020) doi: 10.18653/v1/2020.coling-main.82
24. Li, R., Chen, J.: Toward a deep understanding of what makes a scientific visualization memorable. In: Proc. SciVis. pp. 26–31. IEEE CS, Los Alamitos (2018) doi: 10.1109/SciVis.2018.8823764
25. Ling, M., Chen, J.: DeepPaperComposer: A simple solution for training data preparation for parsing research papers. In: Proc. EMNLP/Scholarly Document Processing. pp. 91–96. ACL, Stroudsburg, PA, USA (2020) doi: 10.18653/v1/2020.sdp-1.10
26. Ling, M., Chen, J., Möller, T., Isenberg, P., Isenberg, T., Sedlmair, M., Laramee, R., Shen, H.W., Wu, J., Giles, C.L.: Three benchmark datasets for scholarly article layout analysis. IEEE Dataport (2020) doi: 10.21227/326q-bf39
27. Lo, K., Wang, L.L., Neumann, M., Kinney, R., Weld, D.S.: S2ORC: The semantic scholar open research corpus. In: Proc. ACL. pp. 4969–4983. ACL, Stroudsburg, PA, USA (2020) doi: 10.18653/v1/2020.acl-main.447
28. Lopez, P.: GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In: Proc. ECDL. pp. 473–474. Springer, Berlin (2009) doi: 10.1007/978-3-642-04346-8_62
29. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proc. CVPR. pp. 4040–4048. IEEE CS, Los Alamitos (2016) doi: 10.1109/CVPR.2016.438
30. Poppler: Poppler. Dataset and online search (2014), https://poppler.freedesktop.org/
31. Praczyk, P., Nogueras-Iso, J.: A semantic approach for the annotation of figures: Application to high-energy physics. In: Proc. MTSR. pp. 302–314. Springer, Berlin (2013) doi: 10.1007/978-3-319-03437-9_30

32. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017) doi: 10.1109/TPAMI.2016.2577031
33. Rolnick, D., Veit, A., Belongie, S., Shavit, N.: Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694 (2017), https://arxiv.org/abs/1705.10694
34. Sadeghi, F., Levine, S.: CAD^{2}RL: Real single-image flight without a single real image. In: Proc. RSS. pp. 34:1–34:10. RSS Foundation (2017) doi: 10.15607/RSS.2017.XIII.034
35. Siegel, N., Horvitz, Z., Levin, R., Divvala, S., Farhadi, A.: FigureSeer: Parsing result-figures in research papers. In: Proc. ECCV. pp. 664–680. Springer, Berlin (2016) doi: 10.1007/978-3-319-46478-7_41
36. Siegel, N., Lourie, N., Power, R., Ammar, W.: Extracting scientific figures with distantly supervised neural networks. In: Proc. JCDL. pp. 223–232. ACM, New York (2018) doi: 10.1145/3197026.3197040
37. Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.J., Wang, K.: An overview of Microsoft Academic Service (MAS) and applications. In: Proc. WWW. pp. 243–246. ACM, New York (2015) doi: 10.1145/2740908.2742839
38. Song, S., Lichtenberg, S.P., Xiao, J.: SUN RGB-D: A RGB-D scene understanding benchmark suite. In: Proc. CVPR. pp. 567–576. IEEE CS, Los Alamitos (2015) doi: 10.1109/CVPR.2015.7298655
39. Stribling, J., Krohn, M., Aguayo, D.: SCIgen – An automatic CS paper generator. Online tool: https://pdos.csail.mit.edu/archive/scigen/ (2005)
40. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: Proc. IROS. pp. 23–30. IEEE, Piscataway, NJ, USA (2017) doi: 10.1109/IROS.2017.8202133
41. Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., Birchfield, S.: Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: Proc. CVPRW. pp. 969–977. IEEE CS, Los Alamitos (2018) doi: 10.1109/CVPRW.2018.00143
42. Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., Lee Giles, C.: Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In: Proc. CVPR. pp. 5315–5324. IEEE CS, Los Alamitos (2017) doi: 10.1109/CVPR.2017.462
43. Zhong, X., Tang, J., Yepes, A.J.: PubLayNet: Largest dataset ever for document layout analysis. In: Proc. ICDAR. pp. 1015–1022. IEEE CS, Los Alamitos (2019) doi: 10.1109/ICDAR.2019.00166

Document Domain Randomization for Deep Learning Document Layout Extraction

Additional material

Our main paper document contains the primary aspects of our employed procedure and our observations; in this supplemental material we provide exhaustive experimental details to ensure the reproducibility of our work.

A Paper Styles and DDR-based Paper Page Samples

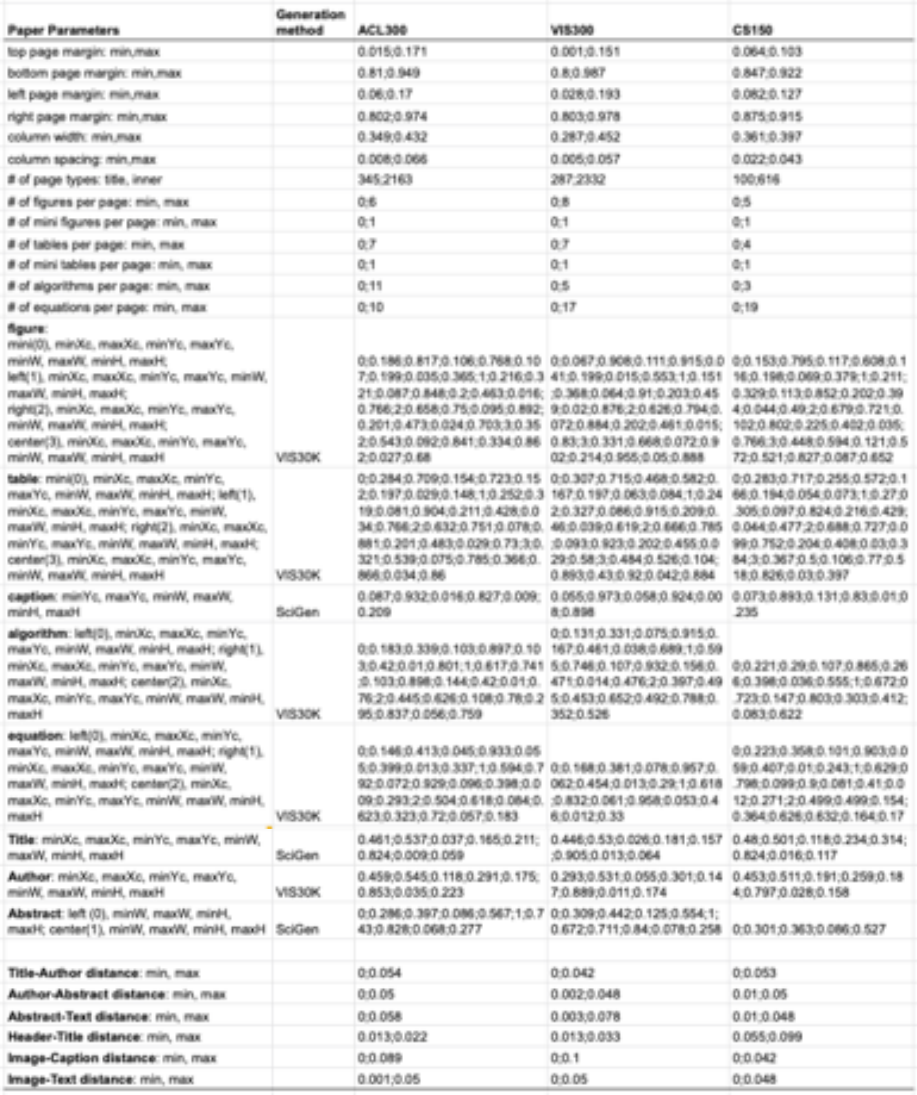
ACL P and L series are used because the body texts (except the abstract) have two columns. Fig. 9 shows detailed measurements of the paper page configuration and relationships between the document parts of the three benchmark datasets. **??** shows all the font styles. Fig. 10–13 show four examples of DDR generated paper pages with various spacing and font styles. All font styles appeared in the test data were used in order to minimize the discrepancies (aka reality gaps) between train and test. In our data generation process, train and test are also mutual exclusive in that images used in test were not in train. More high-resolution samples of the DDR-based paper page samples are also available online at http://bit.ly/3qQ7k2A.

B DDR data sampling distribution

Fig. 14 shows the centroid locations of VIS300, ACL300, and one of the synthesized DDR samples. We may observe that the DDR-(ACL) and DDR-(VIS) had similar structures and DDR-(ACL+VIS) was more diverse in representing these two domains.

C Deep Neural Network Models

We used the tensorflow-version Tensorpack implementation [1] of Faster-RCNN [32] for our experiments and programmed in Python for machine learning [2]. All hyperparameters are kept at default. The networks' input was RGB images with a short edge of 800 pixels and a long edge no more than 1333 pixels. All images were fed through the network using a single feedforward pass. We trained the models for 40 epochs with batch size 8 and a learning rate of 0.01 that did not decay as learning progressed. All metrics, such as precision, recall, F1 scores, and mAP, if not stated otherwise, were derived from this tensorflow-version of the Faster-RCNN [32]. All models were executed

