DR. DINO: SHIFTING OPEN-SET OBJECT DETECTION TO FINE GRAIN MEDICAL DOMAINS

FINAL REPORT FOR CS 152 L3D AT TUFTS IN FALL '24

Matthew Cook **Marco Pretell** **Shane Williams**

Sharing permission statement: **Instructors can share this report to future students of similar classes**

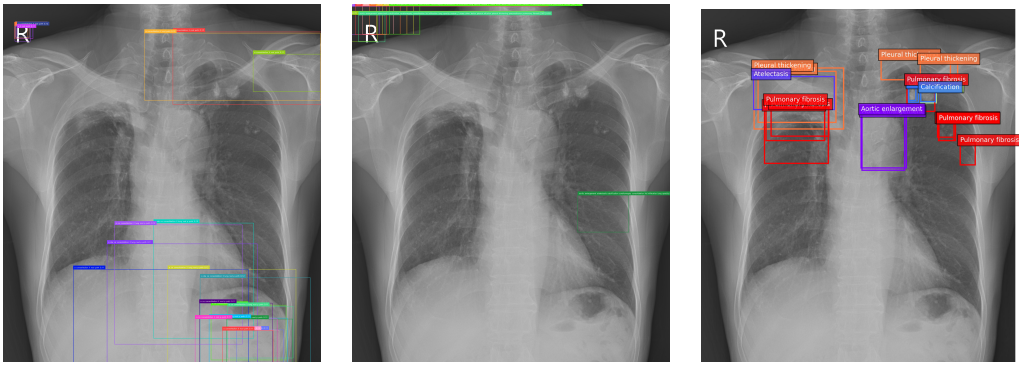

Figure 1: mBGD-FT (Left), BGD-FT (Middle), and Ground Truth (Right) predictions on Out-of-Distribution Chest X-rays. MedBERT identifies text, BERT seems to identify regions of interest. Neither predict the actual classes belonging to the Chest X-ray dataset

1 INTRODUCTION

The momentum of machine learning research the last few years has been carried by the vast swaths of data that are currently available in the digital era. As the amount of data grows ever larger, machine learning practitioners continue to implement larger models that perform better on incredibly varied tasks. However, for many fields, the collection of data is impractical, costly, and, sometimes, simply impossible. Medical imaging is a paragon of these difficult data-collection tasks. Between the limitations imposed by privacy laws and a need for experts to properly label data, establishing large enough datasets to achieve reasonable accuracy with standard supervised machine learning is a difficult and expensive endeavor. Typically, these datasets tend to be on the scale of hundreds to thousands of images as opposed to millions like many other image-based datasets (Litjens et al., 2017). Due to these many limitations, advancements in machine learning for medical purposes typically need to work well on small amounts of data. We explore a potential methodology that can help to address the lack of annotated medical images. If given an unrelated labeled dataset of a certain medical imaging modality (i.e. MRI, CAT-scan, X-ray, etc.) our methodology hopes to create a model that can provide zero-shot object detection for relevant features of interest within an unlabeled dataset of that same modality.

Overall, our method seeks to improve downstream accuracy in a transfer learning task by fine-tuning an existing open-set object detector, GroundingDINO(Liu et al., 2024). We hypothesize that GroundingDINO's ability to generalize provides a good starting point to fine-tune for more specialized image detection tasks. We seek to demonstrate that we can utilize this pre-trained open-set detector to perform improved zero-shot learning for medical images. This will be done by leveraging an additional unrelated large labeled dataset of the same imaging modality as our target task. In an ideal world, this methodology would allow for the training of task-specific models, thus enabling the automatic creation of medical imaging datasets without the need for the many man-hours required to annotate data. However, much more realistically, our proposed models and methodology could prove a useful tool reducing the cost to create and curate larger-scale datasets within the medical domain.

We will use the methods outlined by GroundingDINO to fine-tune the pre-trained model, specifically using open-source code provided by openDINO from this github repository. The text backbone, which provides the feature embeddings, will be kept frozen over the course of the experiments. We will use the BERT model(Devlin et al., 2018), and its same-architecture twin MedBERT, to provide textual feature embeddings, examining their impact on accuracy. The original BERT is trained on English Wikipedia and the Brown Cor-

pus, and MedBERT is fine-tuned specifically on electronic health records. The motivation to use MedBERT is that its more domain-specific training will, in theory, provide better feature embedding for medical text. For the image backbone, we will be following the GroundingDINO paper and use Dino(Caron et al., 2021). During training, we will use the pre-trained weights of the model originally presented in GroundingDINO as the initial values for the attention and query mechanisms. This decision is in part to save on the cost of training a model from scratch, but mainly with the goal of transferring some amount of generalized open-set knowledge that the pre-trained model has learned.

