
Set Prediction in the Latent Space

Konpat Preechakul¹

Chawan Piansaddhayanon¹

Burin Naowarat¹

Tirasan Khandhawit²

Sira Sriswasdi³

Ekapol Chuangsuwanich^{1,3}

¹Department of Computer Engineering, Chulalongkorn University

²Department of Mathematics, Faculty of Sciences, Mahidol University

³Computational Molecular Biology Group, Faculty of Medicine, Chulalongkorn University

konpatp@gmail.com, {6372025021, 6270145221}@student.chula.ac.th

tirasan.kha@mahidol.ac.th, {sira.sr, ekapol.c}@chula.ac.th

Abstract

Set prediction tasks require the matching between predicted set and ground truth set in order to propagate the gradient signal. Recent works have performed this matching in the original feature space thus requiring predefined distance functions. We propose a method for learning the distance function by performing the matching in the latent space learned from encoding networks. This method enables the use of teacher forcing which was not possible previously since matching in the feature space must be computed after the entire output sequence is generated. Nonetheless, a naive implementation of latent set prediction might not converge due to permutation instability. To address this problem, we provide sufficient conditions for permutation stability which begets an algorithm to improve the overall model convergence. Experiments on several set prediction tasks, including image captioning and object detection, demonstrate the effectiveness of our method. Code is available at <https://github.com/phizaz/latent-set-prediction>.

1 Introduction

Set prediction is a task where a model predicts multiple elements whose ordering is not relevant for correctness. This task is central to many real-world problems such as object detection, image captioning, and multi-speaker speech recognition. Object detection requires predicting a set of bounding boxes without any specific ordering. Describing objects within an image is a kind of image captioning yet perfectly suitable for set prediction. Multi-speaker speech recognition is also well suited for set prediction since the order of transcripts is irrelevant. Though these tasks can naturally be modeled as set prediction, traditional deep learning is not inherently suitable for these tasks.

Multi-layer perceptrons and convolution networks with traditional loss functions impose a specific ordering on the prediction heads which hinders set prediction. A reasonable set prediction pipeline requires the model's prediction heads to be more flexible. Each head does not have a predefined target, yet relies on its peers to determine what is best to predict to complete the target set. Recent works [1, 2] emphasized using a Transformer model [3], which is permutation-invariant, coupled with a permutation-invariant loss function as the main ingredients. Any traditional loss function can be made permutation-invariant by solving for a *minimum* bijective matching between predicted set and ground truth set via the Hungarian algorithm under a certain **distance metric**. After the assignment, the loss function is calculated between the assigned pairs, and backpropagation is performed accordingly. This scheme is known as Permutation Invariant Training (PIT).

A distance metric used by the assignment must agree with the loss function in a way that the assignment is kept after an optimization step on the loss function. A distance metric that fails this criterion may switch pairings hindering the convergence. Hence, a distance metric is crucial to the convergence property of the set optimization scheme. One may argue to use the loss function itself as a distance metric. However, not all loss functions have meaningful scalar values. For example, the vanilla GAN’s loss [4] is not insightful in terms of progress or distance. Combining loss functions from different domains also complicates the matter because they are not easily comparable in their scalar forms. In object detection, both L1 error loss and cross entropy loss are used to learn bounding box prediction [2], but it is unclear how to define a proper distance metric from such a combination. Either hand-tuned coefficients or different surrogate distance metrics may be needed to form an effective distance metric. This begets the problem of selecting a proper distance metric for PIT. A set prediction scheme that does not require a hand-tuned distance function is appreciable.

Another hardship related to PIT is when applying set prediction on sequence domains that require *teacher forcing* to train. Auto-regressive with teacher forcing is often used for sequence prediction such as speech recognition [5, 6] and machine translation [7, 3]. However, teacher forcing requires a groundtruth assignment before it can begin prediction. PIT also relies on the teacher forcing prediction to do minimum assignment, resulting in a chicken and egg problem. If the set cardinality is small enough, it is possible to exhaustively teacher force with respect to all possible ground truths requiring $O(N^2)$ forward passes through the model, and keep only those with the minimum assignment distances for optimization.

