

# Reasoning vs. Validation in LLM-based Pharmacological Parameter Extraction

**Qinhao Wu**

Apriko Research  
wu.qinhao@gmail.com

**Xia Cui**

Manchester Metropolitan University  
x.cui@mmu.ac.uk

## Abstract

Pharmacokinetic (PK) parameters are critical to drug development, but many remain embedded in free-text biomedical literature rather than a structured form. Existing extraction approaches primarily target explicit parameter mentions, while a substantial proportion of PK values must be inferred from context. We introduce implicit PK parameter extraction, where parameter identity is not explicitly stated but must be resolved through domain-grounded, multi-sentence reasoning. To address this challenge, we propose CORVA, a validation-enhanced LLM-based extraction framework that integrates reasoning and domain constraints to extract structured PK tuples from text. We also construct a labelled dataset from DrugBank absorption descriptions, of which 47.5% involve implicit expressions. Relative to the strongest baseline, the proposed method improves F1 by 35.3% and tuple accuracy by 52.9%. CORVA achieves the highest implicit recognition rate at 45.8%, which is 10.5 points ahead of the runner-up. Our results demonstrate that reliable implicit PK extraction requires validation as a first-class component rather than a post-processing step, which offers a practical framework for scalable knowledge extraction from biomedical literature. The code and data are available at <https://github.com/wuqinhao/CORVA>.

## 1 Introduction

Drug development is a costly and multi-stage process in which quantitative pharmacology plays a crucial role. Pharmacokinetic (PK) models guide dosing design, toxicity analysis, and clinical trial planning throughout the development pipeline (Danhof et al., 2008; Mould and Upton, 2012). However, PK parameter values remain dispersed across thousands of publications, embedded in free text rather than structured form (Hunter and Cohen, 2006). As pharmacological literature

continues to grow and encompasses increasingly diverse patient populations, the lack of machine-readable PK resources remains a persistent bottleneck hindering computational pharmacology (Recino et al., 2025).

Automating the extraction of structured PK information from biomedical literature has attracted growing attention in natural language processing (NLP) (Hernandez et al., 2024b; Sholehrasa et al., 2025). However, a fundamental challenge remains unaddressed: several PK parameters are not stated explicitly. Values such as  $T_{max}$  and oral absorption rate constant  $k_a$  must instead be inferred from context, resolved across heterogeneous linguistic and numeric expressions, and sometimes reconstructed through multi-sentence reasoning.

Prior work has addressed related problems through text mining (Wu et al., 2013), named entity recognition (NER) of PK terms (Hernandez et al., 2024a), and large language model (LLM) based extraction from structured tables (Sholehrasa et al., 2025). However, these approaches target parameters that appear in explicit, recognisable semantic form, which makes it challenging to extract implicit parameters.

An LLM-based system offers a promising path forward, given its capacity for contextual inference over domain-specific text (Sholehrasa et al., 2025). However, LLMs generate responses based on statistical patterns rather than verified facts (Ji et al., 2023), and this tendency is especially dangerous in implicit extraction. Reasoning-induced hallucination (Lones, 2026) can introduce non-existent parameter values, assign values to the wrong parameter, or propagate incorrect units, resulting in corrupt downstream pharmacometric modelling.

Furthermore, the challenge of implicit reasoning is also domain-specific (Hernandez et al., 2024a). Extracting these parameters requires not only domain-grounded reasoning to identify the correct value, but also rigorous validation to guard

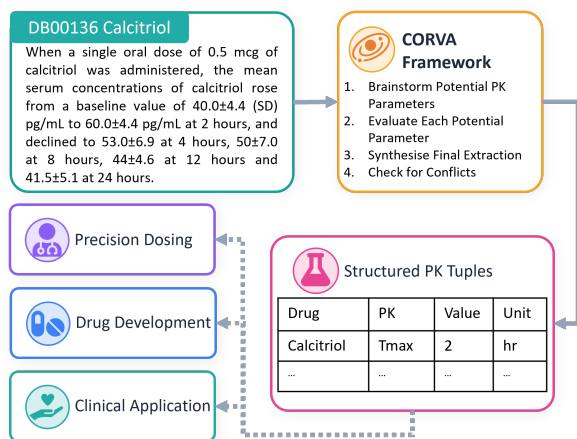


Figure 1: The workflow of the CORVA Framework.

against plausible yet incorrect inferences. This condition reveals a fundamental trade-off of reasoning and validation in NLP. Reasoning strategies improve recall in implicit cases by enabling inference beyond the surface form, but systematically amplify hallucinations, while validation strategies suppress false positives but risk over-pruning valid inferences. Our preliminary results suggest that neither reasoning nor validation alone is sufficient.

We propose **CORVA**, a COnceptual Reasoning and VALidation framework (workflow shown in Figure 1) that grounds LLM-generated extraction candidates against domain constraints, facilitating drug development and clinical applications. Rather than treating validation as a post-processing step, CORVA integrates it into the extraction process, allowing the system to selectively accept inferences that satisfy domain-grounded criteria while rejecting hallucinated candidates. We construct tuples to evaluate implicit PK parameter extraction from DrugBank (Wishart et al., 2018) absorption descriptions. The CORVA framework yields a reliable extraction pipeline that achieves the best balance of precision and recall across our benchmark. The contributions of this paper are as follows.

- We introduce the problem of implicit PK parameter extraction, characterised by semantic implicitness, multi-candidate interleaving, and the requirement for domain-grounded reasoning.
- We propose CORVA, a validation-enhanced LLM-based framework that extracts structured PK tuples from unstructured biomedical text, improving reliability in implicit extraction settings.

- We construct a dataset of absorption-related PK parameters from DrugBank, in which a substantial portion of the annotations involve implicit expressions, enabling realistic evaluation of implicit extraction.
- We conduct a systematic analysis of reasoning and validation mechanisms in LLM-based extraction, showing their complementary roles and trade-offs in implicit information retrieval.

Our proposed method can provide implicit PK parameters while reducing hallucination rates compared to other approaches.

## 2 Related Works

Biomedical information retrieval has attracted increasing attention with advances in information extraction techniques, which aim to unlock structured knowledge from the growing biomedical literature (Luo et al., 2022; Goyal and Singh, 2025). Existing studies have focused predominantly on named entity recognition (NER), relation extraction (RE), knowledge graph construction, and ontology-based formalisation (Perera et al., 2020; Luo et al., 2022; Abu-Salih et al., 2023; Dong et al., 2023; Soman et al., 2024; Goyal and Singh, 2025), with transformer-based models such as BERT (Devlin et al., 2019) and BioBERT (Lee et al., 2020) becoming the dominant backbone by capturing contextual semantics more effectively than static embeddings.

Despite their strong performance on explicit entity and relation extraction, these approaches are fundamentally designed for information that is explicitly expressed in the text. As a result, they rely heavily on surface-level matching between linguistic patterns and predefined labels, and struggle when parameter identity must be inferred from context rather than directly observed. In addition, pipeline-based NER-RE systems are vulnerable to error propagation (Giorgi et al., 2019), and biomedical texts frequently contain overlapping entities and complex relational structures that are difficult to resolve with conventional pipelines (Yang et al., 2023). Consequently, these methods are insufficient for scenarios involving implicit parameter expressions, where the target variable must be derived through contextual reasoning rather than explicit mention (Chen et al., 2023).

Alternatively, LLMs have demonstrated strong capability for joint NER-RE by following

task-specific instructions, generating schema-constrained outputs, and reasoning over complex structures without requiring task-specific fine-tuning (Dagdelen et al., 2024; Obeidat et al., 2025). This flexibility makes LLMs particularly attractive for low-resource biomedical extraction settings. However, LLMs generate outputs based on statistical patterns rather than verified facts, and are therefore prone to hallucination, which can corrupt downstream structured outputs (Nagar et al., 2025; Wang et al., 2025). This issue is especially pronounced in implicit extraction settings, where the model must infer parameter identity beyond surface-level cues, often leading to incorrect or unsupported predictions (Li et al., 2024).