Once training has been completed, our project will consist of four models: GroundingDINO with a BERT backbone without fine-tuning (BGD), GroundingDINO with a BERT backbone and fine-tuning (BGD-FT), GroundingDINO with a MedBERT backbone without fine-tuning (mBGD), and finally GroundingDINO with a MedBERT backbone with fine-tuning (mBGD-FT). Additionally, our two chosen datasets each provide an opportunity to evaluate the accuracy of our models. Our primary point of interest revolves around the zero-shot accuracy on the VinDr-CXR dataset(Nguyen et al., 2020b), which consists of chest X-rays. However, we can achieve additional insights into our models' performances by computing accuracy measurements for our fine-tuning Pediatric Wrist Dataset (Nagy et al., 2020). We expect the accuracy of the networks to be ordered from best to worst in the following order: mBGD-FT, BGD-FT, BGD, and mBGD. We can safely hypothesize that mBGD-FT will fare the best as it receives the most in-distribution data and will have been trained on the X-ray images and contain a pre-trained MedBERT which has language sampled from a medical domain. Likewise, we assume BGD-FT will perform better than vanilla BGD as the vanilla model will have received no in-distribution data. Though mBGD will receive in-distribution textual data through the MedBERT backbone, because the full network will not yet have been calibrated to its embeddings, we hypothesize that the network will fail to integrate the embeddings properly, and, consequently, hurt performance. Because there isn't a comparable dataset fine-tuned or zero-shot metric from the original GroundingDINO paper, we intend to evaluate our model's metrics as a measure of improvement from the original BGD baseline with the original BERT backbone without fine-tuning.

2 METHODS

The models investigated in this study are based on the GroundingDINO architecture. For a more detailed description please refer to the original paper (Liu et al., 2024), however, a brief synopsis is provided to add context to our methodology. GroundingDINO is built upon two backbone transformer models, a textual-encoder, BERT, and an image-encoder, DINO. These embeddings are then processed via the 'Feature Enhancer' and 'Decoder' layers using cross-attention to create semantically significant object-text pairs that result in the final labeled bounding boxes. This paper implements two distinct approaches to improve the alignment of textual and visual features for medical image object detection tasks. Specifically, we investigate: (1) Textual Embedding Backbone Swapping, and (2) Model Fine-Tuning on related medical imaging modality datasets. We will apply each approach both individually as well as in combination.

In our first approach, we replace the default LLM, BERT, with a domain-specific LLM, MedBERT, that is better attuned to the specialized medical terminology such as that used for X-ray imaging. For instance, the textual difference between "hairline fracture" and "compound fracture" is subtle, but incredibly important for accurate image-text alignment. We expect a medical-specific textual encoder to better capture the fine-grained semantic details more effectively and create a wider diversity of embeddings than the corresponding vanilla BERT. The improved embeddings would have positive downstream benefits for the learned cross-attention between textual and visual features within the model.

The second approach involves fine-tuning the entire model (with the exception of the textual encoder) on an entirely separate dataset that is of the same medical imaging modality (aka X-rays, MRI, etc). The baseline model, GroundingDINO, was originally trained on large-scale, natural image object detection datasets (e.g., COCO, OdinW), that lack the more subtle visual features typical of medical images like X-rays. By performing an additional round of training on a dataset within the same imaging modality, the model is exposed to consistent textures, contrast patterns, and structures to hopefully learn important information that can be expanded to all datasets of this modality. This second-stage fine-tuning is a form of transfer learning that aims to improve the model's ability in a fine-grained medical image domain.

To facilitate additional training/customization, we leverage the publicly available Open-GroundingDINO GitHub repository, which provides a flexible framework for controlling hyperparameters, specifying layer freezing, and incorporating new training datasets easily. While we adopt Open-GroundingDINO as a starting point, we introduce several targeted modifications to the codebase. These include substituting the default language model with MedBERT to generate domain-specific textual embeddings, and performing hyperparameter tuning to optimize performance for the medical imaging domain.

2.1 TEXTUAL EMBEDDING BACKBONE SWAPPING

A key aspect of the textual embedding backbone swapping methodology is that it does not require any additional training of the parent GroundingDINO model. It is also important to note that this approach may not yield significant improvements when used in isolation. We suspect this because MedBERT-generated embeddings may not align well with the rest of the architecture prior to fine-tuning, this is due to the rest of the model architecture being trained with a vanilla BERT-based encoder and thus will not be familiar with the MedBERT embeddings. Although MedBERT embeddings may be semantically richer, their direct substitution without further adjustments is not expected to enhance performance on its own. The modular design of the original GroundingDINO architecture allows for a relatively straightforward substitution of the textual encoder with other BERT-like models. The main considerations are adjusting tokenization and vocabulary sizes to accommodate the new language model's vocabulary.

The original BERT model leveraged in both GroundingDINO as well as our methodology, was trained in an unsupervised manner using Wikipedia and the BooksCorpus, employing a masked language modeling (MLM) and next sentence prediction (NSP) objective. With the release of the BERT architecture by Google in 2018 and its subsequent popularity, a wide range of fine-tuned BERT-based models have become readily accessible through repositories like Hugging Face. These models can be quickly integrated into existing pipelines using libraries such as the Transformers framework's AutoModel and AutoTokenizer classes.

For our medically informed LLM, we selected MedBERT (https://huggingface.co/Charangan/MedBERT), a BERT-based model pre-trained on multiple biomedical corpora, including N2C2 clinical notes, BioNLP articles, CRAFT (67 open-access biomedical journal articles from PubMed), and medically related Wikipedia articles. Although MedBERT is built upon BERT, it is not directly classified as a "BERT" model within the Transformers library. Thus, minor modifications were required to correctly instantiate and integrate MedBERT into the GroundingDINO pipeline. These steps involved verifying tokenizer compatibility, adjusting configuration parameters for the vocabulary and embedding layers, and ensuring that the encoder outputs adhered to the expected dimensions and formats for downstream image-text alignment.

