Neurosymbolic Optical Character Recognition

Abstract

Automatically digitizing structured documents requires optical character recognition (OCR) to convert printed or written text into a digital format. Such documents often have known dependencies between various text fields and restrictions on their content that can be expressed as formal constraints. However, OCR-specific neural systems cannot directly incorporate such constraints, while generalist vision language models (VLMs) can be prompted with them but do not explicitly enforce them during text recognition. Ignoring this domain knowledge can lead to invalid predictions and suboptimal behavior. We propose a neurosymbolic approach that integrates data-driven neural perception with symbolic knowledge expressed as constraints via a token-level A*-like search, allowing these constraints to guide inference and learning. Our results show that leveraging available constraints improves recognition accuracy, and that they provide ample supervision to fine-tune the neural text recognition component without ground-truth annotations.

1 Introduction

Optical character recognition (OCR) converts images of text into strings. In many document digitization settings, the input is not a single isolated text snippet but a structured document containing multiple related text fields. Such documents often come with domain knowledge about the extracted text in the form of symbolic constraints, for example: arithmetic relations between monetary amounts, equalities between repeated fields, or formatting requirements.

Both dedicated neural OCR models and generalist vision language models (VLMs) have become increasingly capable at OCR tasks [Li *et al.*, 2023; Bai *et al.*, 2025]. Dedicated OCR models typically take only images as input and therefore cannot directly incorporate symbolic constraints. VLMs can be prompted with domain constraints in natural language, but do not explicitly enforce them during recognition. As a result, neither approach guarantees consistency with known domain constraints. Even a small number of character-level errors can therefore produce inconsistent outputs that invalidate downstream tasks and require costly manual correction.

Neurosymbolic AI provides a natural framework to address this limitation by integrating high-level symbolic knowledge with low-level neural perception [Hitzler and Sarker, 2022; Marra *et al.*, 2024]. In this work, we propose C*OCR, a neurosymbolic approach to OCR that integrates an autoregressive neural perception model with symbolic constraints. We consider a setting in which a document image contains values for multiple variables subject to known domain constraints. Each variable is specified by either a cropped text-field image, produced by an upstream text detection step, or by a natural-language description. Given such a neurosymbolic OCR problem, C*OCR predicts the string variables using an A*-like token-level search guided by neural probabilities, while pruning infeasible partial assignments using the known constraints.

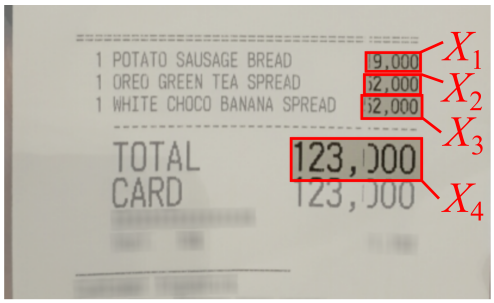
Our experiments show that leveraging domain constraints improves both variable-level and document-level accuracy. Moreover, we show that the same constraints provide supervision to fine-tune the neural OCR model without ground-truth annotations.

Our main contributions are: (1) a formulation of the neurosymbolic OCR problem; (2) C*OCR, a neurosymbolic inference method that integrates an autoregressive neural perception model with symbolic constraints over multiple variables, each represented as a token sequence; (3) a neurosymbolic fine-tuning approach that leverages symbolic constraints as supervision for neural text recognition models in the absence of ground-truth annotations.

2 Background

In the classical OCR setting, one is given a text field image $X \in \mathcal{X}$ that has to be mapped onto the corresponding ground-truth string s^{gt} over an alphabet Σ. Contemporary models operate over a vocabulary $\mathcal{T} \subset \Sigma^*$ of tokens (characters or subwords) rather than individual characters.

Token-level OCR is typically implemented with autoregressive models: given an image X, the model P_θ defines a distribution over token sequences by predicting next-token probabilities [Li *et al.*, 2023]. The probability of a complete token sequence $t_{1:\ell} = t_1 t_2 \cdots t_\ell$ with $t_1 = \texttt{BOS}$ (beginning-of-sequence) and $t_\ell = \texttt{EOS}$ (end-of-sequence) being sampled from P_θ given text field image X is computed as $P_\theta(t_{1:\ell} \mid X) = \prod_{j=2}^{\ell} P_\theta(t_j \mid t_{1:j-1}, X)$.



$$\mathcal{C} = \{\, S_1 + S_2 + S_3 = S_4 \,\}$$

Figure 1: A receipt represented as a neurosymbolic OCR problem. The constraint set $\mathcal{C}$ specifies that the sum of all printed line totals equals the printed overall total.