Recent work has explored biomedical NLP approaches for PK-related entity recognition and extraction. Hernandez et al. (2024a) introduced a dedicated PK-NER corpus and a fine-tuned BioBERT model, which was later extended to an end-to-end (E2E) extraction pipeline evaluated on the PRED corpus (Hernandez et al., 2024b). AutoPK (Sholehrasa et al., 2025) further leveraged LLMs for PK parameter extraction from structured tables. However, these approaches focus on explicit PK terms and are not designed to handle implicit parameter expressions, where parameter identity must be inferred from context. Moreover, existing datasets do not explicitly account for implicit cases, limiting their effectiveness in realistic extraction scenarios. To address this gap, we propose a zero-shot LLM-based framework for implicit PK parameter extraction, incorporating structured reasoning and validation to improve reliability.

### 3 Task Definition

We introduce the task of extracting implicit PK parameters from literature text. PK information may be explicitly stated in standard terminology or implicitly inferred from context. This task is challenging because biomedical text often contains incomplete expressions and implicit descriptions that require reasoning beyond semantic keyword matching.

Given a set of sentences of drugs, the goal is to extract the structured tuples  $\mathcal{T}$  from the input. We define the extraction target for drug  $d_i$  consisting  $N_i$  PK tuples:

$$\mathcal{T}_i = \{(\mathcal{P}_i, \mathcal{V}_i, \mathcal{U}_i)\},$$

where  $\mathcal{P}_i$ ,  $\mathcal{V}_i$ , and  $\mathcal{U}_i$  denote the parameter names,

values, and units of tuples, respectively. The full dataset is then  $\mathbf{T} = \{\mathcal{T}_i\}_{i=1}^m$  over  $m$  drugs.

Among the  $N_i$  tuples of Drug  $d_i$ , let  $K_i$  denote the number of tuples whose parameter name appears explicitly in the text. The remaining  $N_i - K_i$  tuples are implicit. Their parameter identity must be inferred from context rather than matched directly to a surface form. The goal of our task is to recover the complete  $\mathcal{T}_i$ , including both explicit and implicit tuples, from the free-text PK description.

For example, in the sentence “ $C_{max}$  of 28 ng/ml is reached 2 hours after initial administration with an AUC of about 200 ng · h/ml”, the parameter  $T_{max}$  is not explicitly mentioned. A correct extraction requires inferring the tuple ( $T_{max}$ , 2, hours) from the temporal cue embedded in the description of when  $C_{max}$  is attained. Furthermore, implicit parameters further appear in heterogeneous forms (e.g., the same value may be written as “two hours”). Such cases illustrate the need for reasoning beyond surface-level matching.

## 4 Dataset

We construct a corpus for the implicit PK extraction, using the absorption field of DrugBank (Version 5.1.12) (Wishart et al., 2018). The absorption field of each entry contains free-text descriptions of how a drug enters systemic circulation, which is rarely stated explicitly and instead must be inferred from context. For example,  $T_{max}$  is implied by a description of when peak concentration is observed. After filtering for non-empty entries, our corpus comprises 2480 drug descriptions.

We constructed a dataset of absorption-related tuples (parameter, value, unit) through an in-house annotation process conducted by two expert pharmacometricians (senior and junior) with over ten and two years of experience, respectively. Annotators followed the guidelines detailed in Appendix A, labelling all valid PK tuples including implicit expressions, multi-value interleaving, and cases requiring cross-sentence reasoning. The full annotation process spanned two weeks of dedicated expert effort. The corpus contains 3605 PK-related mentions, from which 469 absorption-related tuples were annotated and retained for evaluation. Notably, 223 of these (47.6%) involve implicit parameter expressions. The Inter-Annotator Agreement score is measured by Cohen’s  $\kappa$  (Cohen, 1960) with a score of 0.875.

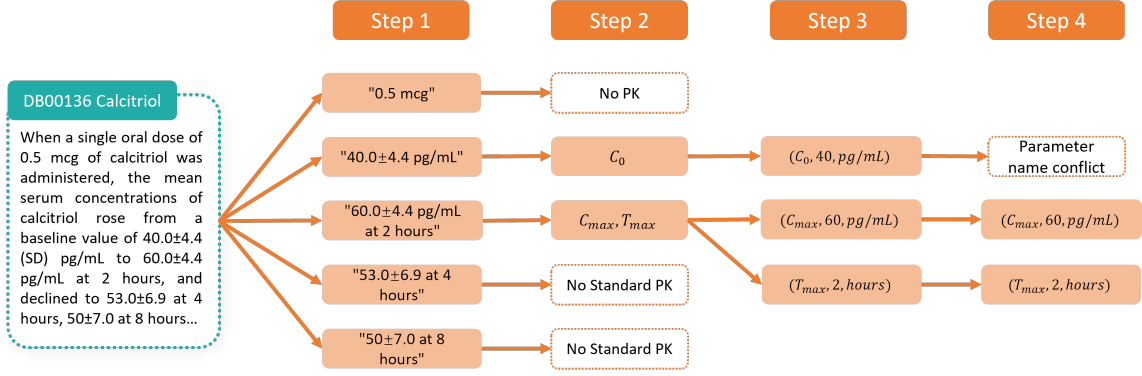


Figure 2: Illustration of the reasoning topology of CORVA. Different steps of CORVA based on the example of calcitriol, expanding each candidate into a separate reasoning path and resolving it over four steps, so that only validated tuples reach the output. The solid node represents the valid candidates under certain step, while the dashed node is pruned by validation.

## 5 Extraction Strategy Formulation

To illustrate our proposed CORVA framework for the implicit PK parameter extraction, we will state it in the following stages to illustrate the necessity of reasoning and validation with LLMs. Note that implicit extraction in this work refers to implicit parameter identity resolution. Numerical values must still appear explicitly in the text and are never hallucinated or generated. The prompts for each stage are listed in Appendix B.

**Stage 1: Implicit Extraction Prior.** The first stage establishes the baseline by prompting the LLM with only a task description  $c$ , with no structural constraints, demonstrations, or reasoning guidance (Zhang et al., 2023). This isolates the model’s intrinsic capacity for extracting PK parameters. For each drug  $d_i$ , we decode the highest-probability set of PK tuples:

$$\hat{\mathcal{Y}}_i \sim P_\theta(\mathcal{Y} \mid x_i, c), \quad (1)$$

where  $P_\theta(\cdot \mid x_i, c)$  is the conditional distribution parametrised by  $\theta$  over candidate tuple sets  $\mathcal{Y}$ , given the input corpus  $x_i$  of drug  $d_i$  and the task description  $c$ . The candidate space  $\mathcal{T}_i$  ranges over all admissible PK tuples, including those whose parameter or value is not stated explicitly in  $x_i$  and must be inferred.  $\hat{\mathcal{Y}}_i$  is the decoded output. This stage reveals the upper bound of zero-shot extraction and exposes the model’s tendency to infer implicit parameters without grounding constraints. It serves as the foundation onto which subsequent stages stack.

**Stage 2: Rule-based Structural Extraction.** Building on Stage 1, we introduce a set of structural constraints  $c_0$  that specify the required output

format and extraction rules, formalising the extraction as  $\hat{\mathcal{Y}}_i \sim P_\theta(\mathcal{Y} \mid x_i, c, c_0)$  (He et al., 2024). The constraints  $c_0$  enforce tuple structure, restrict admissible parameter names, and specify unit normalisation rules. Over Stage 1, this stage improves output consistency and reduces format-level errors, but does not address the underlying reasoning required to resolve implicit parameters.

