FactCheXcker: Mitigating Measurement Hallucinations in Chest X-ray Report Generation Models

Supplementary Material

A. Prompts

Generate Queries from Report Prompt

```
You are an intelligent radiologist tasked with verifying key measurements in radiology reports.

Task: Given a radiology report, your responsibility is to identify if it mentions any of
[endotracheal tube] and formulate a measurement query to verify the results in the report.

**Examples**:

- "Measure the distance between the carina and the endotracheal tube."

- "Measure the diameter of the lung nodules in the upper left lung."

- "Are there any lung masses present in the image?"

**Output Requirements**:

- Make sure the object is present in the image (positive examples). "No pneumothorax" and
"The patient has been extubated" are two negative examples.

- Only output measurement queries, and no validation queries.

- Format each query as a numbered list with no new lines between the queries.

- If there are no relevant queries, output an empty string "".
```

Generate Code from Queries Prompt

````
You are an intelligent code generation agent responsible for transforming user queries into
executable Python code using a specified API reference.

**Task**: When given a query, generate Python code that addresses the query using the provided API.

**Output Requirements**:

- The generated code must be encapsulated in a function named 'get_result,' which takes a
parameter 'cxr_image' of type 'CXRImage.'

- If the method 'find' returns an empty list, the code must return the object that is not
found like this: "[object_name] not present in the image."

- After executing the code, return a string that answers the original query based on the
results obtained.

**Code Template**: Use the following structure for your code:

```python
def get_result(cxr_image: CXRImage):
    # Implement the logic to process the CXRImage and solve the query
    return result  # Replace with the actual result based on the execution
```

**API Reference**:

{api_reference}
````

API Reference

```
class CXRObject:
    """A Python class containing an image object with bounding box information.
    Parameters
    ----------
    object_name : str
        The object name.
    bbox : Tuple[float, float, float, float]
        A tuple representing the bounding box in the format (left, lower, right, upper).
    The object is considered a point if left == right and lower == upper.
    Methods
    -------
    is_point() -> bool
        Returns true if the objects bounding box is a point.
    get_center() -> Tuple[float, float]
        Returns the center of the object's bounding box or the point if it's a single point.
    """

class CXRImage:
    """A Python class containing an image related to a report as well as relevant information.
    Parameters
    ----------
    rid: str
        Id of report object
    reports: dict
        Reports of "GroundTruth" and models.
    image_path: str
        Full image path.
    original_size: List[int]
        The original WxH of the image.
    pixel_spacing: Tuple[float, float]
        Pixel spacing (mm) in the x- and y-directions.
    cache: ModuleCache
        An instance of ModuleCache to retrieve precomputed segmentation and measurement data.
    Methods
    -------
    exists(object_name: str) -> bool
        Returns True if the object is found in the image, and False otherwise.
    find(object_name: str) -> List[CXRObject]
        Returns a list of CXRObjects found in the image matching the object_name.
    segment(object_name: str) -> CXRSegmentation
        Returns a segmentation map of the image based on the object specified.
    within(ob: CXRObject, region: CXRSegmentation) -> bool
        Returns true of if the object center is within the region.
    distance(obj_a: CXRObject  obj_b: CXRObject) -> float
        Returns the distance (in cm) between the center of two objects in the image.
    diameter(obj: CXRObject) -> float
        Returns the diameter (in cm) of the CXRObject.
    dimensions(obj: CXRObject) -> Tuple[float, float]
        Returns the dimensions (in cm) of the CXRObject according to major axis x minor axis.
    width(segmentation: CXRSegmentation) -> float
        Returns the greatest width (in cm) of a segmentation.
    height(segmentation: CXRSegmentation) -> float
        Returns the greatest height (in cm) of a segmentation.
    filter(objects: [CXRObject], region: CXRSegmentation) -> List[CXRObject]
        Returns all objects with their center within the region.
    """

class CXRSegmentation:
    """A Python class containing an anatomical region.
    Parameters
    ----------
    object_name : str
        The object name.
    segmentation_map : List[int, int]
        A binary segmentation map of the anatomical region.
    Methods
    -------
    get_pixel_width() -> float
        Returns the widest pixel width of the segmentation
    get_pixel_height() -> float
        Returns the tallest pixel height of the segmentation
    """
```

