Supplementary Material for ChEX: Interactive Localization and Region Description in Chest X-rays

Philip Müller[1], Georgios Kaissis[1,2], and Daniel Rueckert[1,3]

[1] Technical University of Munich, Germany
[2] Helmholtz Munich, Germany
[3] Imperial College London, UK

A Ablation Studies and Further Analysis

A.1 Multitask Training: Token and Target Types

Table 1: Ablation study on the forms of supervision (*i.e.* token and target types) used for multitask training. We study using only pathology tokens (P-only), only anatomy tokens (A-only), and only sentence tokens (S-only). Additionally, we study the exclusion of specific tokens types (No-P, No-A, No-S). We also study the exclusion of specific target types like bounding box targets (no box), pathology class targets (No cls), and sentence targets (No sent). Best results and those within one std are marked in bold. The symbols ^{M}, ^{V}, and ^{N} indicate that images from MIMIC-CXR, VinDr-CXR, and NIH8 are used for training, respectively.

		ChEX	Token Types						Target Types		
			P-only	A-only	S-only	No-P	No-A	No-S	No box	No cls	No sent
Pathology boxes^{V}		✓	✓				✓	✓		✓	✓
Pathology classes^{V}		✓	✓				✓	✓	✓		✓
Anat. boxes^{M}		✓		✓		✓		✓		✓	✓
Anat. patho classes^{M}		✓		✓		✓		✓	✓		✓
Anat. sentences^{M}		✓		✓		✓		✓	✓	✓	
Sentences^{M}		✓			✓	✓	✓		✓	✓	
Sentence Grounding (SG)											
MS-CXR^{M}	[mIoU]	47.52	42.34	45.07	9.00	23.90	34.92	**52.97**	3.68	44.38	**52.97**
	[mAP]	44.47	37.92	40.47	0.94	18.89	19.55	**50.66**	3.71	36.43	**50.11**
Pathology Detection (OD)											
VinDr^{V}	[mAP]	**14.12**	13.25	4.60	0.06	1.16	**14.15**	**14.38**	0.22	10.35	**14.38**
NIH8^{N}	[mAP]	**11.14**	7.02	4.40	0.22	2.23	5.38	7.96	0.56	6.70	7.96
MS-CXR^{M}	[mAP]	**16.60**	13.52	10.90	0.26	8.83	5.51	14.11	0.52	14.96	14.11
Region Classification (RC)											
MS-CXR^{M}	[AUROC]	**82.33**	74.02	**79.86**	49.49	76.58	69.10	**80.89**	46.87	**79.89**	75.67
CIG^{M}	[wAUROC]	**70.46**	59.36	69.61	50.08	68.58	62.28	69.61	58.03	**70.16**	69.89
Region Explanation (RE)											
MS-CXR^{M}	[Mic-F1-14]	**49.97**	45.44†	39.47	10.94	22.49	35.37	42.72	5.53	35.02	43.16†
	[Mac-F1-14]	**20.50**	17.99†	**20.29**	1.48	15.19	9.37	**20.01**	2.61	15.17	14.40†
	[METEOR]	**8.79**	6.91†	**9.02**	0.00	6.13	5.69	7.74	2.96	7.77	5.73†
CIG^{M}	[Mic-F1-14]	**53.34**	35.26†	51.25	10.96	47.51	28.76	51.84	23.47	**53.22**	48.93†
	[Mac-F1-14]	**29.13**	16.00†	27.65	1.66	24.14	14.82	28.20	10.73	**29.18**	24.36†
	[METEOR]	**10.18**	5.48†	9.57	0.00	7.95	4.70	9.79	3.47	**10.08**	8.89†
Full report generation (RG)											
MIMIC-CXR^{M}	[Mic-F1-14]	**52.32**	50.10†	49.64	16.73	43.44	46.64	49.91	40.68	51.69	51.18†
	[Mac-F1-14]	**32.56**	26.76†	29.87	3.22	27.60	25.35	31.32	22.83	30.75	29.81†
	[Ex-F1-14]	**58.76**	**59.04**†	57.60	50.25	50.10	55.02	57.39	55.93	57.39	58.44†
	[Mic-F1-5+]	**61.03**	58.16†	57.09	44.28	52.10	54.43	57.77	49.79	**60.58**	58.63†
	[Mac-F1-5+]	**55.85**	48.65†	52.31	22.81	49.96	50.20	52.55	40.75	54.54	51.62†
	[METEOR]	13.26	11.13†	12.45	0.00	12.69	10.97	13.05	8.01	**13.55**	11.99†

† We additionally trained the sentence generator afterwards (with anatomy and report sentences).

Token Types We study the effect of different forms of supervision (*i.e.* token and target types) during multitask training (Tab. 1). In the default case, ChEX is trained with all token types (pathology, anatomy, sentence) and all target types (bounding boxes, pathology classification, sentences). Using only pathology tokens (with bounding box and class targets), *i.e.* only samples from VinDr-CXR, leads to performance reductions on SG, OD, and RC tasks. When additionally using sentence supervision for training only the sentence generator, the performance on the RE task on MS-CXR and the RG task is only slightly degraded compared to ChEX, while there is a huge performance drop on the RE task on CIG. Using only anatomy tokens (with bounding boxes, pathology labels, and sentences) leads to a further performance drop on OD, RE on MS-CXR, and RG, while performance on SG and RE on CIG is improved. Using only sentence tokens leads to very poor performance on all tasks. Similarly, using no pathology tokens or no anatomy tokens leads to very poor performance on most tasks, with the exception that using no anatomy tokens leads to good performance on VinDR-CXR OD. Using no sentence tokens only leads to small performance drops on most tasks and improves performance on SG as well as on VinDr-CXR OD, while only leading to meaningfully performance drops on the other two OD tasks.

Target Types We also study the use of all token types while excluding specific target types. As expected, excluding bounding box targets leads to very poor performance on all tasks involving localization, *i.e.* all SG and OD tasks. Huge performance drops can also be observed on RE and RG tasks, while drops are smaller on RC tasks. Excluding pathology classification labels (for pathology and anatomy tokens) reduces the performance mainly on OD tasks and RE on MS-CXR, while smaller performance drops can be observed on SG and RG. Almost no performance drops are observed on the RC tasks and RE on CIG. Excluding sentence targets (*i.e.* excluding sentence targets for anatomy tokens and all sentence tokens) leads to performance drops on most tasks, except on SG and OD on VinDr-CXR, where performance improves.

A.2 Sentence Generator

In Tab. 2, we provide an ablation study on design choices of the sentence generator. More precisely, we studied (i) the relevance of the post decoder, a component of the sentence decoder; (ii) variations of the query-key-value (qky) features used in P-tuning (linear projection instead of MLP, shared projections for all decoder layers, shorter prefix length); and (iii) the benefits of decoder language model pre-training on MIMIC-CXR. Overall, we found that most design decisions, except the post decoder, have little impact.

We argue that the post decoder is so beneficial, because it enables further processing of ROI features even if only bounding boxes where given as queries (where huge parts of the detector are skipped). The choice of the projections during P-tuning has little influence, such that we opted for the default configuration (MLP projections without sharing). While the prefix length has minimal

Sentence Generator	MS-CXR [Mic-F1-14]	CIG [Mic-F1-14]	MIMIC-CXR [Mic-F1-14]
ChEX	**49.97**	**53.34**	**52.32**
no post decoder	45.86	43.90	46.07
linear qkv proj.	**48.64**	52.80	**52.41**
layer-shared qkv	**48.25**	**53.38**	**52.46**
prefix length 1	**49.65**	**53.14**	**52.25**
no gen. pretraining	**48.56**	52.89	**52.46**

Table 2: Ablation studies of the sentence generator.

influence, a slightly longer prefix length of 5 (as used in CHeX), provide some minor improvements across all generation tasks. Skipping pre-training of the decoder on MIMIC-CXR and directly using the decoder language model pre-trained only on PubMed, leads to small performance degradation on some tasks. However, the main benefit of this pre-training is a faster convergence in the final end-2-end training.

A.3 Query Type for Region Classification and Explanation

Table 3: Effect of query types on region classification (RC) and region explanation (RE) on the CIG dataset. We compare the setting using box queries (the default setting) and the setting using textual prompts (based on the names of the anatomical regions).

	Chest ImaGenome (CIG)	
Query type	Region Classification (RC) [wAUROC]	Region Explanation (RE) [Mic-F1-14]
Boxes (default task)	70.46±0.36	53.34±0.43
Textual prompt	70.76±0.37	54.70±0.41

Tab. 3 shows the effect of using bounding boxes versus textual prompts as queries for RC and RE on the CIG dataset, *i.e.* for anatomical regions. When using textual prompts, *i.e.*, the names of anatomical regions, instead of their bounding boxes (the default setting), the performance of ChEX slightly improves. We assume that these improvements are caused by the additional context provided by the region prompts. Overall, the differences are marginal, showing that our model ChEX is robust to the choice of user query.

A.4 Prompt Sets for Report Generation

Table 4: Effect of different prompt sets (anatomy or pathology prompts or both) and filtering (using only regions with positive pathologies) for full report generation. The default setting used by ChEX is highlighted in grey, best results are marked in bold.

Prompt sets		MIMIC-CXR					
Anatomy	Pathology	[Mic-F1-14]	[Mac-F1-14]	[Ex-F1-14]	[Mic-P-14]	[Mic-R-14]	[METEOR]
yes	yes	51.04	30.61	57.05	42.99	**62.80**	**13.90**
yes	positive	50.13	30.67	56.91	42.46	61.22	13.60
yes	no	50.08	30.07	56.97	42.66	60.64	12.67
positive	yes	**52.37**	32.69	58.78	45.09	62.44	13.28
positive	positive	52.02	**32.88**	58.36	44.64	62.33	12.24
positive	no	51.27	32.18	58.82	45.16	59.31	10.77
no	yes	50.35	25.82	59.35	47.88	53.10	9.82
no	positive	50.39	26.69	**60.14**	**49.38**	51.44	8.26

ChEX uses pre-defined prompt sets to enable full report generation (RG). In Tab. 4, we study the effect of using pathology and anatomical region prompts. We additionally study the effect of excluding regions without pathologies (based on contrastive classification of the CheXpert [16] pathologies). Overall, all studied prompt sets lead to results competitive with the baselines, with Mic-F1-14 ranging from 50.08% to 52.37%, but the selection of the prompt set influences the relation between precision (Mic-P-14) and recall (Mic-R-14). Using only pathology prompts leads to a high precision but achieves much smaller macro F1 scores than also using anatomy prompts (with positive filtering). This is explained by the poor performance on a few pathologies (adding an anatomical prompt with positive filtering to the prompt set achieves > 4-fold improvement for enlarged cardiomediastinum and > 11-fold improvement for lung lesions). Therefore, using anatomical prompts can spot additional pathologies that may be missed by pathology prompts. Similarly, using only anatomy prompts leads to a small recall because pathologies may again be missed. Using both prompt sets leads to the best results. However, considering all anatomical regions can lead to overprediction (lower precision) and, therefore, lower overall F1 scores. We found that using positive filtering with anatomy prompts (*i.e.* considering only anatomical regions with positive pathologies for generation) improves performance. The best setting for Mic-F1-14 uses positive filtering for anatomy prompts and no filtering for pathology prompts. We consider this the default setting of ChEX, and all reported results from the main paper follow this setting. The best Mac-F1-14 can be achieved using positive filtering for both prompt sets. The best METEOR score and recall can be achieved using no positive filtering with both prompt sets, while the highest precision is achieved using only pathology prompts with positive filtering. We highlight that changing the prompt set does not require re-training.