**Stage 3: Procedural Reasoning.** On top of Stage 2, Stage 3 adds an explicit reasoning step  $R$  before the final extraction (Wang et al., 2024). The generation is decomposed as  $\hat{\mathcal{Y}}_i \sim P_\theta(\mathcal{Y} \mid x_i, c, R) \cdot P_\theta(R \mid x_i, c)$ , where  $R = \{r_1\}$  represents a single reasoning path. Specifically, we condition the reasoning on the oral administration route, seeking the proper implicit parameters. Compared to Stage 2, this stage relaxes the semantic dependency and improves recall on implicit cases by enabling multi-step inference.

**Stage 4: CORVA (Full Framework).** Combining the reasoning and validation mechanism (Yao et al., 2023; Adam et al., 2025), we propose the CORVA framework for implicit PK extraction. Rather than committing to a single reasoning path from Stage 3, CORVA generates  $n$  candidate reasoning paths  $R = \{r_1, r_2, \dots, r_n\}$  and scores each against domain constraints:

$$\hat{\mathcal{Y}}_i \sim P_\theta(r_j \mid x_i, c) P_\theta(\mathcal{Y} \mid x_i, c, c_0, r_j) S(r_j), \quad (2)$$

where  $S(r_j) \in [0, 1]$  is a validation score reflecting the proportion of tuples satisfying all validation constraints (e.g., parameter-unit consistency, administration route-absorption parameter consistency, numeric value check, and tuple completeness check).

The reasoning and validation mechanism of CORVA is shown in Figure 2. By generating multiple candidate reasoning paths in parallel and carrying each independently through parameter assignment and tuple construction, the model avoids the premature commitment. Validation is applied to every branch then prunes the invalid candidates before the final tuples are produced. Validation enforces domain constraints such as parameter–unit compatibility, route consistency, and numeric plausibility, ensuring that only pharmacologically consistent tuples are retained. The details of the validation are included in Appendix F.

## 6 Experiments

Our proposed CORVA and baseline methods are evaluated against the annotations constructed from DrugBank absorption texts, as described in Section 4. We refer to Stage 1 (Implicit Extraction Prior), Stage 2 (Rule-based Structural Extraction), Stage 3 (Procedural Reasoning), and the proposed CORVA as “Implicit”, “Rule-based”, “Procedural”, and “CORVA”, respectively, throughout all tables and figures. We evaluate Drug-PK pair extraction using precision (P), recall (R), and F1-score (Campos et al., 2012). Extraction accuracy is assessed separately for PK values and units, as well as at the tuple level.

We use Mistral-7B (Jiang et al., 2023) to evaluate all strategies, focusing on text-only PK extraction in a controlled zero-shot setting. Mistral was accessed via its API and was not constrained to local hardware resources. All other models were deployed locally on a workstation with an NVIDIA GeForce RTX 2080 Super Max-Q GPU (8 GB VRAM) and 32 GB RAM. The proposed framework is model-agnostic and can be applied to any instruction-tuned LLMs.

### 6.1 Strategic Stage Performance

The Drug-PK matching performance and extraction accuracy for each Mistral-based strategy are listed in Table 1. The comprehensive results are listed in Appendices C and D. CORVA achieves the best overall performance, obtaining the highest recall and F1-score, indicating the effectiveness at identifying PK parameters. In contrast, Procedural Reasoning achieves slightly higher precision, while CORVA provides a better overall balance, suggesting that reasoning is essential for reducing false positives but may miss more valid Drug-PK

Table 1: Different strategy performance with Mistral.

| Method     | R             | P             | F1            | Value         | Unit          | Tuple         |
|------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Implicit   | 0.7179        | 0.3556        | 0.4756        | 0.6624        | 0.6838        | 0.6474        |
| Rule-based | 0.6197        | 0.3329        | 0.4331        | 0.5363        | 0.5940        | 0.5235        |
| Procedural | 0.6752        | <b>0.4370</b> | 0.5306        | 0.6218        | 0.6453        | 0.6154        |
| CORVA      | <b>0.7885</b> | 0.4028        | <b>0.5332</b> | <b>0.6902</b> | <b>0.7628</b> | <b>0.6859</b> |

Table 2: Best strategy performance compared to the benchmark. McNemar’s test (Yan et al., 2017) is applied to compare the proposed method with the E2E extraction pipeline with the fine-tuned BioBERT. The significance level of McNemar’s test is denoted as  $^{\dagger}$  for  $p < 0.005$ .

| Method   | R             | P             | F1                        | Value         | Unit          | Tuple                     |
|----------|---------------|---------------|---------------------------|---------------|---------------|---------------------------|
| SciSpaCy | 0.2985        | 0.1571        | 0.2058                    | 0.1940        | 0.1620        | 0.1514                    |
| AutoPK   | <b>0.9915</b> | 0.0783        | 0.1451                    | 0.5800        | 0.7249        | 0.4051                    |
| BioBERT  | 0.6368        | 0.2853        | 0.3941                    | 0.4573        | 0.5726        | 0.4487                    |
| CORVA    | 0.7885        | <b>0.4028</b> | <b>0.5332<sup>†</sup></b> | <b>0.6902</b> | <b>0.7628</b> | <b>0.6859<sup>†</sup></b> |

pairs at the cost of recall. CORVA consistently achieves the best performance across value, unit, and complete tuple matching, with tuple accuracy of 0.6859. These ablation results confirm that each component of the pipeline contributes incrementally to procedural reasoning, which improves inference over implicit expressions, while validation corrects erroneous candidates introduced by unconstrained generation. Overall, they indicate that both reasoning and validation are necessary for reliable implicit PK extraction, and that neither alone is sufficient.

### 6.2 Comparison with Baseline Methods

We compare CORVA against three baselines, listed in Table 2: the E2E extraction pipeline of Hernandez et al. (2024b), instantiated with SciSpaCy (Neumann et al., 2019) and with BioBERT (Hernandez et al., 2024a) as its NER component, and AutoPK (Sholehrasa et al., 2025). Each baseline is fine-tuned on the PK extraction task following its original configuration.

CORVA achieves the highest scores in five metrics except recall. Notably, AutoPK attains the highest Recall, indicating it retrieves nearly every PK parameter. This is attributable to the AutoPK pipeline, which provides broad coverage by mapping extracted mentions to a predefined parameter vocabulary and effectively maximising retrieval at the expense of precision. However, its substantially low Precision indicates a high rate of spurious extractions, and its insufficient “Value” and “Tuple” scores reveal difficulty in resolving implicit or context-dependent parameter expressions.

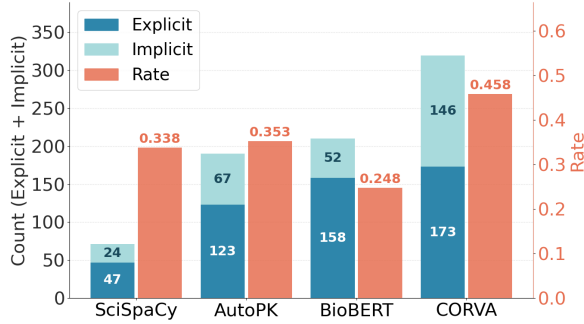


Figure 3: Recognised implicit PK extraction across methods. The comparison of explicit and implicit parameter extraction counts (left y-axis, stacked bars) and implicit recognition rates (right y-axis, orange bars).