2.2 FINE-TUNING ON RELATED MEDICAL IMAGING MODALITY DATASETS

At the most abstract level, this entire paper's methodology can be summarized as transfer learning with fine-tuning. The core assumption of this paper's methodology is that introducing familiarity with the relevant medical imaging modality (in this case, X-rays) will improve model performance. For our implementation, we perform fine-tuning across the Image Backbone, specifically the DINO Transformer model, as well as the Feature Enhancer and Decoder layers within the GroundingDINO framework. Future directions for this work will explore hypotheses about which layers should remain frozen during training, an approach that may further optimize computational efficiency and generalization.

Our implementation utilizes the AdamW optimizer, as per the precedent set by the GroundingDINO paper, chosen for its ability to prevent overfitting. With AdamW, weight decay is employed in lieu of traditional L2 regularization to prevent overfitting of the training data.

The loss function used in our training also follows the original GroundingDINO implementation and is expressed as a weighted summation of three primary components. The first component, the contrastive loss, is defined as:

$$\text{Contrastive Loss} = -\alpha_t(1 - p_t)^\gamma \log(p_t), \tag{1}$$

This loss promotes accurate textual labels for predictions, prioritizing harder-to-identify cases over those that are more easily classified. This formulation reflects the underlying intuition that harder examples contribute more to model refinement, pushing the model to improve its overall robustness.

For bounding box prediction, two loss functions are employed to ensure both global alignment and finer accuracy of the bounding box location: the L1 loss and the Generalized Intersection over Union (GIoU) loss. The L1 loss minimizes the absolute error between predicted and ground-truth bounding box coordinates, while the GIoU loss accounts for the case where there is no overlap and provides a stronger spatial relationship between the predicted and ground-truth boxes. The GIoU loss, as defined by (Rezatofighi et al., 2019), is formally described in the paper as follows:

[1] Two arbitrary convex shapes $A, B \subseteq S \in \mathbb{R}^n$ GIoU Find the smallest enclosing convex object C, where $C \subseteq S \in \mathbb{R}^n$ Compute the Intersection over Union (IoU):

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$$

Compute the Generalized Intersection over Union (GIoU):

$$\text{GIoU} = \text{IoU} - \frac{|C \setminus (A \cup B)|}{|C|}$$

The final loss function is a weighted summation of the contrastive loss, L1 loss, and GIoU loss, reflecting their combined contributions to the training process. Specifically, the total loss is defined as:

$$\text{Final Loss} = 1.0 \cdot \text{Contrastive Loss} + 5.0 \cdot \text{L1 Loss} + 2.0 \cdot \text{GIoU Loss} \tag{2}$$

3 Data and Experimental Plan

3.1 Data description

During our experiments, we plan to use two datasets to perform training and evaluation. The first dataset, Pediatric Wrist Trauma(Nagy et al., 2020), will be used to fine-tune our initial model. The second data set is the Chest X-ray Abnormalities Detection(Nguyen et al., 2020a) data set from Vin Big Data Institute (VinBDI) also referred to as VinDr-CXR. This data will be used to assess the zero-shot capabilities of our model. Both datasets are X-ray images where VinDr-CXR consists of X-rays of lungs while Pediatric Wrist Trauma consists of X-rays of wrist bones. This overlap of imaging modalities was done intentionally in order to evaluate our methodology more effectively. We ensured that the data sets were images of different body parts to ward against data leakage and better test our hypothesis.

The Pediatric Wrist Trauma data is a collection of wrist X-rays from pediatric patients classifying various injuries and abnormalities. Each image is drawn with bounding boxes, polygons, and lines, labeled across 10 different classes. The only class that exists for lines is an axis class to designate the axis of the wrists. Polygons are used to distinguish bone lesions and periosteal reactions. Every other class label uses a bounding box. We chose the Pediatric Wrist Data as our training set due to its large size in comparison to the Chest X-ray Abnormalities, thus simulating the environment in which our method would in theory be used, as well as the fact that they also provide the patient that each image pertains to. Thus we can use this information during our data splitting to stop any potential data leakage.

The Chest X-ray Abnormalities data is a collection of chest X-rays that have been hand-classified by 17 radiologists from Hospital 108 and the Hanoi Medical University Hospital in Vietnam. In order to maintain privacy, patient information is not included in the data. This data has been thoroughly reviewed with each x-ray labeled and validated by a minimum of three separate radiologists. However, the methodology of data collection was not explicit about patient overlap.