As an alternative to OCR-specific models, one can use a generalist VLM P_θ, whose input X is an image-prompt pair, where the prompt specifies which section of text to extract from the document image (e.g., "read the total amount field").

3 Neurosymbolic OCR Problems

We define a **neurosymbolic OCR problem** as a pair of inputs and constraints $(\mathbf{X}, \mathcal{C})$. The input tuple $\mathbf{X} = (X_1, \ldots, X_n)$ contains one input for each string variable S_i, where each X_i is either a text field image or an image-prompt pair containing the value of S_i. The constraint set $\mathcal{C} = \{C_1, \ldots, C_m\}$ contains symbolic constraints over the variables $\{S_1, \ldots, S_n\}$. The goal is to map the inputs to the unknown ground-truth strings $\mathbf{s}^{\text{gt}} = (s_1^{\text{gt}}, \ldots, s_n^{\text{gt}})$, where s_i^{gt} is the value of S_i depicted or specified by X_i, given that these strings satisfy all constraints in $\mathcal{C}$. We write $\mathbf{s} \models \mathcal{C}$ to denote that the assignment $\{S_1 = s_1, \ldots, S_n = s_n\}$ satisfies all constraints in $\mathcal{C}$, so $\mathbf{s}^{\text{gt}} \models \mathcal{C}$. Example constraints could specify that a particular string consists of exactly five characters or that the sum of the numbers represented by S_1, S_2, and S_3 equals S_4. Constraints specify the knowledge that is available to the expert.

We approach neurosymbolic OCR inference as a constrained maximization problem. Given a neurosymbolic OCR problem $(\mathbf{X}, \mathcal{C})$ and an autoregressive neural perception model P_θ, our objective is to compute $\mathbf{s}^*$, the most likely tuple of strings that satisfies the constraints $\mathcal{C}$:

$$\mathbf{s}^* = \underset{\mathbf{s} \in (\Sigma^*)^n \text{ s.t. } \mathbf{s} \models \mathcal{C}}{\arg\max} \; P_\theta(\mathbf{s} \mid \mathbf{X})$$

where the joint probability factorizes into a product of per-variable probabilities: $P_\theta(\mathbf{s}|\mathbf{X}) = \prod_{i=1}^n P_\theta(s_i|X_i)$.

We also investigate how the constraints of neurosymbolic OCR problems can provide supervision for neural training, removing the need for expensive ground-truth annotations. In our **neurosymbolic OCR learning problem**, we are given a dataset of neurosymbolic OCR problems $\mathcal{D} = \{(\mathbf{X}_i, \mathcal{C}_i)\}_{i=1}^N$, consisting of N independent samples from a distribution $\mathcal{P}$ over neurosymbolic OCR problems, together with a pre-trained neural text recognition model P_θ. The goal is to fine-tune the parameters θ using $\mathcal{D}$ such that the resulting model attains higher recognition accuracy on unseen images belonging to samples drawn from $\mathcal{P}$.

4 Method

We perform neurosymbolic OCR inference via a token-level A*-like search, and refer to our method as C*OCR, for *Constrained A*-based OCR*.

4.1 Inference as a Token-level A*-like Search

Given a neurosymbolic OCR problem $(\mathbf{X}, \mathcal{C})$ and a neural model P_θ, the goal of neurosymbolic OCR inference is to find the most likely satisfying tuple of strings $\mathbf{s}^*$, where the probability $P_\theta(\mathbf{s} \mid \mathbf{X})$ marginalizes over multiple possible tokenizations for each string. We approximate $\mathbf{s}^*$ by searching instead for the most likely satisfying tuple of complete token sequences. This is done using an A*-like search over n (partial) token sequences.