Update Report Prompt

```
You are a highly skilled radiologist responsible for updating an existing radiology report
based on a list of new results.

**Task**: Using the provided list of new results, carefully revise the original report to
reflect the new information.

**Requirements**:

- You must use all results in the list.

- If any of the new results contradict details in the original report, update those details
to align with the new results.

- Retain the formatting, tone, and language of the original report as much as possible.

- If the new results indicate that any sentence or phrase in the original report is
incorrect, omit it from the updated version.

- If the results contain no new measurements, you must keep the measurements in the
original report.

**Output Format**:

Return the updated report using the following format:

Updated report: "{updated-report-here}"
```

Extract ETT Mentions Prompt

````
You are an experienced radiologist responsible for finding information on endotracheal
tube (ET) placement in radiology reports. Your performance and accuracy are crucial for
our patient care quality.

To solve this task, perform the following steps:

1. Identify ET Tube Present: Determine if the report explicitly states that an ET tube
is present. Note that mentions of ET tube removal or patient extubation indicate that
the ET tube is no longer present.

2. Extract ET Tube Measurement: If an ET tube is present, extract its relative distance
to the carina in centimeters (cm) if specified. Positive values indicate placement above
the carina, while negative values indicate placement below the carina.

3. Determine ET Tube Placement: If an ET tube is present, determine if the report deems
the placement correct or incorrect. If incorrect, categorize the placement as "too low"
or "too high," if possible.

If you cannot extract a specific category, use "null". There is no need to guess or invent
a value.

Adhere to the following rules:

- Interpret measurements such as "less than [x] cm" as "[x] cm".

- Interpret measurements such as "[x]-[y] cm" as the higher value, outputting "[y] cm"

- Interpret measurements such as "2. 0 cm" as "2.0 cm"

- Interpret terms like "stable" and "unchanged" as "correct."

- If the report does not clarify whether the measurement is above or below the carina,
assume it is above and provide a positive value.

- If you do not find a specific measurement in centimeters (cm) or millimeters (mm),
never infer or approximate a value, simply output 'null'. This is true even if the report
specifies anatomical landmarks.

First, write one sentence describing how you solved the task for each step. Finally, format
your results as a JSON object using the following schema:
```json
{
    'ET_present': bool,
    'ET_measurement': float or null,
    'ET_placement': str or null,
}
```
````

Extract Measurement Mentions Prompt

```
You are an experienced radiologist responsible for finding information on object measurements
in radiology reports. Your performance and accuracy are crucial for our quality of patient care.

To solve this task, perform the following steps:

1. Identify measured object(s): list all objects that the reports include measurements for
using concrete measurements in centimeters (cm) and/or millimeters (mm).

Adhere to the following rules:

- Do not include objects specified in qualitative descriptions or anatomical landmarks, such
as "incorrect position" and "projects into the stomach."

First, write one sentence describing how you solved the task for each step. Finally, format
your answer as a comma-separated string, following this format:

Reasoning: [1 sentence explaining your reasoning]
Answer: object 1, object 2, etc., or an empty string if no objects are measured.

**Examples:**

Report: The tracheostomy tube ends 3.5 cm from the carina. There is a small apical right
pneumothorax. Heart size is normal-the endotracheal tube projects into the T2 region.
Reasoning: Only the tracheostomy tube contains a specific measurement in centimeters or millimeters.
Answer: tracheostomy tube

Report: Ill-defined nodule in the right upper lung measuring 1.3 x 1.4 cm. The endotracheal tube
tip now measures approximately 4.6 cm above the carina-the tip of the right internal
jugular vein catheter projects over the cavoatrial junction.
Reasoning: Both the nodule and the endotracheal tube tip include specific measurements in
centimeters or millimeters, which is why the answer consists of both objects but not the
right internal jugular vein.
Answer: nodule, endotracheal tube
```

B. Baseline Models

CheXagent [5]. CheXagent is an instruction-tuned foundation model specifically designed for chest X-ray interpretation. The model consists of a vision encoder for representing CXR images, and a network to bridge the vision and language modalities. This model is trained on CheXinstruct, a large-scale instruction-tuning dataset curated from 28 publicly-available datasets.

