

Unlocking Public Catalogues: Instruction-Tuning LLMs for ICD Coding of German Tumor Diagnoses

Stefan Lenz^{1,4}, Lakisha Ortiz Rosario¹, Georg Vollmar², Arsenij Ustjanzew¹, Fatma Alickovic¹, Thomas Kindler^{2,3}, Torsten Panholzer¹

¹Institute of Medical Biostatistics, Epidemiology and Informatics, University Medical Center of the Johannes Gutenberg University Mainz

²University Cancer Center of the Johannes Gutenberg University Mainz

³Third Department of Medicine, University Medical Center of the Johannes Gutenberg University Mainz

⁴Corresponding author, e-mail: stefan.lenz@uni-mainz.de

Abstract

Background

Accurate coding of tumor diagnoses with ICD-10-GM and ICD-O-3 is essential for structured cancer documentation in Germany. Smaller open-weight LLMs are especially appealing for privacy-preserving automation of tumor documentation tasks but struggle with coding accuracy in German-language contexts.

Methods

This study investigates whether instruction-based fine-tuning on public datasets improves the coding accuracy of open-weight LLMs for German tumor diagnosis texts. The evaluation is based on a dataset of coded tumor diagnoses from the local tumor documentation system. A systematic data quality check on a random sample of these test data assessed whether the short textual coding descriptions provided sufficient information for unambiguous code assignment. The upper limit for ICD-10 coding performance was determined with 95% confidence intervals as 60-79% for exact code derivation and 81-94% for partial (three-character codes only) derivation. As training data, over 500,000 question-answer pairs were created based on the ICD-10-GM, ICD-O-3, and OPS catalogues. The training instructions cover coding tasks and tumor diagnosis recognition, with OPS entries serving as additional negative examples. We fine-tuned eight open-weight models from the Qwen, Llama, and Mistral model families, ranging from 7 billion (Qwen2.5-7B) to 70 billion parameters (Llama 3.3 70B).

Results

ICD-10-GM coding accuracy rose from values of 1.4-24% for the base models to 41-58% after fine-tuning. Partial accuracy increased from 31-74% to 73-83%. The accuracy of ICD-O-3 topography coding also improved, but started and remained considerably lower with an exact accuracy of 22-40% and a partial accuracy of 56-67% after fine-tuning. Malformed code outputs dropped to 0% for all models. Tumor-diagnosis recognition reached 99%. Accuracy correlated positively with model size, but gaps between small and large models narrowed after fine-tuning. The reasoning mode in Qwen3 generally produced a lower performance than fine-tuning while increasing the response time over 100-fold.

Conclusions

Our findings highlight the underused opportunity of leveraging public catalogues to build instruction datasets that improve LLMs in medical documentation tasks. Future progress

towards reliable clinical use may also come from more diverse instructions addressing the varied challenges of medical documentation.

Our complete training dataset and the best-performing checkpoints of the fine-tuned models are available from <https://huggingface.co/datasets/stefan-m-lenz/ICDOPS-QA-2024>.

Keywords

- Natural Language Processing
- Large Language Models
- Clinical coding
- ICD-10
- ICD-O-3
- German language

1. Background

The semantic coding of medical content from unstructured text sources is an important part of structured information extraction in medicine. As this is a purely text-based task, automating this process has been in the focus of natural language processing (NLP) research [1]. Large language models (LLMs) are a new and promising NLP technique of approaching this complex problem [2]. However, further improvements are needed to make these tools viable for practical use in German-language tumor documentation [3].

Whenever possible, structured information extraction approaches should be based on established classification systems to enhance data reusability. One of the most widely used medical classification systems is the International Classification of Diseases (ICD) [4]. For coding tumor diagnoses there is the International Classification of Diseases for Oncology (ICD-O). The use of both ICD-10 (10th revision of the ICD catalogue) and ICD-O for documenting tumor diagnoses and reporting them to the cancer registries is mandated by law for the tumor documentation in Germany [5].

The ICD-10 codes for tumor diagnoses are single codes in the range of C00.0-D48.9 [4]. The ICD-O code consists of two parts [6]. The first part is the ICD-O topography code, which encodes the localization in the body and assumes values in the range of C00.0-C80.9. Secondly, there is the 5-digit ICD-O morphology code, which requires an examination of tumor tissue to be determined and provides greater detail about the histological properties of the tumor. While there is a substantial semantic overlap and great similarity between ICD-O and ICD-10, there are many cases where the information about a tumor diagnosis cannot be translated from one coding system to the other without losing information. As a result, both systems are used alongside each other rather than one replacing the other.

Assigning ICD codes to textual diagnoses is a difficult task and it has been shown that even powerful LLMs can struggle with it [7]. Thus far, correctly coding tumor diagnoses requires skilled personnel which is familiar with the specific coding requirements of the tumor documentation.

Medical coding is even more challenging for languages other than English due to the limited availability of publicly accessible training data. This study investigates the hypothesis that, despite these limitations, publicly available data has not yet been fully leveraged for training LLMs in medical coding, particularly for non-English languages. Therefore, we aimed to investigate fine-tuning of LLMs with instructions that are tailored to enhance the coding abilities for German tumor diagnoses. To assess the real-world impact of this training, we evaluate the coding abilities of the LLMs with respect to ICD-10 and ICD-O topography codes using actual diagnosis texts from the local tumor documentation system in the university medical center Mainz.

2. Methods

2.1. A training dataset of instructions constructed from public catalogues

Instruction-tuning of large language models is generally performed using question-answer pairs. For improving the documentation capabilities of the models, we designed question-answer pairs that encapsulate knowledge about coding tumor diagnoses. The question-answer pairs are created using the following public catalogues used for medical documentation in Germany:

- Alpha-ID (alphabetical index of the German version of the International Classification of Diseases (ICD-10-GM), 2024 version) [8]
- ICD-O-3 (International Classification of Diseases for Oncology, German version, second revision, 2019) [6]

- OPS (German classification of medical procedures, 2024 version) [9]

The variant of the Alpha-ID catalogue used contains 13,600 tumor diagnosis descriptions corresponding to a total of 809 distinct codes. From the ICD-O-3-catalogue, we extracted 1,338 localization descriptions with 400 distinct ICD-O topography codes for the training and 2,804 tumor diagnosis descriptions with 1,145 distinct ICD-O morphology codes.

It has been shown that increasing the diversity of tasks in instruction-tuning datasets can lead to better training results [10]. The data from the catalogues is therefore presented in the form of different instructions in the training dataset. The tasks focus primarily on coding tumor diagnoses according to ICD-10 and ICD-O using a variety of textual descriptions.

In addition to that, we trained the models to distinguish tumor diagnoses from other diseases or from medical procedures. Operations and procedures may be related to tumor diagnoses but do not constitute diagnoses themselves. Procedures should therefore not be reported as diagnoses in the documentation. An equal number of non-tumor diagnosis codes from the ICD-10 catalogue as well as procedures from the OPS catalogue were therefore added as negative examples to sharpen the understanding of the concept of tumor diagnoses.

An overview of the types of question-answer pairs that are included in the dataset can be seen in Table 1. The instructions are phrased in German. Translated into English, example questions include “What is the ICD-10 code for [diagnosis]?”, or “Which ICD-O topography code fits [diagnosis]?”, with the corresponding code as the answer. Similarly, questions related to tumor diagnosis recognition are variations of “Is ‘[...]’ a tumor diagnosis?”, to which the given answer is either “Yes” or “No”.

Table 1. Distribution of question-answer pairs in the training dataset.