Each node z in the C*OCR search tree is a tuple of n partial token sequences $(t^1_{1:\ell_1}, \ldots, t^n_{1:\ell_n})$, one per input X_i. The root contains only `BOS` in each sequence, and a child is constructed by appending one token to one of the parent's prefixes. A node is *complete* if every sequence ends with `EOS`, *satisfying* if complete and consistent with $\mathcal{C}$, and *satisfiable* if its prefixes can be extended (possibly emptily) into a satisfying node. The search aims to find the most likely satisfying node z^*.

We score nodes with $f(z) = g(z) + h(z)$. At each iteration, C*OCR pops the lowest-f node from the frontier; if it is satisfiable, its children are inserted back. The search terminates when a satisfying node is popped. We define $g(z)$ as the negative log-likelihood of the current prefixes,

$$g(z) = -\sum_{i=1}^{n} \sum_{j=2}^{\ell_i} \log P_\theta\big(t^i_j \mid t^i_{1:j-1}, X_i\big)\,,$$

and $h(z)$ as an estimate of the remaining cost to reach z^*.

4.2 The Geometric Mean Heuristic

To avoid repeated backtracking to short prefixes with lower g-scores than better but longer candidates simply because they are shorter, we use a length-aware heuristic. Let $|z| = \sum_{i=1}^n (\ell_i - 1)$ be the total number of tokens in z (excluding `BOS`). Assuming the per-token cost of the path from z to z^* continues at the current average $g(z)/|z|$, and approximating the unknown $|z^*|$ by the unconstrained neural-OCR length $|z_{\text{neural}}|$, we set

$$h(z) = \max(0,\, |z_{\text{neural}}| - |z|)\; \frac{g(z)}{|z|},$$

following [Manhaeve *et al.*, 2021]. The $\max$ acts as a non-negativity safeguard.

4.3 Computing Satisfiability During Search

C*OCR checks whether a node is satisfiable before expanding it, allowing inconsistent partial assignments to be pruned early while ensuring that its output satisfies the constraints.

A node is satisfiable if the current token prefixes can be extended into complete sequences that satisfy $\mathcal{C}$. Thus, a satisfiability check corresponds to a constraint satisfaction problem, which we model using the CPMpy constraint modeling framework and solve using a constraint solver [Guns, 2019].

4.4 Variable Ordering

During expansion, C*OCR creates k children by appending the top k next-token values to one selected prefix of the parent node $(t^1_{1:\ell_1}, \ldots, t^n_{1:\ell_n})$. Thus, each expansion must choose which variable prefix to extend.

We use a best-child-first heuristic: C*OCR extends the prefix whose most likely next token has the highest probability. This prioritizes high-confidence prefixes, reducing unnecessary backtracking and allowing likely correct assignments to constrain the remaining, more uncertain variables.

5 Fine-tuning with Symbolic Constraints

In this section, we introduce a neurosymbolic fine-tuning algorithm that uses C*OCR inference to tackle the neurosymbolic OCR learning problem introduced in Section 3.

This setting has a dataset $\mathcal{D} = \{(\mathbf{X}_i, \mathcal{C}_i)\}_{i=1}^N$ of neurosymbolic OCR problems, but no direct supervision on the strings depicted in the inputs. Given an instance $(\mathbf{X}_i, \mathcal{C}_i)$ from the dataset, we first apply neurosymbolic inference using C*OCR to obtain the most likely result that satisfies the known constraints $\mathcal{C}_i$. If C*OCR succeeds for its given input, meaning that it does not fall back on the neural OCR result due to a timeout or an empty frontier, its output $\hat{z}$ given as complete token sequences $(\hat{t}^1_{1:\ell_1}, \ldots, \hat{t}^n_{1:\ell_n})$ serves as supervision for the inputs $(X_1, \ldots, X_n)$.

An update for the neural weights θ is then computed by taking the gradient of the negative logarithm of the probability of $(\hat{t}^1_{1:\ell_1}, \ldots, \hat{t}^n_{1:\ell_n})$, where this probability is computed as the product of conditional next-token probabilities predicted by P_θ. By optimizing the joint probability of the n token sequences in $\hat{z}$, we also optimize the individual conditional probabilities $P_\theta(\hat{t}^i_{1:\ell_i} \mid X_i)$.

6 Experiments

We evaluate whether incorporating known domain constraints through C*OCR improves OCR inference and enables neurosymbolic fine-tuning.