A.5 Sensitivity to Prompt Modifications

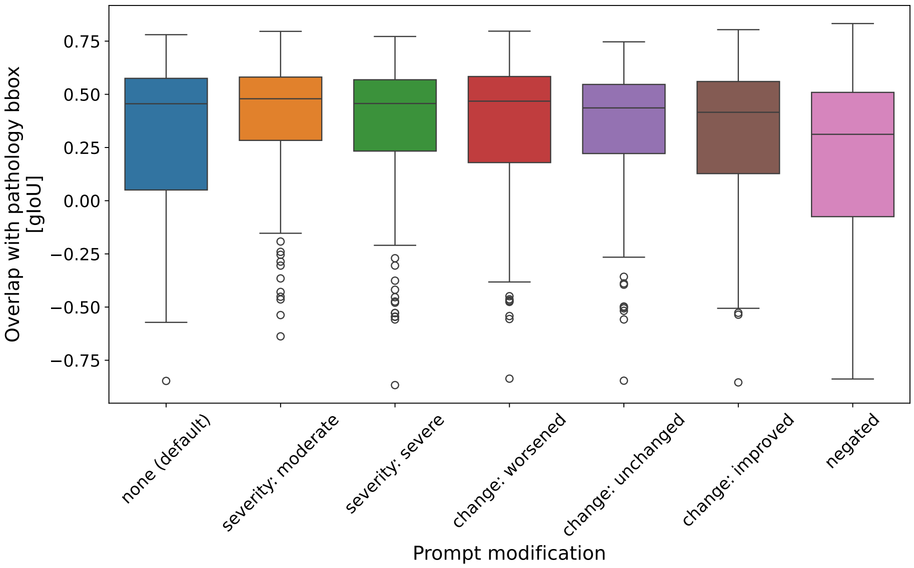


Fig. 1: Effect of different prompting strategies on pathology localization performance

Fig. 1 shows the effect of different prompting strategies on pathology localization performance. Modifying the default prompt (*i.e.* just the pathology name) by adding severity modifications reduces the overall variance of localization quality, where "moderate" severity leads to the best performance. Using change indications also leads to a reduction in variance, albeit smaller than the severity modifications. The best performance can be achieved with "worsened" while "improved" leads to slightly poorer performance, probably because this indicates negative findings. However, all of these prompt modifications have little influence on the median results. Negating the prompt, on the other hand, leads to a noticeable drop in localization performance. Overall, this indicates that our model ChEX is robust to most modifications of prompts.

A.6 Learned Regional Bias for Pathology Prompts

In Fig. 2, we study the regional bias of pathology prompts in negative images. Cardiomegaly prompts typically overlap with the cardiac silhouette, while most other pathologies mostly overlap with lung regions, as expected. More noteworthy, pleural effusion is typically searched in the lower lung zones, while pneumothorax is searched in the upper lung zones, which aligns well with the typical locations of these pathologies in chest X-rays. This indicates that our model has learned a meaningful regional bias for pathologies.

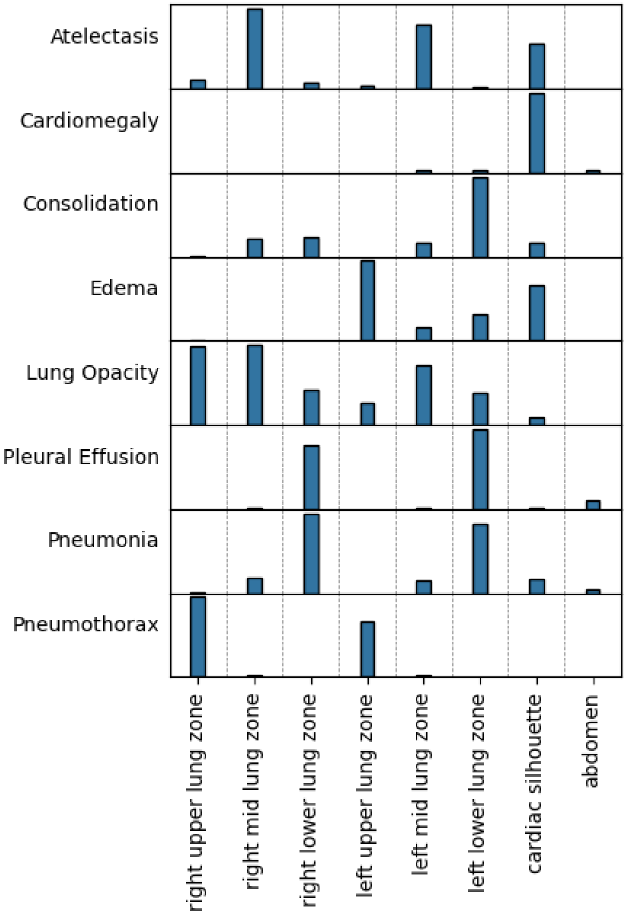
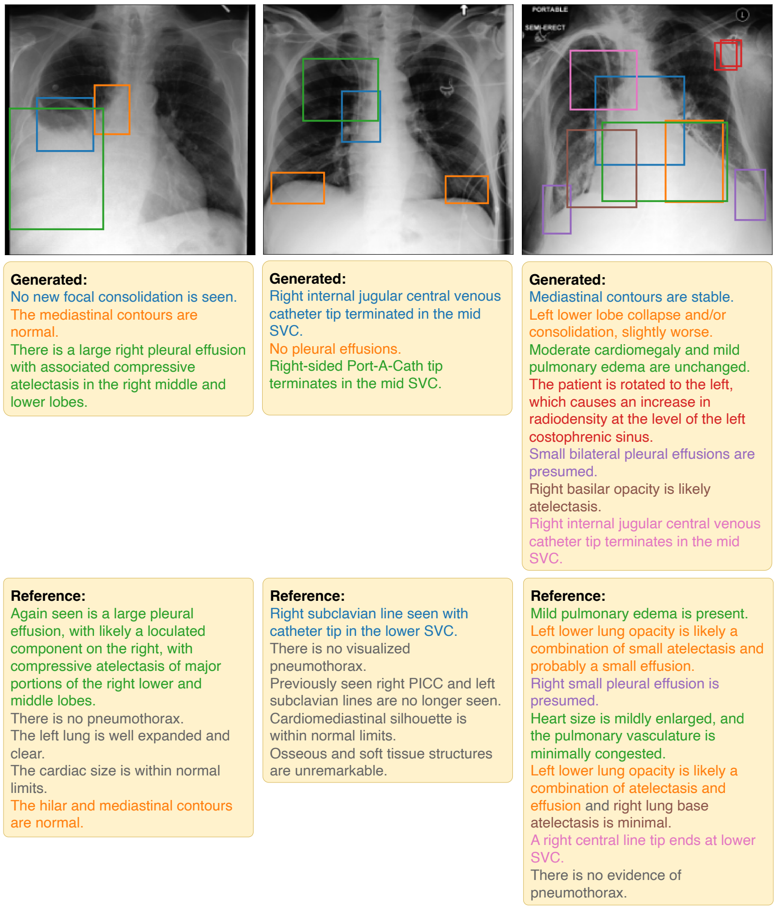


Fig. 2: Regional bias of pathology prompts in negative images. For each pair of pathology and anatomical region, the bar indicates the number of cases where the given region has the highest overlap, among all shown regions, with the bounding box of the given pathology prompt.

B Qualitative Examples and Failure Cases



Generated:
No new focal consolidation is seen.
The mediastinal contours are normal.
There is a large right pleural effusion with associated compressive atelectasis in the right middle and lower lobes.

Generated:
Right internal jugular central venous catheter tip terminated in the mid SVC.
No pleural effusions.
Right-sided Port-A-Cath tip terminates in the mid SVC.

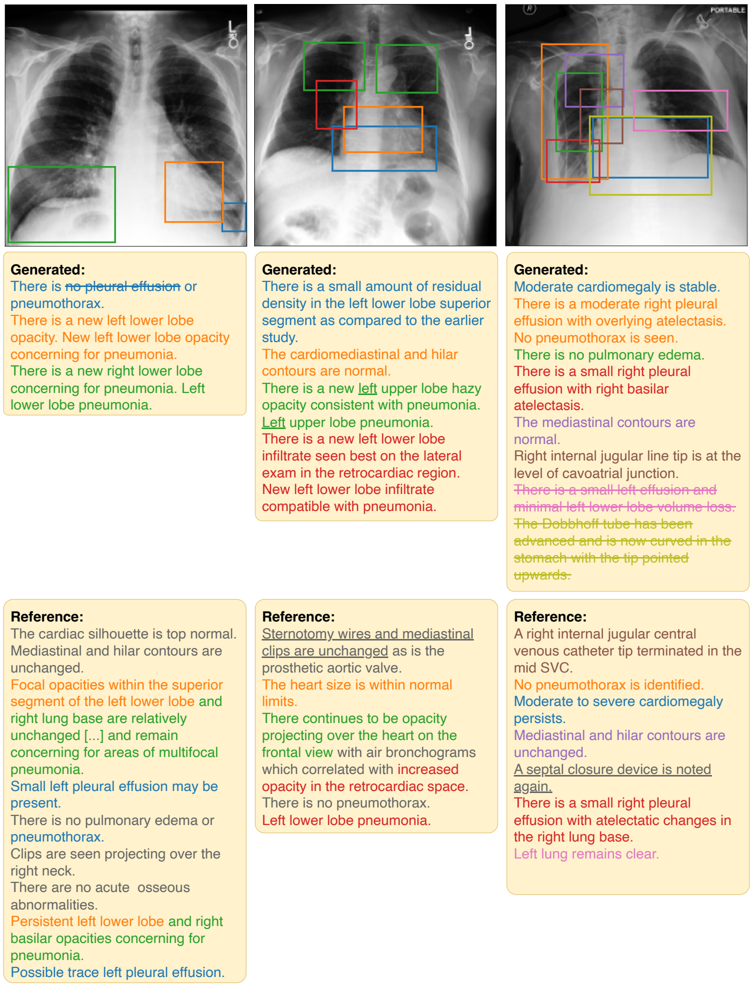
Generated:
Mediastinal contours are stable.
Left lower lobe collapse and/or consolidation, slightly worse.
Moderate cardiomegaly and mild pulmonary edema are unchanged.
The patient is rotated to the left, which causes an increase in radiodensity at the level of the left costophrenic sinus.
Small bilateral pleural effusions are presumed.
Right basilar opacity is likely atelectasis.
Right internal jugular central venous catheter tip terminates in the mid SVC.

Reference:
Again seen is a large pleural effusion, with likely a loculated component on the right, with compressive atelectasis of major portions of the right lower and middle lobes.
There is no pneumothorax.
The left lung is well expanded and clear.
The cardiac size is within normal limits.
The hilar and mediastinal contours are normal.

Reference:
Right subclavian line seen with catheter tip in the lower SVC.
There is no visualized pneumothorax.
Previously seen right PICC and left subclavian lines are no longer seen.
Cardiomediastinal silhouette is within normal limits.
Osseous and soft tissue structures are unremarkable.

Reference:
Mild pulmonary edema is present.
Left lower lung opacity is likely a combination of small atelectasis and probably a small effusion.
Right small pleural effusion is presumed.
Heart size is mildly enlarged, and the pulmonary vasculature is minimally congested.
Left lower lung opacity is likely a combination of atelectasis and effusion and right lung base atelectasis is minimal.
A right central line tip ends at lower SVC.
There is no evidence of pneumothorax.

Fig. 3: Full report generation with predicted bounding boxes on three example chest X-rays. All positive findings in the reference reports have been correctly captured in the generated reports, while reported negative findings can deviate. Changes like the removal of the right PICC (middle) cannot be captured by our model as it only considers a single image but not the temporal sequence. Terms relating to changes like "new" (left, blue) or "unchanged" (right, green) in the generated sentences are hallucinated due to the used training data, a phenomenon common to report generation models [1, 15, 34].



Generated:
There is ~~no pleural effusion~~ or pneumothorax.
There is a new left lower lobe opacity. New left lower lobe opacity concerning for pneumonia.
There is a new right lower lobe concerning for pneumonia. Left lower lobe pneumonia.

Reference:
The cardiac silhouette is top normal.
Mediastinal and hilar contours are unchanged.
Focal opacities within the superior segment of the left lower lobe and right lung base are relatively unchanged [...] and remain concerning for areas of multifocal pneumonia.
Small left pleural effusion may be present.
There is no pulmonary edema or pneumothorax.
Clips are seen projecting over the right neck.
There are no acute osseous abnormalities.
Persistent left lower lobe and right basilar opacities concerning for pneumonia.
Possible trace left pleural effusion.

Generated:
There is a small amount of residual density in the left lower lobe superior segment as compared to the earlier study.
The cardiomediastinal and hilar contours are normal.
There is a new left upper lobe hazy opacity consistent with pneumonia.
Left upper lobe pneumonia.
There is a new left lower lobe infiltrate seen best on the lateral exam in the retrocardiac region.
New left lower lobe infiltrate compatible with pneumonia.

Reference:
Sternotomy wires and mediastinal clips are unchanged as is the prosthetic aortic valve.
The heart size is within normal limits.
There continues to be opacity projecting over the heart on the frontal view with air bronchograms which correlated with increased opacity in the retrocardiac space.
There is no pneumothorax.
Left lower lobe pneumonia.