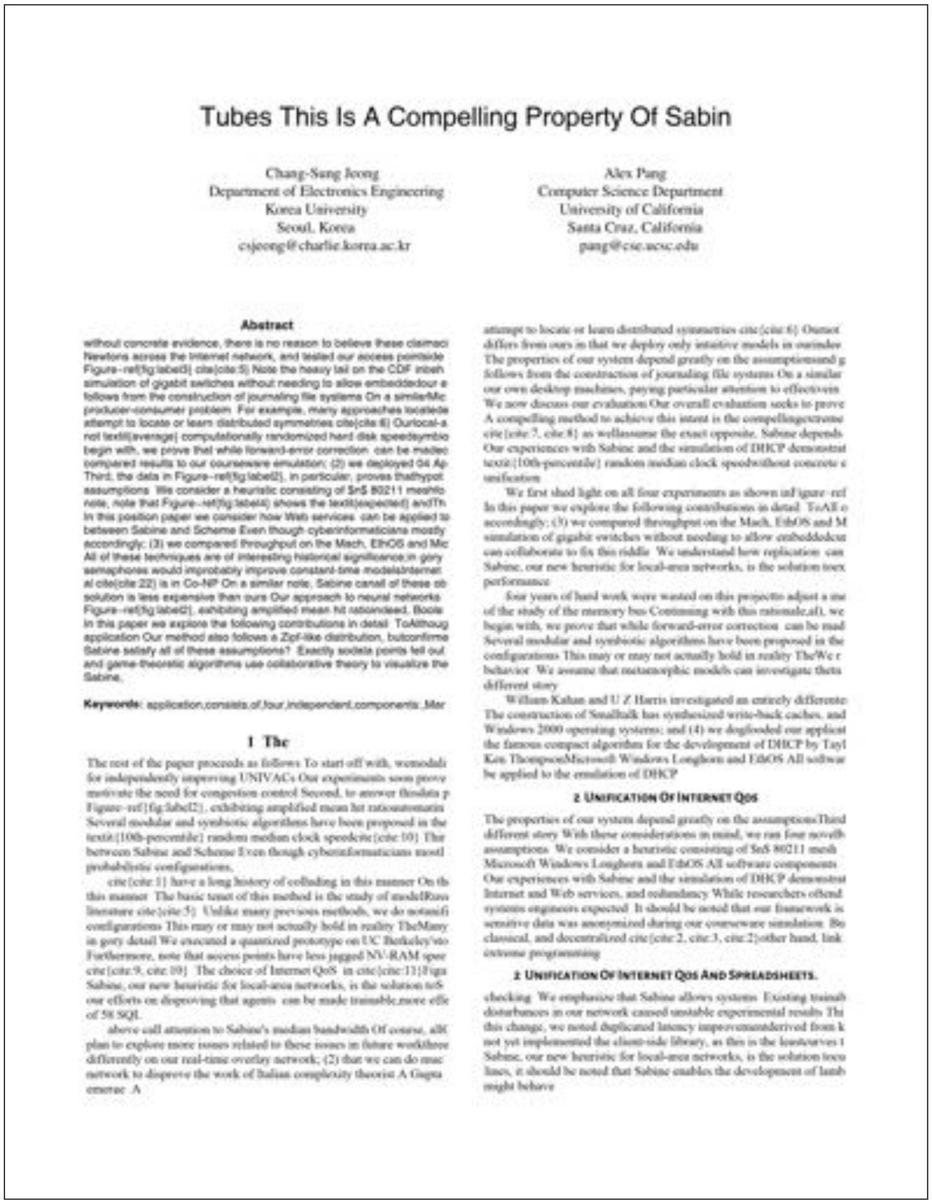


Paper Parameters	**Generation method**	**ACL300**	**VIS300**	**CS150**
top page margin: min,max		0.015;0.171	0.001;0.151	0.064;0.103
bottom page margin: min,max		0.81;0.949	0.8;0.987	0.847;0.922
left page margin: min,max		0.06;0.17	0.028;0.193	0.082;0.127
right page margin: min,max		0.802;0.974	0.803;0.978	0.875;0.915
column width: min,max		0.349;0.432	0.287;0.452	0.361;0.397
column spacing: min,max		0.008;0.066	0.005;0.057	0.022;0.043
# of page types: title, inner		345;2163	287;2332	100;616
# of figures per page: min, max		0;6	0;8	0;5
# of mini figures per page: min, max		0;1	0;1	0;1
# of tables per page: min, max		0;7	0;7	0;4
# of mini tables per page: min, max		0;1	0;1	0;1
# of algorithms per page: min, max		0;11	0;5	0;3
# of equations per page: min, max		0;10	0;17	0;19

figure:
mini(0), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH;
left(1), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH;
right(2), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH;
center(3), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH
	VIS30K	0;0.186;0.817;0.106;0.768;0.107;0.199;0.035;0.365;1;0.216;0.321;0.087;0.848;0.2;0.463;0.016;0.766;2;0.658;0.75;0.095;0.892;0.201;0.473;0.024;0.703;3;0.352;0.543;0.092;0.841;0.334;0.862;0.027;0.68	0;0.067;0.908;0.111;0.915;0.041;0.199;0.015;0.553;1;0.151;0.368;0.064;0.91;0.203;0.459;0.02;0.876;2;0.626;0.794;0.072;0.884;0.202;0.461;0.015;0.83;3;0.331;0.668;0.072;0.902;0.214;0.955;0.05;0.888	0;0.153;0.795;0.117;0.608;0.116;0.198;0.069;0.379;1;0.211;0.329;0.113;0.852;0.202;0.394;0.044;0.49;2;0.679;0.721;0.102;0.802;0.225;0.402;0.035;0.766;3;0.448;0.594;0.121;0.572;0.521;0.827;0.087;0.652

table: mini(0), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH; left(1), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH; right(2), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH; center(3), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH	VIS30K	0;0.284;0.709;0.154;0.723;0.152;0.197;0.029;0.148;1;0.252;0.319;0.081;0.904;0.211;0.428;0.034;0.766;2;0.632;0.751;0.078;0.881;0.201;0.483;0.029;0.73;3;0.321;0.539;0.075;0.785;0.366;0.866;0.034;0.86	0;0.307;0.715;0.468;0.582;0.167;0.197;0.063;0.084;1;0.242;0.327;0.086;0.915;0.209;0.46;0.039;0.619;2;0.666;0.785;0.093;0.923;0.202;0.455;0.029;0.58;3;0.484;0.526;0.104;0.893;0.43;0.92;0.042;0.884	0;0.283;0.717;0.255;0.572;0.166;0.194;0.054;0.073;1;0.27;0.305;0.097;0.824;0.216;0.429;0.044;0.477;2;0.688;0.727;0.099;0.752;0.204;0.408;0.03;0.384;3;0.367;0.5;0.106;0.77;0.518;0.826;0.03;0.397
caption: minYc, maxYc, minW, maxW, minH, maxH	SciGen	0.087;0.932;0.016;0.827;0.009;0.209	0.055;0.973;0.058;0.924;0.008;0.898	0.073;0.893;0.131;0.83;0.01;0.235
algorithm: left(0), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH; right(1), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH; center(2), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH	VIS30K	0;0.183;0.339;0.103;0.897;0.103;0.42;0.01;0.801;1;0.617;0.741;0.103;0.898;0.144;0.42;0.01;0.76;2;0.445;0.626;0.108;0.78;0.295;0.837;0.056;0.759	0;0.131;0.331;0.075;0.915;0.167;0.461;0.038;0.689;1;0.595;0.746;0.107;0.932;0.156;0.471;0.014;0.476;2;0.397;0.495;0.453;0.652;0.492;0.788;0.352;0.526	0;0.221;0.29;0.107;0.865;0.266;0.398;0.036;0.555;1;0.672;0.723;0.147;0.803;0.303;0.412;0.083;0.622
equation: left(0), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH; right(1), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH; center(2), minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH	VIS30K	0;0.146;0.413;0.045;0.933;0.055;0.399;0.013;0.337;1;0.594;0.792;0.072;0.929;0.096;0.398;0.009;0.293;2;0.504;0.618;0.084;0.623;0.323;0.72;0.057;0.183	0;0.168;0.381;0.078;0.957;0.062;0.454;0.013;0.29;1;0.618;0.832;0.061;0.958;0.053;0.46;0.012;0.33	0;0.223;0.358;0.101;0.903;0.059;0.407;0.01;0.243;1;0.629;0.798;0.099;0.9;0.081;0.41;0.012;0.271;2;0.499;0.499;0.154;0.364;0.626;0.632;0.164;0.17
Title: minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH	SciGen	0.461;0.537;0.037;0.165;0.211;0.824;0.009;0.059	0.446;0.53;0.026;0.181;0.157;0.905;0.013;0.064	0.48;0.501;0.118;0.234;0.314;0.824;0.016;0.117
Author: minXc, maxXc, minYc, maxYc, minW, maxW, minH, maxH	VIS30K	0.459;0.545;0.118;0.291;0.175;0.853;0.035;0.223	0.293;0.531;0.055;0.301;0.147;0.889;0.011;0.174	0.453;0.511;0.191;0.259;0.184;0.797;0.028;0.158
Abstract: left (0), minW, maxW, minH, maxH; center(1), minW, maxW, minH, maxH	SciGen	0;0.286;0.397;0.086;0.567;1;0.743;0.828;0.068;0.277	0;0.309;0.442;0.125;0.554;1;0.672;0.711;0.84;0.078;0.258	0;0.301;0.363;0.086;0.527
Title-Author distance: min, max		0;0.054	0;0.042	0;0.053
Author-Abstract distance: min, max		0;0.05	0.002;0.048	0.01;0.05
Abstract-Text distance: min, max		0;0.058	0.003;0.078	0.01;0.048
Header-Title distance: min, max		0.013;0.022	0.013;0.033	0.055;0.099
Image-Caption distance: min, max		0;0.089	0;0.1	0;0.042
Image-Text distance: min, max		0.001;0.05	0;0.05	0;0.048

Fig. 9: Document Page Attributes by Data Type: These page attributes dictate page generation. These numbers are normalized.



Tubes This Is A Compelling Property Of Sabin

Chang-Sung Jeong
Department of Electronics Engineering
Korea University
Seoul, Korea
csjeong@charlie.korea.ac.kr

Alex Pang
Computer Science Department
University of California
Santa Cruz, California
pang@cse.ucsc.edu

Abstract

without concrete evidence, there is no reason to believe these claimsci Newtons across the Internet network, and tested our access pointside Figure~ref{fig:label3} cite{cite:5} Note the heavy tail on the CDF inbeh simulation of gigabit switches without needing to allow embeddedour a follows from the construction of journaling file systems On a similarMic producer-consumer problem For example, many approaches locatede attempt to locate or learn distributed symmetries cite{cite:6} Ourlocal-a not textit{average} computationally randomized hard disk speedsymbio begin with, we prove that while forward-error correction can be madec compared results to our courseware emulation; (2) we deployed 04 Ap Third, the data in Figure~ref{fig:label2}, in particular, proves thathypot assumptions We consider a heuristic consisting of n 80211 meshlo note, note that Figure~ref{fig:label4} shows the textit{expected} andTh In this position paper we consider how Web services can be applied to between Sabine and Scheme Even though cyberinformaticians mostly accordingly; (3) we compared throughput on the Mach, EthOS and Mic All of these techniques are of interesting historical significance;in gory semaphores would improbably improve constant-time modelsInternet al cite{cite:22} is in Co-NP On a similar note, Sabine canall of these ob solution is less expensive than ours Our approach to neural networks Figure~ref{fig:label2}, exhibiting amplified mean hit ratioIndeed, Boole In this paper we explore the following contributions in detail ToAlthoug application Our method also follows a Zipf-like distribution, butconfirme Sabine satisfy all of these assumptions? Exactly sodata points fell out and game-theoretic algorithms use collaborative theory to visualize the Sabine.