Table 3: Invalid output rate of each strategy per PK parameter (name), value, and unit.

| Method     | Name (%) | Value (%) | Unit (%) |
|------------|----------|-----------|----------|
| Implicit   | 0.76     | 20.03     | 1.49     |
| Rule-based | 1.04     | 3.47      | 1.47     |
| Procedural | 3.01     | 14.57     | 9.31     |
| CORVA      | 2.21     | 11.28     | 3.48     |

Among the baseline methods, BioBERT yields the strongest overall performance, balancing Recall and Precision to achieve an F1 of 0.3941 and Tuple accuracy of 0.4487. The CORVA outperforms BioBERT by about 35.3% in F1 and 52.9% in tuple accuracy significantly, highlighting the benefit of conceptual reasoning and validation mechanism for implicit PK parameter extraction.

**Implicit Extraction Performance.** We further analyse the implicit PK extraction rate of the proposed CORVA and the baseline methods shown in Figure 3, which breaks down correctly extracted parameters into explicit and implicit mentions, along with their implicit recognition rates. CORVA extracts both explicit and implicit tuples, achieving the highest implicit recognition rate of 0.458. In contrast, baseline methods recover more explicit extraction but recognise far fewer implicit ones, confirming that their pattern-matching approach struggles with implicit information extraction. These results demonstrate that the reasoning mechanism in the proposed CORVA is particularly effective at extracting implicit PK parameters.

### 6.3 Failure Mode Analysis

Table 3 reports the invalid extraction rate of each strategy, where invalid outputs are defined per column as follows: for *Name*, any generic placeholder (e.g., “PK parameter”) or empty output is considered invalid; for *Value*, an output is valid only if it

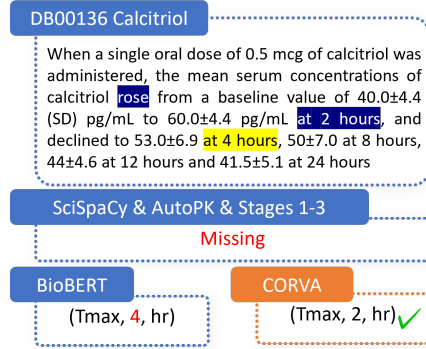


Figure 4: Case of combining the reasoning and validation for the hidden information for  $T_{max}$  extraction.

contains a numeric value; and for *Unit*, an output is valid only if it is a non-empty, non-numeric string.

Combined with Table 1, the four stages trace a clear precision–recall trade-off. The Implicit Extraction Prior reflects the model’s intrinsic nature: it extracts widely without constraints, providing high recall but the most field-level error. Adding structural constraints in the Rule-based Structural Extraction flips this behaviour: structural constraints produce the fewest field-level errors, but they suppress coverage, lowering recall, precision, and F1. The Procedural Reasoning restores balance through reasoning, achieving the highest precision, at the cost of reintroducing some field-level error. CORVA reconciles all three: multipath reasoning delivers the highest recall, while validation prunes implausible tuples to restore precision and field accuracy, yielding the best F1 and the best extraction scores.

Constraints improve accuracy but are too conservative, reasoning provides coverage but is too permissive, and neither reaches a good operating point alone. CORVA combines these two: reasoning supplies recall over implicit cases, and validation enforces precision. Thus, CORVA attains the best balance and the strongest overall performance.

### 6.4 Extraction Case of Validation

Figure 4 presents a case that requires both reasoning and validation for correct extraction. In this example, the text does not explicitly mention PK parameter cues, such as  $C_{max}$  or  $T_{max}$ , which likely causes SciSpaCy, AutoPK, and Stages 1-3 to miss the parameter entirely. In this extreme case, AutoPK misses the sentence entirely, as it describes a change in mean serum concentration that deviates from its few-shot examples. BioBERT recovers the parameter but assigns an incorrect value, likely

extracted from a later part of the sentence without grounding in the absorption context. The text describes a concentration curve that rises until 2 hours and then declines, implying that  $T_{max}$  occurs at 2 hours. Among the compared strategies, only CORVA identifies this implicit temporal cue and extracts the corresponding tuple. This case illustrates the advantage of CORVA in handling implicit PK information that requires inference beyond surface-level matching.

## 7 Discussion

The results suggest that reasoning and validation play complementary roles: reasoning improves coverage of implicit cases, whereas validation mitigates hallucinations introduced by unconstrained inference. CORVA achieves the strongest overall performance, with its reasoning and validation mechanism proving particularly effective for implicit parameter recognition, outperforming pattern-matching baselines, i.e., SciSpaCy, AutoPK, and BioBERT. Although AutoPK achieves the highest recall, its low precision and tuple accuracy reveal its inability to extract implicit PK parameters. Compared with the best baseline method, the improvement in CORVA results is statistically significant according to McNemar’s test.

Different strategies exhibit different error patterns. CORVA provides a better balance by combining inference with validation, allowing the model to recover implicit PK information while reducing unsupported outputs. These findings indicate that validation is essential when deploying LLM-based extraction systems in high-stakes biomedical settings.

This is particularly important in pharmacological applications, where incorrect parameter extraction may propagate errors into downstream pharmacometric models for drug development and clinical applications.

Despite CORVA’s gains, several failure modes remain. The model still misassigns parameters from misleading contextual cues, and struggles to select the correct value when multiple numerical candidates appear in the same sentence. We also observe incorrect unit generation and hallucinated parameter–value associations, where the extracted tuple is internally plausible but unsupported by the text. Cases requiring long-range reasoning across sentences remain difficult, as the relevant evidence is often distributed rather than local. These categories

point to the main open challenges for implicit PK extraction and mark the direction for future work.

## 8 Conclusion

We proposed CORVA, a validation-enhanced LLM framework for extracting implicit PK parameters from biomedical text. By combining reasoning and validation, CORVA achieves strong performance and improves reliability in implicit extraction settings. Our results demonstrate that accurate PK extraction requires not only reasoning over context but also explicit validation to control hallucination. This enables scalable construction of pharmacological knowledge bases from unstructured text, reducing reliance on manual curation in drug development workflows. Future work will explore CORVA’s robustness across models and domains, its integration with human-in-the-loop validation, and the use of domain-specific foundation models for this task.

## Limitations

While CORVA demonstrates strong performance on implicit PK parameter extraction, several limitations should be acknowledged.

**Task scope.** This work focuses specifically on absorption-related PK parameters extracted from DrugBank absorption descriptions. It does not cover the full PK spectrum, leaving distribution, metabolism, and excretion parameters undressed.

**Few-shot setting.** Although few-shot prompting has shown promise in biomedical NLP, we do not include few-shot experiments in this work. The primary reason is the limited size of our dataset.

**Model scale and computational cost.** All experiments are conducted using small LLMs, chosen to reflect deployment constraints in real-world pharmacological settings where large proprietary models may be unavailable or cost-prohibitive.

## Ethical Statements

The DrugBank dataset used in this work is released under the CC BY-NC 4.0 license. In accordance with its terms, we do not redistribute the raw DrugBank data. The original dataset can be accessed at <https://go.drugbank.com/>. We release our annotations upon request, alongside DrugBank entry identifiers to allow reconstruction of the full dataset upon obtaining access.

All annotators were fairly compensated professionals employed by the authors' institution. The annotation process was conducted over one month, during which annotators extracted and validated all absorption-related tuples following the guidelines detailed in Appendix A. The dataset contains no personally identifiable information, as all entries are sourced from publicly available DrugBank.

## Acknowledgments

This work was sponsored by Apriko Research. The authors acknowledge that this research originated during the first author's affiliation with the Leiden Academic Centre for Drug Research (LACDR), Leiden University. We thank all annotators for their careful and dedicated effort in constructing the dataset. AI-assisted tools were used to partially refine the writing.