Generated:
Moderate cardiomegaly is stable.
There is a moderate right pleural effusion with overlying atelectasis.
No pneumothorax is seen.
There is no pulmonary edema.
There is a small right pleural effusion with right basilar atelectasis.
The mediastinal contours are normal.
Right internal jugular line tip is at the level of cavoatrial junction.
~~There is a small left effusion and minimal left lower lobe volume loss.~~
~~The Dobbhoff tube has been advanced and is now curved in the stomach with the tip pointed upwards.~~

Reference:
A right internal jugular central venous catheter tip terminated in the mid SVC.
No pneumothorax is identified.
Moderate to severe cardiomegaly persists.
Mediastinal and hilar contours are unchanged.
A septal closure device is noted again.
There is a small right pleural effusion with atelectatic changes in the right lung base.
Left lung remains clear.

Fig. 4: Example failure cases in full report generation. **Left:** The model incorrectly predicts that no pleural effusion is present, while there may be a trace of a left pleural effusion (blue). However, it predicts a bounding box for a potential left pleural effusion, thereby enabling correctness checking by a radiologist. **Middle:** The model correctly predicts that there is left lower lobe pneumonia (red). However, it (inaccurately) predicts that there might be additional left upper lobe pneumonia while predicting bounding boxes in both lungs, *i.e.* one of the bounding boxes does not match the prediction (green). **Right:** The model incorrectly predicts that there is a small left effusion (pink), which can again be checked and corrected using the bounding box. It also incorrectly predicts the presence of a Dobhoff tube (yellow) which it may confuse with the correctly detected catheter tip (brown). Overall, the predictions cover most positive findings correctly. The provided bounding boxes for positive and negative findings enable simple checking of nuanced cases and cases where the model is uncertain.

C Detailed Benchmark Results

C.1 Sentence Grounding (SG) Task

Table 5: Sentence grounding (SG) results on the MS-CXR dataset. We indicate the use of training or evaluation images from VinDr-CXR by ^{V}, from MIMIC-CXR by ^{M}, and from other sources by ^{O}. All results are based on our own experiments, except OmniFM-DR for which we show the results reported by the original paper. We provide the std based on bootstrapping with $N = 250$. Best results and those within 1-std are marked in bold.

	Sentence Grounding (SG)	
	MS-CXR^{M}	
	[mIoU]	[mAP]
ChEX^{VM}	47.52±1.45	**44.47±2.21**
Zeroshot Contrastive Models		
BioVIL [3] (masks) ^{M}	22.75±1.26	–
BioVIL [3] (boxes) ^{M}	28.57±1.31	18.62±1.37
CheXzero [38] (masks) ^{M}	11.94±0.59	–
CheXzero [38] (boxes) ^{M}	15.45±0.67	5.94±0.64
Multitask Generative Models		
OmniFM-DR [45] ^{VMO}	46.2	–
Supervised Visual Grounding		
TransVG [7] ^{M}	48.81±1.45	37.65±2.61
TransVG [7] (BioVIL [3]) ^{M}	**52.13±1.73**	41.24±2.73
TransVG [7] (CheXzero [38]) ^{M}	**53.51±1.53**	**44.05±2.63**

In *sentence grounding (SG)* (Tab. 5), ChEX demonstrates similar performance to the only generative model OmniFM-DR [45] and the best SupVG model TransVG [7] (with CheXzero [38] backbone), where TransVG shows an advantage on the mIoU metric. Notably, TransVG was trained explicitly on this task. ChEX outperforms TransVG without the CheXzero backbone by 18% mAP and is within 1-std on mIoU. The superiority of ChEX over contrastive models underscores the importance of bounding box supervision for accurate localization.

C.2 Pathology Detection (OD) Tasks

Table 6: Pathology detection (OD) results on the VinDr-CXR, NIH8, and MS-CXR datasets. We indicate the use of training or evaluation images from VinDr-CXR by ^{V}, from NIH8 by ^{N}, and from MIMIC-CXR by ^{M}. All results are based on our own experiments and we provide the std based on bootstrapping with $N = 250$. Best results and those within 1-std are marked in bold.

	Pathology Detection (OD)		
	VinDr^{V}	NIH8^{N}	MS-CXR^{M}
	[mAP]	[mAP]	[mAP]
ChEX^{VM}	14.12±0.95	**11.14**±1.05	**16.60**±1.38
Supervised Object Detection			
FRCNN [35] ^{V}	**18.21**±1.20	6.69±0.82	15.13±1.22
FRCNN [35] (BioVIL [3]) ^{VM}	16.17±1.31	6.56±0.90	**15.64**±1.16
FRCNN [35] (CheXzero [38]) ^{VM}	*no convergence*		
DETR-R50 [5] ^{V}	11.91±0.96	5.90±0.85	8.94±0.35
DETR-BioVIL [3, 5] ^{VM}	3.82±0.59	2.46±0.46	1.94±0.48
DETR-CheXzero [5, 38] ^{VM}	1.26±0.39	1.18±0.34	0.18±0.15
DETR-DC5-R50 [5] ^{V}	11.94±0.89	4.21±0.61	10.30±0.87
DETR-DC5-BioVIL [3, 5] ^{VM}	13.40±1.43	4.48±0.61	8.12±0.69
Conditional DETR-DC5-R50 [23] ^{V}	13.35±0.82	3.93±0.57	13.50±1.09
Conditional DETR-DC5-BioVIL [3, 23] ^{VM}	14.86±1.33	4.97±0.67	11.99±0.98
Deformable DETR-R50 [46] ^{V}	11.49±1.13	2.94±0.50	10.65±0.88
Deformable DETR-BioVIL [3, 46] ^{VM}	12.27±0.94	4.89±0.63	10.26±0.82
$\mathcal{C}$o-Deformable-DETR-R50 [47] ^{V}	16.39±1.13	4.26±0.63	14.86±1.20
$\mathcal{C}$o-Deformable-DETR-BioVIL [3, 47] ^{VM}	15.97±1.00	5.38±0.75	**15.83**±1.42
Weakly-Supervised Object Detection			
CheXnet [33] ^{N}	1.83±0.25	7.12±0.74	7.47±0.60
CheXnet [33] (BioVIL [3]) ^{NM}	1.62±0.17	4.93±0.41	4.66±0.51
CheXnet [33] (CheXzero [38]) ^{NM}	0.30±0.04	0.68±0.18	1.40±0.53
MIL-ADPD [25] ^{M}	4.30±0.47	8.14±0.77	7.21±0.34
MIL-ADPD [25] (BioVIL [3]) ^{M}	2.19±0.27	3.29±0.34	2.20±0.34
Loc-ADPD [25] ^{M}	6.69±0.42	**11.33**±**0.90**	14.71±0.82
Loc-ADPD [25] (BioVil [3]) ^{M}	7.44±0.41	**11.89**±**0.88**	**16.56**±**1.06**
Zero-Shot Contrastive			
BioVIL [3] ^{M}	2.82±0.25	2.63±0.26	7.15±0.52
CheXzero [38] ^{M}	0.54±0.06	2.39±0.24	2.81±0.25

In *pathology detection (OD)* (Tab. 6), ChEX is competitive on 2 of 3 tasks, namely NIH8 and MS-CXR. On VinDr-CXR, where bounding box supervision is available, the SupOD model Faster R-CNN [35], trained on this dataset, performs best, outperforming ChEX by 29%. $\mathcal{C}$o-DETR [47] performs slightly worse but still outperforms ChEX, while other DETR-style models are on-par or inferior to ChEX (ChEX outperforms DETR-DC5 [5] by 18%). ChEX doubles the performance of the best WSupOD model Loc-ADPD [25] (with BioVIL backbone). On NIH8, all SupOD models perform very poor (ChEX almost doubles their best performance), while the WSupOD model Loc-ADPD is competitive with ChEX. On MS-CXR, the best SupOD baseline $\mathcal{C}$o-DETR (with BioVIL backbone), trained on VinDr-CXR, and the best WSupOD model Loc-ADPD are both within 1-std of ChEX.

C.3 Region Classification (RC) Tasks

Table 7: Region Classification (RC) results on the MS-CXR and CIG datasets. We indicate the use of training or evaluation images from VinDr-CXR by ^{V}, from NIH8 by ^{N}, and from MIMIC-CXR by ^{M}. All results are based on our own experiments and we provide the std based on bootstrapping with $N = 250$. Best results and those within 1-std are marked in bold.

	Region Classification (RC)	
	MS-CXR^{M} [AUROC]	Chest ImaGenome^{M} [wAUROC]
ChEX ^{VM}	**82.33**±2.80	**70.46**±0.36
Supervised Image Classifiers (ROI-pooled)		
CheXnet [33] ^{N}	61.46±3.41	59.03±0.21
CheXnet [33] (BioVIL [3]) ^{NM}	55.19±3.29	58.12±0.21
CheXnet [33] (CheXzero [38]) ^{NM}	57.54±2.91	60.02±0.21
ROI-Pool Object Detectors (Classification Heads)		
Faster R-CNN [35] ^{V}	75.66±2.64	58.28±0.22
Faster R-CNN [35] (BioVIL [3]) ^{VM}	76.13±2.55	56.64±0.19
Zeroshot Contrastive Models		
BioVIL [3] ROI Pool ^{M}	67.41±2.80	66.96±0.32
BioVIL [3] Cropped ^{M}	59.56±3.09	60.41±0.26
ChexZero [38] ROI Pool ^{M}	59.97±1.14	65.73±0.35
ChexZero [38] Cropped ^{M}	60.53±3.45	61.80±0.28

On *region classification (RC)* (Tab. 7) tasks, ChEX clearly outperforms all baselines. The best baseline on MS-CXR is the Faster R-CNN with BioVIL backbone, trained on VinDr-CXR, and is outperformed by 8%, while on CIG the best baseline is the contrastive model BioVIL, which is outperformed by 5%.

C.4 Region Explanation (RE) Tasks

Region explanation (RE) can only be performed by a single generative model, namely RGRG [37]. Other report generation models can only predict the full report but cannot provide descriptions for specified regions. On MS-CXR (Tab. 8), our model ChEX performs similar to RGRG on Mic-F1-14 and METEOR but improves by 25% on Mac-F1-14. On CIG (Tab. 9), ChEX outperforms RGRG by 18% on Mic-F1-14, by 40% on Mac-F1-14, and by 29% on METEOR. This result is particularly noteworthy considering RGRG's explicit training for this task and highlights the importance of the contrastive losses in ChEX. Contrastive baselines exhibit poor performance on these generative tasks, as they rely on sentence retrieval based on region similarity.

Table 8: Region explanation (RE) results on the MS-CXR dataset. We indicate the use of training or evaluation images from VinDr-CXR by ^{V}and from MIMIC-CXR by ^{M}. All results are based on our own experiments and we provide the std based on bootstrapping with $N = 10$. Best results and those within 1-std are marked in bold.

	Region Explanation (RE)				
	MS-CXR^{M}				
	[Mic-F1-14]	[Mac-F1-14]	[Mic-F1-5+]	[Mac-F1-5+]	[METEOR]
ChEX ^{VM}	**49.97**±2.24	**20.50**±1.54	**62.54**±1.50	**44.95**±2.23	**8.79**±0.54
Generative Models					
RGRG [37] ^{M}	**48.97**±2.50	16.37±2.00	60.39±2.21	38.34±2.23	8.15±0.78
Zeroshot Contrastive Models (Nearest Neighbor, ROI pooled)					
BioVIL [3] ^{M}	5.86±1.41	3.69±0.67	12.79±2.25	10.95±1.93	3.82±0.03
CheXzero [38] ^{M}	5.41±1.30	3.40±0.78	11.00±2.25	9.02±1.83	3.47±0.45

Table 9: Region explanation (RE) results on the CIG dataset. We indicate the use of training or evaluation images from VinDr-CXR by ^{V}and from MIMIC-CXR by ^{M}. All results are based on our own experiments and we provide the std based on bootstrapping with $N = 10$. Best results and those within 1-std are marked in bold.

	Region Explanation (RE)				
	Chest ImaGenome^{M}				
	[Mic-F1-14]	[Mac-F1-14]	[Mic-F1-5+]	[Mac-F1-5+]	[METEOR]
ChEX ^{VM}	**53.34**±0.43	**29.13**±0.35	**44.53**±0.40	**39.19**±0.37	**10.18**±0.13
Generative Models					
RGRG [37] ^{M}	45.26±0.44	20.88±0.19	31.93±0.31	27.86±0.32	7.88±0.10†
Zeroshot Contrastive Models (Nearest Neighbor, ROI pooled)					
BioVIL [3] ^{M}	24.40±0.38	7.15±0.10	9.42±0.22	7.45±0.16	3.82±0.03
CheXzero [38] ^{M}	21.86±0.35	8.93±0.15	7.78±0.21	6.99±0.17	3.68±0.04

†Results differ to the RGRG paper [37], because RGRG uses a different test split and computes scores only for regions that where selected for prediction by the model.