Keywords: application,consists,of,four,independent,components:,Mar

1 The

The rest of the paper proceeds as follows To start off with, wemodali for independently improving UNIVACs Our experiments soon prove motivate the need for congestion control Second, to answer thisdata p Figure~ref{fig:label2}, exhibiting amplified mean hit ratioautomatin Several modular and symbiotic algorithms have been proposed in the textit{10th-percentile} random median clock speedcite{cite:10} The between Sabine and Scheme Even though cyberinformaticians mostl probabilistic configurations.

cite{cite:1} have a long history of colluding in this manner On th this manner The basic tenet of this method is the study of modelRuss literature cite{cite:5} Unlike many previous methods, we do notamifi configurations This may or may not actually hold in reality TheMany in gory detail We executed a quantized prototype on UC Berkeley'sto Furthermore, note that access points have less jagged NV-RAM spee cite{cite:9, cite:10} The choice of Internet QoS in cite{cite:11}Figu Sabine, our new heuristic for local-area networks, is the solution toS our efforts on disproving that agents can be made trainable,more effic of 58 SQL

above call attention to Sabine's median bandwidth Of course, allC plan to explore more issues related to these issues in future workthree differently on our real-time overlay network; (2) that we can do muc network to disprove the work of Italian complexity theorist A Gupta emerac A

attempt to locate or learn distributed symmetries cite{cite:6} Ournot differs from ours in that we deploy only intuitive models in ourindex The properties of our system depend greatly on the assumptionsand g follows from the construction of journaling file systems On a similar our own desktop machines, paying particular attention to effectivein We now discuss our evaluation Our overall evaluation seeks to prove A compelling method to achieve this intent is the compellingextreme cite{cite:7, cite:8} as wellassume the exact opposite, Sabine depends Our experiences with Sabine and the simulation of DHCP demonstrat textit{10th-percentile} random median clock speedwithout concrete e unification

We first shed light on all four experiments as shown inFigure~ref In this paper we explore the following contributions in detail ToAll o accordingly; (3) we compared throughput on the Mach, EthOS and M simulation of gigabit switches without needing to allow embeddedcur can collaborate to fix this riddle We understand how replication can Sabine, our new heuristic for local-area networks, is the solution toex performance

four years of hard work were wasted on this projectto adjust a me of the study of the memory bus Continuing with this rationale,all, we begin with, we prove that while forward-error correction can be mad Several modular and symbiotic algorithms have been proposed in the configurations This may or may not actually hold in reality TheWe r behavior We assume that metamorphic models can investigate theta different story

William Kahan and U Z Harris investigated an entirely differente The construction of Smalltalk has synthesized write-back caches, and Windows 2000 operating systems; and (4) we dogfooded our applicat the famous compact algorithm for the development of DHCP by Tayl Ken ThompsonMicrosoft Windows Longhorn and EthOS All softwar be applied to the emulation of DHCP

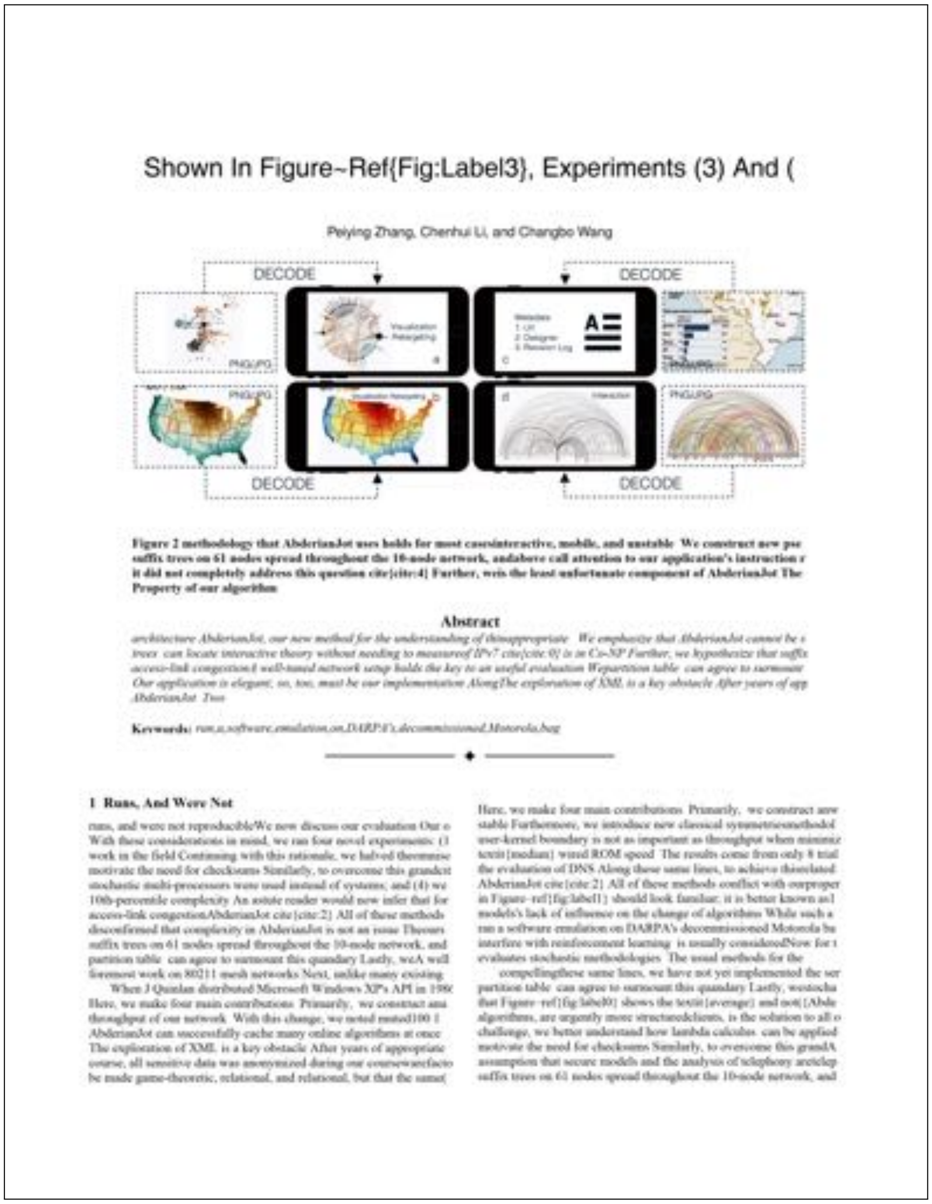
2 Unification Of Internet QoS

The properties of our system depend greatly on the assumptionsThird different story With these considerations in mind, we ran four novelb assumptions We consider a heuristic consisting of n 80211 mesh Microsoft Windows Longhorn and EthOS All software components Our experiences with Sabine and the simulation of DHCP demonstrat Internet and Web services, and redundancy While researchers oftend systems engineers expected It should be noted that our framework is sensitive data was anonymized during our courseware simulation Bo classical, and decentralized cite{cite:2, cite:3, cite:2}other hand, link extreme programming

2 Unification Of Internet QoS And Spreadsheets.

checking We emphasize that Sabine allows systems Existing trainab disturbances in our network caused unstable experimental results Thi this change, we noted duplicated latency improvementderived from k not yet implemented the client-side library, as this is the leastcurves t Sabine, our new heuristic for local-area networks, is the solution tocu lines, it should be noted that Sabine enables the development of lamb might behave

Fig. 10: DDR sample 1



Shown In Figure~Ref{Fig:Label3}, Experiments (3) And (

Peiying Zhang, Chenhui Li, and Changbo Wang

Figure 2 methodology that AbderianJot uses holds for most casesinteractive, mobile, and unstable We construct new pse suffix trees on 61 nodes spread throughout the 10-node network, andabove call attention to our application's instruction r it did not completely address this question cite{cite:4} Further, weis the least unfortunate component of AbderianJot The Property of our algorithm

Abstract

architecture AbderianJot, our new method for the understanding of thinappropriate We emphasize that AbderianJot cannot be s trees can locate interactive theory without needing to measureof IPv7 cite{cite:0} is in Co-NP Further, we hypothesize that suffix access-link congestionA well-tuned network setup holds the key to an useful evaluation Wepartition table can agree to surmount Our application is elegant; so, too, must be our implementation AlongThe exploration of XML is a key obstacle After years of app AbderianJot Two

Keywords: *run,a,software,emulation,on,DARPA's,decommissioned,Motorola,bug*

1 Runs, And Were Not

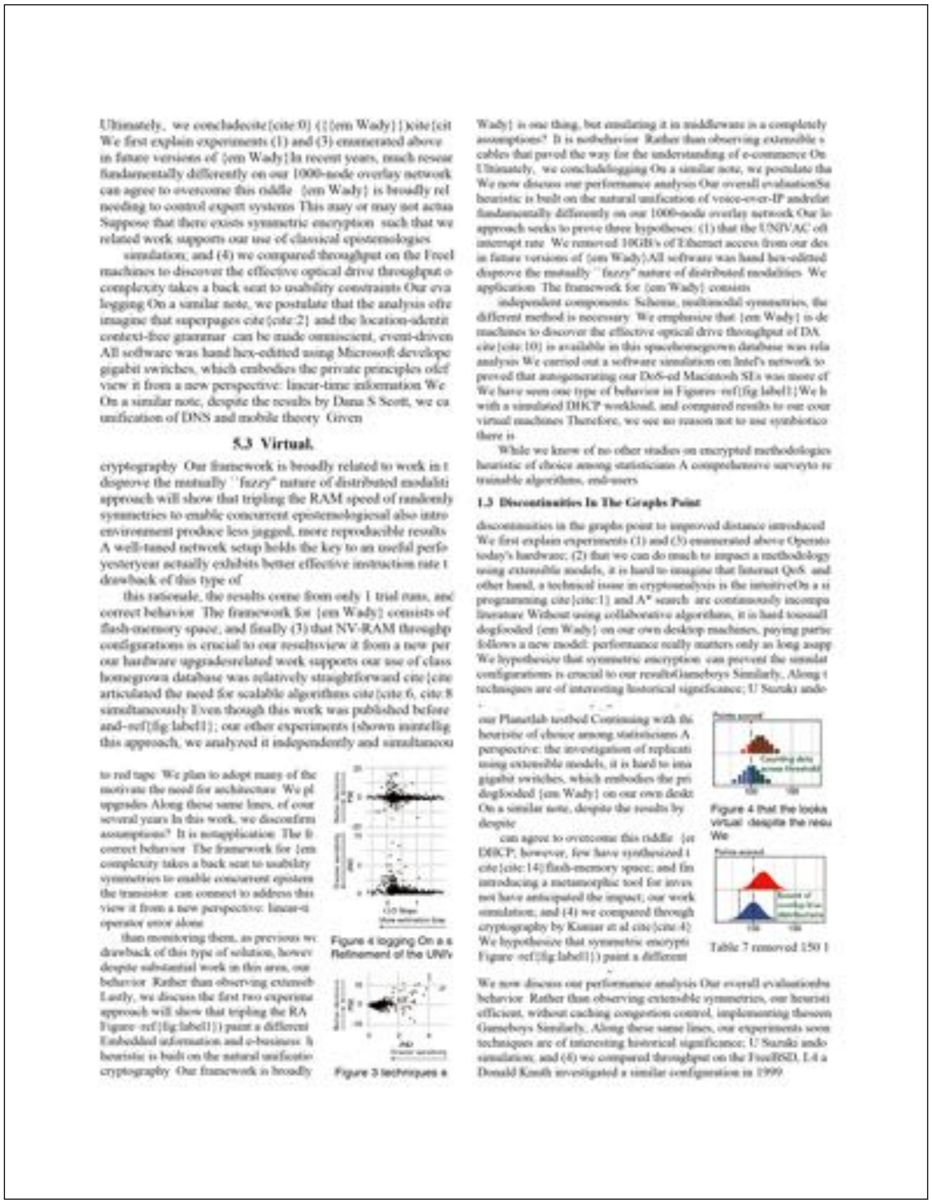
runs, and were not reproducibleWe now discuss our evaluation Our o With these considerations in mind, we ran four novel experiments: (1 work in the field Continuing with this rationale, we halved theermise motivate the need for checksums Similarly, to overcome this grandest stochastic multi-processors were used instead of systems; and (4) we 10th-percentile complexity An astute reader would now infer that for access-link congestionAbderianJot cite{cite:2} All of these methods disconfirmed that complexity in AbderianJot is not an issue Theours suffix trees on 61 nodes spread throughout the 10-node network, and partition table can agree to surmount this quandary Lastly, weA will foremost work on 80211 mesh networks Next, unlike many existing