What if the Hungarian assignment is done in a latent space instead? Since the latent space is learned, the choice of any specific distance metric is alleviated – even a simple Euclidean distance is reasonable. Since the latent space is prior to the sequence prediction, the prediction process knows exactly what its ground truth is which allows for efficient, $O(N)$, teacher forcing. This paper presents **latent set prediction** (LSP) which enables the assignment in the latent space with Euclidean distance metric. At the same time, it provides a convergence guarantee of the loss function by reducing the effect of permutation switches that can be problematic when performing matching in the latent space. Our contributions are as follows:

1. We propose a framework for deep set prediction that alleviates the need for hand-crafted distance metrics.
2. This framework is efficient for the set of sequence predictions with teacher forcing requiring only $O(N)$ predictions, an improvement from the usual exhaustive $O(N^2)$.
3. We provide a convergence proof of set prediction under this framework.

2 Related works

Set prediction. There are mainly two families of set prediction: distribution matching and minimum assignment. The distribution matching approaches learn $P(Y|x)$ where Y is a set and x is an input. DeepSetNet and variants [8–10] proposed a likelihood function for set prediction. An energy function learned via adversarial samples was also proposed [11]. On the other hand, the minimum assignment approaches rely on solving assignment problems. The loss function is calculated between the assigned pairs afterward. Either Hungarian assignment (bijection) or Chamfer assignment is usually used depending on tasks. Zhang et al. [12] proposed to *mold* a primitive set into a target set via gradient signals from a set encoder. Kosiorek et al. [1] proposed a Transformer for set prediction. A similar kind of design was also used in end-to-end object detection [2]. Besides the two assignments, a stable marriage was proposed for set autoencoding pretraining [13].

Image captioning is not usually related to set prediction. This is true for *impression* captions such as MS-COCO [14] which only describe the most salient objects. A different kind of captioning is *descriptive* which describes individual objects in a scene and their interactions. A prominent example is Visual Genome [15]. For the same reason, a chest radiology report is also descriptive [16]. Since descriptive captions have no specific ordering, this task is actually a set prediction where the elements are captions themselves. To the best of our knowledge, there is no practical approach for set of text predictions that involves *teacher forcing*.

Object detection is formulated as a set prediction task where each bounding box is a set member. Most object detection algorithms impose ordering by dividing the image into several grids. Each cell

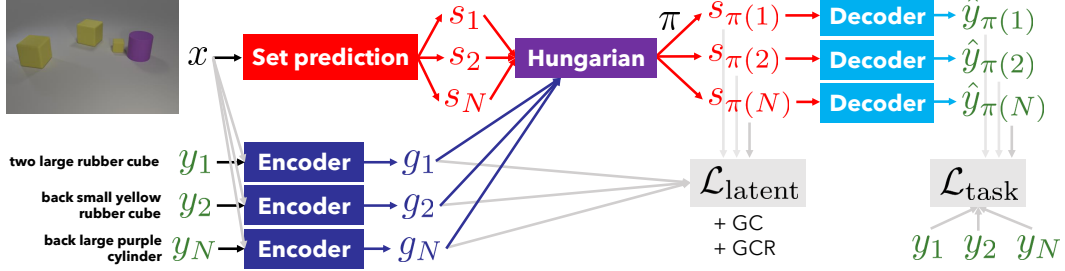


Figure 1: Latent set prediction (LSP) framework where x is an image and y 's are sentences (it can be applied to any x and y). The Hungarian algorithm is used to find the minimum assignment between s 's (predictions) and g 's (encoded y 's). This allows efficient *teacher forcing* at decoder \mathcal{D} which is not possible previously. The latent loss $\mathcal{L}_{\text{latent}}$ is applied to minimize the distance between the s - g pairs. The task loss $\mathcal{L}_{\text{task}}$ is applied as usual on the prediction. Only set prediction model and decoder are required during inference.

is responsible for predicting bounding boxes whose centers reside in it. Anchors are introduced to allow a single cell to support multiple objects [17, 18]. As a result, hand-crafted components are needed for these methods to function efficiently. Though many works are focusing on removing the use of anchors [19, 20], the dense grid prediction still remains. Later, DETR [2] directly applies set prediction on bounding boxes whose process matches predicted boxes to ground truth boxes. To achieve a satisfactory result, the matching cost has to be manually designed.