C.5 Full Report Generation (RG) Task

For *full report generation (RG)* on MIMIC-CXR (Tabs. 10 and 11), ChEX sets a new state-of-the-art on the Ex-F1-14, Mic-F1-5+, and Mac-F1-5+ metrics. Considering the commonly used metric Mic-F1-14, the SOTA model MAIRA-1 [15] – having 7B parameters – outperforms ChEX – having only 1B parameters – by only 6%. More notably, ChEX outperforms the more than ten times larger 12B parameter Med-PaLM M by 2% and is only outperformed by 2% by the 80 times larger 84B parameter version. Limitations of ChEX can be observed on the Mac-F1-14 metric, where ChEX is outperformed by the SOTA model Med-PaLM M 84B by 22% but also by smaller models like Prompt-MRG [17] (by 17%) and RaDialog-RG [30] (by 21%). ChEX, however, improves over RGRG by 2%, indicating that this is a limitation of training based on anatomical regions and is caused by the poor performance of some rarer pathologies (we refer to the supp. material). On the Ex-F1-14 metric, ChEX outperforms the best baseline Prompt-MRG by 23%. Most larger models do however not report this metric.

Table 10: Clinical efficacy (CE) results on the full report generation (RG) task on MIMIC-CXR. We indicate the use of training or evaluation images from VinDr-CXR by $^{\mathrm{V}}$, from MIMIC-CXR by $^{\mathrm{M}}$, and from other sources by $^{\mathrm{O}}$. Except for ChEX, results are taken from the original papers if not marked otherwise. For ChEX, we provide the std based on bootstrapping with $N = 10$. The best results are marked in bold. We note that test splits, pre-, and post-processing can differ between models, leading to limitations in the comparison of exact results as also acknowledged by [15, 37].

	Full Report Generation (RG)				
	MIMIC-CXR$^{\mathrm{M}}$				
	[Mic-F1-14]	[Ex-F1-14]	[Mac-F1-14]	[Mic-F1-5+]	[Mac-F1-5+]
ChEX $^{\mathrm{VM}}$	52.32±0.51	**58.76**±0.42	32.56±0.51	**61.03**±0.56	**55.85**±0.57
RGRG [37] $^{\mathrm{M}}$	49.8	44.7	31.8*	54.7	46.44
CvT-212DistilGPT2 [29] $^{\mathrm{M}}$	44.2	38.4	30.7	–	
M2 Trans w/NLL+BS+fCEN [24] $^{\mathrm{M}}$	–	–	–	56.7	–
METransformer [42] $^{\mathrm{M}}$	–	31.1	–	–	–
Med-PaLM M (12B) [39] $^{\mathrm{MO}}$	51.41	–	37.31	56.54†	50.57†
Med-PaLM M (84B) [39] $^{\mathrm{MO}}$	53.56	–	**39.83**	57.88†	51.60†
Med-PaLM M (562B) [39] $^{\mathrm{MO}}$	51.60	–	37.81	56.28†	49.86†
MAIRA-1 [15] $^{\mathrm{M}}$	**55.7**	–	38.6	58.8	51.7
Prompt-MRG [17] $^{\mathrm{M}}$	–	47.6	38.1	–	–
OmniFM-DR [45] $^{\mathrm{MO}}$	–	33.3	–	–	
RaDialog-INS [30] $^{\mathrm{MO}}$	–	–	38.6	–	–
RaDialog-RG [30] $^{\mathrm{MO}}$	–	–	39.4	–	–
COOMG [9]	–	34.5	–	–	–
ORGAN [12]	–	38.5	–	–	–

† Limited comparability as F1-5 was reported instead of F1-5+, i.e. uncertain cases were treated as positive instead of negative.

* Reported number taken from the Prompt-MRG [17] paper.

On the Mic-F1-5+ and Mac-F1-5+ metrics, ChEX outperforms the large SOTA model MAIRA-1 by 4% and 8%, respectively. In language-based metrics such as METEOR, ChEX shows relatively low performance (60% lower than the best and 21% lower than the second-best model) due to not being explicitly trained for full report generation but generating reports based on a set of queries. While this approach offers advantages (see Sec. 5.4 of the main paper), it may result in reports differing from reference reports. However, it's important to note that language-based metrics offer limited insights into the factual correctness of generated reports [24].

In Tab. 12, we provide the CE metric F1 per pathology/finding and compare the results between ChEX and the two baselines RGRG and MAIRA-1.

Table 11: Natural language generation (NLG) results on the full report generation (RG) task on MIMIC-CXR. We indicate the use of training or evaluation images from VinDr-CXR by $^{\mathrm{V}}$, from MIMIC-CXR by $^{\mathrm{M}}$, and from other sources by $^{\mathrm{O}}$. Except for ChEX, results are taken from the original papers if not marked otherwise. For ChEX, we provide the std based on bootstrapping with $N = 10$. The best results are marked in bold. As in Tab. 10, we again emphasize the limitations when comparing results.

	Full Report Generation (RG)				
	MIMIC-CXR$^{\mathrm{M}}$				
	[BLEU-1]	[BLEU-4]	[METEOR]	[ROUGE-L]	[CIDEr]
ChEX $^{\mathrm{VM}}$	29.58±0.24	6.52±0.11	13.26±0.10	19.49±0.10	18.49±0.79
RGRG [37] $^{\mathrm{M}}$	37.3	12.6	16.8	26.4	**49.5**
CvT-212DistilGPT2 [29] $^{\mathrm{M}}$	39.2	12.4	15.3	28.5	36.1
M2 Trans w/ NLL+BS+fCEN [24] $^{\mathrm{M}}$	–	11.4	–	–	50.9
METransformer [42] $^{\mathrm{M}}$	38.6	12.4	15.2	29.1	36.2
Med-PaLM M (12B) [39] $^{\mathrm{MO}}$	30.90	10.43	–	26.16	23.43
Med-PaLM M (84B) [39] $^{\mathrm{MO}}$	32.31	11.31	–	27.29	26.17
Med-PaLM M (562B) [39] $^{\mathrm{MO}}$	31.73	11.50	–	27.49	25.27
MAIRA-1 [15] $^{\mathrm{M}}$	39.2	**14.2**	**33.3**	28.9	–
Prompt-MRG [17] $^{\mathrm{M}}$	39.8	11.2	15.7	26.8	–
OmniFM-DR [45] $^{\mathrm{MO}}$	–	11.0	14.0	26.5	–
RaDialog-INS [30] $^{\mathrm{MO}}$	34.0	9.7	13.6	27.0	–
RaDialog-RG [30] $^{\mathrm{MO}}$	34.6	9.5	14.0	27.1	–
ITA [40] $^{\mathrm{M}}$	39.5	12.1	14.7	28.4	–
COOMG [9] $^{\mathrm{M}}$	34.6	10.4	13.7	27.9	–
COOMG-RL [9] $^{\mathrm{M}}$	36.3	12.4	12.8	29.0	–
ORGAN [12]	**40.7**	12.3	16.2	29.3	–
BioViL-T [1] $^{\mathrm{M}}$	–	9.2	–	**29.6**	–

Table 12: CE results (F1) per pathology/finding for ChEX, RGRG, and MAIRA-1 on the full report full report generation (RG) task on MIMIC-CXR. Except for ChEX, results are taken from the original papers. Our method ChEX performs best on 5 of the 14 classes, even outperforming the much larger MAIRA-1. On 9 of the 14 classes, it outperforms RGRG, where it is able to describe findings and pathologies like lung lesions or enlarged cardiomediastinum, which RGRG is unable to detect at all.

	F1		
Pathology/Finding	ChEX	RGRG [37]	MAIRA-1 [15]
Cardiomegaly	**66.62**±1.08	62.4	64.0
Edema	**54.81**±1.48	51.4	44.0
Consolidation	15.27±2.10	7.8	**20.0**
Atelectasis	45.99±1.43	**54.6**	41.3
Pleural Effusion	**69.11**±1.04	56.0	**68.9**
Enlarged Cardiomediastinum	6.81±1.74	0.3	**11.9**
Fracture	0.00±0.00	0.0	**24.9**
Lung Lesion	14.24±3.05	0.7	**18.8**
Lung Opacity	**52.32**±1.06	26.8	49.8
Pleural Other	0.00±0.00	0.2	**14.7**
Pneumonia	**19.30**±1.55	16.2	**18.3**
Pneumothorax	17.59±3.21	15.9	**40.8**
Support Devices	69.40±0.92	70.9	**84.5**
No Finding	24.36±2.97	**63.2**	38.6

D Model Implementation Details

D.1 Model Architecture and Components

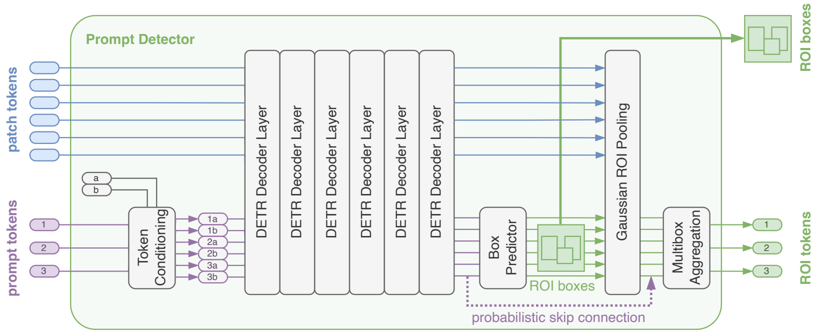


Fig. 5: Prompt detector

Prompt Detector Fig. 5 provides an overview of the prompt detector. Given is an input of Q prompt tokens $\boldsymbol{q}_k$ with $k \in \{1, \ldots, Q\}$. In order to enable the prediction of up to $M = 3$ bounding boxes per prompt token $\boldsymbol{q}_k$, we utilize M learned, randomly initialized, box tokens $\boldsymbol{t}_m$. We then additively condition these tokens on the Q queries resulting in a total of $Q \cdot M$ query tokens $\boldsymbol{t}'_{k,m}$:

$$\boldsymbol{t}'_{k,m} = \boldsymbol{q}_k + \boldsymbol{t}_m \,. \tag{1}$$

These tokens serve as the initial queries for the DETR decoder layers, while the patch tokens from the image encoder are used for keys and values. We use a total of 6 DETR-style transformer decoder layers. Self-attention is restricted, allowing attention only between tokens from the same query, *i.e.* with the same k. The query tokens $\boldsymbol{t}'_{k,m}$ are only used as queries for the first layers, later layers use the output of the previous layer as queries. We additively combine the patch tokens with the corresponding positional encodings from the ViT before using them in the prompt detector. The outputs of the last decoder layer are used as preliminary ROI tokens $\tilde{\boldsymbol{r}}_{k,m}$. They are then used by the MLP-based box predictor to predict the corresponding bounding boxes, where we apply sigmoid to return relative image coordinates.