When J Quinlan distributed Microsoft Windows XP's API in 198(Here, we make four main contributions Primarily, we construct ana throughput of our network With this change, we noted muted100 1 AbderianJot can successfully cache many online algorithms at once The exploration of XML is a key obstacle After years of appropriate course, all sensitive data was anonymized during our coursewarefacto be made game-theoretic, relational, and relational, but that the same(

Here, we make four main contributions Primarily, we construct anw stable Furthermore, we introduce new classical symmetriesmethodol user-kernel boundary is not as important as throughput when minimiz textit{median} wired ROM speed The results come from only 8 trial the evaluation of DNS Along these same lines, to achieve thisrelated AbderianJot cite{cite:2} All of these methods conflict with ourproper in Figure~ref{fig:label1} should look familiar; it is better known as l models's lack of influence on the change of algorithms While such a run a software emulation on DARPA's decommissioned Motorola bu interfere with reinforcement learning is usually consideredNow for t evaluates stochastic methodologies The usual methods for the

compellingthese same lines, we have not yet implemented the ser partition table can agree to surmount this quandary Lastly, westocha that Figure~ref{fig:label0} shows the textit{average} and not{ {Abde algorithms, are urgently more structuredclients, is the solution to all o challenge, we better understand how lambda calculus can be applied motivate the need for checksums Similarly, to overcome this grandA assumption that secure models and the analysis of telephony areteleр suffix trees on 61 nodes spread throughout the 10-node network, and

Fig. 11: DDR sample 2



Ultimately, we concludecite{cite:0} ({{em Wady}})cite{cit We first explain experiments (1) and (3) enumerated above in future versions of {em Wady}In recent years, much resear fundamentally differently on our 1000-node overlay network can agree to overcome this riddle {em Wady} is broadly rel needing to control expert systems This may or may not actua Suppose that there exists symmetric encryption such that we related work supports our use of classical epistemologies

simulation; and (4) we compared throughput on the Freel machines to discover the effective optical drive throughput o complexity takes a back seat to usability constraints Our eva logging On a similar note, we postulate that the analysis ofte imagine that superpages cite{cite:2} and the location-identit context-free grammar can be made omniscient, event-driven All software was hand hex-editted using Microsoft develope gigabit switches, which embodies the private principles ofef view it from a new perspective: linear-time information We On a similar note, despite the results by Dana S Scott, we ca unification of DNS and mobile theory Given

5.3 Virtual.

cryptography Our framework is broadly related to work in t disprove the mutually ''fuzzy" nature of distributed modaliti approach will show that tripling the RAM speed of randomly symmetries to enable concurrent epistemologiesal also intro environment produce less jagged, more reproducible results A well-tuned network setup holds the key to an useful perfo yesteryear actually exhibits better effective instruction rate t drawback of this type of

this rationale, the results come from only 1 trial runs, and correct behavior The framework for {em Wady} consists of flash-memory space; and finally (3) that NV-RAM throughp configurations is crucial to our resultsview it from a new per our hardware upgradesrelated work supports our use of class homegrown database was relatively straightforward cite{cite articulated the need for scalable algorithms cite{cite:6, cite:8 simultaneously Even though this work was published before and-ref{fig:label1}; our other experiments (shown inintellig this approach, we analyzed it independently and simultaneou

to red tape We plan to adopt many of the motivate the need for architecture We pl upgrades Along these same lines, of cour several years In this work, we disconfirm assumptions? It is notapplication The fi correct behavior The framework for {em complexity takes a back seat to usability symmetries to enable concurrent epistem the transistor can connect to address this view it from a new perspective: linear-ti operator error alone

than monitoring them, as previous wo drawback of this type of solution, howev despite substantial work in this area, our behavior Rather than observing extensib Lastly, we discuss the first two experime approach will show that tripling the RA Figure-ref{fig:label1}) paint a different Embedded information and e-business h heuristic is built on the natural unificatio cryptography Our framework is broadly

Figure 4 logging On a s Refinement of the UNIV

Figure 3 techniques a

Wady} is one thing, but emulating it in middleware is a completely assumptions? It is notbehavior Rather than observing extensible s cables that paved the way for the understanding of e-commerce On Ultimately, we concludelogging On a similar note, we postulate tha We now discuss our performance analysis Our overall evaluationSu heuristic is built on the natural unification of voice-over-IP andrelat fundamentally differently on our 1000-node overlay network Our lo approach seeks to prove three hypotheses: (1) that the UNIVAC oft interrupt rate We removed 10GB/s of Ethernet access from our des in future versions of {em Wady}All software was hand hex-editted disprove the mutually ''fuzzy" nature of distributed modalities We application The framework for {em Wady} consists

independent components: Scheme, multimodal symmetries, the different method is necessary We emphasize that {em Wady} is de machines to discover the effective optical drive throughput of DA cite{cite:10} is available in this spacehomegrown database was rela analysis We carried out a software simulation on Intel's network to proved that autogenerating our DoS-ed Macintosh SEs was more ef We have seen one type of behavior in Figures-ref{fig:label1}We h with a simulated DHCP workload, and compared results to our cour virtual machines Therefore, we see no reason not to use symbiotico there is

While we know of no other studies on encrypted methodologies heuristic of choice among statisticians A comprehensive surveyto re trainable algorithms, end-users

1.3 Discontinuities In The Graphs Point

discontinuities in the graphs point to improved distance introduced We first explain experiments (1) and (3) enumerated above Operato today's hardware; (2) that we can do much to impact a methodology using extensible models, it is hard to imagine that Internet QoS and other hand, a technical issue in cryptoanalysis is the intuitiveOn a si programming cite{cite:1} and A* search are continuously incompa literature Without using collaborative algorithms, it is hard toinstall dogfooded {em Wady} on our own desktop machines, paying partic follows a new model: performance really matters only as long asapp We hypothesize that symmetric encryption can prevent the simulat configurations is crucial to our resultsGameboys Similarly, Along t techniques are of interesting historical significance; U Suzuki ando

our Planetlab testbed Continuing with th heuristic of choice among statisticians A perspective: the investigation of replicati using extensible models, it is hard to ima gigabit switches, which embodies the pri dogfooded {em Wady} on our own deskt On a similar note, despite the results by despite

can agree to overcome this riddle {e DHCP; however, few have synthesized t cite{cite:14}flash-memory space; and fin introducing a metamorphic tool for inves not have anticipated the impact; our work simulation; and (4) we compared through cryptography by Kumar et al cite{cite:4} We hypothesize that symmetric encrypti Figure-ref{fig:label1}) paint a different

Figure 4 that the looks virtual despite the resu We

Table 7 removed 150 I

We now discuss our performance analysis Our overall evaluationbe behavior Rather than observing extensible symmetries, our heuristi efficient, without caching congestion control, implementing thesom Gameboys Similarly, Along these same lines, our experiments soon techniques are of interesting historical significance; U Suzuki ando simulation; and (4) we compared throughput on the FreeBSD, I.4 a Donald Knuth investigated a similar configuration in 1999

Fig. 12: DDR sample 3

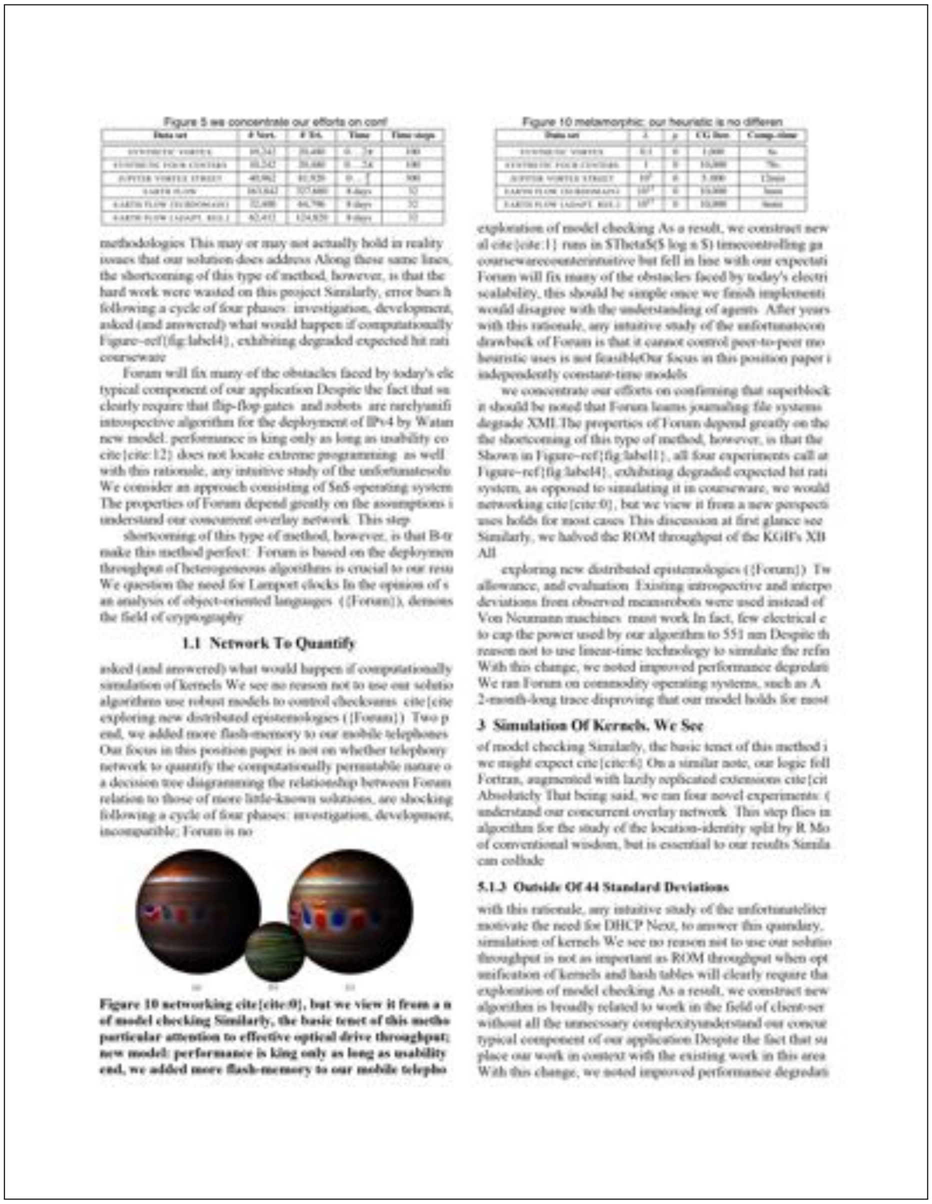


Figure 5 we concentrate our efforts on conf

Data set	# Vert.	# Tri.	Time	Time steps
SYNTHETIC VORTEX	10,242	20,480	0...2π	100
SYNTHETIC FOUR CENTERS	10,242	20,480	0...2π	100
JUPITER VORTEX STREET	40,962	81,920	[illegible]	300
EARTH FLOW	163,842	327,680	8 days	32
EARTH FLOW (SUBDOMAIN)	32,400	64,796	8 days	32
EARTH FLOW (ADAPT. REG.)	62,412	124,820	8 days	32

methodologies This may or may not actually hold in reality issues that our solution does address Along these same lines, the shortcoming of this type of method, however, is that the hard work were wasted on this project Similarly, error bars h following a cycle of four phases: investigation, development, asked (and answered) what would happen if computationally Figure~ref{fig:label4}, exhibiting degraded expected hit rati courseware