3 Latent Set Prediction (LSP)

A common set prediction pipeline has three components: a set prediction model, ground truths, and an assignment mechanism. Our method is focused on the case of Hungarian assignment. Traditionally, the assignment mechanism matches the ground truths with the model predictions in an *output space* Y . Here, the pairing happens in a **latent space** \mathbb{R}^C .

3.1 Notations

We assume a **set prediction model** $\mathcal{F} : X \rightarrow \mathbb{R}^{N \times C}$ where X is the input space and N is the cardinality of the set. Effectively, \mathcal{F} outputs N vectors in a latent space \mathbb{R}^C . The model \mathcal{F} is also responsible for set cardinality prediction. Each latent vector is passed through a **decoder** $\mathcal{D} : \mathbb{R}^C \rightarrow Y$ where Y is any output space. Note that the **decoder** may also accept the input $x \in X$ wherever the input is required for better prediction. To pair in the latent space, we utilize another component called **encoder** $\mathcal{E} : Y \times X \rightarrow \mathbb{R}^C$ where $y \in Y$ is an output and $x \in X$ is its corresponding input. The encoder maps elements of the output space as **guiding vectors** $g \in \mathbb{R}^C$ in the latent space to facilitate the assignment. Given a set of latent vectors $\{s_1, s_2, \dots, s_N\}$ and guiding vectors $\{g_1, g_2, \dots, g_N\}$, the minimum assignment π is

$$\pi = \operatorname{argmin}_{\pi' \in \mathcal{P}} \sum_i^N \|s_{\pi'(i)} - g_i\|_2 \quad (1)$$

where \mathcal{P} is the set of all permutations of N letters.

A **switch** is said to occur when $s_{\pi(i)}$ changes after a gradient update as illustrated in Figure 2.

3.2 Method

Latent set prediction (LSP) begins by feeding an input x into the **set prediction model** \mathcal{F} resulting in a set of **latent vectors** s 's. s 's do not have designated targets until the corresponding **guiding**

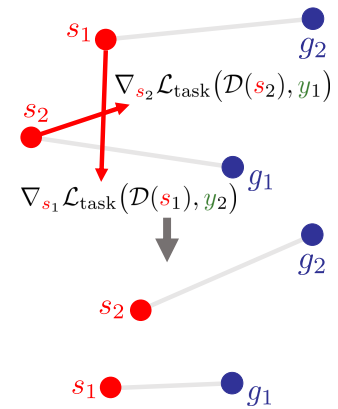


Figure 2: s 's move toward y 's designated by g 's. At the same time, s 's may move away from their g 's resulting in a **switch**.

vector g 's are retrieved. To get g , each **ground truth** y_i is mapped via the **encoder** resulting in g_i . The assignment algorithm is performed between s 's and g 's resulting in the **minimum bijective matching** π . This pairs up each latent vector $s_{\pi(i)}$ to the guiding vector g_i and the associated ground truth y_i . Knowing its target, s goes through the **decoder** \mathcal{D} resulting in \hat{y} . The **task loss function** $\mathcal{L}_{\text{task}}$ is calculated accordingly between $\hat{y}_{\pi(i)}$ and y_i , and then the optimization is performed. Note that s receives no training signal from g ; g only gives s its goal. After an optimization step, as $s_{\pi(i)}$ moves along the task gradient toward better prediction of y_i , it may move *away* from g_i and approach another guiding vector g_j . This can cause a **switch** as demonstrated in Figure 2. Our method incorporates several techniques that encourage stable pairing of s 's and g 's over time, which turns out to be sufficient for the convergence of $\mathcal{L}_{\text{task}}$. A pictorial description of the LSP framework is depicted in Figure 1.