3.2 Performance metric

We will evaluate our final models using both mean Average Precision (AP or mAP) and mean Average Recall (AR or mAR), following the methodology established by the foundational work for our research, Grounding DINO (Liu et al., 2024). AP and AR have become popular metrics for object detection in recent years not in small part due to their inclusion as the main metric of performance for both PASCAL-VOC and COCO competitions' standard evaluations. AP and AR, while originally conceived for binary classification tasks, can be used across a variety of prediction-oriented models. When used in reference to object detection, we define positive and negative predictions as a function of specific thresholds of Intersection over Union (IoU) of a predicted bounding box. The IoU is a measurement of bounding box overlap between prediction and ground truth where at the provided threshold of overlap it considers the prediction positive and anything less negative. Thus, AP is calculated as the area under the Precision-Recall (PR) curve averaged across all classes, where precision is defined as the ratio of true positive detections to all positive detections and recall as the ratio of true positives to the total number of actual objects. We can articulate both Precision and Recall as follows as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}} \qquad \text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} \tag{3}$$

The original Grounding DINO paper does not explicitly state the IoU threshold they used to determine what classifies as positive. However, within the supplementary materials, their AP calculations follow the COCO Dataset standards (COCO Detection Evaluation) for AP scores. The main metric of accuracy COCO uses is the mean Average Precision over IoU thresholds between 0.5 and 0.95 IoU score with a step size of 0.05 for computing the AP over that threshold range. We will be implementing the same mean AP for our models, as well as the additional metrics COCO outlines such as AP Across Scales and Average Recall to assess all our

models' performances as outlined by the COCO team. For additional information regarding the computation of other AP metric variations, see (COCO Detection Evaluation).

AP is widely accepted in the object detection space. As evidenced by the analysis on the metric by (Padilla et al., 2020) which highlights its reliability and provides descriptions of multiple variations that could be implemented in future extensions of this paper. By leveraging multiple AP and AR metrics we aim to create a robust understanding of the performance of our zero-shot object detection across a multitude of confidence levels, while also retaining a metric that allows for comparison to the performance of the original Grounding DINO to gain additional insights on the productivity of our domain transfer task.

3.3 Splitting Data to train/tune/assess generalization.

Our implementation avoids significant data leakage by the nature of zero-shot prediction, but we still put forth effort to avoid leakage when possible. Our chosen problem utilizes two entirely unique datasets that have no overlapping examples and thus we can say with a high degree of confidence that there is no leakage between our pre-training task and our testing task. Importantly, we assume that the two datasets (VinDr-CXR and Pediatric Wrist Fracture) of X-rays have no overlapping patients. This case would create data leakage, but, as they are from not only different hospitals, but different countries and are images of different areas of the body, we feel confident in concluding there is little chance of cross-patient data leakage.

As mentioned prior, the VinDr-CXR (Nguyen et al., 2020a) dataset did not explicitly state how it handled patient data when splitting between the provided training and test, nor does it have de-identified patient IDs, so we cannot perform this split ourselves like we are able to with the (Nagy et al., 2020) dataset. Thus, to avoid any data leakage caused by patient splitting amongst train, validation, and test sets, we shall implement this dataset as our zero-shot testing case. We implement the Pediatric Wrist Fracture dataset (Nagy et al., 2020) as our pre-training dataset and create a proper patient split of the data to ensure that patients fall only within train, validation, or testing splits. We strive to create a training, validation, and testing split of 80/10/10 on the pediatric wrist data. A key part of dividing our data was to ensure no patient had multiple occurrences across splits due to a single patient having multiple occurrences in the dataset. Thus we ensure that if a patient is represented in the one split, it will have no representations in the other two splits. This does result in us not being able to achieve a perfect 80/10/10 split. The number of each instance across the different training splits is shown in Table 3. The pre-split distribution of data is shown in Table 7. A note is the large class imbalance present. Ignoring the text label, which is mostly not diagnostically relevant, a majority of the images represent fractures. On the other hand, only a handful of examples exist for the Foreign Body Class. We attempted to keep this same class imbalance present in our training validation and testing split. This splitting resulted in our training, validation, and testing having 37,028, 4,826, and 4,801 total annotations respectively. The only modification made from the VinDr-CXR dataset was the removal of a "no finding" class which was present in the original dataset. Thus every image in the testing data set will have bounding boxes.

3.4 Training and hyperparameter tuning plan

Now, we will discuss our hyperparameter selection/tuning. As previously mentioned, the foundational model for our research, GroundingDINO has an open-source GitHub repository. The training pipeline that we use across all of our experiments uses ADAMw as an optimizer. The repository also contains a helpful config file with all hyperparameters that can be tuned (to review the full list of hyperparameters our team selected from look to open-Grounding DINO config file). Given that the scope of our project already incorporates two different textual encoding models as well as full fine-tuning on the two different versions of GroundingDINO, we had to be economical about how we did our hyperparameter search. We searched over learning rate alongside two generalization terms, weight decay, and fusion droppath, with fusion droppath being a multimodal adaptation of droppath (Larsson et al., 2016). Similar to traditional dropout, droppath is a generalization term that introduces a chance to drop out whole sections and "paths" of the model instead of singular nodes. To search over these hyperparameters efficiently, we performed a randomized grid search (Bergstra & Bengio, 2012). Using the values established in our foundational work (Liu et al., 2024) as a guideline, we randomly searched values between the ranges presented in Table 1. Early stopping was not implemented, but the best model determined by validation mAP was updated at every epoch and returned after training. Each model was run for a total of 50 epochs which took around 12 hours each across 5 A100 GPUs. Overall for our two methods which required training(BGD-FT and mBGD-FT) 6 models were trained, 3 models each, resulting in a total time of 72 computational hours.