Questions asking for ...	No. of formulations	No. of questions	Proportion of questions
ICD-10 codes (only tumor diagnoses)	12	163,200	31.5%
Recognizing tumor diagnoses from the Alpha-ID catalogue, yes/no answers (50% tumor and non-tumor diagnoses)	6	163,200	31.5%
Recognizing tumor diagnoses from the Alpha-ID catalogue, answers with yes/no and ICD-10 code (50% tumor and non-tumor diagnoses)	4	108,800	21.0%
OPS main categories	10	29,640	5.7%
Recognizing operations/procedures from the OPS catalogue as distinct from tumor diagnosis	4	11,856	2.3%
ICD-O morphology codes from the ICD-O-3 catalogue	10	28,040	5.4%
ICD-O topography codes from the ICD-O-3 catalogue	10	13,380	2.6%
Total		518,116	100%

Each type of question is provided in a number of different formulations with identical meaning (see second column in Table 1). This way, the model is trained to respond to a wider range of prompt formulations in the desired manner.

The constructed answers are kept short to instruct the model to return short answers, which are faster to generate and easier to process than longer answers.

2.2. Real-world test data from the tumor documentation system

To evaluate the effect of training the models on the data assembled from the catalogues, we use a real-world test data set from the local tumor documentation system of the University Cancer Center at the University Medical Center Mainz.

The test data were extracted on June 27, 2024. We used only recent diagnoses from the years 2023 and 2024. This way, potential differences with older versions of the ICD-10-GM catalogue, which is updated yearly, can be avoided. Restricting the number of diagnoses this way also keeps the computation time for the evaluation manageable. Furthermore, duplicated textual descriptions were filtered out.

The resulting data used for testing the models (test data) comprises 2024 unique textual descriptions of diagnoses in total, which are labeled with the corresponding ICD-10 code and the ICD-O topography code. The three different tasks of assigning an ICD-10 code, assigning an ICD-O code, and recognizing the description as a description of a tumor diagnosis were evaluated separately with three different sets of evaluation questions that are presented to the models. The questions for evaluating the models were constructed in the same manner as the training questions.

2.3. Data quality assessment

The available data were extracted from only three columns of the comprehensive database of the local tumor documentation system, representing a narrow subset of the full patient information. In routine use, documenters enter short free-text descriptions of tumor diagnoses in the context of extensive additional clinical information. As a result, these descriptions are not primarily designed to capture all details required for code derivation. Nevertheless, in most cases the short description is sufficient to derive a code. To account for the issue of incomplete information, we carried out an analysis of the quality of the descriptions with respect to their suitability for coding.

We identified two main reasons why codes cannot be derived from the textual description: First, if the tumor behavior (benign/malign/uncertain) is not specified, an appropriate ICD-10 code cannot be assigned. Second, a missing anatomical localization can hinder accurate ICD-O topography coding.

An example for missing information about the tumor behavior in the data is the description “Nierentumor der rechten Niere” (English: renal tumor of the right kidney), which is coded as C64 in the data. The localization is described in detail here, but the behavior is unclear from the description alone. Therefore, several ICD-10 codes could match this description: C64 (if malignant), D30.0 (if benign), or D41.0 (if uncertain behavior).

An example where the ICD-O localization cannot be derived fully from the text is the diagnosis description “DLBCL Stadium IIIA”. DLBCL (diffuse large B-cell lymphoma) can be readily mapped to the ICD-10 code C83.3. However, the localization cannot be derived from this description alone. Lymphomas, although primarily arising in lymph nodes, may also occur in extranodal sites containing lymphatic cells, such as the gastrointestinal tract, the central nervous system, or even bone [11]. Without explicit mention of the primary site, an accurate ICD-O topography code cannot be assigned.

Due to such cases of incomplete information as illustrated in the examples above, perfect model performance (i.e., 100% accuracy) cannot be expected for ICD-10 coding, ICD-O coding,

or tumor diagnosis recognition. To estimate the theoretical upper bound of achievable performance, a data quality analysis was conducted. For this purpose, a random sample of 100 cases (4.94% of the dataset) was selected and annotated independently by two reviewers. The annotations addressed whether the code specified for the diagnosis could be derived from the tumor diagnosis text and classified the diagnosis coding using the categories fully/partially/not derivable. A code was considered fully derivable if the exact code could be determined, and partially derivable if at least the three-character category could be inferred. This classification was performed separately for ICD-10 and ICD-O. In addition, the reviewers documented the reasons for unclear mappings (missing tumor behavior/missing localization/other).

To evaluate the consistency of these annotations, inter-rater agreement was calculated. Cohen’s kappa (calculated with R package ‘irr’, version 0.84.1) indicated substantial agreement among the two annotators, with values of 0.63 for ICD-10 code derivability, 0.68 for ICD-O localization derivability, and 0.70 for the attribution of reasons for unclear mappings. After a joint discussion of the two reviewers and a third researcher, cases that remained uncertain were referred to an experienced specialist. Finally, a consensus version was established.

Table 2 shows the consensus results of the quality assessment for the derivability of the codes. In four cases for the ICD-10 and two of the ICD-O codes, there was a complete mismatch between code and text, i.e., the given code did not fit the textual description. In Table 2, these cases were counted as “not derivable” along with instances where multiple codes fit the text. Overall, a considerable proportion of cases were only partially or not derivable, primarily due to missing localization (31 cases) or non-derivable tumor behavior (3 cases).

Table 2. Results of data quality assessment on 100 randomly selected diagnosis descriptions (95% confidence intervals in brackets). “Partially derivable”: three-character category can be inferred but not the exact code; “Not derivable”: multiple possible codes in different three-character categories or mismatch between text and code.

Code	Fully derivable (%)	Partially derivable (%)	Not derivable (%)	Total (%)
ICD-10	70 (60–79)	19 (12–28)	11 (6–19)	100
ICD-O	65 (55–74)	26 (18–36)	9 (4–16)	100

In conclusion, the data quality analysis indicates that the attainable accuracy is bounded at approximately 70% for complete ICD-10-GM code prediction and 65% for exact ICD-O localization. Likewise, the upper bounds for the partial accuracy (prediction of the three-character category) lie around 89% for ICD-10-GM and 91% for ICD-O.

2.4. Model selection

For this evaluation, we selected eight different LLMs from the “Llama” model family from Meta, the “Qwen” model family from Alibaba, and two models from Mistral AI, ranging from 7 billion model parameters (suffix “7B”) to 70 billion model parameters (suffix “70B”). An overview of the models can be found in Table 3.

Table 3. Overview of evaluated models with model size (number of parameters), quantization, number of GPUs (Nvidia A40), and fine-tuning time. The models are sorted by size, defined as the number of learned parameters (weights and biases), as specified in their respective model cards on HuggingFace.

Model name	No. of parameters / 10^9	Quantization (fine-tuning / test)	No. of GPUs (fine-tuning / test)	Fine-tuning time (hours)	Model card
Qwen2.5-7B	7.61	No / No	1 / 1	102	[12]
Llama 3.1 8B	8.03	No / No	1 / 1	96	[13]
Qwen3-8B	8.19	No / No	1 / 1	110	[14]
Mistral NeMo	12.2	No / 8-bit	2 / 1	118	[15]
Qwen3-14B	14.8	8-bit / 8-bit	1 / 1	86	[16]
Qwen3-32B	32.8	8-bit / 8-bit	2 / 1	179	[17]
Mixtral 8x7B	46.7	4-bit / 4-bit	2 / 2	315	[18]
Llama 3.3 70B	70.6	4-bit / 4-bit	2 / 2	508	[19]

While all these models have primarily been trained on English texts, they officially support multiple languages, including German. We have already tested some of these models for ICD-10 classification and found the results promising but with room for improvement [20]. To assess the generalizability of the results, we included models from different model families and vendors here. Moreover, we used models of different sizes to examine how the model size affects the baseline performance of the models and the potential to improve the models via fine-tuning.