CheXpertPlus [3]. CheXpertPlus, introduced in the CheXpert Plus paper, utilizes a Swinv2 [25] architecture with a two-layer BERT decoder [18] for medical report generation.

GPT-4V [45]. GPT-4V (GPT-4 with vision) is a multimodal LLM released by OpenAI, which enables users to instruct GPT-4 to analyze image inputs provided by the user. In our evaluation, we used the API of model "gpt4o05132024" and followed the official evaluation protocols to assess its performance. The prompt we used is "You are a helpful assistant. Please generate a report for the given images, including both findings and impressions. Return the report in the following format: Findings: {} Impression: {}. ".

LLM-CXR [19]. LLM-CXR is a multimodal large language model that utilizes VQ-GAN to tokenize images, integrating both image and text tokens as input to its base LLM architecture. This model enables CXR-to-report generation, report-to-CXR generation, and CXR-related VQA.

RGRG [39]. RGRG (Region-Guided Radiology Report Generation) employs object detection to extract localized visual features from 29 anatomical regions in chest X-rays. It uses binary classifiers to select salient features and encode abnormalities, followed by a language model generating sentences for each selected region. RGRG was trained on the Chest ImaGenome dataset [43].

MAIRA-2 [2]. MAIRA-2 is a large multimodal model that combines a radiology-specific image encoder with a Large Language Model (LLM), trained for grounded report generation from chest X-rays. For input, the model accepts X-ray images along with indication, comparison, and technique information. For studies containing both frontal and lateral views, we input the technique that "PA and lateral views of the chest were obtained.". For studies with only frontal views, we use "PA view of the chest was obtained.".

MedVersa [48]. MedVersa is a compound medical AI system that can coordinate multimodal inputs, orchestrate models and tools for varying tasks, and generate multimodal outputs. MedVersa was trained on the MIMIC-CXR train and valid dataset for medical report generation tasks.

RadFM [42]. RadFM is a versatile radiology foundation model trained on large-scale multi-modal datasets. It supports both 2D and 3D scans, multi-image input, and visual-language interleaving cases. The model's training included the MIMIC-CXR dataset.

RaDialog [31]. RaDialog is a large vision-language model for radiology report generation and interactive dialogue. It integrates visual image features and structured pathology findings with a large language model (LLM), adapted to radiology using parameter-efficient fine-tuning. RaDialog was trained on the MIMIC-CXR.

RGRG [39]. RGRG (Region-Guided Radiology Report Generation) employs object detection to extract localized visual features from 29 anatomical regions in chest X-rays. It uses binary classifiers to select salient features and encode abnormalities, followed by a language model generating sentences for each selected region. RGRG was trained on the Chest ImaGenome dataset [43].

VLCI [4]. VLCI (Visual-Linguistic Causal Intervention) combines Visual linguistic pre-training using a multiway transformer for cross-modal alignment with Visual-linguistic causal intervention, integrating a pre-trained transformer and Visual and linguistic de-confounding Modules to mitigate cross-modal bias through local and global visual sampling and linguistic estimation using a vocabulary dictionary and visual features.

C. CarinaNet Fine-tuning

We fine-tuned CarinaNet to develop CarinaNet+, leveraging a private dataset of 1,100 chest X-rays collected from intensive care units across 22 hospitals. The dataset was split into 770 images for training and 330 for validation, with final testing performed on the MIMIC-CXR test set (Table 2). Training was conducted using the AdamW optimizer [26] with OneCycleLR scheduling [37]. The hyperparameters were configured with a batch size of 32, initial learning rate of 7.28e-5, weight decay of 0.044323, maximum learning rate of 3.74e-4, and percentage start of 0.48062. The model was trained for 1,000 total steps with early stopping patience of 6 epochs.