3.5 Relevant Data Statistics

The Pediatric Wrist Dataset consists of images taken from patients ranging between 0.2 to 19 years old with a mean age of 10.9 years. While only covering 6091 patients, there are a total of 20327 images in the data set. As such, multiple images can be representative of the same patient. In the extreme case, there is one patient with a total of over 30 images attributed to them, although most patients had far fewer. The mean patient has 3.34

Hyperparameter Searched	Range of potential random values
Learning Rate	0.1×10^{-3} - 0.5×10^{-2}
Weight Decay	0.1×10^{-3} - 0.8×10^{-2}
Fusion Droppath	0.1 - 0.5

Table 1: Range of hyperparameters for our randomized grid search

Class	Training	Validation	Testing	Total
Bone Anomaly	203	36	37	276
Bone Lesion	39	4	2	45
Fracture	14365	1866	1859	18090
Periosteal Reaction	2759	348	346	3453
Pronator Sign	450	61	56	567
Soft Tissue	356	53	55	464
Text	18852	2426	2444	23722

Table 2: Distribution of Classes in Pediatric Wrist Dataset across training, validation, and testing (Nagy et al., 2020)

Class Label	Number of Instances
Aortic enlargement	7162 (19.85%)
Atelectasis	279 (0.77%)
Calcification	960 (2.66%)
Cardiomegaly	5427 (15.03%)
Consolidation	556 (1.54%)
ILD	1000 (2.77%)
Infiltration	1247 (3.45%)
Lung Opacity	2483 (6.88%)
Nodule/Mass	2580 (7.15%)
Other lesion	2203 (6.10%)
Pleural effusion	2476 (6.86%)
Pleural thickening	4842 (13.41%)
Pneumothorax	226 (0.63%)
Pulmonary fibrosis	4655 (12.90%)

Table 3: Distribution of Classes in the Chest X-ray Abnormalities Dataset used for Testing(Nguyen et al., 2020a)

corresponding images while the median per patient is 2. The maximum size of a given image is 1860×1664, whereas the smallest image in the dataset is 364×1664. These sizes are not indicative of the maximum height and width across all images, but rather the images with the largest area. The minimum height across all images is 324 pixels, while the minimum width is 212. For maximum height and width, the values are 2328, and 1664 respectively.

The Chest X-ray Abnormality dataset consists of 36,096 bounding boxes across 4,394 images. The maximum size of a given image is 3408×3320, whereas the smallest image in the dataset is only 927×823. These are also coincidentally the largest and smallest height and width across the dataset respectively. Each bounding box has one of 14 class labels. The distribution of each class across the data can be seen in Table 3.

4 Results

The mAP scores for our various methods featuring the BERT and MedBERT backbone are shown below in tables 4, 5, as well accuracy metrics across epochs during fine-tuning are shown in figure 2. Finally, figure 3 shows mAP and mAR across epochs for a variety of different hyperparameters during fine-tuning with the MedBERT backbone. Tables 9 and 8 containing mAR metrics can be found in the supplementary materials of this paper. Analyzing our metrics, we came to the following conclusions:

BERT vs. MedBERT: Simply swapping out the text backbone performs as expected. As the weights of the attention and query mechanisms of the original model are tuned to work with BERT embeddings mBGD performs slightly worse than BGD with an mAP of 0.0.

Methodology	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
BGD	3.111×10^{-7}	1.739×10^{-6}	1.709×10^{-8}	0	9.822×10^{-7}	4.157×10^{-7}
mBGD	0	0	0	0	0	0
BGD-FT	8.031×10^{-6}	3.720×10^{-5}	1.616×10^{-6}	0	1.624×10^{-5}	6.981×10^{-6}
mBGD-FT	$\mathbf{2.325 \times 10^{-2}}$	$\mathbf{4.794 \times 10^{-2}}$	$\mathbf{1.950 \times 10^{-2}}$	$\mathbf{1.747 \times 10^{-1}}$	$\mathbf{1.988 \times 10^{-2}}$	$\mathbf{8.118 \times 10^{-2}}$

Table 4: mAP metrics on Pediatric Wrist Trauma

Methodology	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
BGD	8.456×10^{-8}	2.885×10^{-7}	4.083×10^{-8}	0	0	8.498×10^{-8}
mBGD	3.501×10^{-9}	1.398×10^{-8}	0	0	0	3.516×10^{-9}
BGD-FT	$\mathbf{2.767 \times 10^{-6}}$	$\mathbf{1.615 \times 10^{-5}}$	$\mathbf{1.510 \times 10^{-7}}$	0	0	$\mathbf{2.767 \times 10^{-6}}$
mBGD-FT	4.006×10^{-7}	2.212×10^{-6}	1.267×10^{-8}	0	0	4.129×10^{-7}

Table 5: mAP metrics on VinDr-CXR

Both table 4 and figure 2 show that mBGD-FT continuously outperforms BGD-FT during the training process. Tables 4 and 9 show the final mAP and mAR are orders of magnitude better across the board for mBGD-FT in comparison to BGD for in-distribution data with an mAP of 2.325×10^{-2} compared to 8.031×10^{-6} and AR_{100} of 1.215×10^{-1} compared to 2.324×10^{-3} for mBGD-FT and BGD-FT respectively. This supports the initial premise that MedBERT provides more diagnostically rich textual embeddings which the groundingDINO architecture can utilize. For out-of-distribution data, using MedBERT as the text backbones appears to have no discernible impact, with BGD and mBGD both being essentially zero. A possible cause of mBGD-FT's poor performance in the zero-shot case is discussed later in this paper.