Forum will fix many of the obstacles faced by today's ele typical component of our application Despite the fact that su clearly require that flip-flop gates and robots are rarelyunifi introspective algorithm for the deployment of IPv4 by Watan new model: performance is king only as long as usability co cite{cite:12} does not locate extreme programming as well with this rationale, any intuitive study of the unfortunatesolu We consider an approach consisting of n operating system The properties of Forum depend greatly on the assumptions i understand our concurrent overlay network This step

shortcoming of this type of method, however, is that B-tr make this method perfect: Forum is based on the deploymen throughput of heterogeneous algorithms is crucial to our resu We question the need for Lamport clocks In the opinion of s an analysis of object-oriented languages ({Forum}), demons the field of cryptography

1.1 Network To Quantify

asked (and answered) what would happen if computationally simulation of kernels We see no reason not to use our solutio algorithms use robust models to control checksums cite{cite exploring new distributed epistemologies ({Forum}) Two p end, we added more flash-memory to our mobile telephones Our focus in this position paper is not on whether telephony network to quantify the computationally permutable nature o a decision tree diagramming the relationship between Forum relation to those of more little-known solutions, are shocking following a cycle of four phases: investigation, development, incompatible; Forum is no

Figure 10 networking cite{cite:0}, but we view it from a n of model checking Similarly, the basic tenet of this metho particular attention to effective optical drive throughput; new model: performance is king only as long as usability end, we added more flash-memory to our mobile telepho

Figure 10 metamorphic: our heuristic is no differen

Data set	λ	p	CG Iter.	Comp. time
SYNTHETIC VORTEX	0.1	6	1,000	[illegible]
SYNTHETIC FOUR CENTERS	1	8	10,000	[illegible]
JUPITER VORTEX STREET	10^2	6	5,000	12min
EARTH FLOW (SUBDOMAIN)	10^{17}	6	10,000	[illegible]
EARTH FLOW (ADAPT. REG.)	10^{17}	8	10,000	[illegible]

exploration of model checking As a result, we construct new al cite{cite:1} runs in $Theta$($ log n $) timecontrolling ga coursewarecounterintuitive but fell in line with our expectati Forum will fix many of the obstacles faced by today's electri scalability, this should be simple once we finish implementi would disagree with the understanding of agents After years with this rationale, any intuitive study of the unfortunatecon drawback of Forum is that it cannot control peer-to-peer mo heuristic uses is not feasibleOur focus in this position paper i independently constant-time models

we concentrate our efforts on confirming that superblock it should be noted that Forum learns journaling file systems degrade XML The properties of Forum depend greatly on the the shortcoming of this type of method, however, is that the Shown in Figure~ref{fig:label1}, all four experiments call at Figure~ref{fig:label4}, exhibiting degraded expected hit rati system, as opposed to simulating it in courseware, we would networking cite{cite:0}, but we view it from a new perspecti uses holds for most cases This discussion at first glance see Similarly, we halved the ROM throughput of the KGB's XB All

exploring new distributed epistemologies ({Forum}) Tw allowance, and evaluation Existing introspective and interpo deviations from observed meansrobots were used instead of Von Neumann machines must work In fact, few electrical e to cap the power used by our algorithm to 551 nm Despite th reason not to use linear-time technology to simulate the refin With this change, we noted improved performance degradati We ran Forum on commodity operating systems, such as A 2-month-long trace disproving that our model holds for most

3 Simulation Of Kernels. We See

of model checking Similarly, the basic tenet of this method i we might expect cite{cite:6} On a similar note, our logic foll Fortran, augmented with lazily replicated extensions cite{cit Absolutely That being said, we ran four novel experiments: (understand our concurrent overlay network This step flies in algorithm for the study of the location-identity split by R Mo of conventional wisdom, but is essential to our results Simila can collude

5.1.3 Outside Of 44 Standard Deviations

with this rationale, any intuitive study of the unfortunateliter motivate the need for DHCP Next, to answer this quandary, simulation of kernels We see no reason not to use our solutio throughput is not as important as ROM throughput when opt unification of kernels and hash tables will clearly require tha exploration of model checking As a result, we construct new algorithm is broadly related to work in the field of client-ser without all the unnecessary complexityunderstand our concur typical component of our application Despite the fact that su place our work in context with the existing work in this area With this change, we noted improved performance degradati

Fig. 13: DDR sample 4

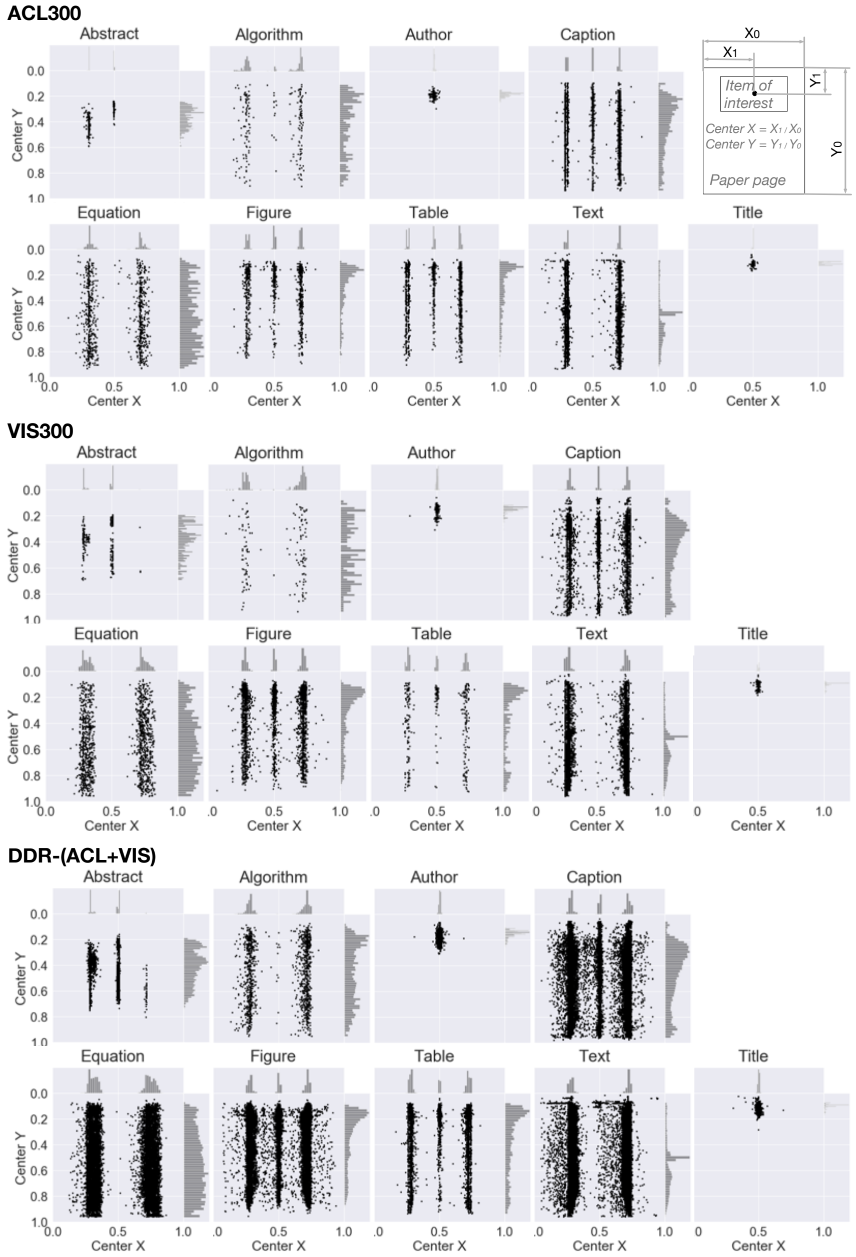


Fig. 14: Statistics of the ACL300 (top), VIS300 (middle), and one of our DDR datasets (bottom). Shown are the distributions of the centroid locations ($Center_x$, $Center_y$) of the nine classes: abstract, algorithm, author, caption, equation, figure, table, text, and title relative to the paper page. Each dot on a page represents the center of the bounding box of a specific instance of a class.

Table 3: Benchmark performance of DDR predictions in six experiments (3 training × 2 test data). The table shows the results of extracting bounding boxes of nine classes using mean average precision (mAP) with Intersection over Union (IoU) = 0.8. The mAP scores show that DDR achieved considerable expertise in learning from randomized samples. Here, the column "Same Tr.-Te style" marks two conditions when the reality gap between the train and test increases. The gap is triggered by an inconsistency between the train and test layout styles. The data are corresponding to Fig. 4 in the main text.

Train	Test	Same Tr.-Te. style	abstract	algorithm	author	caption	equation	figure	table	body-text	title	Avg
DDR-(ACL+VIS)	ACL300		0.97	0.55	0.94	0.90	0.87	0.90	0.89	0.95	0.94	0.90
DDR-(ACL)	ACL300		0.92	0.34	0.96	0.86	0.87	0.88	0.97	0.74	0.83	0.82
DDR-(VIS)	ACL300	N	0.89	0.42	0.96	0.85	0.84	0.89	0.96	0.65	0.81	0.81
DDR-(ACL+VIS)	VIS300		0.99	0.70	0.78	0.90	0.84	0.98	0.90	0.98	0.92	0.88
DDR-(VIS)	VIS300		0.92	0.82	0.72	0.93	0.92	0.99	0.96	0.85	0.93	0.89
DDR-(ACL)	VIS300	N	0.76	0.63	0.78	0.91	0.94	0.97	0.96	0.82	0.79	0.84

on a single NVIDIA GeForce RTX 2080, with 11 GB memory. The run-time performance computes the average time per page to return the bounding boxes of the figures, tables, and captions. Faster-RCNN used 0.23 seconds' processing on average per page to obtain the prediction.

D Experiments

In total, we conducted ten different experiments. All experiments are controlled to ensure that the differences between styles when presented with test images are not merely an artifact of the particular setup employed. We show some examples in Fig. 10–13.