2.5. Training and evaluation

In all cases, instruction-tuned models were used, i.e., models that had been fine-tuned on question-answer pairs to enhance their conversational abilities. Building on this, additional instruction-based fine-tuning was attempted with the training dataset of tailored question-answer pairs, aiming at enhancing the ICD coding expertise of the models. The scripts for implementing the fine-tuning and the evaluation were developed using Python (version 3.12) and the ‘transformers’ package [21].

The fine-tuning was performed using Low-Rank Adaptation (LoRA), a parameter-efficient technique that updates only low-rank matrices added to specific layers of the models. This technique significantly reduces the computational and memory requirements compared to a full fine-tuning of all model weights [22]. LoRA training was configured with a low-rank dimension of $r=8$, a scaling factor of $\alpha=16$, and a dropout probability of 0.1. Further training hyperparameters were the learning rate (0.00005), the weight decay (0.01), and the batch size per device (6). Due to the extensive computational cost and training time, no optimization of these hyperparameters was conducted.

During training the loss values and the gradients were monitored. For the models Qwen3-14B and Qwen3-32B, NaN values for the gradients occurred in training epoch 3 when training with a reduced batch size of 2 to fit the models on a single GPU. In these cases, training was resumed from the last valid checkpoint in epoch 2. For Qwen3-14B, resuming the training was sufficient to get a stable result, while the instability persisted with Qwen3-32B. Subsequently, two GPUs were used to train Qwen3-32B, but a reduced learning rate of 10^{-5} and gradient norm clipping with a maximum norm of 1.0 were applied.

The textual answers returned from the trained models were compared to the labels and converted to truth values with simple regular expression matching. The models were evaluated

before and after each of the five fine-tuning epochs. For a direct comparison to the training data, the performance on the training data from the catalogue is evaluated in the same manner.

While the base models generally tend to give longer answers, the fine-tuned models return short answers as they were trained to do this with the question-answer pairs. To shorten the answers of the base models, we added a sentence in the prompt that guides the models to also answer shortly. This reduced the amount of output and increased the success of parsing the answer to a well-defined code. The values of the metrics for the base models are based on these modified prompts to alleviate this problem and make a fairer comparison possible.

A property unique to the Qwen3 models is their ability to operate in “thinking mode” as well as in “non-thinking mode” [23]. In thinking mode, they generate intermediate reasoning steps before providing an answer, as opposed to responding directly. We used this property to examine the extent to which the reasoning capabilities of LLMs can improve performance in the specific task of ICD coding, and to estimate the additional computational cost introduced by the required reasoning. The probabilistic token outputs from the models were not further sampled for the base models (temperature 0) to avoid the introduction of additional noise. For the thinking process, sampling is needed, and we followed the recommendation from the Hugging Face model cards of the three Qwen3 models ([14,16,17]) to use temperature 0.6, TopP 0.95, TopK 20, and MinP 0.

3. Results

The loss curves in Figure 1 indicate that the chosen learning hyperparameters were adequate and worked for most models because the curves are smooth and drop rapidly, especially in the first training epoch. In subsequent training epochs, the loss continued to decline on the training data but at a much slower rate. Apart from the instabilities observed for Qwen3-14B and Qwen3-32B, the models behaved quite similarly in the training.

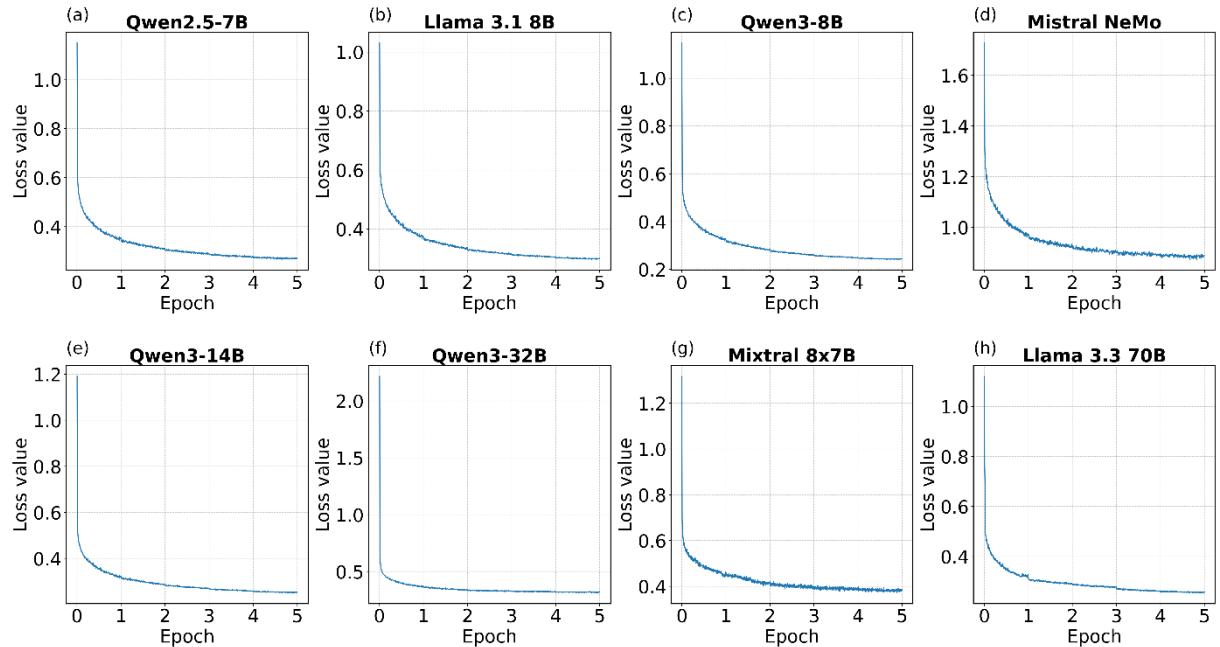


Figure 1. Loss curves for the model training.

Figure 2 shows the performance of the models on the training and test datasets. Almost all models improved substantially after the first training epoch. The performance on the training data also improved further for ICD-10 and ICD-O coding after the first training epoch. However, there were no substantial gains on the test data after the first epoch for ICD-10 classification,

and in some cases, there was even a slight deterioration. With the much smaller part of questions covering the ICD-O catalogue in the training dataset, improvements could also be gained in the subsequent training epochs, especially with the largest model, Llama 3.3 70B.