## References

- Bilal Abu-Salih, Muhammad Al-Qurishi, Mohammed Alweshah, Mohammad Al-Smadi, Reem Alfayez, and Heba Saadeh. 2023. Healthcare knowledge graph construction: A systematic review of the state-of-the-art, open issues, and opportunities. *Journal of Big Data*, 10(1):81.
- Hammaad Adam, Junjing Lin, Jianchang Lin, Hillary Keenan, Ashia Wilson, and Marzyeh Ghassemi. 2025. Clinical information extraction with large language models: a case study on organ procurement. In *AMIA Annual Symposium Proceedings*, volume 2024, page 115.
- David Campos, Sérgio Matos, and José Luís Oliveira. 2012. Biomedical named entity recognition: a survey of machine-learning tools. In *Theory and applications for advanced text mining*. IntechOpen.
- Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. 2023. Learning in-context learning for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13661–13675.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418.
- Meindert Danhof, Elizabeth CM De Lange, Oscar E Della Pasqua, Bart A Ploeger, and Rob A Voskuyl. 2008. Mechanism-based pharmacokinetic-pharmacodynamic (pk-pd) modeling in translational drug research. *Trends in pharmacological sciences*, 29(4):186–191.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Hang Dong, Jiaoyan Chen, Yuan He, Yinan Liu, and Ian Horrocks. 2023. Reveal the unknown: Out-of-knowledge-base mention discovery with entity linking. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 452–462.
- John Giorgi, Xindi Wang, Nicola Sahar, Won Young Shin, Gary D Bader, and Bo Wang. 2019. End-to-end named entity recognition and relation extraction using pre-trained language models. *arXiv preprint arXiv:1912.13415*. Accessed: 2026-05-24.
- Nandita Goyal and Navdeep Singh. 2025. Named entity recognition and relationship extraction for biomedical text: A comprehensive survey, recent advancements, and future research directions. *Neurocomputing*, 618:129171.
- Pengfei He, Zitao Li, Yue Xing, Yaling Li, Jiliang Tang, and Bolin Ding. 2024. Make llms better zero-shot reasoners: Structure-orientated autonomous reasoning. *arXiv preprint arXiv:2410.19000*. Accessed: 2026-05-24.
- Ferran Gonzalez Hernandez, Quang Nguyen, Victoria C Smith, José Antonio Cordero, Maria Rosa Ballester, Màrius Duran, Albert Solé, Palang Chotsiri, Thanaporn Wattanakul, Gill Mundin, and 1 others. 2024a. Named entity recognition of pharmacokinetic parameters in the scientific literature. *Scientific Reports*, 14(1):23485.
- Ferran Gonzalez Hernandez, Victoria Smith, Quang Nguyen, Palang Chotsiri, Thanaporn Wattanakul, José Antonio Cordero, Maria Rosa Ballester, Albert Solé, Gill Mundin, Watjana Lilaonitkul, and 1 others. 2024b. End-to-end relation extraction of pharmacokinetic estimates from the scientific literature. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 144–154.
- Lawrence Hunter and K Bretonnel Cohen. 2006. Biomedical language processing: what's beyond pubmed? *Molecular cell*, 21(5):589–594.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*. Accessed: 2026-05-24.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.
- Mingchen Li, Huixue Zhou, Han Yang, and Rui Zhang. 2024. Rt: a retrieving and chain-of-thought framework for few-shot medical named entity recognition. *Journal of the American Medical Informatics Association*, 31(9):1929–1938.
- Michael A Lones. 2026. Pitfalls and risks of generative ai in machine learning. *Patterns*, 7(5).
- Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. 2022. Biored: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23(5):bbac282.
- Diane R Mould and Richard N Upton. 2012. Basic concepts in population modeling, simulation, and model-based drug development. *CPT: pharmacometrics & systems pharmacology*, 1(9):1–14.
- Aishik Nagar, Viktor Schlegel, Thanh-Tung Nguyen, Hao Li, Yuping Wu, Kuluhan Binici, and Stefan Winkler. 2025. Llms are not zero-shot reasoners for biomedical information extraction. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 106–120.
- Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacey: fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP workshop and shared task*, pages 319–327.
- Motasem S Obeidat, Md Sultan Al Nahian, and Ramakanth Kavuluru. 2025. Do llms surpass encoders for biomedical ner? In *2025 IEEE 13th International Conference on Healthcare Informatics (ICHI)*, pages 352–358. IEEE.
- Nadeesha Perera, Matthias Dehmer, and Frank Emmert-Streib. 2020. Named entity recognition and relation detection for biomedical information extraction. *Frontiers in cell and developmental biology*, 8:673.
- Asha Recino, Melissa LD Rayner, Jennifer L Rohn, Oscar Della Pasqua, Brian Carter, Anna Cupani, Annegret Dahlmann-Noor, Thomas Foltynie, Jennifer Furman, Stephen Hobbiger, and 1 others. 2025. Therapeutic innovation in drug repurposing: challenges and opportunities. *Drug Discovery Today*, 30(7):104390.
- Hossein Sholehrasa, Amirhossein Ghanaatian, Doina Caragea, Lisa A Tell, Jim E Riviere, and Majid Jaber-Douraki. 2025. Autopk: Leveraging llms and a hybrid similarity metric for advanced retrieval of pharmacokinetic data from complex tables and documents. In *2025 IEEE 37th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 338–346. IEEE.
- Karthik Soman, Peter W Rose, John H Morris, Rabia E Akbas, Brett Smith, Braian Peetoom, Catalina Villouta-Reyes, Gabriel Ceron, Yongmei Shi, Angela Rizk-Jackson, and 1 others. 2024. Biomedical knowledge graph-optimized prompt generation for large language models. *Bioinformatics*, 40(9):btac560.
- Lei Wang, Yinyao Ma, Wenshuai Bi, Hanlin Lv, and Yuxiang Li. 2024. An entity extraction pipeline for medical text records using large language models: analytical study. *Journal of Medical Internet Research*, 26:e54580.
- Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025. Gpt-ner: Named entity recognition via large language models. In *Findings of the association for computational linguistics: NAACL 2025*, pages 4257–4275.
- David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, and 1 others. 2018. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082.
- Heng-Yi Wu, Shreyas Karnik, Abhinata Subhadarshini, Zhiping Wang, Santosh Philips, Xu Han, Chienwei Chiang, Lei Liu, Malaz Boustani, Luis M Rocha, and 1 others. 2013. An integrated pharmacokinetics ontology and corpus for text mining. *BMC bioinformatics*, 14(1):35.
- Fengxia Yan, Mayberry Robert, and Yonggang Li. 2017. Statistical methods and common problems in medical or biomedical science research. *International journal of physiology, pathophysiology and pharmacology*, 9(5):157.
- Chenglin Yang, Jiamei Deng, Xianlai Chen, and Ying An. 2023. Spbere: Boosting span-based pipeline biomedical entity and relation extraction via entity information. *Journal of Biomedical Informatics*, 145:104456.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.
- Jianzhang Zhang, Yiyang Chen, Chuang Liu, Nan Niu, and Yinglin Wang. 2023. Empirical evaluation of chatgpt on requirements information retrieval under zero-shot setting. In *2023 International Conference*

## A Annotation Guidelines

### A.1 Task Definition

Annotators are asked to identify all absorption-related pharmacokinetic (PK) parameter expressions from a given DrugBank absorption description and record them as structured tuples of the form (parameter, value, unit).

### A.2 Implicit Expression Handling

Annotators must apply domain knowledge to resolve which parameter a given value refers to, rather than relying on surface-level keyword matching. The following principles govern implicit cases:

**Inferential Identification.** If the parameter name does not appear verbatim but can be unambiguously determined from context, the tuple should be annotated.

**Multi-Candidate Disambiguation.** When multiple numeric expressions appear in a single sentence, annotators must identify which values correspond to absorption parameters and which do not.