6.1 Experimental Setup

C*OCR inference is evaluated on the CORD dataset, which contains smartphone photographs of store and restaurant receipts [Park *et al.*, 2019]. Each receipt has a single arithmetic consistency constraint: the sum of the line totals must equal the overall total printed on the receipt, as illustrated in Figure 1. To assess robustness to visual degradation, we evaluate multiple JPEG compression levels by lowering the quality factor. We consider C*OCR in combination with two neural models: TrOCR [Li *et al.*, 2023], an OCR-specific model given cropped text field images for all line totals and the overall total; and Qwen3-VL-Instruct [Bai *et al.*, 2025], a VLM given the full receipt image together with two prompts, one for all line totals and one for the overall total. A timeout is imposed on each C*OCR inference call; if the timeout is exceeded, C*OCR returns the prediction of its underlying neural model to ensure fair evaluation. All baseline VLMs are provided with the known arithmetic constraint in natural language as part of their prompt. Because VLM-based methods may omit some line totals, we report text field recall over the target fields, namely all line totals and the overall total, rather than text field accuracy.

To evaluate the effect of neurosymbolic fine-tuning, we fine-tune TrOCR using the proposed algorithm without ground-truth text annotations and report TrOCR's text field accuracy before and after fine-tuning. We use the FinTabNet dataset, which contains document images of financial tables from corporate reports [Smock *et al.*, 2023]. For each table, the predicted variables are the numerical entries, constrained by summation or subtraction relationships within each column. Synthetic visual degradations are applied to both training and test images to create out-of-distribution inputs for the pre-trained text recognition model [Groleau *et al.*, 2023].

6.2 Results

The neurosymbolic inference results in Table 1 show that C*OCR consistently improves receipt recognition accuracy and text field recall over its underlying neural model. This holds both when C*OCR is combined with TrOCR and when it is combined with the open-source VLM Qwen3-VL-Instruct. Compared to the neural models alone, C*OCR has increased inference time per receipt, which grows as the underlying neural model becomes less accurate. The best absolute results are achieved by the proprietary VLM Gemini-2.5-Flash, although this should not be interpreted as a like-for-like comparison: Gemini-2.5-Flash has an undisclosed model size and is expected to be significantly larger than the tested open-source alternatives. All VLM-based results in this table should be considered preliminary, as they are based on a single evaluation run; additional runs are needed to quantify the effect of VLM non-determinism.

The results in Table 2 demonstrate that the proposed neurosymbolic fine-tuning procedure can fine-tune a neural text recognition model in the absence of ground-truth annotations. Fine-tuning is effective for both $k = 1$ and $k = 20$, with the additional search enabled by a larger k yielding consistent, albeit sometimes modest, accuracy gains. The benefit of C*OCR search is most pronounced in the low-data regime, where each additional training instance derived from a satisfying prediction provides a relatively larger learning signal.

7 Related Work

C*OCR is related to perception-based constraint solving methods [Mulamba *et al.*, 2024], where neural perception is used to solve constraint satisfaction problems with visual inputs. Existing approaches perform neural inference before symbolic solving, whereas neurosymbolic OCR requires interleaving token-level neural prediction with constraint reasoning due to the large space of possible strings.

C*OCR is also inspired by DPLA* [Manhaeve *et al.*, 2021], an A*-like approximate inference method for DeepProbLog [Manhaeve *et al.*, 2018]. Unlike DPLA*, C*OCR

Text recognition method	Requires text detection	Size	JPEG quality = 95			JPEG quality = 20			JPEG quality = 10		
			Receipt acc.	Field recall	Inference time (s)	Receipt acc.	Field recall	Inference time (s)	Receipt acc.	Field recall	Inference time (s)
TrOCR	yes	0.3B	0.971	0.990	0.06	0.909	0.960	0.06	0.804	0.883	0.06
C*OCR (TrOCR)	yes	0.3B	0.993	0.997	0.07	0.975	0.980	0.11	0.893	0.916	0.20
Qwen3-VL-Instruct	no	4B	0.763	0.850	0.51	0.709	0.835	0.52	0.643	0.772	0.49
Qwen3-VL-Thinking	no	4B	0.765	0.856	13.54	0.713	0.837	14.21	0.652	0.761	14.81
C*OCR (Qwen3-VL-Instruct)	no	4B	0.867	0.877	2.46	0.829	0.866	2.99	0.744	0.797	3.72
Gemini-2.5-Flash	no	>4B*	0.963	0.963	–	0.947	0.956	–	0.885	0.899	–