D. Additional Pipeline Metrics

Table 5. Performance statistics for the tasks of ETT Presence, Measurement, and Placement using the **original** reports.

	ETT Presence				ETT Measurement		ETT Placement			
Model	Precision	Recall	F1	BACC	MAE	MSE	Precision	Recall	F1	BACC
CheXagent	0.70	0.36	0.48	0.67	1.98	9.44	0.85	1.00	0.92	0.50
CheXpertPlus	0.64	0.76	0.69	0.86	1.44	3.47	0.90	0.67	0.77	0.70
Cvt2distilgpt2	0.71	0.53	0.60	0.75	3.82	101.58	0.73	0.91	0.81	0.48
GPT4V	0.20	0.22	0.21	0.57	2.35	7.35	1.00	0.50	0.67	0.50
LLM-CXR	0.08	0.53	0.13	0.49	2.41	10.38	0.74	0.50	0.60	0.46
MAIRA-2	0.63	0.82	0.71	0.89	1.43	3.91	0.76	0.97	0.85	0.50
MedVersa	0.73	0.72	0.72	0.85	1.07	2.29	0.84	0.94	0.89	0.60
RGRG	0.50	0.84	0.63	0.88	1.58	5.44	0.77	0.94	0.85	0.50
RaDialog	0.67	0.70	0.69	0.84	0.99	1.54	0.81	0.91	0.85	0.60
RadFM	0.28	0.05	0.08	0.52	0.65	0.64	1.00	1.00	1.00	1.00
VLCI	0.25	0.12	0.16	0.54	3.50	58.02	0.80	0.89	0.84	0.44

Table 6. Performance statistics for the tasks of ETT Presence, Measurement, and Placement using the **updated** reports.

	ETT Presence				ETT Measurement		ETT Placement			
Model	Precision	Recall	F1	BACC	MAE	MSE	Precision	Recall	F1	BACC
CheXagent	0.76	0.36	0.49	0.68	0.68	0.93	0.90	0.93	0.91	0.66
CheXpertPlus	0.68	0.76	0.72	0.86	0.66	0.76	0.94	0.88	0.91	0.84
Cvt2distilgpt2	0.72	0.53	0.61	0.75	0.89	2.92	0.88	0.82	0.85	0.76
GPT4V	0.58	0.22	0.32	0.60	0.39	0.16	1.00	1.00	1.00	1.00
LLM-CXR	0.58	0.53	0.55	0.75	0.91	1.95	0.97	0.82	0.89	0.87
MAIRA-2	0.68	0.82	0.74	0.89	0.99	3.05	0.91	0.83	0.87	0.80
MedVersa	0.80	0.72	0.76	0.85	0.86	1.98	0.94	0.85	0.89	0.81
RGRG	0.60	0.84	0.70	0.89	0.76	1.10	0.96	0.90	0.93	0.88
RaDialog	0.69	0.70	0.70	0.84	0.86	2.36	0.92	0.84	0.88	0.81
RadFM	0.47	0.05	0.09	0.52	1.04	3.49	1.00	0.80	0.89	0.80
VLCI	0.54	0.12	0.20	0.56	0.98	1.96	0.88	0.78	0.82	0.64

E. Robustness Analysis

We stratify our analysis by patient demographics (sex, age groups) and clinical contexts (presence of comparison studies, clinical indications). We filter for the most common clinical indications when an endotracheal tube is present and categorize them into respiratory, intubation, or other. We report the average performance of 11 models in Table 7.

Table 7. Performance for the tasks of ETT Presence, Measurement, and Placement using the updated reports stratified by clinical context.