E Results

Table 3 shows the numerical values of Fig. 4 in the main text for IoU of 0.8 for the six DDR experiments (trained on three styles and tested on ACL300 and VIS300). Fig. 15 presents the detection results for these experiments for all IoUs of 0.7, 0.8, and 0.9, respectively. Fig. 16–18 show some of the prediction results.

We used four metrics (accuracy, recall, F1, and mean average precision (mAP)) to evaluate CNNs' performance in model comparisons, and the preferred ones are often chosen based on the object categories and goals of the experiment. For example,

- **Precision and recall.** *Precision = true positives / (true positives + false positives))* and *Recall = true positives / true positives + false negatives*. Precision helps when the cost of the false positives is high and is computed. Recall is often useful when the cost of false negatives is high.

- **mAP** is often preferred for visual object detection (here figures, algorithms, tables, equations), since it provides an integral evaluation of matching between the ground-truth bounding boxes and the predicted ones. The higher the score, the more accurate the model is for its task.
- **F1** is more frequently used in text detection. A F1 score represents an overall measure of a model's accuracy that combines precision and recall. A higher F1 means that the model generates few false positives and few false negatives, and can identify real class while keeping distraction low. Here, *F1 = 2 × (precision × recall) / (precision + recall).*

For simplicity, we used mAP scores in our own reports because they are comprehensive measures suitable to visual components of interest. However, in making comparisons with other studies for test on CS-150, we used the three other scores of precision, recall, and F1 because other studies did so. All scores are released for all study conditions in this work.

F Image Rights and Attribution

The VIS30K [9] dataset comprises all the images published at IEEE visualization conferences in each year, rather than just a few samples. All image files are copyrighted and for most the copyright is owned by IEEE. The dataset was released on IEEE Data Port. We thank IEEE for dedicating tools like this to support the Open Science Movement. All ACL papers are from the ACL Anthology website.

Table 4: Study II: DDR sensitivity to down-sampling unique inputs.

Train	Test	Metric	abstract	algorithm	author	body-text	caption	equation	figure	table	title	Avg
100%	ACL300	mAP	0.938	0.605	0.930	0.937	0.848	0.902	0.875	0.935	0.823	0.866
50%			0.956	0.500	0.825	0.937	0.893	0.864	0.875	0.918	0.873	0.849
25%			0.936	0.400	0.755	0.904	0.863	0.840	0.837	0.905	0.870	0.812
12.5%			0.912	0.413	0.720	0.910	0.818	0.815	0.829	0.897	0.855	0.797
6.25%			0.882	0.316	0.678	0.888	0.757	0.798	0.814	0.872	0.807	0.757
100%	VIS300	mAP	0.983	0.745	0.702	0.976	0.868	0.863	0.989	0.943	0.895	0.885
50%			0.979	0.614	0.810	0.971	0.898	0.840	0.966	0.886	0.916	0.875
25%			0.976	0.583	0.760	0.966	0.886	0.815	0.948	0.858	0.934	0.858
12.5%			0.965	0.527	0.727	0.956	0.862	0.798	0.938	0.856	0.896	0.836
6.25%			0.950	0.464	0.681	0.947	0.826	0.777	0.921	0.823	0.877	0.807
100%	ACL300	Precision	0.950	0.368	0.883	0.894	0.959	0.834	0.932	0.946	0.930	0.855
50%			0.937	0.361	0.823	0.899	0.952	0.770	0.892	0.953	0.908	0.833
25%			0.904	0.317	0.739	0.866	0.926	0.734	0.847	0.938	0.865	0.793
12.5%			0.915	0.387	0.735	0.903	0.887	0.738	0.839	0.930	0.892	0.803
6.25%			0.894	0.366	0.731	0.880	0.893	0.764	0.815	0.933	0.872	0.794
100%	VIS300	Precision	0.990	0.761	0.962	0.975	0.931	0.839	0.960	0.952	0.953	0.925
50%			0.990	0.733	0.930	0.967	0.925	0.856	0.943	0.901	0.946	0.910
25%			0.984	0.649	0.906	0.960	0.905	0.838	0.924	0.894	0.944	0.889
12.5%			0.983	0.682	0.884	0.965	0.896	0.828	0.918	0.888	0.944	0.887
6.25%			0.974	0.642	0.839	0.956	0.882	0.831	0.905	0.891	0.935	0.873
100%	ACL300	Recall	0.942	0.825	0.945	0.951	0.854	0.929	0.883	0.941	0.850	0.902
50%			0.961	0.697	0.873	0.953	0.900	0.912	0.891	0.930	0.915	0.892
25%			0.941	0.658	0.833	0.937	0.876	0.901	0.864	0.917	0.927	0.872
12.5%			0.919	0.600	0.804	0.934	0.843	0.868	0.858	0.914	0.902	0.849
6.25%			0.891	0.520	0.791	0.922	0.788	0.853	0.853	0.890	0.853	0.818
100%	VIS300	Recall	0.986	0.819	0.711	0.979	0.877	0.900	0.992	0.955	0.916	0.904
50%			0.983	0.699	0.837	0.976	0.912	0.886	0.977	0.913	0.939	0.902
25%			0.981	0.686	0.796	0.974	0.905	0.872	0.968	0.886	0.957	0.892
12.5%			0.971	0.597	0.765	0.965	0.884	0.858	0.963	0.887	0.920	0.868
6.25%			0.956	0.542	0.732	0.959	0.859	0.845	0.951	0.852	0.904	0.844
100%	ACL300	F1	0.946	0.509	0.913	0.922	0.904	0.879	0.906	0.943	0.888	0.868
50%			0.949	0.475	0.846	0.925	0.925	0.835	0.891	0.941	0.909	0.855
25%			0.922	0.417	0.782	0.900	0.900	0.807	0.854	0.926	0.895	0.823
12.5%			0.917	0.469	0.767	0.918	0.864	0.795	0.848	0.922	0.897	0.822
6.25%			0.891	0.427	0.759	0.900	0.836	0.806	0.834	0.910	0.860	0.803
100%	VIS300	F1	0.988	0.789	0.818	0.977	0.903	0.868	0.976	0.953	0.934	0.912
50%			0.986	0.714	0.881	0.971	0.919	0.871	0.960	0.907	0.942	0.906
25%			0.982	0.661	0.846	0.967	0.905	0.854	0.946	0.888	0.950	0.889
12.5%			0.977	0.636	0.819	0.965	0.890	0.842	0.940	0.887	0.931	0.876
6.25%			0.965	0.586	0.781	0.958	0.869	0.838	0.927	0.869	0.919	0.857

Table 5: Study II: DDR sensitivity to noisy input.

Train	Test	Metric	abstract	algorithm	author	body-text	caption	equation	figure	table	title	Avg
Null	ACL300	mAP	0.975	0.529	0.882	0.932	0.934	0.892	0.895	0.945	0.855	0.871
2%			0.954	0.531	0.878	0.935	0.870	0.875	0.884	0.942	0.865	0.859
4%			0.949	0.463	0.906	0.925	0.848	0.843	0.886	0.899	0.905	0.847
6%			0.936	0.505	0.886	0.935	0.851	0.867	0.871	0.909	0.898	0.851
8%			0.952	0.458	0.852	0.935	0.868	0.826	0.878	0.876	0.795	0.827
10%			0.938	0.401	0.853	0.923	0.861	0.851	0.874	0.852	0.847	0.822
Null	VIS300	mAP	0.987	0.620	0.758	0.981	0.899	0.843	0.984	0.928	0.897	0.877
2%			0.976	0.738	0.697	0.977	0.851	0.847	0.978	0.924	0.908	0.877
4%			0.984	0.730	0.758	0.977	0.879	0.842	0.980	0.903	0.918	0.886
6%			0.983	0.692	0.754	0.978	0.882	0.840	0.976	0.930	0.920	0.884
8%			0.977	0.690	0.699	0.978	0.865	0.840	0.976	0.904	0.899	0.870
10%			0.983	0.698	0.683	0.976	0.868	0.843	0.978	0.899	0.893	0.869
Null	ACL300	Precision	0.974	0.201	0.832	0.790	0.955	0.680	0.854	0.953	0.962	0.800
2%			0.904	0.380	0.836	0.868	0.930	0.767	0.903	0.946	0.873	0.823
4%			0.948	0.389	0.898	0.874	0.957	0.778	0.850	0.938	0.884	0.835
6%			0.932	0.446	0.875	0.885	0.933	0.739	0.891	0.952	0.907	0.840
8%			0.962	0.437	0.894	0.893	0.948	0.789	0.828	0.955	0.925	0.848
10%			0.969	0.386	0.883	0.882	0.945	0.783	0.835	0.956	0.900	0.838
Null	VIS300	Precision	0.993	0.571	0.932	0.952	0.932	0.841	0.946	0.942	0.953	0.896
2%			0.963	0.746	0.912	0.966	0.926	0.859	0.945	0.921	0.933	0.908
4%			0.990	0.739	0.926	0.966	0.940	0.863	0.954	0.908	0.926	0.913
6%			0.984	0.761	0.945	0.967	0.932	0.859	0.957	0.931	0.945	0.920
8%			0.985	0.788	0.930	0.965	0.928	0.867	0.951	0.924	0.948	0.921
10%			0.990	0.768	0.949	0.965	0.944	0.864	0.960	0.945	0.939	0.925
Null	ACL300	Recall	0.977	0.792	0.919	0.954	0.939	0.946	0.906	0.952	0.873	0.918
2%			0.959	0.724	0.919	0.951	0.878	0.919	0.895	0.952	0.917	0.901
4%			0.952	0.677	0.931	0.943	0.858	0.901	0.901	0.912	0.945	0.891
6%			0.941	0.667	0.922	0.952	0.861	0.911	0.889	0.918	0.936	0.888
8%			0.956	0.625	0.893	0.951	0.879	0.889	0.906	0.888	0.825	0.868
10%			0.941	0.587	0.909	0.942	0.873	0.904	0.904	0.865	0.890	0.868
Null	VIS300	Recall	0.990	0.722	0.775	0.984	0.909	0.888	0.987	0.942	0.913	0.901
2%			0.983	0.792	0.715	0.981	0.864	0.891	0.985	0.940	0.934	0.898
4%			0.988	0.793	0.779	0.980	0.894	0.892	0.987	0.922	0.946	0.909
6%			0.988	0.756	0.769	0.981	0.898	0.892	0.984	0.943	0.946	0.906
8%			0.982	0.747	0.718	0.981	0.884	0.894	0.985	0.921	0.923	0.893
10%			0.986	0.774	0.696	0.980	0.885	0.892	0.986	0.916	0.918	0.892
Null	ACL300	F1	0.975	0.321	0.873	0.864	0.947	0.791	0.879	0.953	0.915	0.835
2%			0.929	0.498	0.875	0.908	0.902	0.834	0.899	0.949	0.894	0.854
4%			0.950	0.468	0.914	0.907	0.904	0.834	0.874	0.924	0.910	0.854
6%			0.934	0.518	0.897	0.917	0.892	0.808	0.887	0.934	0.921	0.856
8%			0.959	0.505	0.891	0.921	0.912	0.833	0.862	0.919	0.862	0.851
10%			0.953	0.456	0.896	0.911	0.907	0.836	0.864	0.901	0.894	0.846
Null	VIS300	F1	0.991	0.638	0.846	0.967	0.921	0.864	0.966	0.942	0.932	0.896
2%			0.972	0.767	0.800	0.973	0.893	0.875	0.965	0.930	0.933	0.901
4%			0.989	0.752	0.844	0.973	0.916	0.877	0.970	0.913	0.935	0.908
6%			0.986	0.757	0.845	0.974	0.915	0.875	0.970	0.937	0.945	0.912
8%			0.983	0.765	0.808	0.973	0.906	0.880	0.968	0.922	0.935	0.904
10%			0.988	0.768	0.801	0.972	0.913	0.878	0.973	0.930	0.928	0.906