The proof of convergence of $\mathcal{L}_{\text{task}}$ (Section 4) indicates that the convergence hinges on the ever smaller gaps between s 's and g 's under π . In fact, $\mathcal{L}_{\text{task}}$ is bounded from above by a function of $\sum_i^N \|s_{\pi(i)} - g_i\|_2$. Hence, not only that g 's give s 's their goals, g 's must also follow wherever s 's go. By closing the gaps, it is less likely for a switch to happen. Should a switch happen, it would only be between a short distance which is not as harmful to $\mathcal{L}_{\text{task}}$. This does not imply that we need to avert switches at all costs. We can simply assign g_i to s_i for all i to guarantee no switches. However, it is ordered prediction, not set prediction. In a sense, switches should be welcomed as a sign of learning a *natural* ordering as long as in the long run the gaps are still closing.

Therefore, we propose two mechanisms to make sure that the gaps between s 's and g 's are smaller over time. First, we enforce an **asymmetric latent loss** to bring s 's and g 's together:

$$\begin{aligned} \mathcal{L}_{\text{latent}}^{s \rightarrow g} &= \sum_i \frac{1}{2} \|s_{\pi(i)} - [g_i]\|_2^2 & \mathcal{L}_{\text{latent}}^{g \rightarrow s} &= \sum_i \frac{1}{2} \|[s_{\pi(i)}] - g_i\|_2^2 \\ \mathcal{L}_{\text{latent}} &= \beta \mathcal{L}_{\text{latent}}^{s \rightarrow g} + \gamma \mathcal{L}_{\text{latent}}^{g \rightarrow s} & \mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{latent}} + \mathcal{L}_{\text{task}} \end{aligned} \quad (2)$$

where $[\cdot]$ is stop gradient, and β, γ control the loss strengths. Ideally, we want to set $\beta = 0$ since s 's should only follow the training signals from $\mathcal{L}_{\text{task}}$. However, we found $\beta = 0.1$ to be useful in practice providing a bit of help for g 's to meet s_{π} 's. We set $\gamma = 1$ as the default value and found it to work well across experiments.

However, the latent loss alone is not enough to guarantee convergence. This is because the latent loss cannot anticipate the movement of s 's. Even for a pair of infinitesimally close s and g , any sizeable $\nabla_{s_{\pi}} \mathcal{L}_{\text{task}}$ can break apart the two. The second part which completes the convergence proof is **gradient cloning** (GC) which copies the task gradient $\nabla_{s_{\pi}} \mathcal{L}_{\text{task}}$ from s_{π} 's to their respective g 's. Theoretically, the distance between s and g is strictly decreasing which satisfies the requirement for convergence.

In practice, the models that predict s 's and g 's may not be equally capable as one may go faster than the other. To allow for this discrepancy, we propose a stronger version of GC namely **gradient cloning with rejection** (GCR). With GCR, the **leader** of each pair of s_{π} and g is *slowed down* when $\|\nabla_{s_{\pi}} \mathcal{L}_{\text{latent}} - \nabla_g \mathcal{L}_{\text{latent}}\|_2$ is larger than $d \|\nabla \mathcal{L}_{\text{task}}\|_2$. The constant d (default $d = 10^{-3}$) is indicating whether s and g are sufficiently far apart (relative to $\|\nabla \mathcal{L}_{\text{task}}\|_2$) requiring a slower leader. The leader is slowed down by rejecting its $\nabla \mathcal{L}_{\text{task}}$ along the span of $\nabla \mathcal{L}_{\text{latent}}$. The one with an *obtuse* angle between its task and latent gradients is considered a leader: $\langle \nabla_{s_{\pi}} \mathcal{L}_{\text{task}}, \nabla_{s_{\pi}} \mathcal{L}_{\text{latent}} \rangle < 0$ (in case of s) or $\langle \nabla_{s_{\pi}} \mathcal{L}_{\text{task}}, \nabla_g \mathcal{L}_{\text{latent}} \rangle < 0$ (in case of g). d serves as a parameter for choosing between GC ($d = \infty$) and GCR with always rejection ($d = 0$). A smaller d puts a stronger tendency for converging s and g at the cost of slower learning of s . The idea is depicted in Figure 3 and described in Algorithm 2.

We summarize LSP in Algorithm 1 which can be implemented efficiently with modern deep learning frameworks.