	Gender			Age			Comparison		Indication			All
	Female	Male	Unknown	0-59	60+	Unknown	True	False	Respiration	Intubation	Other	All
Presence Precision	0.62	0.62	0.78	0.66	0.57	0.78	0.72	0.60	0.67	0.74	0.59	0.64
Measurement MAE	0.64	0.72	1.15	0.60	0.90	1.15	0.73	0.78	0.95	0.65	0.87	0.82
Placement Precision	0.96	0.82	0.73	0.95	0.89	0.73	0.76	0.94	0.84	0.83	0.95	0.94

We validate our approach on the CheXpert Plus (Stanford-CXR) test set. The average results of 11 models are summarized in Table 8. However, note that the number of cases with endotracheal tubes is significantly fewer than for MIMIC-CXR 2.0.0, with overlapping model predictions and ground truths only ranging from 4-18 reports.

Table 8. Average performance for the tasks of ETT Presence, Measurement, and Placement using the updated reports of CheXpert Plus

ETT Presence (Precision)		ETT Measurement (Composite)			ETT Placement (Precision)	
Original	Updated	Original	Updated	Improvement	Original	Updated
0.52	**0.68**	5.05	**2.03**	149.0%	0.58	**0.68**

We conduct an ablation study using MedKLIP backbones for the ResNet-50+ endotracheal tube detection module. Since the original ImageNet backbone outperforms MedKLIP (Table 9), we use ImageNet in the final pipeline.

Table 9. Performance of the ResNet-50+ Module using ImageNet and MedKLIP backbones.

Pretraining	ACC	BACC	F1	Prec.	Rec.	AUC
ImageNet	**0.94**	**0.97**	**0.73**	**0.57**	**0.99**	**0.97**
MedKLIP	0.91	0.94	0.63	0.46	**0.99**	0.94

Table 10 shows the performance of running the FactCheXcker pipeline on all reports regardless of the mention of ETT. Although this approach improves false negatives, the precision score significantly drops.

Table 10. Performance comparison between the Original Model and the Updated Model with the FactCheXcker modules applied to all reports, regardless of ETT mention.

Model	ETT Presense		ETT Measurement						ETT Placement	
	Precision		MAE			Composite			Precision	
	Original	Updated	Original	Updated	Improvement	Original	Updated	Improvement	Original	Updated
CheXagent	**0.70**	0.57	1.98	**0.68**	191.0%	4.12	**0.93**	343.0%	0.85	**0.90**
CheXpertPlus	**0.64**	0.57	1.44	**0.66**	118.0%	2.09	**0.90**	132.0%	0.90	**0.94**
Cvt2distilgpt2	**0.71**	0.57	3.82	**0.89**	329.0%	6.37	**1.22**	422.0%	0.73	**0.88**
GPT4V	0.20	**0.57**	2.35	**0.39**	503.0%	11.19	**0.53**	2011.0%	**1.00**	**1.00**
LLM-CXR	0.08	**0.57**	2.41	**0.91**	165.0%	18.54	**1.25**	1383.0%	0.74	**0.97**
MAIRA-2	**0.63**	0.57	1.43	**0.99**	44.0%	2.01	**1.36**	48.0%	0.76	**0.91**
MedVersa	**0.73**	0.57	1.07	**0.86**	24.0%	1.49	**1.18**	26.0%	0.84	**0.94**
RGRG	0.50	**0.57**	1.58	**0.76**	108.0%	2.51	**1.04**	141.0%	0.77	**0.96**
RaDialog	**0.67**	0.57	0.99	**0.86**	15.0%	1.43	**1.18**	21.0%	0.81	**0.92**
RadFM	0.28	**0.57**	**0.65**	1.04	-38.0%	8.12	**1.42**	472.0%	**1.00**	**1.00**
VLCI	0.25	**0.57**	3.50	**0.98**	257.0%	21.88	**1.34**	1533.0%	0.80	**0.88**
Average	0.49	**0.57**	1.93	**0.82**	135.0%	7.25	**1.12**	546.0%	0.84	**0.94**