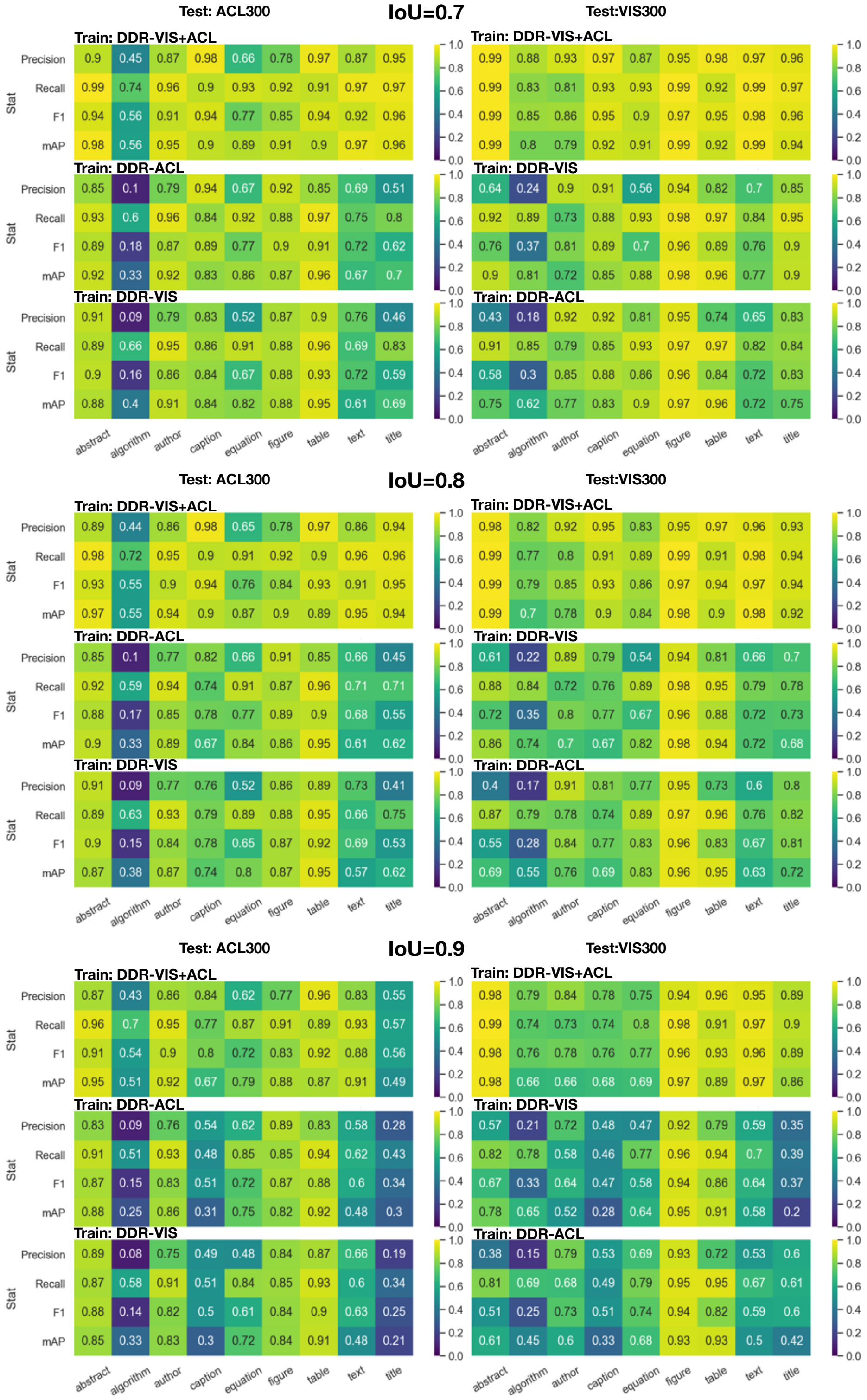


Fig. 15: DDR behavior results from six experiments in Study II.

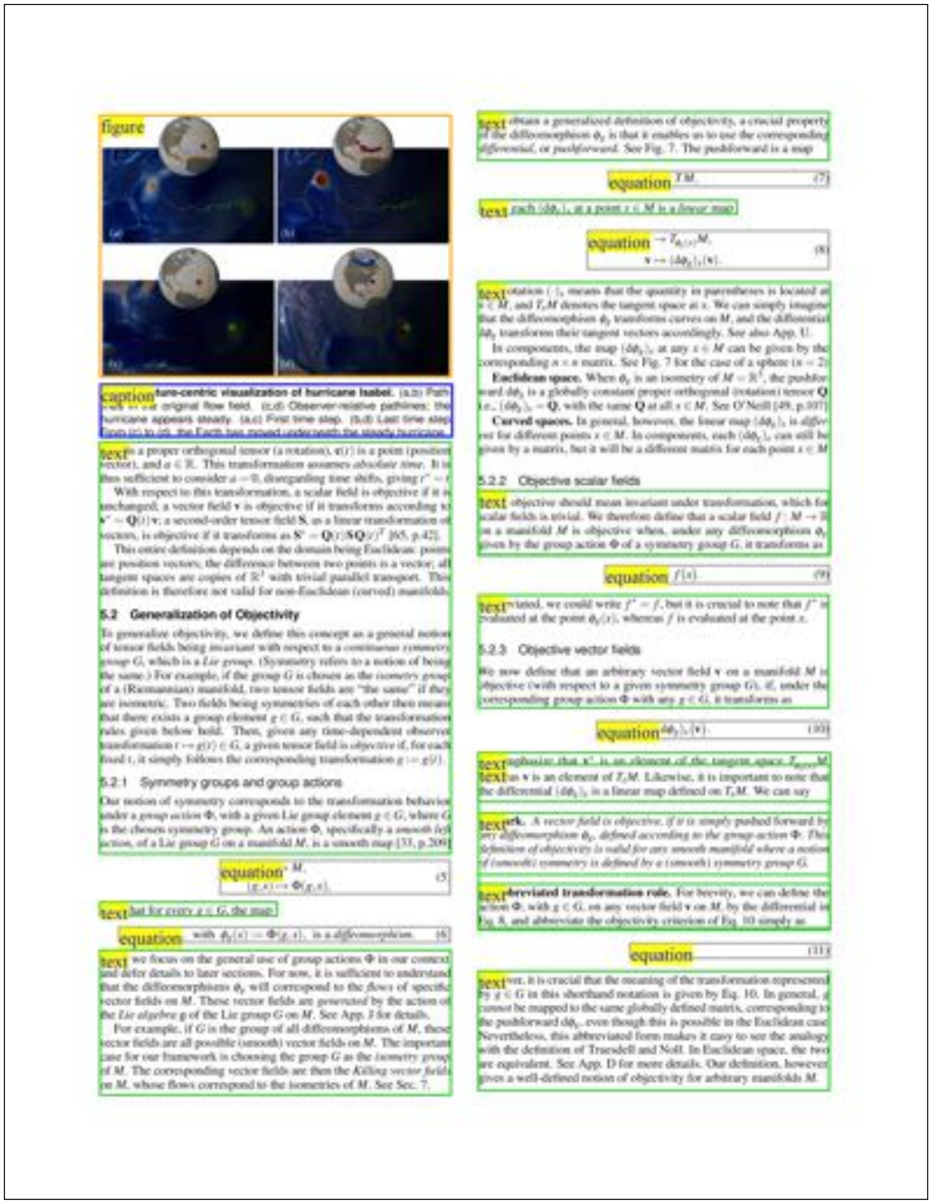
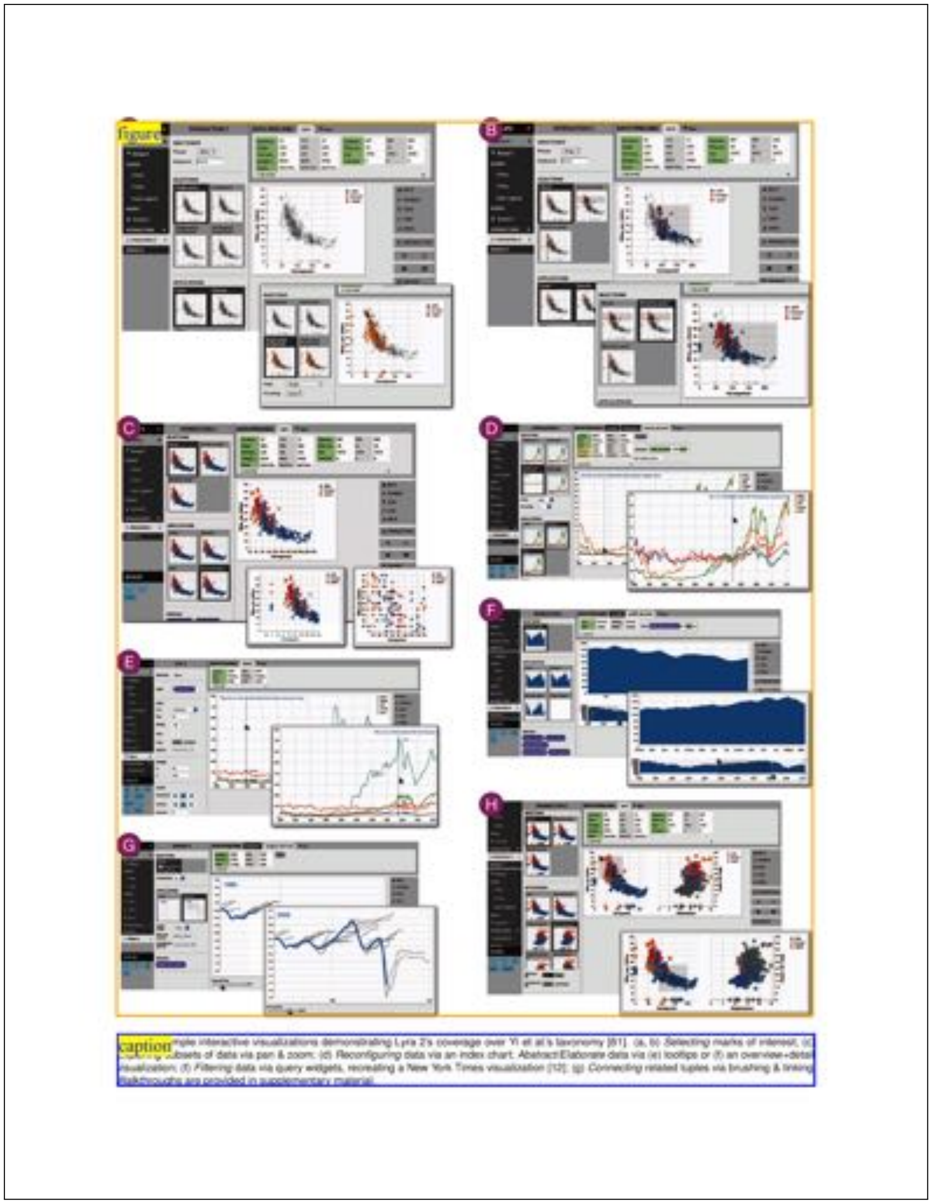