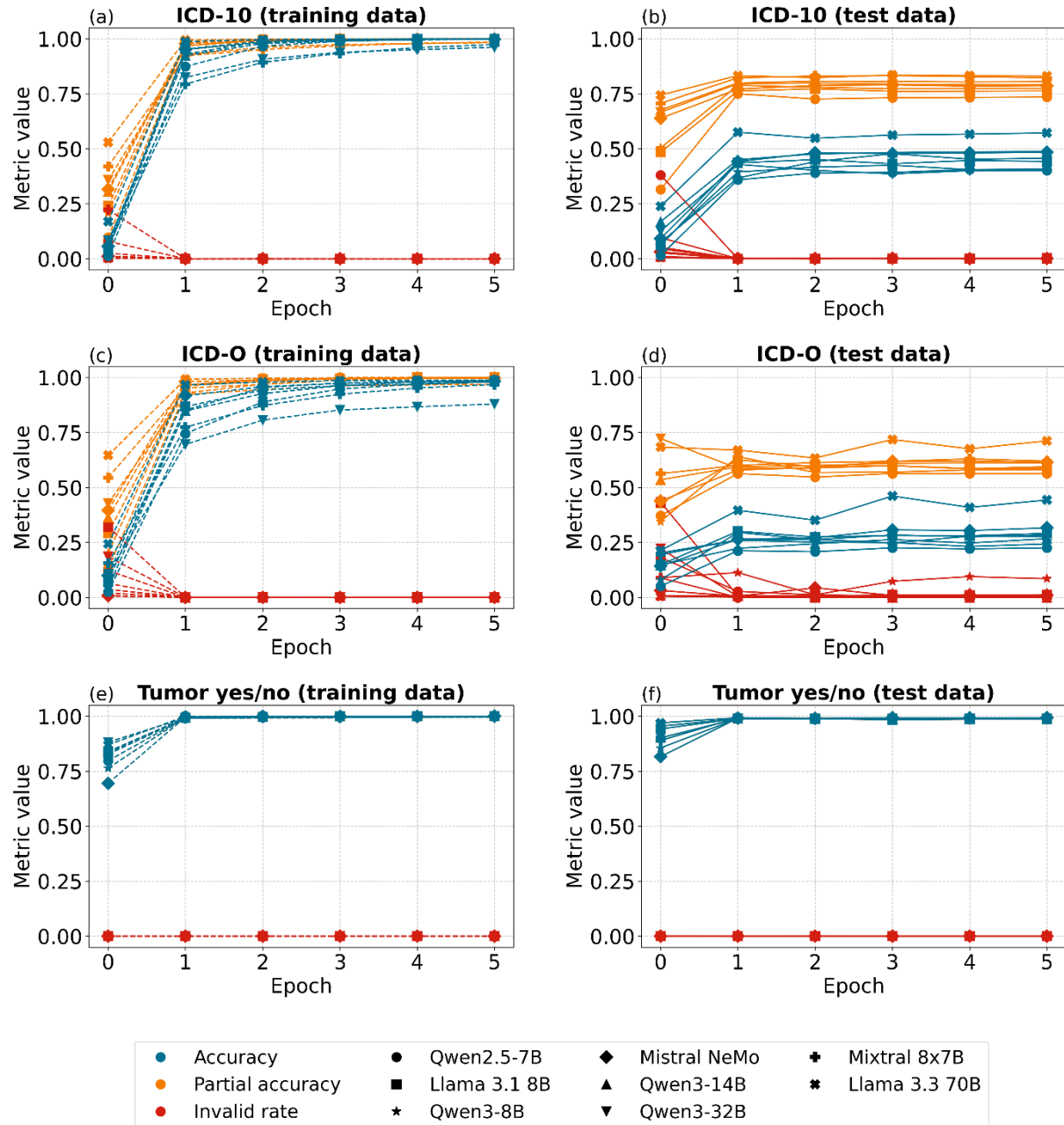


Figure 2. Performance of base models (epoch 0) and fine-tuned models (epochs 1-5) on the training dataset (panels a, c, e) and the test dataset (panels b, d, f). The base models and the fine-tuned models are evaluated after each training epoch with respect to the accuracy (rate of completely correct ICD-10 code, correct ICD-O topography code, or correct recognition as tumor diagnosis), partial accuracy (rate of correct first three characters of the ICD-10 or ICD-O topography code), and the rate of invalid, i.e., not interpretable, answers from the models. The accuracy and partial accuracy are measured relative to the proportion of valid answers.

For ICD-10 coding, the fine-tuning improved the accuracy by around 40% for most models and at least by 31.9% for Qwen3-14B. The rate of invalid answers, e.g., codes that could not be interpreted as ICD codes via regular-expression based pattern matching of the answers, was reduced to near zero after fine-tuning for all of the best models.

All models performed rather poorly on ICD-O coding compared to ICD-10 coding, even though the fine-tuning clearly improved the performance. Producing well-formed ICD-O codes was more difficult for the models, which can also be seen in a higher rate of malformed answers for the base models and for some fine-tuned training checkpoints. These invalid responses for ICD-O mostly consisted of morphology codes or completely invented codes that are an invalid mashup of topography and morphology codes or a similarly looking combination of characters and numbers. This indicates that the concept of ICD-O codes in general and ICD-O topography codes in particular is less familiar to the models compared to ICD-10 codes.

Table 4. Comparison of the performances of the base models vs. the fine-tuned models on the test dataset. Only the values for the best training checkpoints with respect to the accuracy of the ICD-10 coding are shown. The number X in “(best=X)” indicates the training epoch with the best-performing training checkpoints with respect to ICD-10 accuracy.

	ICD-10			ICD-O			Tumor diagnosis (y/n)	
	Acc.	P. acc.	Inv.	Acc.	P. acc.	Inv.	Acc.	Inv.
Qwen2.5-7B								
Base model	1.43%	31.46%	38.05%	5.09%	37.17%	18.17%	90.16%	0.00%
Trained (best=4)	40.57%	73.35%	0.02%	22.07%	56.33%	0.64%	98.64%	0.00%
Llama 3.1 8B								
Base model	5.86%	48.48%	5.21%	14.86%	42.85%	43.35%	94.19%	0.01%
Trained (best=2)	45.08%	77.28%	0.00%	27.47%	59.43%	0.73%	98.77%	0.00%
Qwen3-8B								
Base model	6.69%	50.35%	2.64%	8.02%	34.52%	8.98%	85.57%	0.00%
Trained (best=3)	42.52%	77.30%	0.00%	28.38%	56.99%	7.34%	98.73%	0.00%
Mistral NeMo								
Base model	9.17%	63.84%	3.17%	14.26%	43.87%	3.12%	81.62%	0.00%
Trained (best=5)	48.42%	78.66%	0.08%	31.63%	61.45%	1.04%	99.29%	0.00%
Qwen3-14B								
Base model	16.84%	67.63%	0.96%	14.73%	53.52%	0.80%	90.13%	0.00%
Trained (best=5)	48.72%	80.54%	0.00%	26.69%	61.09%	0.00%	98.71%	0.00%
Qwen3-32B								
Base model	12.58%	66.57%	4.49%	20.29%	72.24%	22.18%	95.50%	0.00%
Trained (best=1)	42.87%	79.59%	0.01%	26.05%	58.61%	0.42%	98.83%	0.00%
Mixtral 8x7B								
Base model	7.38%	70.66%	9.53%	19.42%	56.28%	8.87%	88.93%	0.02%
Trained (best=3)	47.75%	83.27%	0.06%	24.75%	61.89%	0.07%	98.59%	0.00%
Llama 3.3 70B								
Base model	23.81%	74.41%	0.57%	21.35%	68.33%	0.51%	96.95%	0.00%
Trained (best=1)	57.54%	83.26%	0.04%	39.64%	66.98%	0.11%	99.26%	0.00%

Distinguishing tumor diagnosis from non-tumor diagnosis is a much simpler task than coding diagnoses. This is reflected in the much better performance of the models in this task compared to the other two. Interestingly, the base models were not able to get this simple task right in all cases. But already after the first training epoch, all models reached an accuracy of 99% on training and test data. This shows the utility of including non-tumor diagnoses as negative examples in the training dataset, which improved the understanding of the definition of tumor diagnoses by the models.

Table 4 summarizes the performance of the best performing training checkpoints of the models. The performance values of the best performing fine-tuned versions are also shown in Figure 3.