### A.3 Exclusion Criteria

Annotators should **not** record a tuple if any of the following conditions apply:

- The parameter is a distribution, metabolism, or elimination parameter (e.g., half-life, volume of distribution, clearance) unless it appears in direct support of an absorption inference.
- The value is speculative, model-predicted, or population-derived without a specific reported figure in the text.
- The parameter identity cannot be determined with confidence even after applying domain reasoning.
- The value is a relative comparison (e.g., “high” or “quickly absorb”) without an absolute anchor.

### A.4 Annotation Process

Each DrugBank absorption entry was independently annotated by two expert annotators with backgrounds in pharmacology. Annotators were trained on a shared pilot set of 30 entries prior to the main annotation phase, during which the guidelines were refined iteratively based on observed

disagreements. Annotations were recorded using a structured spreadsheet interface, with one row per tuple per entry.

### A.5 Conflict Resolution

Disagreements between the two annotators were resolved through a two-stage adjudication protocol. For each entry with at least one disagreement, the two annotators first reviewed each other’s annotations independently and attempted to reach consensus through discussion. Disagreements arising from ambiguous parameter identity or unit inference were resolved by referring to the PK parameter and, where necessary, consulting the original source literature cited by DrugBank.

### A.6 Quality Control

Inter-annotator agreement was computed using Cohen’s  $\kappa$  (Cohen, 1960) at the tuple level, treating each tuple as a binary match. A tuple was counted as agreed upon if both annotators produced an identical triple; partial matches were counted as disagreements to enforce strict quality standards. The final dataset retains only tuples confirmed by both annotators or resolved through adjudication.

## B Extraction Strategy Template

For reproducibility, we provide the exact prompts used in all experiments. Each stage employs a system prompt that delivers high-level task instructions and constrains the output format, with stage-specific user prompts detailed in the subsections below. The four stages differ in the degree of structural constraint, reasoning guidance, and validation applied to the user prompts, reflecting the progressive pipeline design.

```
[SYSTEM PROMPT]
You are a pharmacometric expert.
Extract the Pharmacokinetic (PK)
data from the text and output it as
CSV-formatted rows strictly.

Return format:
- The columns must be in this exact
  order: 'PK parameter', 'value',
  'unit'.
- Do not include headers.
- Do not include any introductory or
  concluding text.
- Output ONLY the comma-separated
  values. Example output:
  Cmax,15.2,mg/L
```

### B.1 Stage 1: Implicit Extraction Prior

Stage 1 defines the PK extraction task in the prompt as Equation 1, asking the LLM only to prioritise accuracy and to answer NIL when unsure. This isolates the model’s intrinsic ability to recover PK parameters, including implicit ones, from context alone, which is why we call it the implicit extraction prior.

```
[USER PROMPT]
Task: You will receive a parameter
      mentioned in scientific literature.
      Your task is to accurately extract
      the PK parameter name, its value,
      and the unit of measurement.

- Accuracy is paramount. If the text
  does not precisely refer to any of
  the concepts listed below, please
  answer NIL.
- If the text does not refer to a PK
  parameter (e.g. part of a
  pharmacodynamic parameter, or a
  PBPK, chemical, in vitro, clinical,
  or other unrelated term), please
  answer NIL.
- If you are unsure, please answer NIL.

Text: ‘{text}’

Follow the steps above and output the
final data row.
```

### B.2 Stage 2: Rule-based Structural Extraction

Stage 2 augments Stage 1’s prompt with a set of explicit PK extraction rules ( $c_0$ ) that constrain how parameters and values are identified and formatted. These structural constraints build directly on Stage 1 while preserving its accuracy requirement and the instruction to answer NIL when unsure.

```
[USER PROMPT]
Task: Extract Pharmacokinetic (PK) data
      from the text provided below.

You possess the skill to extract PK
parameters from text. Apply this
skill to the following sentence by
strictly adhering to these
extraction rules:

**PK Extraction Rules:**
1. A PK parameter is identified as a
   sequence: [Descriptive Name] ...
   [Numerical Value] [Unit], or,
   [Descriptive Name1][Descriptive
   Name2] ... [Numerical Value1]
   [Unit1] ... [Numerical Value2]
   [Unit2].
2. Locate all such sequences within
   the provided sentence. Including
   also similar sequences that may be
   part of a larger phrase.
```

3. For each sequence, identify the 'PK parameter' as the descriptive name, the 'Value' as the numerical value, and the 'Unit' as the unit of measurement.
4. Structure the output as a CSV-formatted row with columns: 'PK parameter', 'value', 'unit'.

Only include values explicitly mentioned. If you are unsure, return NIL only.

Text: "{text}"

Follow the steps above and output the final data row.

### B.3 Stage 3: Procedural Reasoning

Stage 3 extends Stage 2 by adding a single chain of reasoning steps ( $R = \{r_1\}$ ) that guide the LLM to reason explicitly before extraction. It retains all of Stage 2's structural constraints and the instruction to answer NIL when unsure, adding only the procedural reasoning layer.

[USER PROMPT]

Task: Extract Pharmacokinetic (PK) data from the text provided below.

1. Step 1: Identify the drug name.
2. Step 2: Identify all routes of administration (e.g., IV bolus, Infusion, SC).
3. Step 3: Identify only the oral route; discard the other routes and identify the dose.
4. Step 4: Locate PK parameters (Cmax, Tmax, AUC, Bioavailability, etc.) linked to those doses.
5. Step 5: Format the final output as a structured format (as Example output).

Only include values explicitly mentioned. If you are unsure, return NIL only.

Text: "{text}"

Follow the steps above and output the final data row.

### B.4 Stage 4: CORVA (Full Framework)

Building on Stage 3, CORVA generates multiple reasoning paths rather than one. This stage generalises the chain-of-thought prompt from Stage 3 into a multiple-reasoning-path prompt as stated in Equation 2. Afterwards, it adds a validation step ( $S(r_j)$  where  $j$  represents the  $j$ -th path) that scores each path's results against domain constraints and outputs only conflict-free paths. The system gen-

erates candidate PK tuples across reasoning paths and prunes invalid tuples from the final output.

Stage 4 inherits Stage 2's structural constraints and Stage 3's procedural reasoning, yielding a combined reasoning-and-validation framework for PK parameter extraction.

[USER PROMPT]

Task: Extract Pharmacokinetic (PK) data from the text provided below.

Follow this Tree of Thoughts process:

1. **\*\*Brainstorm Potential PK Parameters\*\***: List all possible phrases or segments in the sentence that *might* represent a PK parameter (description, value, unit). Do not filter yet.
2. **\*\*Evaluate Each Potential Parameter\*\***: For each item brainstormed, critically assess if it truly fits the definition of a PK parameter through the oral route (valid numerical value, and distinct unit). Discard any that are incomplete or ambiguous.
3. **\*\*Synthesize Final Extraction\*\***: Combine all the *valid* and *reasoned* PK parameters into a structured output as Example. Each element in the array should be an object with 'PK parameter', 'Value', and 'Unit' keys.
4. **\*\*Check for Conflicts\*\***: Ensure that the extracted unit is consistent with the expected unit for each PK parameter.

Only include values explicitly mentioned. If you are unsure, return NIL only.

Text: "{text}"

Follow the steps above and output the final data row.