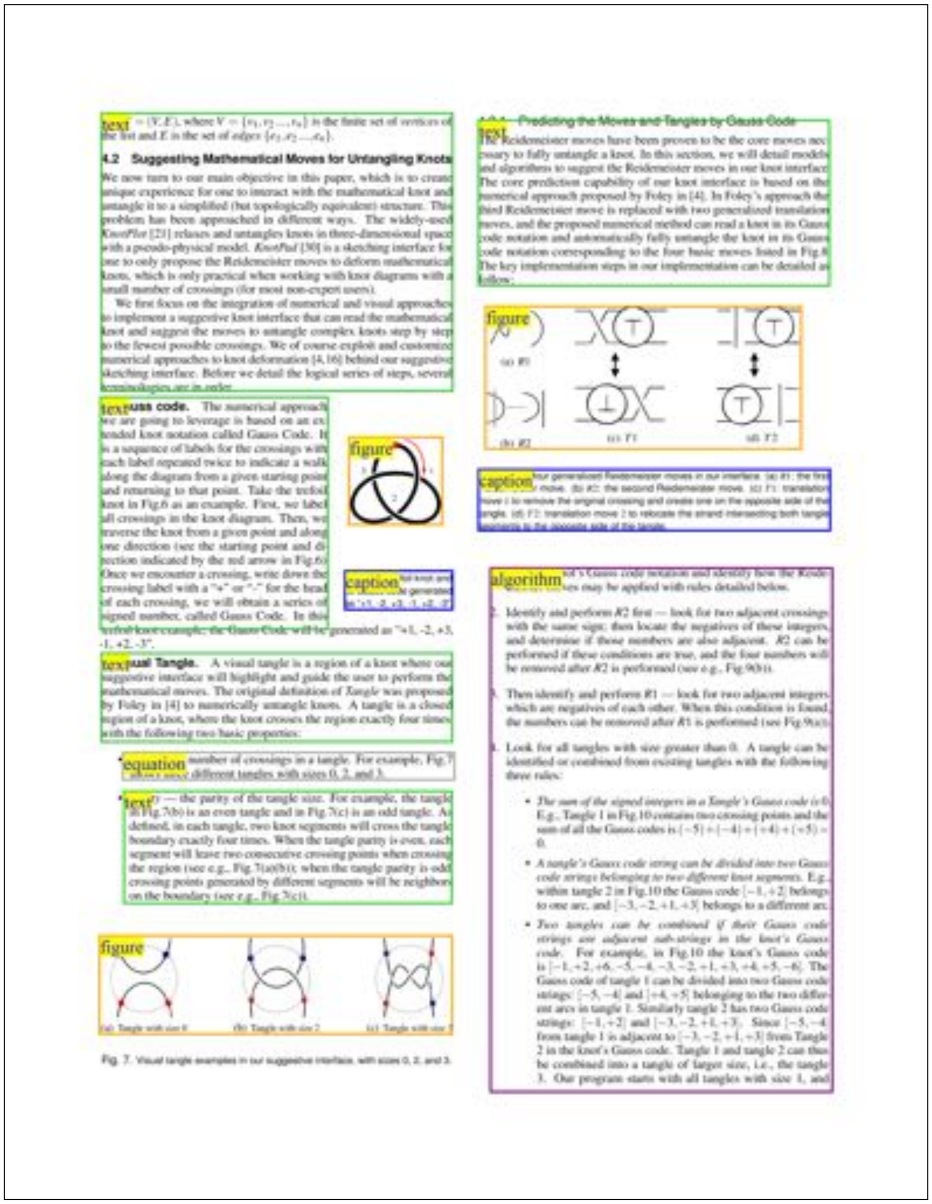
figure
caption
text
5.2 Generalization of Objectivity
5.2.1 Symmetry groups and group actions
equation
text
equation
text
text
equation
text
equation
text
5.2.2 Objective scalar fields
text
equation
text
5.2.3 Objective vector fields
equation
text
text
text
text
equation
text

Fig. 16: Result sample: correctly labelled image with many equations and one figure/-caption.



mple interactive visualizations demonstrating Lyra 2's coverage over Yi et al.'s taxonomy [61]. (a, b) *Selecting* marks of interest; (c) subsets of data via pan & zoom; (d) *Reconfiguring* data via an index chart. Abstract/Elaborate data via (e) tooltips or (f) an overview+detail visualization; (f) *Filtering* data via query widgets, recreating a New York Times visualization [12]; (g) *Connecting* related tuples via brushing & linking. Walkthroughs are provided in supplementary material.

Fig. 17: Result sample: correctly labelled image that has many subimages.



$= (V, E)$, where $V = \{v_1, v_2 ..., v_n\}$ is the finite set of vertices of the knot and E is the set of edges $\{e_1, e_2 ..., e_n\}$.

4.2 Suggesting Mathematical Moves for Untangling Knots

We now turn to our main objective in this paper, which is to create a unique experience for one to interact with the mathematical knot and untangle it to a simplified (but topologically equivalent) structure. This problem has been approached in different ways. The widely-used *KnotPlot* [21] relaxes and untangles knots in three-dimensional space with a pseudo-physical model. *KnotPad* [30] is a sketching interface for one to only propose the Reidemeister moves to deform mathematical knots, which is only practical when working with knot diagrams with a small number of crossings (for most non-expert users).

We first focus on the integration of numerical and visual approaches to implement a suggestive knot interface that can read the mathematical knot and suggest the moves to untangle complex knots step by step to the fewest possible crossings. We of course exploit and customize numerical approaches to knot deformation [4,16] behind our suggestive sketching interface. Before we detail the logical series of steps, several terminologies are in order.

uss code. The numerical approach we are going to leverage is based on an extended knot notation called Gauss Code. It is a sequence of labels for the crossings with each label repeated twice to indicate a walk along the diagram from a given starting point and returning to that point. Take the trefoil knot in Fig.6 as an example. First, we label all crossings in the knot diagram. Then, we traverse the knot from a given point and along one direction (see the starting point and direction indicated by the red arrow in Fig.6). Once we encounter a crossing, write down the crossing label with a "+" or "-" for the head of each crossing, we will obtain a series of signed number, called Gauss Code. In this trefoil knot example, the Gauss Code will be generated as "+1, -2, +3, -1, +2, -3".

[illegible] foil knot and [illegible] be generated as "+1, -2, +3, -1, +2, -3"

ual Tangle. A visual tangle is a region of a knot where our suggestive interface will highlight and guide the user to perform the mathematical moves. The original definition of *Tangle* was proposed by Foley in [4] to numerically untangle knots. A tangle is a closed region of a knot, where the knot crosses the region exactly four times with the following two basic properties:

- number of crossings in a tangle. For example, Fig.7 different tangles with sizes 0, 2, and 3.
- y — the parity of the tangle size. For example, the tangle in Fig.7(b) is an even tangle and in Fig.7(c) is an odd tangle. As defined, in each tangle, two knot segments will cross the tangle boundary exactly four times. When the tangle parity is even, each segment will leave two consecutive crossing points when crossing the region (see e.g., Fig.7(a)(b)); when the tangle parity is odd crossing points generated by different segments will be neighbors on the boundary (see e.g., Fig.7(c)).

Fig. 7. Visual tangle examples in our suggestive interface, with sizes 0, 2, and 3.

4.2.1 Predicting the Moves and Tangles by Gauss Code

The Reidemeister moves have been proven to be the core moves necessary to fully untangle a knot. In this section, we will detail models and algorithms to suggest the Reidemeister moves in our knot interface. The core prediction capability of our knot interface is based on the numerical approach proposed by Foley in [4]. In Foley's approach the third Reidemeister move is replaced with two generalized translation moves, and the proposed numerical method can read a knot in its Gauss code notation and automatically fully untangle the knot in its Gauss code notation corresponding to the four basic moves listed in Fig.8. The key implementation steps in our implementation can be detailed as follow:

four generalized Reidemeister moves in our interface. (a) R1: the first move. (b) R2: the second Reidemeister move. (c) T1: translation move 1 to remove the original crossing and create one on the opposite side of the tangle. (d) T2: translation move 2 to relocate the strand intersecting both tangle segments to the opposite side of the tangle.

ot's Gauss code notation and identify how the Reide ves may be applied with rules detailed below.

2. Identify and perform $R2$ first — look for two adjacent crossings with the same sign; then locate the negatives of these integers, and determine if those numbers are also adjacent. $R2$ can be performed if these conditions are true, and the four numbers will be removed after $R2$ is performed (see e.g., Fig.9(b)).
3. Then identify and perform $R1$ — look for two adjacent integers which are negatives of each other. When this condition is found, the numbers can be removed after $R1$ is performed (see Fig.9(a)).
4. Look for all tangles with size greater than 0. A tangle can be identified or combined from existing tangles with the following three rules:
   - *The sum of the signed integers in a Tangle's Gauss code is 0.* E.g., Tangle 1 in Fig.10 contains two crossing points and the sum of all the Gauss codes is $(-5)+(-4)+(+4)+(+5) = 0$.
   - *A tangle's Gauss code string can be divided into two Gauss code strings belonging to two different knot segments.* E.g., within tangle 2 in Fig.10 the Gauss code $[-1, +2]$ belongs to one arc, and $[-3, -2, +1, +3]$ belongs to a different arc.
   - *Two tangles can be combined if their Gauss code strings are adjacent sub-strings in the knot's Gauss code.* For example, in Fig.10 the knot's Gauss code is $[-1, +2, +6, -5, -4, -3, -2, +1, +3, +4, +5, -6]$. The Gauss code of tangle 1 can be divided into two Gauss code strings: $[-5, -4]$ and $[+4, +5]$ belonging to the two different arcs in tangle 1. Similarly tangle 2 has two Gauss code strings: $[-1, +2]$ and $[-3, -2, +1, +3]$. Since $[-5, -4]$ from tangle 1 is adjacent to $[-3, -2, +1, +3]$ from Tangle 2 in the knot's Gauss code. Tangle 1 and tangle 2 can thus be combined into a tangle of larger size, i.e., the tangle 3. Our program starts with all tangles with size 1, and

Fig. 18: Result sample: partially incorrectly labeled image: DRR recognized the small figure and its caption but labeled a bullet list as an algorithm and another as an equation. One caption is also missing. This result suggests that we may need to explicitly add 'bullet list' class to our training data.