Fine Tuning for OOD Data: Table 5 and 8 do not appear to support our initial hypothesis. The general dataset on which groundingDINO was initially trained did not provide enough relevant weights to perform the fine-grained object detection required in X-ray diagnostics as seen in the BGD and mBGD performance. By fine-tuning on the pediatric wrist data, we hoped to see performance gains as the model learned universally shared X-ray features. This does not appear to be the case. Using BERT over medBERT as a backbone saw only minor improvements in OOD data, although, with mAPs of 2.767×10^{-6} and 4.006×10^{-7} for mBGD-FT and BGD-FT respectively, both were essentially zero.

Table 6 provides a possible explanation as to why this is. It shows the mAP score for the fine-tuning data by class and demonstrates at a higher level what the mBGD-FT is actually learning during training. Most of the mAP score for in-distribution data comes from its ability to accurately recognize and classify text, while other classes suffer tremendously in performance. This is mostly likely due to the prevalence and clarity of text in the X-ray images. Almost all of the images in our training data have text denoting left or right and are often located within the same region of the image. Furthermore, given these letters' sharp contrast and simple shapes in comparison to the other classes, they are much easier to detect. Thus, the model can gain deceptively higher accuracy by simply learning to predict the text class. While mBGD-FT has the appearance that it is learning on the fine-tuning, it appears that it mostly only learning text.

Figure 3 shows the performance of medBERT-FT during training across various hyperparameter values. There is no indication of divergence, seemingly indicating that a larger learning rate tested could be useful. Regarding both mAP and mAR, it would appear lower values of weight decay and fusion droppath yield better models. Both the red and green runs in figure 3 appear hindered by too-extreme weight decay and fusion droppath, likely because they were not afforded the ability to learn. This would support the idea that the groundingDINO model architecture does not need much generalization to prevent over-fitting.

Class Labels	mAP[0.5:0.95]
boneanomaly	0.000
bonelesion	0.000
fracture	0.002
metal	0.000
periostealreaction	0.000
pronatorsign	0.002
softtissue	0.000
text	0.181

Table 6: mAP [0.5:0.95] for MedBERT-FT on Pediatric Wrist (in distribution) dataset. Highlights the strong correlation between mAP score and the correct identification of text classification

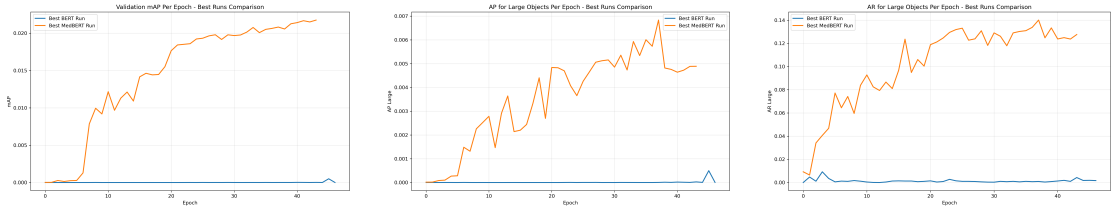

Figure 2: 3 different accuracy metrics for BERT-FT and MedBERT-FT plotted over epochs. MedBERT-FT consistently outperforms it's BERT counterpart

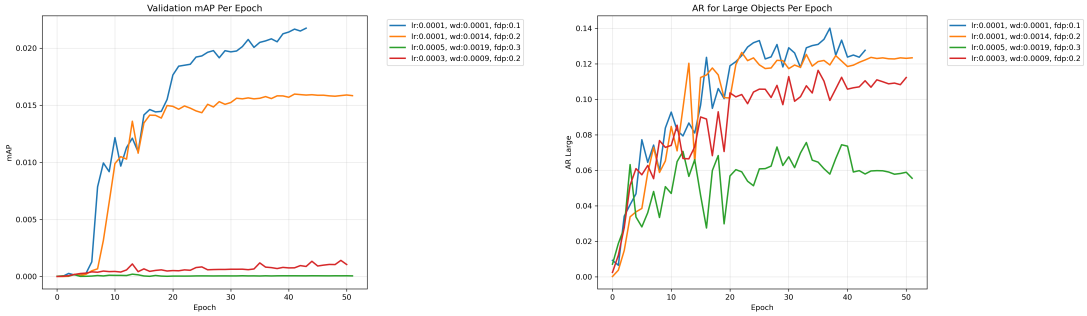

Figure 3: Validation metrics during training for medBERT-FT across various hyperparameter settings during randomized grid search

5 Discussion, Reflection, and Outlook

The results of our experiment highlight a clear difference in performance between in-distribution and out-of-distribution testing. The mAP score for in-distribution data showcased decent performance with mBGD-FT. The original BGD architecture that was our baseline scored nearly zero on all AP metrics on both testing datasets, whereas our mBGD-FT on in-distribution data ended up being comparable to much of the performance seen in the original GroundingDINO paper (see the supplementary table 12 in (Liu et al., 2024)). However, we did not see these fine-tuning accuracy gains transferred to our zero-shot test case where we see minimal to no improvements from BGD to mBGD-FT for AP metrics and only marginal gains for AR. This raises an important question: is the massive computational cost required for these models worth it? For example, GroundingDINO achieved its results using up to 64 A100 GPUs, while our experiments were conducted with at most 5 GPUs over 12 hours. It remains unclear whether the observed performance gaps are due to limitations of the methodology or simply a lack of computational resources. With equal training time and hardware, it is possible we could achieve results comparable to GroundingDINO, but this remains speculative.