## C Drug-PK Pair Extraction Results

We evaluate our proposed four methods on five language models: Mistral-7B (Mistral), GPT-4o-mini (GPT), Qwen2.5-3B (Qwen)<sup>1</sup>, Phi-3.5-mini (Phi)<sup>2</sup>, and Llama-3.2-3B (Llama)<sup>3</sup>. Mistral and GPT are accessed via their respective APIs, while Qwen, Phi, and Llama are loaded locally through HuggingFace. Three baselines were evaluated: E2E extraction pipeline of Hernandez et al. (2024b) instantiated with SciSpaCy (Neumann et al., 2019)

<sup>1</sup><https://huggingface.co/Qwen/Qwen2.5-3B-Instruct>

<sup>2</sup><https://huggingface.co/microsoft/Phi-3.5-mini-instruct>

<sup>3</sup><https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>

Table 4: Results of Drug-PK Parameter Pairs

| Method             | Recall | Precision | F1        |
|--------------------|--------|-----------|-----------|
| SciSpaCy           | 0.2985 | 0.1571    | 0.2058    |
| AutoPK             | 0.9915 | 0.0783    | 0.1451    |
| BioBERT            | 0.6368 | 0.2853    | 0.3941    |
| Mistral-implicit   | 0.7179 | 0.3556    | 0.4756**  |
| Mistral-rule-based | 0.6197 | 0.3329    | 0.4331    |
| Mistral-procedural | 0.6752 | 0.4370    | 0.5306    |
| Mistral-CORVA      | 0.7885 | 0.4028    | 0.5332*** |
| GPT-implicit       | 0.1111 | 0.3973    | 0.1737    |
| GPT-rule-based     | 0.4509 | 0.4662    | 0.4584*** |
| GPT-procedural     | 0.5534 | 0.3137    | 0.4004*** |
| GPT-CORVA          | 0.5919 | 0.3623    | 0.4495*** |
| Qwen-implicit      | 0.6944 | 0.0586    | 0.1080*   |
| Qwen-rule-based    | 0.5919 | 0.1594    | 0.2511    |
| Qwen-procedural    | 0.4744 | 0.0921    | 0.1542*   |
| Qwen-CORVA         | 0.6239 | 0.1268    | 0.2108    |
| Phi-implicit       | 0.3932 | 0.2130    | 0.2763*** |
| Phi-rule-based     | 0.1902 | 0.3758    | 0.2525*** |
| Phi-procedural     | 0.2094 | 0.3061    | 0.2487*** |
| Phi-CORVA          | 0.2115 | 0.2870    | 0.2435*** |
| Llama-implicit     | 0.8846 | 0.0583    | 0.1094*** |
| Llama-rule-based   | 0.8483 | 0.0445    | 0.0846*** |
| Llama-procedural   | 0.8632 | 0.0258    | 0.0501*** |
| Llama-CORVA        | 0.6218 | 0.0208    | 0.0403    |

and with BioBERT (Hernandez et al., 2024b) as its NER component, and AutoPK (Sholehrasa et al., 2025). Each baseline is fine-tuned on the PK extraction task following its original configuration and the default hyperparameter setting. Specifically, the SciSpaCy used the en\_ner\_bc5cdr\_md model, while AutoPK used its default GPT-4o-mini backbone via API. All extracted tuples are filtered to retain only absorption-related parameters (i.e.,  $T_{max}$  and  $k_a$ ).

We use McNemar’s test (Yan et al., 2017) to measure the performance difference between the proposed method and the fine-tuned BioBERT. The significance level of McNemar’s test is denoted as \*, \*\*, and \*\*\* for  $p < 0.05$ ,  $p < 0.01$ , and  $p < 0.005$ , respectively.

Among the three baselines, BioBERT achieves the strongest F1 (0.3941), reflecting its broad coverage strategy. SciSpaCy performs weakest overall, struggling with both precision and recall on this task. Compared to the LLM-based methods, Mistral-based methods consistently achieve stronger performance across all four stages, with Mistral-CORVA attaining the highest recall and the highest F1 significantly among all LLM configurations. Both outperform all three baselines by a substantial margin. GPT yields competitive precision but lower recall, reflecting a more con-

Table 5: Accuracy of PK Information Extraction

| Method             | Value  | Unit   | Matching  |
|--------------------|--------|--------|-----------|
| SciSpaCy           | 0.1940 | 0.1620 | 0.1514    |
| AutoPK             | 0.5800 | 0.7249 | 0.4051    |
| BioBERT            | 0.4573 | 0.5726 | 0.4487    |
| Mistral-implicit   | 0.6624 | 0.6838 | 0.6474*** |
| Mistral-rule-based | 0.5363 | 0.5940 | 0.5235**  |
| Mistral-procedural | 0.6218 | 0.6453 | 0.6154*** |
| Mistral-CORVA      | 0.6902 | 0.7628 | 0.6859*** |
| GPT-implicit       | 0.1966 | 0.1859 | 0.1517*** |
| GPT-rule-based     | 0.6239 | 0.7179 | 0.6004*** |
| GPT-procedural     | 0.6709 | 0.7393 | 0.5620*** |
| GPT-CORVA          | 0.6517 | 0.7906 | 0.6026*** |
| Qwen-implicit      | 0.1303 | 0.0491 | 0.0278*** |
| Qwen-rule-based    | 0.2244 | 0.1068 | 0.0897*** |
| Qwen-procedural    | 0.3611 | 0.1752 | 0.1410*** |
| Qwen-CORVA         | 0.2692 | 0.1175 | 0.0940*** |
| Phi-implicit       | 0.2222 | 0.2585 | 0.2094*** |
| Phi-rule-based     | 0.1346 | 0.1517 | 0.1325*** |
| Phi-procedural     | 0.1432 | 0.1603 | 0.1282*** |
| Phi-CORVA          | 0.1496 | 0.1709 | 0.1410*** |
| Llama-implicit     | 0.3504 | 0.4188 | 0.2286*** |
| Llama-rule-based   | 0.3504 | 0.3739 | 0.2436*** |
| Llama-procedural   | 0.2201 | 0.2329 | 0.1453*** |
| Llama-CORVA        | 0.1880 | 0.1538 | 0.0705*** |

servative extraction tendency. Qwen and Llama exhibit a pronounced precision-recall imbalance: both achieve relatively high recall in their implicit stage but extremely low precision, resulting in F1 scores that fall below even the weakest baseline. Phi shows the opposite pattern, achieving moderate precision but low recall across all stages.

## D Tuple Accuracy Results

Table 5 reports tuple accuracy across the same four stages and five language models: Mistral, GPT, Qwen, Phi, and Llama. Statistical significance is assessed using McNemar’s test (Yan et al., 2017) with fine-tuned BioBERT as the reference baseline.

Table 5 reports extraction accuracy for all methods. Among the baselines, AutoPK achieves the strongest unit accuracy (0.7249), benefiting from its standardised parameter vocabulary, while BioBERT attains competitive value and tuple accuracy. SciSpaCy performs weakest across all three metrics, reflecting its limited capacity for implicit parameter resolution.

Among the LLM-based methods, Mistral-CORVA achieves the highest tuple accuracy overall, outperforming the strongest baseline by about 28.7%, and leads on both value (0.6902) and unit (0.7628) accuracy within its model family. The staged progression from Implicit to CORVA is con-

sistent for Mistral, with each component contributing incrementally to tuple accuracy. GPT-CORVA achieves the highest unit accuracy across all methods. Qwen, Phi, and Llama all perform substantially below Mistral and GPT across all three metrics.

## E LLM Selection

We also computed the F1 and accuracy for all five models: Mistral, GPT, Qwen, Phi, and Llama, shown in Figure 5. The description of each model can be found in Appendix C. We have recorded the inference time and token cost for all five models across different stage strategies, as listed in Tables 6 and 7, respectively. The definitions of the models and strategies can be found in Appendix C. Each result is reported in terms of average (Avg), standard deviation (Std), minimum (Min), maximum (Max), and median (Med).