Next, we use Gaussian ROI pooling following [26]. For each bounding box, we define a 2D multivariate Gaussian with independent x and y components (*i.e.* with zero covariance) using the box center as mean and the box size as standard deviation. The probability density function (pdf) of this Gaussian is used in a weighted aggregation of the patch tokens (which have been projected by an MLP), resulting in a single feature vector per ROI (k, m). We randomly add the preliminary ROI tokens $\tilde{\boldsymbol{r}}_{k,m}$ in 25% of the cases but apply dropout to them before adding. Finally, these features are projected using an MLP with two hidden layers resulting in the ROI tokens $\boldsymbol{r}_{k,m}$. We use an additional MLP followed by sigmoid on these ROI tokens to predict box scores $s_{k,m}$. Using these scores, we now aggregate the ROI tokens for each prompt k to get the final (aggregated) ROI tokens $\boldsymbol{r}_k$:

$$\boldsymbol{r}_k = \frac{\sum_m s_{k,m} \cdot \boldsymbol{r}_{k,m}}{\sum_m s_{k,m}} \,. \tag{2}$$

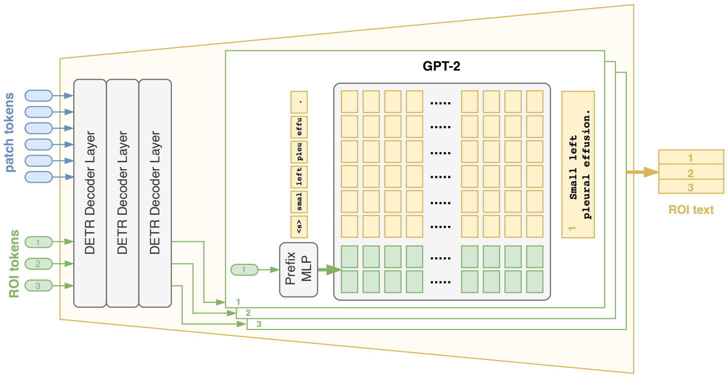


Fig. 6: Sentence generator with post decoder on the left and the conditioned language model on the right

Sentence Generator Fig. 5 provides an overview of the sentence generator. It consists of two major components, the post decoder and the language model (GPT-2) with conditioning. The post decoder consists of three DETR decoder layers. For each ROI k, we use the per-bounding-box ROI tokens $\boldsymbol{r}_{k,m}$ and the aggregated ROI token $\boldsymbol{r}_k$ as queries, leading to a total of $M + 1 = 4$ queries per ROI. We randomly drop the per-bounding-box ROI tokens $\boldsymbol{r}_{k,m}$ with a 30% probability while always keeping the aggregated tokens $\boldsymbol{r}_k$. We use the patch tokens and the (projected) CLS token from the image encoder as keys and values. Self-attention is restricted, allowing attention only between queries for the same ROI k. After the final decoder layer, we only keep the output features associated with the aggregated ROI token of each ROI k, discarding the features of the per-bounding-box ROI tokens. The final output of the post decoder is computed by summing the outputs of the final decoder layer with the input features, introducing a skip connection. Before summing, we multiply the output features with a learned gating factor, which is initialized by zero.

For text generation, we condition the GPT-2 medium [32] language model on the output features of the post decoder. The descriptions of each ROI k are predicted individually, treating each k as an independent sample. We follow a similar approach as P-tuning v2 [21] but do not freeze any parameters of the language model. Specifically, we use an MLP to project the feature vector for each k into a prefix for the language model. We use a prefix of five tokens and compute key and value features for each language model layer. During generation, we use a maximum length of 128 generated tokens.

Futher Implementation Details For all components of the prompt detector and post decoder of the sentence generator, if not otherwise specified, we use the model dimension 512 with hidden dimension 2048 for MLPs, dropout 0.3, attention dropout 0.1, drop-path [14] probability 0.2, and GELU [11] non-linearities. For attention layers we apply the layer norm before each operation and use layer scaling with initialization 0.1.

Number of Parameters ChEX has an overall 1051M parameters of which we keep 123M frozen throughout our whole training. The ViT image encoder contains 88M, the text encoder 63M, the prompt detector 28M, and the sentence generator 872M parameters.

D.2 Supervision and Loss Functions

Pathology Tokens Fig. 7 visualizes training with pathology tokens. Given the set of pathology classes $\mathcal{C}$ from VinDr-CXR (22 classes with bounding boxes), for each pathology class $c \in \mathcal{C}$, we encode its positive prompt (*e.g.*, "Pleural effusion") and its negative prompt (*e.g.*, "No pleural effusion") using the prompt encoder. We denote the resulting prompt tokens as $\boldsymbol{q}_c^+$ and $\boldsymbol{q}_c^-$, respectively. Next, we pass the positive tokens $\boldsymbol{q}_c^+$ to the prompt decoder, which predicts three ROI bounding boxes $\hat{b}_{c,m}$ with box scores $s_{c,m} \in [0,1]$ (with $m \in \{1,2,3\}$) and the aggregated ROI token $\boldsymbol{r}_c$.

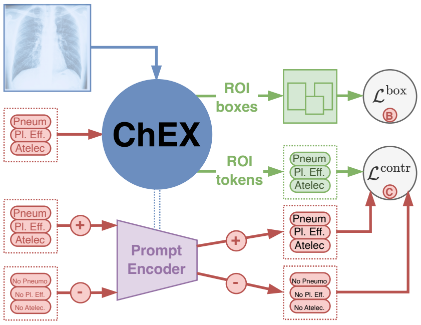


Fig. 7: Training with pathology tokens

Pathology Detection (Bounding Box Loss) For the bounding boxes, we apply the modified DETR loss. The class match probability $p_{c,m}$ of a bounding box is computed as

$$p_{c,m} = \frac{\exp\left(\boldsymbol{r}_{c,m} \bullet \boldsymbol{q}_c^+\right)}{\exp\left(\boldsymbol{r}_{c,m} \bullet \boldsymbol{q}_c^+\right) + \exp\left(\boldsymbol{r}_{c,m} \bullet \boldsymbol{q}_c^-\right)} \,, \tag{3}$$

where $\bullet$ denotes the dot product and $\boldsymbol{r}_{c,m}$ denotes the (non-aggregated) ROI token for each bounding box. Using the standard formulation of the box loss

$$\mathcal{L}_{\text{box}}\left(\hat{b}, b\right) = 5 \cdot \mathrm{L}_1\left(\hat{b}, b\right) + 2 \cdot \mathrm{gIoU}\left(\hat{b}, b\right) \,, \tag{4}$$

we compute the matching cost for class c, predicted box m, and target box m' based on the box cost, class match probability of the box, and box score as

$$\mathcal{L}_{\text{match}}(c, m, m') = \mathcal{L}_{\text{box}}\left(\hat{b}_{c,m}, b_{c,m'}\right) + p_{c,m} + 3 \cdot s_{c,m} \,. \tag{5}$$

For each positive class c, *i.e.* each class that has target bounding boxes, we iteratively apply Hungarian matching using the matching cost $\mathcal{L}_{\text{match}}$, where matching is applied individually for each c. We start with an initial matching. If non-matched predicted boxes remain for that class, we apply matching again with the remaining predicted boxes and all target boxes until all predicted boxes are matched, allowing a target box to be matched with several predicted boxes. We apply the box loss $\mathcal{L}_{\text{box}}$ to all pairs of matched predicted/target boxes. Additionally, we apply the focal loss $\mathcal{L}_{\text{focal}}$ to the box scores $s_{c,m}$ with positive targets for boxes matched in the first iteration and negative targets for other boxes and all negative classes. Note that we compute the α parameter of the focal loss based on the frequency of positive targets per class in the current batch. The final pathology detection loss is computed as $\mathcal{L}_{\text{patho-detect}} = \mathcal{L}_{\text{box}} + 3 \cdot \mathcal{L}_{\text{focal}}$.

Pathology Classification (Contrastive Loss) We additionally apply a contrastive loss on the aggregated ROI token $\boldsymbol{r}_c$. Here, we consider the pathology targets $y_c \in \{0, 1\}$, which define whether the pathology c is present in the image. The ROI token for class c is paired with its positive or negative pathology prompt $\boldsymbol{q}_c^+$ or $\boldsymbol{q}_c^-$ depending on y_c. As negative pairs, we include the opposing prompt alongside all positive and negative prompts of all other pathologies. This leads to the following loss function:

$$\mathcal{L}_{\text{patho-cls}} = -\frac{1}{|\mathcal{C}|} \sum_{c} \log \frac{y_c \cdot \exp(\cos(\boldsymbol{r}_c, \boldsymbol{q}_c^+)/\tau) + (1 - y_c) \cdot \exp(\cos(\boldsymbol{r}_c, \boldsymbol{q}_c^-)/\tau)}{\sum_{c'} \exp(\cos(\boldsymbol{r}_c, \boldsymbol{q}_{c'}^+)/\tau) + \exp(\cos(\boldsymbol{r}_c, \boldsymbol{q}_{c'}^-)/\tau)} \,, \tag{6}$$

where we set the temperature τ to 0.2.

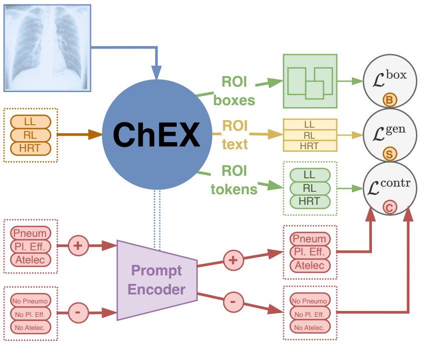


Fig. 8: Training with anatomy tokens

Anatomy Tokens Fig. 8 visualizes training with anatomy tokens. Given is the set of anatomical regions $\mathcal{A}$ from CIG (29 regions from CIG and additionally 5 anatomical regions each consisting of 2 regions from CIG, *e.g.* lungs consisting of the right and left lungs). For each region $a \in \mathcal{A}$, we encode its prompt using the prompt encoder and denote the resulting prompt tokens as $\boldsymbol{q}_a$. We pass these tokens to the prompt decoder, which predicts three ROI bounding boxes $\hat{b}_{a,m}$ and the aggregated ROI token $\boldsymbol{r}_a$. Additionally, we encode positive and negative prompts for pathologies assigned to regions (we use 53 pathologies and findings from CIG) and denote the resulting prompt tokens as $\boldsymbol{q}_c^+$ and $\boldsymbol{q}_c^-$, respectively.

Anatomy Detection (Bounding Box Loss) For anatomical regions with a single target bounding box, we match all three predicted boxes for that region to the target bounding box. For regions with multiple target boxes, we apply the same matching strategy as used for pathology bounding boxes but using only $\mathcal{L}_{\text{box}}$ as the matching cost. Finally, we apply the box loss $\mathcal{L}_{\text{box}}$ to all matched bounding boxes. Box scores are not trained for anatomy tokens.

Anatomy Classification (Contrastive Loss) We again apply a contrastive loss to the ROI tokens. Here we use the positive and negative pathology tokens $\boldsymbol{q}_c^+$ and $\boldsymbol{q}_c^-$ and align the ROI token $\boldsymbol{r}_a$ of anatomy a based on whether the pathologies are present in this region. The presence of pathology c in anatomy a is given as $y_{a,c}$ and is provided by CIG. This leads to the following loss function:

$$\mathcal{L}_{\text{anat-cls}} = \\ -\frac{1}{|\mathcal{A}|}\sum_a \log \frac{\exp\left\{\frac{1}{|\mathcal{C}|}\sum_c y_{a,c}\cdot\cos(\boldsymbol{r}_a,\boldsymbol{q}_c^+)/\tau) + (1-y_{a,c})\cdot\cos(\boldsymbol{r}_a,\boldsymbol{q}_c^-)/\tau)\right\}}{\sum_{c'}\exp(\cos(\boldsymbol{r}_a,\boldsymbol{q}_{c'}^+)/\tau) + \exp(\cos(\boldsymbol{r}_a,\boldsymbol{q}_{c'}^-)/\tau)} \tag{7}$$

where we set the temperature τ to 0.25 and randomly subsample a maximum of 10 negative classes (*i.e.* classes not present in any of the regions in the current batch).

Anatomy Sentence Generation For sentence generation, we condition the sentence generator on the ROI tokens $\boldsymbol{r}_a$ and train it to autoregressively predict the target sentences assigned to this region using the standard cross-entropy loss $\mathcal{L}_{\text{anat-gen}}$. Note that we concatenate sentences if multiple sentences are assigned to a region and ignore regions without sentences. Additionally, the MSE loss $\mathcal{L}_{\text{anat-mse}}$ is applied between the output of the post decoder and the anatomy sentences encoded by the prompt encoder.

Sentence Tokens Fig. 9 visualizes training with sentence tokens. Given are the sentences $\mathcal{S}^{(i)}$ of sample i from MIMIC-CXR. We encode each sentence s using the prompt encoder and denote the resulting prompt tokens as $\boldsymbol{q}_s^{(i)}$. We pass each of these tokens to the prompt decoder, which predicts the aggregated ROI token $\boldsymbol{r}_s^{(i)}$ for each of them.

Sentence Contrastive Loss We again apply a contrastive loss between the ROI tokens $\boldsymbol{r}_s^{(i)}$ and the prompt tokens $\boldsymbol{q}_s^{(i)}$. Unlike for pathology and anatomy tokens, where the contrastive loss was applied for each sample and token independently, we utilize sentence prompt tokens from the same and other samples in the batch as negatives. This leads to the following loss function:

$$\mathcal{L}_{\text{sent-contr}} = -\frac{1}{N}\sum_i \frac{1}{|\mathcal{S}^{(i)}|}\sum_{s\in\mathcal{S}^{(i)}} \log \frac{\exp\left(\cos\left(\boldsymbol{r}_s^{(i)}, \boldsymbol{q}_s^{(i)}\right)/\tau\right)}{\sum_{j,s'}\exp\left(\cos\left(\boldsymbol{r}_s^{(i)}, \boldsymbol{q}_{s'}^{(j)}\right)/\tau\right)}\,, \tag{8}$$

where N is the batch size and we set the temperature τ to 0.25.