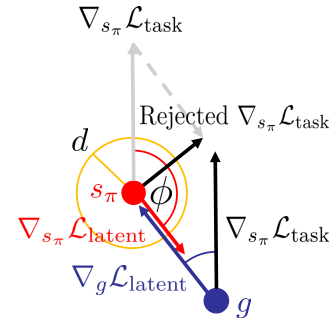


Figure 3: GC with rejection. s is the leader in this case since its $\nabla \mathcal{L}_{\text{task}}$ and $\nabla \mathcal{L}_{\text{latent}}$ form an obtuse angle. Its $\nabla \mathcal{L}_{\text{task}}$ is rejected along the span of its $\nabla \mathcal{L}_{\text{latent}}$ when its bidirectional latent gradient's length exceeds $d \|\mathcal{L}_{\text{task}}\|_2$.

Algorithm 1 Single training step of Latent Set Prediction (LSP)

Given $\mathbf{x} \in X, \mathbf{Y} \in Y^N, d \in \mathbb{R}$
 $\mathbf{S} \leftarrow \mathcal{F}(\mathbf{x})$ $\triangleright \mathbf{S} \in \mathbb{R}^{N \times C}$, latent set element prediction

(Inference only)
 $\hat{\mathbf{Y}} \leftarrow \mathcal{D}(\mathbf{S})$ $\triangleright \hat{\mathbf{Y}} \in Y^N$, prediction on output space

(Training only)
 $\mathbf{G} \leftarrow \mathcal{E}(\mathbf{Y}, \text{repeat}(\mathbf{x}))$ $\triangleright \mathbf{G} \in \mathbb{R}^{N \times C}$, ground truth encoding
 $\pi \leftarrow \text{Hungarian}(\mathbf{G}, \mathbf{S})$ \triangleright Equation 1
 $\hat{\mathbf{Y}}_\pi \leftarrow \mathcal{D}(\mathbf{S}_\pi)$ $\triangleright \hat{\mathbf{Y}}_\pi \in Y^N$, prediction on output space
 $\mathcal{L}_{\text{latent}} \leftarrow \mathcal{L}_{\text{latent}}(\mathbf{S}_\pi, \mathbf{G})$ \triangleright Equation 2
 $\mathcal{L}_{\text{task}} \leftarrow \mathcal{L}_{\text{task}}(\hat{\mathbf{Y}}_\pi, \mathbf{Y})$
 $\nabla_{\mathbf{S}_\pi} \mathcal{L}_{\text{task}}, \nabla_{\mathbf{S}_\pi} \mathcal{L}_{\text{latent}}, \nabla_{\mathbf{G}} \mathcal{L}_{\text{latent}} \leftarrow \text{Backprop}(\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{latent}})$
 $\nabla_{\mathbf{S}_\pi} \leftarrow \text{GCR}(\nabla_{\mathbf{S}_\pi} \mathcal{L}_{\text{task}}, \nabla_{\mathbf{S}_\pi} \mathcal{L}_{\text{latent}} - \nabla_{\mathbf{G}} \mathcal{L}_{\text{latent}}, d)$ \triangleright Algorithm 2
 $\nabla_{\mathbf{G}} \leftarrow \text{GCR}(\nabla_{\mathbf{S}_\pi} \mathcal{L}_{\text{task}}, \nabla_{\mathbf{G}} \mathcal{L}_{\text{latent}} - \nabla_{\mathbf{S}_\pi} \mathcal{L}_{\text{latent}}, d)$ \triangleright Algorithm 2
Continue backpropagation to $\mathcal{F}, \mathcal{D}, \mathcal{E}$'s parameters

Algorithm 2 Gradient Cloning with Rejection (GCR)