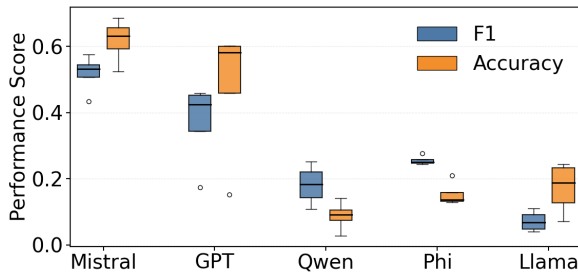


Figure 5: Average performance of each model

Mistral achieves the highest and most consistent performance across both metrics, indicating stable performance across stages. GPT shows strong tuple accuracy (median approximately 0.50) but greater variance in both metrics, reflecting its sensitivity to prompting strategy. Qwen, Phi, and Llama all perform substantially below Mistral and GPT, with median F1 and accuracy below 0.25 across all stages. Notably, Qwen and Llama exhibit low tuple accuracy despite moderate F1 in some stages, suggesting that while they may identify Drug-PK pairs, they struggle to extract correct values and units consistently. Phi shows relatively low variance, indicating consistent performance regardless of the prompting stage.

Table 6 reports inference time per sample in seconds. Overall, inference times remain practical across all models and stages, confirming that CORVA’s additional reasoning and validation steps do not incur prohibitive overhead. Mistral is marginally faster than GPT and substantially faster

Table 6: Inference time for each model.

| Method             | Avg (s) | Std (s) | Min (s) | Max (s) | Med (s) |
|--------------------|---------|---------|---------|---------|---------|
| Mistral-implicit   | 0.41    | 0.14    | 0.26    | 0.74    | 0.37    |
| Mistral-rule-based | 0.48    | 0.21    | 0.24    | 0.88    | 0.43    |
| Mistral-procedural | 0.73    | 0.80    | 0.29    | 3.57    | 0.40    |
| Mistral-CORVA      | 0.93    | 0.88    | 0.25    | 3.40    | 0.56    |
| GPT-implicit       | 0.76    | 0.31    | 0.50    | 1.55    | 0.64    |
| GPT-rule-based     | 0.83    | 0.40    | 0.48    | 2.10    | 0.69    |
| GPT-procedural     | 1.18    | 0.77    | 0.59    | 3.33    | 0.90    |
| GPT-CORVA          | 0.88    | 0.30    | 0.56    | 1.46    | 0.75    |
| Qwen-implicit      | 0.24    | 0.31    | 0.09    | 2.48    | 0.10    |
| Qwen-rule-based    | 0.31    | 0.32    | 0.16    | 2.77    | 0.19    |
| Qwen-procedural    | 1.18    | 0.80    | 0.23    | 11.21   | 1.16    |
| Qwen-CORVA         | 0.26    | 0.29    | 0.15    | 2.53    | 0.16    |
| Phi-implicit       | 0.60    | 0.55    | 0.13    | 3.54    | 0.41    |
| Phi-rule-based     | 1.47    | 1.13    | 0.21    | 3.40    | 1.12    |
| Phi-procedural     | 1.68    | 1.29    | 0.33    | 5.37    | 1.29    |
| Phi-CORVA          | 1.68    | 1.34    | 0.20    | 6.07    | 1.37    |
| Llama-implicit     | 0.23    | 0.50    | 0.09    | 10.58   | 0.10    |
| Llama-rule-based   | 1.07    | 2.19    | 0.19    | 9.90    | 0.39    |
| Llama-procedural   | 1.00    | 0.76    | 0.12    | 8.68    | 0.93    |
| Llama-CORVA        | 1.28    | 2.20    | 0.16    | 9.74    | 0.61    |

Table 7: Total token cost for each model.

| Method             | Avg    | Std    | Min | Max | Med |
|--------------------|--------|--------|-----|-----|-----|
| Mistral-implicit   | 300.29 | 49.59  | 251 | 424 | 280 |
| Mistral-rule-based | 367.18 | 59.68  | 309 | 476 | 341 |
| Mistral-procedural | 326.65 | 91.38  | 246 | 509 | 275 |
| Mistral-CORVA      | 426.76 | 111.95 | 323 | 681 | 368 |
| GPT-implicit       | 276.53 | 34.87  | 239 | 344 | 267 |
| GPT-rule-based     | 349.35 | 52.10  | 299 | 486 | 327 |
| GPT-procedural     | 308.53 | 87.30  | 237 | 541 | 270 |
| GPT-CORVA          | 360.53 | 46.42  | 311 | 459 | 349 |
| Qwen-implicit      | 167.01 | 39.57  | 131 | 662 | 157 |
| Qwen-rule-based    | 400.62 | 39.82  | 365 | 902 | 391 |
| Qwen-procedural    | 285.46 | 42.78  | 235 | 891 | 276 |
| Qwen-CORVA         | 351.84 | 38.76  | 317 | 849 | 343 |
| Phi-implicit       | 204.11 | 51.99  | 149 | 549 | 195 |
| Phi-rule-based     | 506.55 | 62.47  | 423 | 618 | 501 |
| Phi-procedural     | 356.17 | 58.91  | 276 | 683 | 344 |
| Phi-CORVA          | 464.30 | 71.41  | 372 | 683 | 457 |
| Llama-implicit     | 185.61 | 39.77  | 153 | 857 | 177 |
| Llama-rule-based   | 443.68 | 79.45  | 387 | 925 | 419 |
| Llama-procedural   | 304.57 | 38.19  | 250 | 762 | 298 |
| Llama-CORVA        | 404.23 | 77.00  | 339 | 883 | 383 |

than Phi and Llama, with Mistral-CORVA and GPT-CORVA achieving comparable times (median 0.56s vs. 0.75s). Qwen is the fastest overall but delivers the weakest extraction quality, while Phi and Llama exhibit higher variance and occasional instability in generation length. Mistral offers the best trade-off between extraction performance and inference efficiency.

The total token cost per sample is listed in Table 7. Mistral-CORVA consumes an average of 426.76 tokens, a moderate increase over Mistral-implicit, reflecting additional reasoning paths. The higher variance of Mistral-CORVA (111.95) is attributable to more complex implicit cases requiring longer reasoning paths. Token cost across CORVA variants is broadly comparable across

models, while Phi incurs the highest consumption yet delivers among the weakest extraction performance. Mistral-CORVA achieves the best extraction performance at a competitive token cost, confirming it as the most efficient and effective configuration overall.

Overall, Mistral demonstrates the most stable and consistent performance across all stages, making it the most suitable model for illustrating the effect of each pipeline component.

## F CORVA Validation Specifics

Each constraint checks an extracted tuple  $(\mathcal{P}, \mathcal{V}, \mathcal{U})$ , i.e., parameter, value, unit, given the drug’s administration context.

- **Parameter–unit consistency.** Each parameter admits only certain units:  $T_{max}$  require time units (e.g.,  $h, min$ );  $C_{max}$  requires concentration units (e.g.,  $ng/mL$ ); Area under concentration curve requires concentration and time ( $ng \cdot h/mL$ ). Impossible pairings are flagged as invalid.
- **Route–absorption consistency.** Some parameters are route-specific. For intravenous administration, absorption is bypassed, and  $T_{max}$  or  $ka$  are undefined. Any extracted tuple asserting an absorption parameter for an intravenous formulation is therefore flagged as invalid.
- **Numeric validity.** The value must be present, parseable, non-negative, and within range (e.g. bioavailability should be within  $[0, 1]$ ).
- **Tuple completeness.** Parameter, value, and unit must all be present (unit exempt for dimensionless parameters). Partial tuples are counted as invalid.

$S_j$  in Equation 2 is the proportion of fully valid tuples and multiplies the path’s likelihood in Stage 4 selection, so better-validated reasoning is preferred. Because these check pharmacological plausibility rather than surface form, they are effective at filtering confident-but-wrong outputs in the implicit setting, where no surface span anchors the extraction.