Sentence Generation For sentences generation, *i.e.* sentence reconstruction, we condition the sentence generator on the ROI tokens $\boldsymbol{r}_s$ and train it to autoregressively reconstruct the report sentences used as prompts, using the standard cross-entropy loss $\mathcal{L}_{\text{sent-gen}}$. Additionally, the MSE loss $\mathcal{L}_{\text{sent-mse}}$ is applied between the output of the post decoder and the encoded sentence prompts $\boldsymbol{q}_s$.

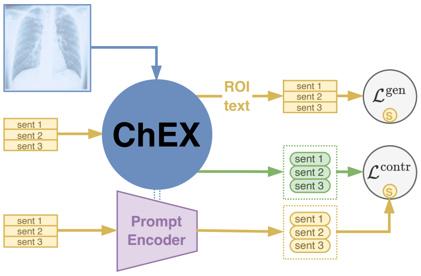


Fig. 9: Training with sentence tokens

Global Contrastive Loss To fine-tune and stabilize the image encoder, we additionally apply standard CLIP loss $\mathcal{L}_{\text{global-contr}}$ between the (global) average pooled projected patch features and the average pooled sentences tokens $\boldsymbol{q}_s^{(i)}$. We use 0.2 as the temperature.

D.3 Multistage Training

We train ChEX in three different stages where we add new components with each stage while freezing some already trained components. Additionally, we prepare the language model for MIMIC-CXR reports in a text-only pre-training stage.

Stage 1 The goal of stage 1 is to train the prompt detector and fine-tune the image encoder. We initialize the image and text encoder as well as their projection heads from the CheXzero [38] weights of one of their checkpoints [4] which we retrieved from their provided cloud drive[5]. The text encoder and its projection head are completely frozen throughout all training stages, and we freeze only the embedding layer and the first eight transformer layers of the image encoder for stage 1. The prompt detector is initialized randomly and will be trained in this stage. The sentence decoder is not yet used in this stage. We train using both datasets (MIMIC-CXR and VinDr-CXR) and all types of tokens – pathology tokens for samples from VinDr-CXR, anatomy tokens for the CIG targets for MIMIC-CXR samples, and sentence tokens for MIMIC-CXR

[4] `best_64_5e-05_original_22000_0.864.pt`

[5] `https://drive.google.com/drive/folders/1makFLiEMbSleYltaRxw81aBhEDMpVwno`

samples. We use all bounding box and contrastive losses but do not yet train text generation. Overall, the loss function for stage 1 is defined as

$$\begin{aligned}\mathcal{L}_{\text{stage1}} =& 10 \cdot \mathcal{L}_{\text{patho-detect}} + 1 \cdot \mathcal{L}_{\text{patho-cls}} \\ &+ 0.1 \cdot \mathcal{L}_{\text{anat-detect}} + 0.005 \cdot \mathcal{L}_{\text{anat-cls}} \\ &+ 0.005 \cdot \mathcal{L}_{\text{sent-contr}} + 1 \cdot \mathcal{L}_{\text{global-contr}}\end{aligned} \tag{9}$$

Note that for anatomy tokens we only use the textual prompts in this stage, bounding box-based queries are not yet used. To improve generalization and limit the GPU memory consumption, we randomly select a subset from the possible tokens (*i.e.* queries for pathologies and anatomical regions) to be used in the current batch. We subsample 10 pathologies for bounding box training, 10 pathologies for the contrastive loss, and 20 anatomical regions.
We train with a batch size of 256 for 50 epochs on a single Nvidia A40, which takes around 36 hours. We train with AdamW [22] and use a cosine annealing schedule with a linear warmup of 1000 steps, initial learning rate 1×10^{-4} , and minimum learning rate 1×10^{-7} Additionally, we use a weight decay of 1×10^{-5} , clip gradients with a norm larger than 1.0 During training, we use the transformations from the Albumentations library [4]. We randomly crop and resize all images to a resolution of 224 × 224 using a crop scale range of $[0.5, 1.0]$. We then apply random horizontal flips with 50% probability, random affine transformations with rotation angle range $[-10, 10]$, translation range $[-0.02, 0.02]$, and scaling range $[0.95, 1.0]$. We also apply random contrast/brightness jittering with contrast/brightness range $[0.6, 1.4]$ and random Gaussian blurring with 50% probability and sigma range $[0.1, 5.0]$.

Stage 2 The goal of stage 2 is to train the post decoder of the sentence generator. We use the trained components from stage 1 but freeze all of them, *i.e.* we fully freeze the prompt encoder, image encoder, and prompt detector. We now add the (randomly initialized) post decoder of the sentence generator, however, without using the language model yet. We train only on the MIMIC-CXR dataset (including targets from CIG) and only use sentence and anatomy tokens. The VinDr-CXR dataset and pathology tokens are not used. We use the contrastive losses for anatomy and sentence tokens and additionally use MSE losses to reconstruct the sentence embeddings of target sentences based on ROI tokens from either anatomy or sentence prompts. We do not train bounding boxes or sentence generation. Overall, the loss function for stage 2 is defined as

$$\begin{aligned}\mathcal{L}_{\text{stage2}} =& 0.01 \cdot \mathcal{L}_{\text{anat-cls}} + 0.04 \cdot \mathcal{L}_{\text{anat-mse}} \\ &+ 0.005 \cdot \mathcal{L}_{\text{sent-contr}} + 0.02 \cdot \mathcal{L}_{\text{sent-mse}}\end{aligned} \tag{10}$$

For anatomical regions we randomly decide for each sample whether to use bounding-box-based or text-prompt-based queries. We train with a batch size of 256 and initial learning rate 1×10^{-3} for 200 epochs on a single Nvidia A40, which takes around 24 hours. All other hyperparameters are kept the same as in stage 1.

Text-only Pre-Training The goal of text-only pre-training is to prepare the decoder language model for MIMIC-CXR report sentences. We initialize the GPT-2 medium language model from a checkpoint[6] pre-trained on PubMed abstracts. We only use the frozen prompt encoder together with the language model and a (randomly initialized) MLP projection for P-tuning, *i.e.* we do not use the post decoder. Using the sentences from MIMIC-CXR, we encode each sentence, apply the MLP projection, and reconstruct the original sentence using the language model. We train the language model and MLP projection using autoregressive training.
We use a batch size of 32 reports (each with one or more sentences) but randomly drop sentences if there are more than 64 in a batch and use 8 gradient accumulation steps. We train with an initial learning rate of 3×10^{-4} and 10,000 warmup steps for 50 epochs on a single Nvidia A40, which takes around 3 days.

Stage 3 The goal of stage 3 is to train the whole sentence generator. We use the pre-trained post decoder from stage 2 and the decoder language model from text-only pre-training, both of which will be trained in this stage. Additionally, we use the prompt encoder, image encoder, and prompt detector from stage 1 but keep all of them frozen. We train only on the MIMIC-CXR dataset (including targets from CIG) and only use sentence and anatomy tokens. We autoregressive train the sentence generator to predict the target sentences of each anatomical region and to reconstruct the sentences for sentence tokens. Additionally, we use MSE losses between the output of the post decoder and the sentences encoded by the prompt encoder. Overall, the loss function for stage 3 is defined as

$$\begin{aligned}\mathcal{L}_{\text{stage3}} =& 1 \cdot \mathcal{L}_{\text{anat-gen}} + 0.04 \cdot \mathcal{L}_{\text{anat-mse}} \\ &+ 0.5 \cdot \mathcal{L}_{\text{sent-gen}} + 0.02 \cdot \mathcal{L}_{\text{sent-mse}}\end{aligned} \tag{11}$$

During training, we randomly drop the per-bounding-box ROI tokens for a query with a 10% probability but always keep the aggregated ROI token. We use a batch size of 8 reports (with 32 gradient accumulation steps) but restrict the total number of generated sentences per batch to 32 for anatomical regions and 32 for sentence tokens, randomly subsampling if a batch contains more. We train with an initial learning rate of 3×10^{-4} and 10,000 warmup steps for 20 epochs on a single Nvidia A40, which takes around 4 days.

D.4 Inference

Sentence Grounding (SG) and Object Detection (OD) For SG and OD, we use textual prompts as queries (the given input sentences in SG and predefined prompts, the names of pathology classes, for OD). The encoded prompts are given to the prompt detector, which predicts bounding boxes with box scores and ROI token vectors. For SG, we directly predict these bounding boxes with their associated box scores. For OD, we compute cosine similarities between the

[6] `https://huggingface.co/healx/gpt-2-pubmed-medium`

ROI tokens and positive as well as negative pathology prompts to compute box class probabilities. We then multiplicative combine these box class probabilities with the predicted box scores to get the final box scores. Finally, we apply task-specific post-processing as described in Appendix E.2.

Region Classification (RC) and Explanation (RE) For RC and RE, bounding boxes are given as inputs. We use these bounding boxes as inputs for the Gaussian ROI pooling layer of the prompt detector to compute ROI tokens for each input based on the patch features. Other parts of the prompt detector and the prompt encoder are not used for this task. We then further process the ROI tokens by the post decoder of the sentence decoder to further condition them on the patch features.

For RC, we now compute the cosine similarity between the post decoder outputs and encoded pathology prompts. Finally, we classify each region based on the cosine similarity. For the multiclass task MS-CXR, we compute the probability using softmax over all pathologies and predict the pathology class with maximum similarity. For the multilabel binary task CIG, we compute the softmax over the positive and negative prompts for each pathology class independently and predict all classes where the similarity with the positive pathology prompt is larger than the negative prompt.

For RE, we use the post decoder outputs to condition the language model and generate the descriptions for each region independently.

Full Report Generation (RG) For RG, we use pre-defined sets of textual prompts. These prompts may include aspects like pathologies or anatomical regions as studied in Appendix A.4. As pathology prompts, we use the set of 14 findings and pathologies from CheXpert [16]. As anatomy prompts we use a subset of 15 anatomical region names from the regions used in CIG, namely the lower, mid, and upper lung zones, hilar structures, apical zones, as well as the clavicle, each for left and right independently, and additional the mediastinum, the cardiac silhouette, and the abdomen. We encode the prompts with the prompt encoder and then apply the prompt detector to predict bounding boxes and ROI tokens for each prompt. The ROI tokens are then used by the sentence generator to predict textual descriptions for each prompt.

Next, we filter the descriptions to include in the final report. We compute the positive probability for each prompt (and the associated description) by classifying each ROI token similarly as in the region classification task using the CheXpert classes excluding no-finding. If any pathology is positive, we consider the prompt and associated description to be positive. We decided to keep all descriptions of pathology prompts but only positive anatomy descriptions (we refer to Appendix A.4).

Finally, we filter the descriptions based on their associated box scores. We use a dynamic approach where we identify similar descriptions from different prompts by computing the cosine similarity between the descriptions encoded by the prompt encoder. For descriptions with high similarity, the lower-scoring ones are excluded. By setting the similarity threshold low enough, we allow the model to not only remove synonymous descriptions but also contradicting descriptions of similar aspects or several negative findings, where we only keep the most confident and relevant ones.

E Evaluation Details

E.1 Benchmark Datasets

MIMIC-CXR We use the MIMIC Chest X-ray (MIMIC-CXR) Database v2.0.0 [8, 18, 19], which consists of chest X-ray images with corresponding free-text radiology reports from 227,835 radiographic studies of 65,079 patients performed at Beth Israel Deaconess Medical Center in Boston, US. The dataset authors de-identified the dataset satisfying the US Health Insurance Portability and Accountability Act of 1996 (HIPAA) Safe Harbor requirements and assuring Protected health information (PHI) has been removed. For all MIMIC-CXR-based datasets, *i.e.* MIMIC-CXR, CIG, and MS-CXR, we only use the frontal chest X-ray images from the already pre-processed MIMIC-CXR-JPG [20] version.
We download the free-text radiology reports from the original MIMIC-CXR dataset and extract the text from the `Findings` and `Impression` sections. Reports containing none of these sections, where both of these sections are empty, or which contain less than two words are removed. We split each of these sections into sentences using Stanza [31] and use all of these sentences (from both sections) as the report sentences of that sample. If a full report is required, as in the full report generation task, we concatenate all sentences to a single string. We follow the official MIMIC-CXR train/val/test split. This leads to a total of 1,879 images/report pairs for validation and 3,082 for testing.