Table 1: Receipt-level accuracy, text field-level recall, and average inference time per text field across JPEG quality levels. Text recognition methods that require text detection are given cropped text field images marked as line total or overall total amounts. *Gemini-2.5-Flash has an undisclosed model size; we expect it to be significantly larger than 4B parameters.

Setting	Pre-trained	C*OCR fine-tuning ($k = 1$)	C*OCR fine-tuning ($k = 20$)
Noise 0.25, 25 tables	0.880 ± 0.000	0.952 ± 0.001	0.983 ± 0.004
Noise 0.50, 25 tables	0.701 ± 0.000	0.797 ± 0.029	0.868 ± 0.057
Noise 0.75, 25 tables	0.573 ± 0.000	0.754 ± 0.028	0.821 ± 0.013
Noise 0.75, 500 tables	0.573 ± 0.000	0.940 ± 0.005	0.942 ± 0.015

Table 2: Best neural-only TrOCR text field accuracy on FinTabNet after fine-tuning for five epochs across noise intensities (mean $\pm$ std over 5 seeds). We compare the pre-trained model to versions that are fine-tuned using C*OCR with $k = 1$ (greedy decoding) and $k = 20$.

targets autoregressive OCR models with multiple string variables decoded as token sequences and a non-fixed number of neural token predictions.

8 Conclusion

We introduced the neurosymbolic OCR problem and presented C*OCR, a neurosymbolic inference method that integrates autoregressive neural text recognition with symbolic constraints. C*OCR leverages known domain constraints to improve both text field and document-level accuracy over its underlying neural model. We further showed that constraints can provide supervision for learning, enabling fine-tuning of a pre-trained OCR model on unlabeled constrained instances and improving robustness to out-of-distribution noise.

References

[Bai *et al.*, 2025] Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-vl technical report, 2025. arXiv:2511.21631.

[Groleau *et al.*, 2023] Alexander Groleau, Kok Wei Chee, Stefan Larson, Samay Maini, and Jonathan Boarman. Augraphy: A data augmentation library for document images. In *Proceedings of the 17th International Conference on Document Analysis and Recognition (ICDAR)*, 2023.

[Guns, 2019] Tias Guns. Increasing modeling language convenience with a universal n-dimensional array, cppy as python-embedded example. In *Proceedings of the 18th workshop on Constraint Modelling and Reformulation at CP (Modref 2019)*, volume 19, 2019.

[Hitzler and Sarker, 2022] Pascal Hitzler and Md Kamruzzaman Sarker. *Neuro-symbolic Artificial Intelligence: The State of the Art*. Frontiers in artificial intelligence and applications. IOS Press, 2022.

[Li *et al.*, 2023] Minghao Li, Tengchao Lv, Jingye Chen, et al. Trocr: Transformer-based optical character recognition with pre-trained models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11), 2023.

[Manhaeve *et al.*, 2018] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, et al. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[Manhaeve *et al.*, 2021] Robin Manhaeve, Giuseppe Marra, and Luc De Raedt. Approximate Inference for Neural Probabilistic Logic Programming. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, pages 475–486, 11 2021.

[Marra *et al.*, 2024] Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, 328:104062, 2024.

[Mulamba *et al.*, 2024] Maxime Mulamba, Jayanta Mandi, Ali Mahmutoğulları, and Tias Guns. Perception-based constraint solving for sudoku images. *Constraints*, 29:112–151, 10 2024.

[Park *et al.*, 2019] Seunghyun Park, Seung Shin, Bado Lee, et al. Cord: A consolidated receipt dataset for post-ocr parsing. In *Document Intelligence Workshop at Neural Information Processing Systems*, 2019.

[Smock *et al.*, 2023] Brandon Smock, Rohith Pesala, and Robin Abraham. Aligning benchmark datasets for table structure recognition. In *International Conference on Document Analysis and Recognition*, pages 371–386. Springer, 2023.