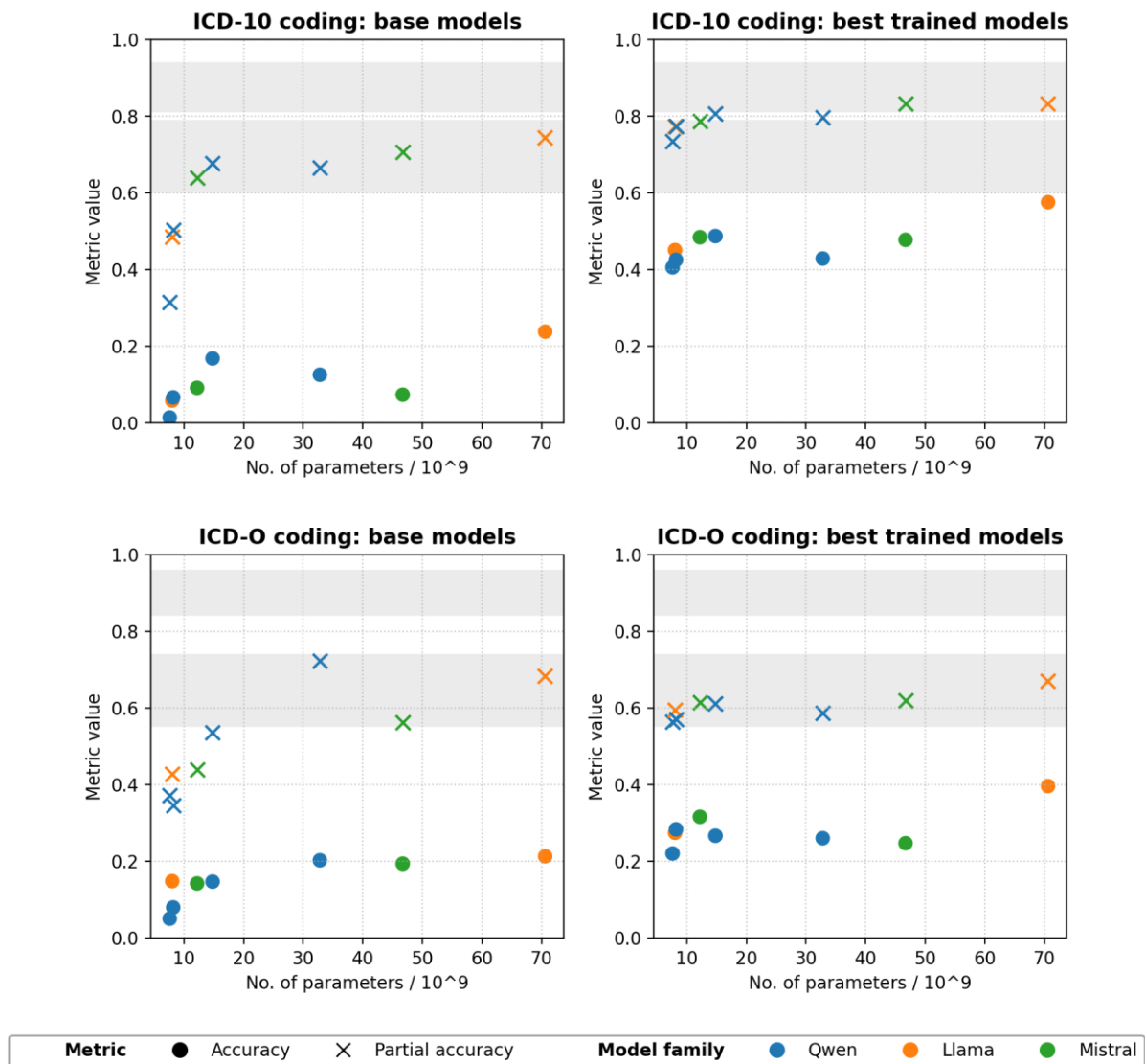


Figure 3. Performance of base models and best trained models for coding ICD-10 coding and ICD-O topography coding on the test dataset. The shaded grey bands represent 95% confidence intervals for the estimated upper limits of accuracy (lower grey band) and partial accuracy (upper grey band), which are derived from the data quality analysis.

Figure 3 is designed to facilitate a deeper look at how model size relates to baseline performance and the ability to improve via fine-tuning. The graphical display suggests a generally positive, approximately linear association between the number of parameters and accuracy, both for the

base models and for the best fine-tuned models. An exception can be seen for the partial accuracy, which shows a sharp increase up to 14B parameters, particularly pronounced in ICD-10 coding. In contrast, the best fine-tuned models follow a more consistent linear trend. These patterns suggest that larger models capture more general information about codes and their meaning, even without detailed knowledge of subcategories. Fine-tuning reduces the initial performance gap between smaller and larger models, although larger fine-tuned models maintain a small advantage. For partially matching ICD-10 codes, the performance of the best trained models gets closest to the upper limit of achievable performance (indicated by the confidence bands in Figure 3), with only small room for improvement with the two strongest models, Llama 3 70B and Qwen3-14B. The performance gap is larger for exact ICD-10 code matches, and even greater for both exact and partial matches in ICD-O coding, indicating substantial potential for improvement in these tasks.

A further point of investigation was the use of the thinking mode in the Qwen3 models. For this purpose, Figure 4 compares the performance of the Qwen3 base models with and without thinking and the best fine-tuned version of the models. (Results for the combination of fine-tuning and thinking mode are not shown there, as our experiments showed that the models do not employ the thinking process after fine-tuning for answering the questions.) Figure 4 shows that the accuracy did not improve when the thinking mode was activated. On the contrary, the accuracy often even decreased compared to the base model in non-thinking mode. Interestingly, the partial accuracy improved at least slightly when using the thinking process. This indicates that the thinking process steered the answers of the base models in the right direction but was not sufficient to get to the completely right conclusion. In almost all cases, the fine-tuned models performed better than the base models with or without thinking. The only exception to this was Qwen3-32B, where the fine-tuned model had a better performance with respect to the accuracy but a worse performance in the partial accuracy.

After fine-tuning, the rate of malformed answers for ICD-O questions dropped to zero from relatively high values of 18% and 22% for Qwen3-32B with and without thinking, respectively. This means that the model became more confident to give a straight-up answer in more cases but was not actually knowledgeable enough to pick the right one in many of them. The thinking process generally did not seem to help in producing more valid codes. In some cases, the rate of invalid answers was even higher using the thinking mode, especially in the case of ICD-O coding.

The response time is also a crucial factor to consider. Generating answers in thinking mode was more than 100 times slower than providing direct answers (7.9–37 s/question vs. 0.05–0.19 s/question on average across datasets and models). The fine-tuned models responded even faster than the base models (0.04–0.15 s/question). This is consistent with the training objective, as the models have learned to respond directly with the code rather than providing more verbose answers.