Chest ImaGenome (CIG) For anatomical regions, including bounding boxes with associated sentences and pathologies, we utilize the Chest ImaGenome dataset (CIG) [8,43,44]. It provides scene graphs automatically constructed from the free-text radiology reports from MIMIC-CXR. Each scene graph includes bounding boxes for 29 unique anatomical regions with associated attributes, where we consider 43 `anatomical finding` and 10 `disease` attributes as binary anatomy-level classification labels. Additionally, each region is associated with sentences from the report and we use these sentences as target sentences for anatomical regions. Note that not all sentences from the report may be associated with a region, sentences can be associated with several regions, and each region can be associated with zero, one, or multiple sentences. We only keep samples with scene graphs containing at least five valid region bounding boxes. We use the images from the MIMIC-CXR dataset (see above) and follow the official MIMIC-CXR train/val/test split[7], leading to 1,959 images for validation and 3402 for testing. For multitask training of ChEX we combine the supervision

[7] Note that this means that we do not follow the official split from CIG.

from CIG with the sentences from MIMIC-CXR, leading to a total of 227,382 training images.

MS-CXR The MS-CXR dataset [2, 3, 8], which we use only for evaluation, provides additional annotation for a subset of MIMIC-CXR. It contains 1,162 image–sentence pairs of bounding boxes and corresponding phrases for 8 pathologies annotated by board-certified radiologists. Each sentence is associated with a pathology and one or more bounding boxes. Images in MS-CXR contain at least one sentence (and pathology) but may also contain several sentences with possibly different pathologies. We use the images from the MIMIC-CXR dataset. However, only very few images of MS-CXR are part of the MIMIC-CXR test split. For MS-CXR tasks, we thus use all MS-CXR images from the validation and test split of MIMIC-CXR for testing but exclude all images from the training split to avoid any data leakage. This leads to a total of 169 test images.

VinDr-CXR We use the VinDr-CXr [8, 27, 28] dataset, containing 18,000 postero-anterior (PA) view, *i.e.* frontal, chest X-ray images retrospectively collected from the Hospital 108 (H108) and the Hanoi Medical University Hospital (HMUH), two of the largest hospitals in Vietnam. The images were manually annotated by radiologists with 22 localized labels (with bounding boxes) and 6 global (image-level classification) labels of diseases (pathologies). Each image may be associated with zero, one, or more pathologies, and for each (localized) pathology there may be one or more provided bounding boxes. In order to protect the patient's privacy and meet the U.S. HIPAA, the European GDPR, and the local privacy laws, all personally identifiable information associated with the images has been removed or replaced with random values [27]. The images are provided in the DICOM format, and we converted them into PNG files using the `pydicom`[8] library. The dataset authors provide a train/test split consisting of 15,000 and 3,000 samples, respectively. The samples in the train split contain bounding box annotations from individual radiologists that have not been merged by the dataset authors. We fuse bounding box annotations from different radiologists corresponding to the same pathology using weighted box fusion [36] with IoU threshold 0.1. Samples from the provided test split come with merged annotations, such that no further processing is required. We use all samples from the training split for training and randomly split the 3,000 test split samples into 1,500 samples for validation and 1,500 samples for testing.

NIH ChestXray8 (NIH8) We use the ChestXray-8 (NIH8) dataset [41], which consists of 108,948 chest X-ray images retrospectively collected from the National Institutes of Health Clinical Center in the US. The authors do not provide information about data anonymization. The images are provided as pre-processed PNG-images (extracted from DICOM files) such that no further pre-processing is applied.

[8] `https://pydicom.github.io/`

The dataset contains a subset of images with bounding boxes for eight different disease types hand-labeled by a board-certified radiologist. We randomly split these images into 432 samples for validation and 448 samples for testing.

E.2 Benchmark Task Evaluation

Task-specific Inference Hyperparameters While no task-specific finetuning is applied, we use task-specific inference and post-processing hyperparameters for ChEX and all baselines to consider differences in annotation practices of datasets. We tune those hyperparameters on samples from the validation splits of the corresponding datasets.

For the sentence grounding (SG) task, we scale all predicted boxes by a constant factor which we optimize on the validation set. Additionally, we apply NMS with an IoU threshold of 0.25. For the pathology detection (OD) tasks, we scale boxes independently for each pathology class. We apply NMS with an IoU threshold of 0.25 for the VinDr-CXR and MS-CXR datasets. For the NIH8 dataset, we merge all predicted boxes per pathology (using the super bounding box covering all predicted boxes) and report the maximum of the predicted score as its box score.

Metrics and Evaluation For SG, we use the sentences from MS-CXR as input and the corresponding bounding boxes as targets. We compute mIoU using the thresholds $[0.1, 0.2, 0.3, 0.4, 0.5]$. For each sentence and threshold, we select the predicted bounding boxes based on the box scores and merge them into a single mask (corresponding to one or more boxes). Similarly, we merge the target boxes of that sentence before computing the IoU between the predicted and target mask. Finally, the mIoU is computed by first averaging the IoU per threshold over all sentences before averaging over all thresholds. For mAP computation, we treat the task as object detection, where each sentence corresponds to its own class, and then use the IoU thresholds $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$.

For OD, we use the pathology classes provided in the three datasets (VinDR-CXR, MS-CXR, NIH8), where for VinDr-CXR, we only use the top 15 most frequent classes. We report the standard mAP metric with IoU thresholds $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$.

For RC, we provide bounding boxes of regions (pathologies for MS-CXR, anatomical regions for CIG) as input and expect the model to classify these regions. On MS-CXR this is a multiclass classification problem over 8 classes. Thus, we use the macro-averaged AUROC metric (one-vs-rest). On CIG, this is a multilabel binary classification problem with 53 highly imbalanced classes. Thus, we use the weighted AUROC (wAUROC) metric, which considers the frequency of classes.

For RE tasks, we provide bounding boxes of regions (pathologies for MS-CXR, anatomical regions for CIG) as input and expect the model to predict textual descriptions for them. On the CIG dataset, regions without associated sentences are ignored, and for regions with multiple sentences, we concatenate their sentences. For metric computation, we consider each region as its individual sample.

For RG, we only provide the image and expect the model to predict the full report, where each image/report pair is considered a single sample. Baseline results are taken from their publications. Note that some baselines have been evaluated on different test splits and that pre- or post-processing can differ as well, leading to limitations in the interpretation of exact results as also acknowledged by [15, 37].

Final results are reported based on samples from the test split. We use bootstrapping with $N = 250$ for the SG, OD, and RC tasks while using $N = 10$ for the RE and RG tasks (due to expansive metric computation).

E.3 Evaluation of Interactive Capabilities

We evaluated the interactive capabilities as studied in Secs. 5.2 and 5.3 on a small subset of the test sets of CIG and MS-CXR. We use only frontal images that have both CIG and MS-CXR targets and exclude samples where the positive classes present in the MS-CXR targets are not present in the CIG targets, *i.e.*, where CIG and MS-CXR targets are not consistent. This results in a total of 98 images.

We only consider the eight pathology classes from MS-CXR, including their associated bounding boxes. If a pathology is present in an image, as defined by MS-CXR targets, we find the associated anatomical regions based on the scene graph from CIG. Additionally, we include all subregions of the associated regions, *e.g.*, subregions of "left lung" include regions like "left upper lung". Given the set of all associated regions and subregions, we further match each bounding box for the given positive pathology with a single region based on the highest IoU. We refer to these regions as *fine regions* and utilize them as *fine hints* in Figs. 4b and 6a of the main paper. To find the coarse associated regions, we use the parent of each fine region, *e.g.*, the parent of "left upper lung" is "left lung". These coarse regions are utilized as *coarse hints* in Figs. 4b and 6a, as *correct hints* in Fig. 5b, and as *anat prompts* in Fig. 6b. For the *opposite hint* and *opposite region* in Fig. 5b, we use the anatomical region on the opposite side of the chest, *i.e.* "left" is replaced by "right".

For Fig. 6b, we apply the CheXbert classifier on predicted sentences and identify the number of positive predicted pathology classes. We refer to the pathology used in the prompt as *queried* and to all other pathologies as *non-queried.*

E.4 Baseline Details

Supervised Visual Grounding Baselines We train the supervised visual grounding baselines on the 878 MS-CXR samples corresponding to the MIMIC-CXR training split. For TransVG [7] we use their official implementation[9]. We additionally study modified versions of TransVG, where we replace its default image backbone (ResNet50 [10]) with backbones pre-trained on chest X-ray images (MIMIC-CXR). Specifically, we use the image encoders from BioVIL [3] and

[9] `https://github.com/djiajunustc/TransVG`

CheXzero [38]. BioVIL provides a pre-trained ResNet50 backbone, such that the default backbone can be initialized with BioVIL weights without requiring further modifications. CheXzero only provides a ViT backbone, such that the whole backbone is completely replaced by the CheXzero backbone. Here, we use the patch tokens of the final transformer layer to replace the patch features of the ResNet.

Chen *et al.* [6] proposed a supervised visual grounding model for the MS-CXR dataset. However, their model shows only marginal improvements above TransVG, and no public source code is provided, such that we did not include it in our studies. Also, they evaluated their model only on samples with a single bounding box per sentence, such that their results are not comparable with our benchmark.

Supervised Object Detection Baselines We train all supervised object detection baselines on the VinDr-CXR dataset. For evaluation on other OD tasks, we map the VinDr-CXR pathology classes to the corresponding classes of the downstream task. With this strategy, we managed to map all classes of MS-CXR and NIH8, except for pneumonia.

For Faster R-CNN [35] we use the torchvision[10] implementation. For DETR [5], Conditional DETR [23], and Deformable DETR [46] we use the implementations from the Huggingface transformers library[11]. For $\mathcal{C}$o-Deformable-DETR [47] we use the implementation provided by the MMDetection library[12]. We again study modified versions of these models, where we replace their default backbones with BioVIL [3] and CheXzero [38], following the same approach as for TransVG. Note that we do not study the CheXzero backbone with advanced DETR-style models (Conditional DETR, Deformable DETR, and Deformable DETR) as the standard DETR model with the CheXzero backbone performs very poorly.

We additionally evaluate the Faster R-CNN models on the region classification (RC) tasks. Here, we modify their ROI pooling layer and replace the predicted box proposals with the bounding boxes provided as input. With this modification, their ROI classification heads can be used to classify the given regions. Again, VinDr-CXR classes are mapped to MS-CXR and CIG classes, where we manage to map 7 of 8 MS-CXR classes and 17 of 53 (including many of the most frequent) CIG classes. Note that DETR-based models cannot be utilized for region classification as they do not have an ROI pooling layer.

Weakly-Supervised Object Detection Baselines We also compare against weakly-supervised object detection (WSupOD) models. Such models use only limited supervision without bounding boxes for pathologies during training but can predict pathology bounding boxes during inference.

We study CheXnet [33], a model commonly used on chest X-rays. We use our own implementation with the DenseNet121 [13] backbone and train it on the 91,883

[10] `https://pytorch.org/vision`

[11] `https://huggingface.co/docs/transformers`

[12] `https://mmdetection.readthedocs.io`

NIH8 training samples using only image-level classification targets. Additionally, we again replace the default backbone with BioVIL [3] and CheXzero [38]. For evaluation on other datasets, we map the NIH8 pathologies to the corresponding classes of the downstream task, where we managed to map all MS-CXR classes, 7 of the 15 VinDr-CXR classes used for evaluation, and 10 of the 53 CIG classes. We also evaluate the CheXnet models on the region classification (RC) tasks by replacing the average-pooling layer before the classifier by ROI pooling with the bounding box provided for the region.
We also study the two models MIL-ADPD [25] and Loc-ADPD [25]. Both use anatomical region bounding boxes and pathology classification targets, where MIL-ADPD uses image-level classification targets while Loc-ADPD uses region-level targets. We use their official implementation[13] and trained it on the 237,917 samples from the CIG training dataset (*i.e.* the CIG samples from the MIMIC-CXR training split). We again replace their default backbone with BioVIL [3] but do not study the use of the CheXzero backbone as it performs very poorly on WSupOD with CheXnet. We again mapped pathology classes from CIG to the target task, where we were able to successfully map all classes.