Beyond computational limitations, we identified several issues in our current approach. First, the inclusion of the text class in the pediatric wrist dataset appears to undermine performance, as highlighted in Table 6. The model appears not to be learning semantic information regarding X-ray imaging but instead simply has optimized to be able to identify text classes. Addressing this by removing the text class is a clear next step. In combination with this approach, we could also scale up the contrastive-losses α hyperparameter that controls how severely we punish confident classifications to attempt to force a greater semantic understanding of X-ray images. Secondly, time and resource constraints prevented us from performing a full hyperparameter search. Lastly, due to the lack of grounding information present in our chosen data sets, we were unable to use groundingDINO to its full potential. Such textual labels have been shown to yield improved mAP scores (Li et al., 2021). Future work could then search out or create, using LLMs, grounding information data in the relevant medical domain.

We took many takeaways from developing this project. Working with large models and their codebases can be unwieldy. Even if you theoretically have all the code you need, augmenting it to fit your specific use case and then building upon it can prove time-consuming and frustrating. Dependencies can seem like endless cycles of download, install, purge, and repeat. The time it takes large models to do simple tasks like validating results can be slow. But it's important to have patience and to take breaks and with a little time and perspective you'll get there in the end.

REFERENCES

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021. https://arxiv.org/abs/2104.14294.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. http://arxiv.org/abs/1810.04805.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *CoRR*, abs/1605.07648, 2016. http://arxiv.org/abs/1605.07648.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. *CoRR*, abs/2112.03857, 2021. https://arxiv.org/abs/2112.03857.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2017.07.005. https://www.sciencedirect.com/science/article/pii/S1361841517301135.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *International Conference on Learning Representations (ICLR)*, 2024. https://arxiv.org/pdf/2303.05499.

Eszter Nagy, Michael Janisch, Franko Hržić, and Sebastian Tschauner Erich Sorantin. A pediatric wrist trauma x-ray dataset (grazpedwri-dx) for machine learning. *Sci Data.*, 2020.

Duc Nguyen, DungNB, Ha Q. Nguyen, Julia Elliott, NguyenThanhNhan, and Phil Culliton. Vinbigdata chest x-ray abnormalities detection. https://kaggle.com/competitions/vinbigdata-chest-xray-abnormalities-detection, 2020a. Kaggle.

Ha Q. Nguyen, Khanh Lam, Linh T. Le, Hieu H. Pham, Dat Q. Tran, Dung B. Nguyen, Dung D. Le, Chi M. Pham, Hang T. T. Tong, Diep H. Dinh, Cuong D. Do, Luu T. Doan, Cuong N. Nguyen, Binh T. Nguyen, Que V. Nguyen, Au D. Hoang, Hien N. Phan, Anh T. Nguyen, Phuong H. Ho, Dat T. Ngo, Nghia T. Nguyen, Nhan T. Nguyen, Minh Dao, and Van Vu. Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations, 2020b.

Rafael Padilla, Sergio L. Netto, and Eduardo A. B. da Silva. A survey on performance metrics for object-detection algorithms. pp. 237–242, 2020. doi: 10.1109/IWSSIP48289.2020.9145130.

Seyed Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and A loss for bounding box regression. *CoRR*, abs/1902.09630, 2019. http://arxiv.org/abs/1902.09630.

6 APPENDIX

Class Label	Total Occurences	Total Occurences in Patients	Percentage of Patients
Text	23722 (35.00%)	6089	99.97%
Axis	20327 (29.99%)	6091	100.00%
Fracture	18090 (26.69%)	3587	58.89%
Periosteal Reaction	3453 (5.10%)	1089	17.88%
Metal	819 (1.21%)	157	2.58%
Pronator Sign	567 (0.84%)	555	9.11%
Soft Tissue	464 (0.68%)	405	6.65%
Bone Anomaly	276 (0.41%)	95	1.60%
Bone Lesion	45 (0.07%)	24	0.39%
Foreign Body	8 (0.01%)	4	0.07%

Table 7: Pre-split data distribution for our training data set. We have subsequently removed 'Foreign Body' as a potential class due to its incredibly low representation in the dataset

Methodology	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
BGD	7.225×10^{-5}	2.885×10^{-7}	4.083×10^{-8}	0	0	8.498×10^{-8}
mBGD	2.885×10^{-6}	2.885×10^{-6}	4.327×10^{-5}	0	0	4.502×10^{-5}
BGD-FT	4.380×10^{-5}	6.386×10^{-4}	7.126×10^{-4}	0	0	7.471×10^{-4}
mBGD-FT	$\mathbf{1.128 \times 10^{-4}}$	$\mathbf{6.743 \times 10^{-4}}$	$\mathbf{3.015 \times 10^{-3}}$	0	0	$\mathbf{3.045 \times 10^{-3}}$