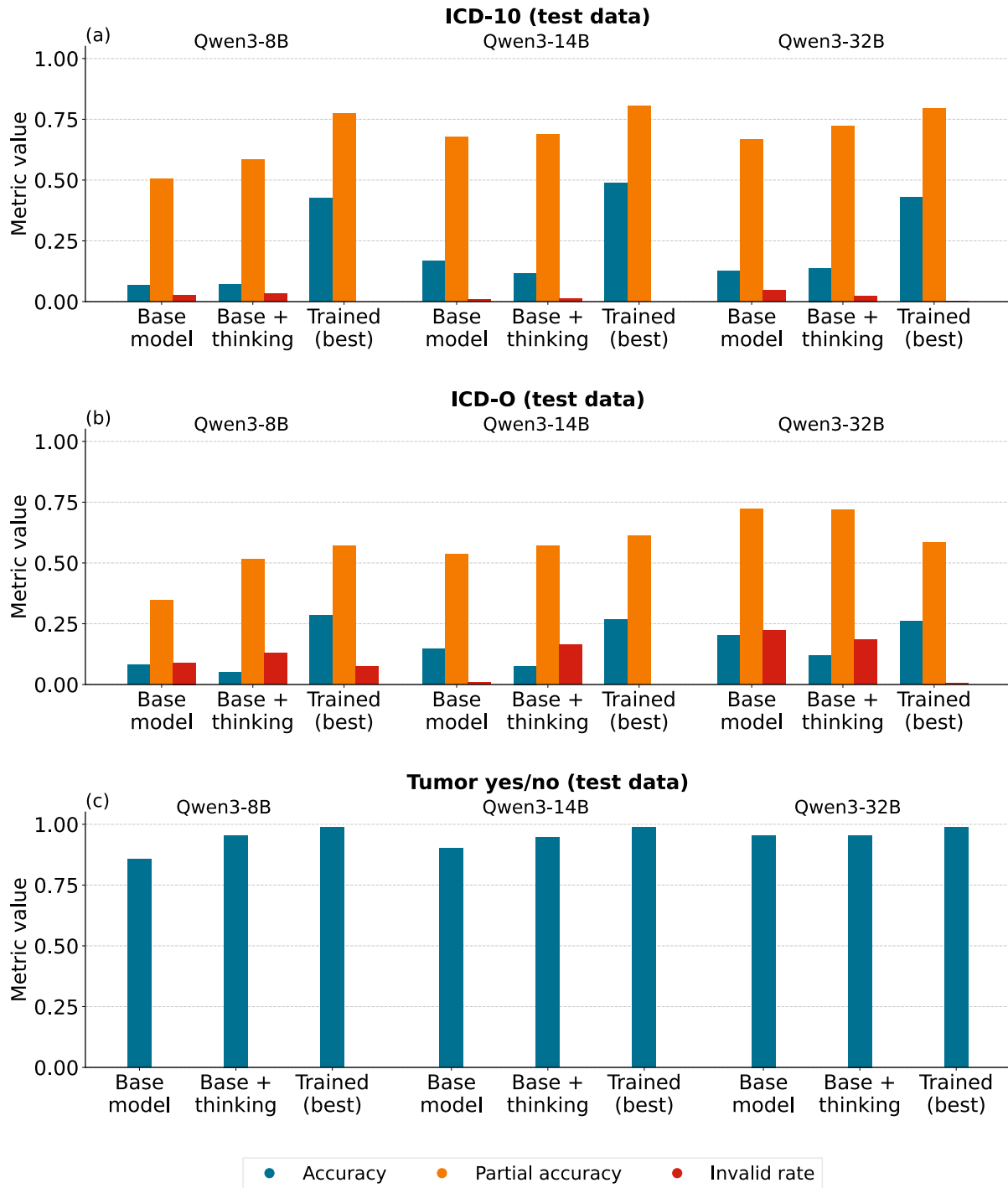


Figure 4. Comparing the ICD coding performance of Qwen3 models using thinking mode to the base models and the fine-tuned (trained) models. The values for the base models and the fine-tuned models were obtained using the complete test data set. Only 10% of the test data was evaluated using thinking mode, as answer generation in this mode is more than 100 times slower. The combination of trained models and thinking is not shown as the models do not use the thinking process after the fine-tuning.

4. Discussion

4.1. Limitations

A central limitation of this study is the limited information contained in the tumor diagnosis descriptions in the real-world test dataset. The short free-text diagnosis descriptions were seen by the LLMs in isolation, whereas human coders have access to the clinical context detailed in doctors' notes and other types of medical documents. As a result, it is not guaranteed that codes can be derived from the short descriptions alone. We tried to quantify this constraint of incomplete information (see Section 2.3) and showed that only about 70% of ICD-10 codes and 65% of ICD-O codes are fully derivable, setting empirical upper bounds for the achievable accuracy below 100%. When employing LLMs for real-world coding tasks, providing richer contextual data could help models overcome the problem of insufficient information. This way, model performance may exceed what is reflected in our evaluation. The main result of this study, which is the demonstration that the coding accuracy can be improved substantially by using instructions from public data, is not undermined by the limitation of data quality.

There are more studies on automatic coding with LLMs that also used real-world datasets of short diagnosis descriptions [7,24]. While these studies did not report an estimation of an upper bound of achievable accuracy, we consider such an estimate very valuable, as it helps to interpret results more realistically and enables fairer comparisons of models and approaches across datasets.

Using bigger models than the ones investigated here might improve the performance. Due to privacy concerns, hosting the models locally is necessary for a routine deployment [25]. For this reason, we only investigated models with a size that makes them usable with local resources inside the hospital in practice.

In this study, models of different sizes were compared with respect to their capability to be improved through fine-tuning. Figure 3 was created with the specific intention to correlate model performance with model size. There, a slight trend could be observed indicating that bigger models kept a small advantage even after fine-tuning that increases with the model size. However, this does not necessarily indicate that larger models must be superior to small models in all cases. A reason why this conclusion should be viewed with caution is the implementation of the fine-tuning with low-rank adaptation (LoRA) [22] here. We applied the same LoRA configuration to all models, but this results in a different number of trainable parameters for different model sizes, as the affected matrices vary with model size. Consequently, larger models are not only stronger at baseline, but they are also given greater adaptation capacity during fine-tuning. One way to mitigate this issue would be to normalize the number of LoRA parameters across models. This could be done, e.g., by increasing the LoRA rank in smaller models so that their effective adaptation space becomes more comparable to that of larger models. However, such adjustments introduce their own trade-offs: higher ranks in smaller models increase training cost and may also raise the risk of overfitting. This could shed light on the influence of the number of adjusted parameters as a confounding factor in the model comparison but it would not necessarily reflect optimal fine-tuning practice.

The limitations discussed above highlight the importance of considering both data quality and model configuration when interpreting results. Nevertheless, the demonstrated potential of instruction-based fine-tuning to enhance coding performance in LLMs remains unaffected.

4.2. ICD-10 coding vs. ICD-O topography coding

The models consistently performed worse on ICD-O topography coding compared to ICD-10 coding, although the relevant categories in ICD-O are fewer than in ICD-10. Figure 3 shows that ICD-O performance falls short even when considering the lower theoretical ceiling

imposed by data quality. This indicates that weaker training coverage, rather than data quality alone, is the main limiting factor.

The base models started with lower prior knowledge of ICD-O, which is probably the case because the ICD-O-3 standard is less represented in general training corpora than ICD-10. This is also reflected by the fact that the base models showed systematic difficulty in distinguishing the concept of localization (ICD-O topography) from histological detail (ICD-O morphology). Including morphology codes in addition to localization codes in the training data was aimed at reducing such confusion. The training succeeded here, as the number of malformed answers was reduced to near zero. Still, the models were less able to generalize ICD-O coding to unseen diagnosis descriptions after fine-tuning compared to ICD-10 coding. This was likely the case because the ICD-O catalogue used for training offers fewer synonyms and linguistic variations than the Alpha-ID catalogue for ICD-10. As a result, the models could not improve as much as they could for ICD-10 coding.

In summary, ICD-O performance lags behind not only because of the slightly lower performance ceiling set by data quality, but primarily because of weaker prior knowledge and lower training coverage. Future work should therefore focus on broadening the diversity of ICD-O coding examples. Targeted instruction-based training data for ICD-O coding should strengthen the understanding of concepts of localization and morphology, as well as the distinction and linkage between the concepts.

4.3. Comparison with additional methods for improving automatic coding

In our study, the textual output of the LLMs was used to infer the ICD codes. Other techniques for inferring codes make predictions based directly on the higher-level numeric representations (“embedding vectors”) in neural networks.