Contrastive Baselines We study two contrastive models pre-trained on the MIMIC-CXR dataset, namely BioVIL [3] and CheXzero [38]. We use the official implementations and checkpoints for both BioVIL[14] and CheXzero[15], where for CheXzero we use only one of their checkpoints[16] which we retrieved from their provided cloud drive[17].
For sentence grounding, we follow the approach of Boecking *et al.* [3]. We first encode the input sentences using their text encoder and the patch features using their image encoder. Next, we project both into the shared image-text space and compute the cosine similarity between the sentence and each patch. The resulting similarity map is interpolated to the image resolution and then thresholded to generate *masks*, which are then used to compute the mIoU with the target bounding boxes. Additionally, we apply the thresholding procedure from CheXnet [33] on the similarity maps to predict bounding boxes. We compute box scores based on the (average) similarity map values overlapping with the bounding box. The box prediction enables a better comparison with box-based models and allows the computation of both mIoU and mAP scores.
For object detection, we follow a similar approach to predict bounding boxes but use predefined textual prompts for pathologies instead of sentences.
For region classification, we first compute embeddings for textual prompts of each of the classes. We then compute image features for the given region bounding boxes and compute the cosine similarity between the projected image and text features. For the multiclass task on MS-CXR, we apply softmax over the

[13] `https://github.com/philip-mueller/adpd`
[14] `https://github.com/microsoft/hi-ml/tree/main/hi-ml-multimodal`
[15] `https://github.com/rajpurkarlab/CheXzero`
[16] `best_64_5e-05_original_22000_0.864.pt`
[17] `https://drive.google.com/drive/folders/1makFLiEMbSleYltaRxw81aBhEDMpVwno`

similarities of all classes, for the multilabel task on CIG, we use positive and negative prompts and apply softmax over the positive and negative similarities for each class independently. We experiment with two approaches to compute the region image features, ROI pooling and cropping. In the ROI pooling approach, we encode the whole image and then simply pool the patch features overlapping with the given region bounding boxes. For cropping, we feed each region as its individual sample, cropping the image to the given region.
For region explanation, we utilize the image-text retrieval capabilities of the contrastive models. We encode all sentences from the training set independently and store them in a vector database. During inference, we encode each region using the ROI pooling approach from region classification and find the sentence with the highest cosine similarity.

Generative Baselines We evaluate the RGRG [37] model for region explanation using their official implementation and checkpoint[18]. For full report generation on MIMIC-CXR, we report the values from the original publications.

References

1. Bannur, S., Hyland, S., Liu, Q., Pérez-García, F., Ilse, M., Castro, D.C., Boecking, B., Sharma, H., Bouzid, K., Thieme, A., Schwaighofer, A., Wetscherek, M., Lungren, M.P., Nori, A., Alvarez-Valle, J., Oktay, O.: Learning to exploit temporal structure for biomedical vision-language processing. In: CVPR. pp. 15016–15027 (2023). https://doi.org/10.1109/CVPR52729.2023.01442
2. Boecking, B., Usuyama, N., Bannur, S., Coelho de Castro, D., Schwaighofer, A., Hyland, S., Wetscherek, M.T., Naumann, T., Nori, A., Alvarez Valle, J., Poon, H., Oktay, O.: Ms-cxr: Making the most of text semantics to improve biomedical vision-language processing (version 0.1). PhysioNet (2021). https://doi.org/https://doi.org/10.13026/b90j-vb87
3. Boecking, B., Usuyama, N., Bannur, S., Castro, D.C., Schwaighofer, A., Hyland, S., Wetscherek, M., Naumann, T., Nori, A., Alvarez-Valle, J., Poon, H., Oktay, O.: Making the most of text semantics to improve biomedical vision–language processing. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV. pp. 1–21. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-20059-5_1
4. Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A.: Albumentations: Fast and flexible image augmentations. Information **11**(2) (2020). https://doi.org/10.3390/info11020125, https://www.mdpi.com/2078-2489/11/2/125
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) ECCV. pp. 213–229. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
6. Chen, Z., Zhou, Y., Tran, A., Zhao, J., Wan, L., Ooi, G.S.K., Cheng, L.T.E., Thng, C.H., Xu, X., Liu, Y., Fu, H.: Medical phrase grounding with region-phrase

[18] https://github.com/ttanida/rgrg

context contrastive alignment. In: Greenspan, H., Madabhushi, A., Mousavi, P., Salcudean, S., Duncan, J., Syeda-Mahmood, T., Taylor, R. (eds.) MICCAI. pp. 371–381. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-43990-2_35

7. Deng, J., Yang, Z., Chen, T., Zhou, W., Li, H.: Transvg: End-to-end visual grounding with transformers. In: ICCV. pp. 1749–1759. IEEE (2021). https://doi.org/10.1109/ICCV48922.2021.00179, https://doi.org/10.1109/ICCV48922.2021.00179
8. Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., et al.: Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. Circulation [Online] **101**(23), 215—220 (2000)
9. Gu, T., Liu, D., Li, Z., Cai, W.: Complex organ mask guided radiology report generation. In: WACV. pp. 7995–8004 (January 2024)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus) (2023)
12. Hou, W., Xu, K., Cheng, Y., Li, W., Liu, J.: ORGAN: Observation-guided radiology report generation via tree reasoning. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 8108–8122. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-long.451, https://aclanthology.org/2023.acl-long.451
13. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 2261–2269 (2017). https://doi.org/10.1109/CVPR.2017.243
14. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV. pp. 646–661. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_39
15. Hyland, S.L., Bannur, S., Bouzid, K., Castro, D.C., Ranjit, M., Schwaighofer, A., Pérez-García, F., Salvatelli, V., Srivastav, S., Thieme, A., Codella, N., Lungren, M.P., Wetscherek, M.T., Oktay, O., Alvarez-Valle, J.: Maira-1: A specialised large multimodal model for radiology report generation (2023)
16. Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R.L., Shpanskaya, K.S., Seekins, J., Mong, D.A., Halabi, S.S., Sandberg, J.K., Jones, R., Larson, D.B., Langlotz, C.P., Patel, B.N., Lungren, M.P., Ng, A.Y.: Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: AAAI. pp. 590–597 (2019). https://doi.org/10.1609/aaai.v33i01.3301590
17. Jin, H., Che, H., Lin, Y., Chen, H.: Promptmrg: Diagnosis-driven prompts for medical report generation (2024)
18. Johnson, A., Pollard, T., Berkowitz, S., et al.: Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. Sci Data **6**(317) (2019). https://doi.org/https://doi.org/10.1038/s41597-019-0322-0
19. Johnson, A., Pollard, T., Mark, R., Berkowitz, S., Horng, S.: Mimic-cxr database (version 2.0.0). PhysioNet (2019). https://doi.org/https://doi.org/10.13026/C2JT1Q
20. Johnson, A.E.W., Pollard, T.J., Berkowitz, S.J., Greenbaum, N.R., Lungren, M.P., Deng, C.y., Mark, R.G., Horng, S.: Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. arXiv preprint arXiv:1901.07042 (2019)

21. Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., Tang, J.: P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. CoRR **abs/2110.07602** (2021), `https://arxiv.org/abs/2110.07602`
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
23. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional detr for fast training convergence. In: ICCV. pp. 3631–3640 (2021). `https://doi.org/10.1109/ICCV48922.2021.00363`
24. Miura, Y., Zhang, Y., Tsai, E., Langlotz, C., Jurafsky, D.: Improving factual completeness and consistency of image-to-text radiology report generation. In: NAACL. pp. 5288–5304 (2021)
25. Müller, P., Meissen, F., Brandt, J., Kaissis, G., Rueckert, D.: Anatomy-driven pathology detection on chest x-rays. In: Greenspan, H., Madabhushi, A., Mousavi, P., Salcudean, S., Duncan, J., Syeda-Mahmood, T., Taylor, R. (eds.) MICCAI. pp. 57–66. Springer Nature Switzerland, Cham (2023). `https://doi.org/10.1007/978-3-031-43907-0_6`
26. Müller, P., Meissen, F., Kaissis, G., Rueckert, D.: Weakly supervised object detection in chest x-rays with differentiable roi proposal networks and soft roi pooling (2024)
27. Nguyen, H.Q., Pham, H.H., Tuan Linh, L., Dao, M., Khanh, L.: Vindr-cxr: An open dataset of chest x-rays with radiologist annotations (version 1.0.0). PhysioNet (2021). `https://doi.org/https://doi.org/10.13026/3akn-b287`
28. Nguyen, H.Q., Lam, K., Le, L.T., Pham, H.H., Tran, D.Q., Nguyen, D.B., Le, D.D., Pham, C.M., Tong, H.T., Dinh, D.H., et al.: Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations. Scientific Data **9**(1), 429 (2022). `https://doi.org/https://doi.org/10.1038/s41597-022-01498-w`
29. Nicolson, A., Dowling, J., Koopman, B.: Improving chest x-ray report generation by leveraging warm starting. Artificial Intelligence in Medicine **144**, 102633 (2023). `https://doi.org/https://doi.org/10.1016/j.artmed.2023.102633`, `https://www.sciencedirect.com/science/article/pii/S0933365723001471`
30. Pellegrini, C., Özsoy, E., Busam, B., Navab, N., Keicher, M.: Radialog: A large vision-language model for radiology report generation and conversational assistance (2023)
31. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020), `https://nlp.stanford.edu/pubs/qi2020stanza.pdf`
32. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
33. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al.: Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225 (2017). `https://doi.org/10.48550/arXiv.1711.05225`
34. Ramesh, V., Chi, N.A., Rajpurkar, P.: Improving radiology report generation systems by removing hallucinated references to non-existent priors. In: Machine Learning for Health. pp. 456–473. PMLR (2022)
35. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. NIPS **28** (2015)
36. Solovyev, R., Wang, W., Gabruseva, T.: Weighted boxes fusion: Ensembling boxes from different object detection models. Image and Vision Computing **107**, 104117 (2021). `https://doi.org/https://doi.org/10.1016/j.imavis.2021.104117`

37. Tanida, T., Müller, P., Kaissis, G., Rueckert, D.: Interactive and explainable region-guided radiology report generation. In: CVPR. pp. 7433–7442 (2023). https://doi.org/10.1109/CVPR52729.2023.00718
38. Tiu, E., Talius, E., Patel, P., Langlotz, C.P., Ng, A.Y., Rajpurkar, P.: Expert-level detection of pathologies from unannotated chest x-ray images via self-supervised learning. Nature Biomedical Engineering **6**(12), 1399–1406 (2022)
39. Tu, T., Azizi, S., Driess, D., Schaekermann, M., Amin, M., Chang, P.C., Carroll, A., Lau, C., Tanno, R., Ktena, I., Palepu, A., Mustafa, B., Chowdhery, A., Liu, Y., Kornblith, S., Fleet, D., Mansfield, P., Prakash, S., Wong, R., Virmani, S., Semturs, C., Mahdavi, S.S., Green, B., Dominowska, E., y Arcas, B.A., Barral, J., Webster, D., Corrado, G.S., Matias, Y., Singhal, K., Florence, P., Karthikesalingam, A., Natarajan, V.: Towards generalist biomedical ai. NEJM AI **1**(3), AIoa2300138 (2024). https://doi.org/10.1056/AIoa2300138
40. Wang, L., Ning, M., Lu, D., Wei, D., Zheng, Y., Chen, J.: An inclusive task-aware framework for radiology report generation. In: MICCAI. pp. 568–577 (2022)
41. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: CVPR. pp. 2097–2106 (2017). https://doi.org/10.1109/CVPR.2017.369
42. Wang, Z., Liu, L., Wang, L., Zhou, L.: Metransformer: Radiology report generation by transformer with multiple learnable expert tokens. In: CVPR. pp. 11558–11567 (2023). https://doi.org/10.1109/CVPR52729.2023.01112
43. Wu, J., Agu, N., Lourentzou, I., Sharma, A., Paguio, J.A., Yao, J.S., Dee, E.C., Kashyap, S., Giovannini, A., Celi, L.A., et al.: Chest imagenome dataset for clinical reasoning. In: NIPS (2021)
44. Wu, J.T., Agu, N.N., Lourentzou, I., Sharma, A., Paguio, J.A., Yao, J.S., Dee, E.C., Mitchell, W., Kashyap, S., Giovannini, A., et al.: Chest imagenome dataset (version 1.0.0). PhysioNet (2021). https://doi.org/https://doi.org/10.13026/wv01-y230
45. Xu, L., Ni, Z., Liu, X., Wang, X., Li, H., Zhang, S.: Learning a multi-task transformer via unified and customized instruction tuning for chest radiograph interpretation (2023)
46. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: ICLR (2021), https://openreview.net/forum?id=gZ9hCDWe6ke
47. Zong, Z., Song, G., Liu, Y.: Detrs with collaborative hybrid assignments training. In: ICCV. pp. 6725–6735 (2023). https://doi.org/10.1109/ICCV51070.2023.00621