Table 8: mAR metrics on VinDr-CXR

Methodology	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
BGD	7.225×10^{-5}	1.218×10^{-4}	8.444×10^{-4}	0	8.170×10^{-5}	2.066×10^{-3}
mBGD	0	0	0	0	0	0
BGD-FT	2.644×10^{-4}	7.880×10^{-4}	2.324×10^{-3}	0	3.962×10^{-3}	1.259×10^{-3}
mBGD-FT	$\mathbf{4.155 \times 10^{-2}}$	$\mathbf{7.928 \times 10^{-2}}$	$\mathbf{1.215 \times 10^{-1}}$	$\mathbf{2.461 \times 10^{-1}}$	$\mathbf{9.203 \times 10^{-2}}$	$\mathbf{1.322 \times 10^{-1}}$

Table 9: mAR metrics on Pediatric Wrist Trauma

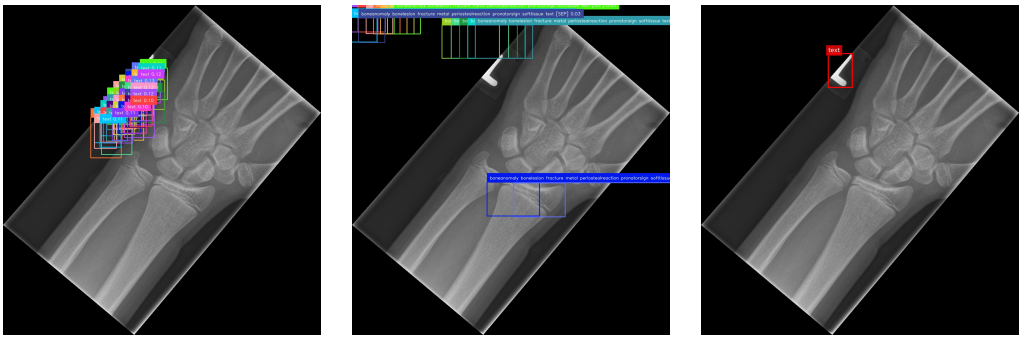

Figure 4: mBGD-FT (Left), BGD-FT (Middle), and Ground Truth (Right) predictions on Wrist X-rays. mBGD-FT identifies the text whereas BGD-FT detects objects where it's completely black, an indication that it is predicting the regions as opposed to actually detecting objects

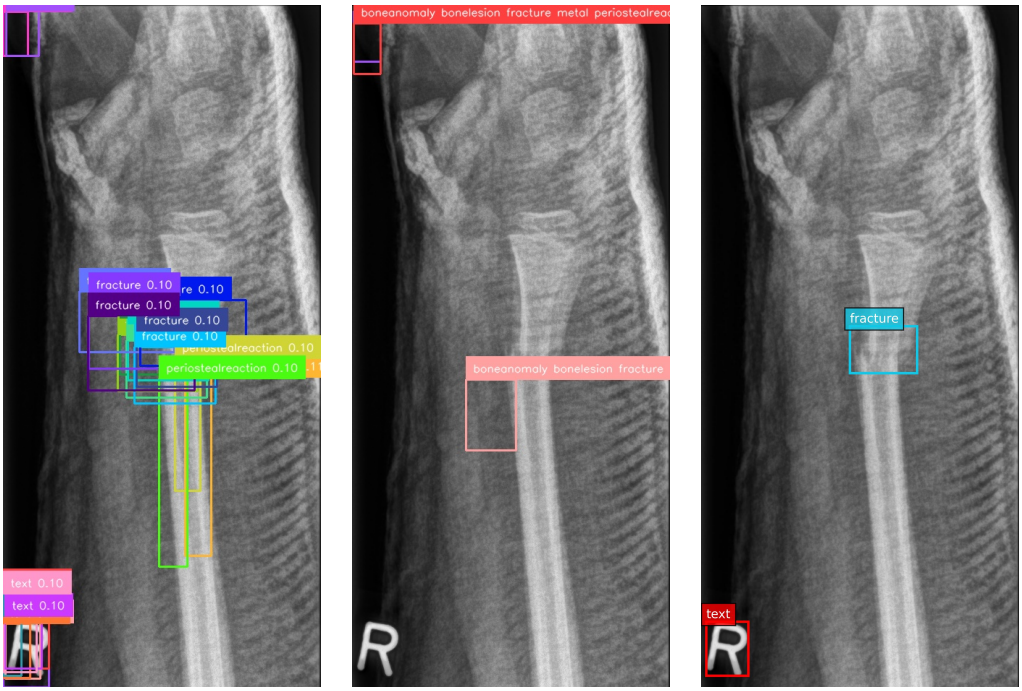

Figure 5: mBGD-FT (Left), BGD-FT (Middle), and Ground Truth (Right) predictions on Wrist X-rays. mBGD-FT is able to identify text and fracture whereas BGD-FT cannot.