In a study by Mittermeier et al. [26] concerning automatic ICD-10 coding of German diagnoses, the models GermanBERT and FlanT5 were trained on German radiology reports, and achieved accuracies of 58.8% and 65.2%, respectively, for predicting the ICD-10 category (i.e., “partial match” here) on the 100 most prevalent codes in the held-out test data set. Compared to this study, we achieved a higher accuracy with 83% partial match accuracy. Since we trained only on public data, our results are presumably more easily generalizable to other datasets of tumor diagnoses.

Kreuzthaler et al. [24] employed the BERT [27] based models medBERT.de and SapBERT-UMLS as embedding models for predicting ICD-10 codes (Austrian version) from short German diagnosis descriptions. Their large real-world dataset comprised over 20 million diagnosis/code pairs, which were split 80:10:10 into training, validation, and test sets. The coded diagnosis descriptions were transformed into embeddings with the two pretrained language models and used to classify diagnosis descriptions through similarity search. The achieved F1 scores for medBERT.de and SapBERT-UMLS on the test data were 0.86 and 0.87, respectively. If only public catalogue data were used to create the database of embedding vectors, Kreuzthaler et al. observed F1 scores of 0.55 and 0.61 on the test data. These values are in a similar range to the accuracy of 58% in our study. The work by Kreuzthaler et al. clearly demonstrates the potential of utilizing large-scale real-world data and embedding models.

In a very similar way, Böhringer et al. [28] conducted a study with three German hospitals, using a database of embedding vectors created with word2vec [29] to infer ICD-10 codes for ophthalmological diagnoses. Using data from the Alpha-ID catalogue and also real-world data from one hospital, they achieved accuracies for the exact matches of 69% and 69% on the data of two other hospitals, and partial accuracies of 86% and 91%. In comparison, the best accuracies achieved here were 58% for the exact match and 83% for the partial match. In contrast, our approach relied solely on the public catalogue and did not yet make use of tumor documentation data for training. It can be assumed that the performance can be improved further via instruction-based fine-tuning with more real-world data bringing in more variation.

Furthermore, we used test data that was directly extracted from a database and conducted a data quality analysis but did not revise the data, while Böhringer et al. [28] tested the approach on diagnosis descriptions that were automatically extracted from 100 physicians' reports with regular expressions that were manually reviewed afterwards. Notably, the word2vec approach employed by Böhringer et al. is much more resource-efficient than the LLM approach, as it relies on a shallow single-layer neural network, which stands in contrast to the complex deep neural architectures used in LLMs. A hybrid strategy that combines such resource-parsimonious methods with LLMs for more complex cases may represent a promising direction for practical application.

The approach of training generative models, which was pursued here, and the approach using embedding vectors can also be combined to achieve a better performance than the methods applied separately. This is done in retrieval-augmented generation (RAG) [30], where the embedding vectors are used to retrieve additional information from a database, such as a database of descriptions and ICD codes. Then the retrieved information, e.g., possible ICD codes and their descriptions, can be used to improve the queries for the LLMs and give them more information to reason about. Such a RAG approach has already been tested with the American coding system ICD-10-CM and English clinical notes from an emergency department [31]. Combining fine-tuned LLMs and RAG could further improve the performance compared to RAG that employs the basic pre-trained models.

Taken together, these studies show different approaches to medical coding from textual descriptions of diagnoses. Comparisons of performance are limited by differences in datasets and medical disciplines, and by the absence of a quantitative data quality analysis in the other studies. While accuracy is the most critical measure for clinical deployment, resource efficiency is also important for many real-world use cases. Notably, some of these methods are more resource-efficient than the way LLMs are employed here. This suggests that hybrid approaches that combine more lightweight models with LLMs for complex cases could be a promising direction.

5. Conclusions

We presented an approach for fine-tuning LLMs with question-answer pairs generated from German-language catalogues covering ICD-10 and ICD-O-3 localization coding. The main result of this study is that instructions crafted from public catalogue data can greatly enhance the coding abilities of current language models. This indicates that the potential of public resources has not been fully exploited for training open-weight LLMs and that there still is a considerable potential for improving the models with respect to specific medical documentation tasks by fine-tuning them with targeted instructions. Using public and standardized data for training allows for an easier generalization across different sites. The resulting models can also be directly re-used and serve as a foundation for site-specific training. Furthermore, the approach is also transferable to other languages and similar catalogues. Our complete dataset of the question-answer pairs and the best-performing training checkpoints of the fine-tuned models are available on Hugging Face¹.

We systematically analyzed the data quality as a decisive factor for judging LLM performance. Because routine diagnosis texts often lack essential details, perfect accuracy cannot be expected. By estimating realistic performance ceilings, we aimed to provide a fair interpretation of results. Our findings suggest that similar evaluations of coding short-form textual description should be accompanied by an assessment of the completeness of the information in the texts to enable a more meaningful comparison with other studies.

¹ <https://huggingface.co/datasets/stefan-m-lenz/ICDOPS-QA-2024>

We further analyzed the association between model size and model performance before and after training. Larger models achieve slightly better baseline results and retain an advantage after fine-tuning. The improvements are approximately linear, but much of the initial gap between smaller and larger base models can be closed through instruction-tuning. Adjusting fine-tuning strategies or diversifying the instruction set could further reduce this difference. Exploiting the reasoning mode in the Qwen3 models did not yield clear performance benefits in this use case, while it drastically increased the runtime. Nevertheless, the reasoning did steer predictions closer to the right categories, suggesting potential for integrating targeted instructions with guided reasoning in the future.

We believe that future progress will come from integrating LLMs with supporting methods, but also from improving the domain knowledge of LLMs, which was the focus of this study. Our findings demonstrate that there is an untapped potential for improving the coding performance of open-weight LLMs via instruction-based fine-tuning with publicly available data. The creation of more comprehensive, open instruction datasets, similar to the one in the approach here, is a promising way to advance the capabilities of LLMs with respect to medical documentation tasks.

References

- [1] Eguia H, Sánchez-Bocanegra CL, Vinciarelli F, Alvarez-Lopez F, Saigí-Rubió F. Clinical Decision Support and Natural Language Processing in Medicine: Systematic Literature Review. *J Med Internet Res*. 2024 Sept 30;26:e55315.
- [2] Wiest IC, Ferber D, Zhu J, Treeck M van, Meyer SK, Juglan R, et al. From Text to Tables: A Local Privacy Preserving Large Language Model for Structured Information Retrieval from Medical Documents [Internet]. *medRxiv*; 2023 [cited 2024 Oct 24]. p. 2023.12.07.23299648. Available from: <https://www.medrxiv.org/content/10.1101/2023.12.07.23299648v1>
- [3] Lenz S, Ustjanzew A, Jeray M, Ressing M, Panholzer T. Can open source large language models be used for tumor documentation in Germany?—An evaluation on urological doctors' notes. *BioData Min*. 2025 July 24;18(1):48.
- [4] Federal Institute for Drugs and Medical Devices (BfArM). ICD-10-GM: International Statistical Classification of Diseases, German Modification [Internet]. 2024 [cited 2025 Mar 7]. Available from: https://www.bfarm.de/EN/Code-systems/Classifications/ICD/ICD-10-GM/_node.html
- [5] Holland J. Bekanntmachung - Aktualisierter einheitlicher onkologischer Basisdatensatz der Arbeitsgemeinschaft Deutscher Tumorzentren e. V. (ADT) und der Gesellschaft der epidemiologischen Krebsregister in Deutschland e. V. (GEKID). *Bundesanzeiger*. 2021 May 10;BANz AT 12.07.2021 B4.
- [6] Federal Institute for Drugs and Medical Devices (BfArM). ICD-O-3: International Classification of Diseases for Oncology [Internet]. 2024 [cited 2025 Mar 7]. Available from: https://www.bfarm.de/EN/Code-systems/Classifications/ICD/ICD-O-3/_node.html
- [7] Soroush A, Glicksberg BS, Zimlichman E, Barash Y, Freeman R, Charney AW, et al. Large Language Models Are Poor Medical Coders — Benchmarking of Medical Code Querying. *NEJM AI*. 2024 Apr 25;1(5):A1dbp2300040.
- [8] Federal Institute for Drugs and Medical Devices (BfArM). ICD-10-GM Alphabetical index [Internet]. 2024 [cited 2025 Mar 7]. Available from: https://www.bfarm.de/EN/Code-systems/Classifications/ICD/ICD-10-GM/Alphabetical-index/_node.html
- [9] Federal Institute for Drugs and Medical Devices (BfArM). German procedure classification (Operationen- und Prozedurenschlüssel - OPS) [Internet]. 2024 [cited 2025 Mar 7]. Available from: https://www.bfarm.de/EN/Code-systems/Classifications/OPS-ICHI/OPS/_node.html

- [10] Bukharin A, Li S, Wang Z, Yang J, Yin B, Li X, et al. Data Diversity Matters for Robust Instruction Tuning. In: Al-Onaizan Y, Bansal M, Chen YN, editors. Findings of the Association for Computational Linguistics: EMNLP 2024 [Internet]. Miami, Florida, USA: Association for Computational Linguistics; 2024 [cited 2025 Mar 11]. p. 3411–25. Available from: <https://aclanthology.org/2024.findings-emnlp.195/>
- [11] Vannata B, Zucca E. Primary extranodal B-cell lymphoma: current concepts and treatment strategies. *Chin Clin Oncol*. 2015 Mar;4(1):10–10.
- [12] Qwen Team. Model card of Qwen2.5-7B-1M on Hugging Face [Internet]. 2025 [cited 2025 Aug 15]. Available from: <https://huggingface.co/Qwen/Qwen2.5-7B-Instruct-1M>
- [13] Meta, Inc. Model card of Llama 3.1 8B on Hugging Face [Internet]. 2024 [cited 2025 Jan 15]. Available from: <https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>
- [14] Qwen Team. Model card of Qwen3-8B on Hugging Face [Internet]. 2025 [cited 2025 Aug 19]. Available from: <https://huggingface.co/Qwen/Qwen3-8B>
- [15] Jiang A, Sablayrolles A, Tacnet A, Kothari A, Roux A, Mensch A, et al. Model card of Mistral Nemo on HuggingFace [Internet]. 2024 [cited 2024 Sept 25]. Available from: <https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407>
- [16] Qwen Team. Model card of Qwen3-14B on Hugging Face [Internet]. 2025 [cited 2025 Aug 15]. Available from: <https://huggingface.co/Qwen/Qwen3-14B>
- [17] Qwen Team. Model card of Qwen3-32B on Hugging Face [Internet]. 2025 [cited 2025 Aug 15]. Available from: <https://huggingface.co/Qwen/Qwen3-32B>
- [18] Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas D de las, et al. Model card of Mixtral 8x7B on Hugging Face [Internet]. 2025 [cited 2025 Jan 15]. Available from: <https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1>
- [19] Meta AI. Model card of Llama 3.3 70B on Hugging Face [Internet]. 2024 [cited 2025 Aug 19]. Available from: <https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>
- [20] Lenz S, Ustjanzew A, Jeray M, Rensing M, Panholzer T. Can open source large language models be used for tumor documentation in Germany? - An evaluation on urological doctors' notes [Internet]. arXiv; 2025. Available from: <http://arxiv.org/abs/2501.12106>
- [21] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: State-of-the-Art Natural Language Processing. In: Liu Q, Schlangen D, editors. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations [Internet]. Online: Association for Computational Linguistics; 2020 [cited 2024 Sept 5]. p. 38–45. Available from: <https://aclanthology.org/2020.emnlp-demos.6>
- [22] Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: Low-Rank Adaptation of Large Language Models. In: International Conference on Learning Representations (ICLR) 2022 [Internet]. 2022 [cited 2025 Mar 11]. Available from: <https://openreview.net/forum?id=nZeVKeeFYf9>
- [23] Yang A, Li A, Yang B, Zhang B, Hui B, Zheng B, et al. Qwen3 Technical Report [Internet]. arXiv; 2025 [cited 2025 May 22]. Available from: <http://arxiv.org/abs/2505.09388>
- [24] Kreuzthaler M, Pfeifer B, Schulz S. Secondary Use of Clinical Problem List Descriptions for Bi-Encoder Based ICD-10 Classification. *AMIA Annu Symp Proc*. 2025 May 22;2024:620–7.
- [25] Wiest IC, Ferber D, Zhu J, van Treeck M, Meyer SK, Juglan R, et al. Privacy-preserving large language models for structured medical information retrieval. *Npj Digit Med*. 2024 Sept 20;7(1):1–9.
- [26] Mittermeier A, Aßenmacher M, Schachtner B, Grosu S, Dakovic V, Kandratovich V, et al. [Automatic ICD-10 coding: Natural language processing for German MRI reports]. *Radiol Heidelb Ger*. 2024 Oct;64(10):793–800.

- [27] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Internet]. arXiv; 2019 [cited 2024 Oct 17]. Available from: <http://arxiv.org/abs/1810.04805>
- [28] Böhringer D, Angelova P, Fuhrmann L, Zimmermann J, Schargus M, Eter N, et al. Automatic inference of ICD-10 codes from German ophthalmologic physicians' letters using natural language processing. *Sci Rep.* 2024 Apr 19;14(1):9035.
- [29] Mikolov T, Chen K, Corrado G, Dean J. arXiv.org. 2013 [cited 2025 Mar 19]. Efficient Estimation of Word Representations in Vector Space. Available from: <https://arxiv.org/abs/1301.3781v3>
- [30] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [Internet]. arXiv; 2021 [cited 2025 Mar 19]. Available from: <http://arxiv.org/abs/2005.11401>
- [31] Klang E, Tessler I, Apakama DU, Abbott E, Glicksberg BS, Arnold M, et al. Assessing Retrieval-Augmented Large Language Model Performance in Emergency Department ICD-10-CM Coding Compared to Human Coders [Internet]. medRxiv; 2024 [cited 2024 Oct 22]. p. 2024.10.15.24315526. Available from: <https://www.medrxiv.org/content/10.1101/2024.10.15.24315526v1>

Acknowledgements

We thank Dr. Meike Ressing for the discussion of selected difficult coding cases in the dataset, and we thank Marco Jeray for providing legal guidance on matters related to data extraction and analysis.

Funding

This research has been supported by the Federal Ministry of Research, Technology and Space (BMFTR) in Germany in the project “Digitaler FortschrittsHub Gesundheit – DECIDE” (FKZ 01ZZ2106A) of the German national medical informatics initiative. The project aims to facilitate data transfer and interoperability among different stakeholders in the healthcare sector and to provide decision support based on structured patient information.

Ethics declarations

The requirement for an ethics approval was waived by the responsible ethics commission on 06 September 2024 because only short diagnosis descriptions and codes without identifying information were used for this study.

Authors' contributions

S.L. conceived the study, designed the experiments and wrote the manuscript. L.O.R. and S.L. carried out the experiments, and created the tables and plots. T.K. provided the data from the tumor documentation system, and G.V. performed the data extraction. A.U. and F.A. created the annotations for the data quality analysis, and A.U. analyzed them. T.P. provided the funding and supervised the work. All authors read and approved the final version of the manuscript.