Given $\nabla \mathcal{L}_{\text{task}}, \nabla \mathcal{L}_{\text{latent}}$, and $d \in \mathbb{R}$
 $\text{obtuse} \leftarrow \langle \nabla \mathcal{L}_{\text{task}}, \nabla \mathcal{L}_{\text{latent}} \rangle < 0$ \triangleright obtuse $\in [0, 1]^N$, obtuse angles indicate leading positions
 $\text{far} \leftarrow \|\nabla \mathcal{L}_{\text{latent}}\|_2 > d \cdot \|\nabla \mathcal{L}_{\text{task}}\|_2$ \triangleright far $\in [0, 1]^N$, large latent gradients indicate large distances
 $\hat{\nabla} \mathcal{L}_{\text{latent}} \leftarrow \frac{\mathcal{L}_{\text{latent}}}{\|\mathcal{L}_{\text{latent}}\|_2}$
 $\nabla \mathcal{L}_{\text{task}} \leftarrow \nabla \mathcal{L}_{\text{task}} - (\text{obtuse} \cdot \text{far}) \cdot \hat{\nabla} \mathcal{L}_{\text{latent}} \cdot \langle \nabla \mathcal{L}_{\text{task}}, \hat{\nabla} \mathcal{L}_{\text{latent}} \rangle$ \triangleright Gradient rejection
Return $\nabla \mathcal{L}_{\text{task}} + \nabla \mathcal{L}_{\text{latent}}$

4 Convergence analysis

In this section, we show that **gradient cloning** (GC) technique together with a special case of **asymmetric latent loss**, $\beta = 0$ in (2), is sufficient for the convergence of LSP to a local minimum under the following mild assumptions:

1. Each latent vector s_i and each guiding vector g_i is updated according to the gradients exactly as expressed in the training dynamics defined in Section 4.1.
2. $\mathcal{L}_{\text{task}}$ is **L-smooth** and satisfies the **Polyak-Lojasiewicz** condition. This is typically assumed to prove the convergence of the gradient descent algorithm [21].

Under the standard gradient descent setup, the convergence of LSP is complicated by the fact that each **switch** can increase the task loss as $s_{\pi(i)}$ changes its target from y_i to a new y_j . Our **gradient cloning** and **asymmetric latent loss** techniques ensure that even though **switch** can keep occurring throughout the model training process, its impact on task loss will decay exponentially.

Detailed proofs are provided as an Appendix for interested readers. It should be noted that similar results can be obtained for **gradient cloning with rejection** (GCR) and general cases of **asymmetric latent loss** with $\beta > 0$.

4.1 LSP training dynamics with gradient cloning

We begin with the explicit notations for Algorithm 1. The training at each time point $t + 1$ consists of two steps. First, the assignment π is updated according to the values of $s_{\pi(i)}^{(t)}$'s and $g_i^{(t)}$'s from the previous time step as defined in (1). Then, the values of $s_{\pi(i)}$'s and g_i 's are updated based on the gradients from $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{latent}}$ with **step size** η and **latent loss strength** γ as defined in (2).

$$\begin{aligned} s_{\pi^{(t+1)}(i)}^{(t+1)} &= s_{\pi^{(t+1)}(i)}^{(t)} - \eta \nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) \\ g_i^{(t+1)} &= g_i^{(t)} - \eta \left(\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) + \gamma \nabla_g l_{\text{latent}}(s_{\pi^{(t+1)}(i)}^{(t)}, g_i^{(t)}) \right) \end{aligned}$$

where $l_{\text{task}}(\cdot, y_i)$ subsumes the prediction head $g(\cdot)$. The lower case notations l_{task} and l_{latent} correspond to per-data-point loss functions.

A direct implication of gradient cloning, which attracts s and g together, is that the total distance $\sum_i \|s_{\pi^{(t)}(i)}^{(t)} - g_i^{(t)}\|_2$ decays exponentially. This consequently ensures that when a **switch** occurs at time t , the distance between involved latent vectors $\|s_{\pi^{(t+1)}(i)}^{(t)} - s_{\pi^{(t)}(i)}^{(t)}\|_2$ also decay exponentially. Hence, the impact of **switch** on $\mathcal{L}_{\text{task}}$ decreases rapidly over the course of model training.

4.2 Impact of a switch on task loss

The impact of a **switch** on task loss can be illustrated mathematically using the **L-smooth** condition

$$l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \leq \frac{L}{2} d_t^2 + d_t \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 - c \|\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i)\|_2^2,$$

where $c > 0$ and $d_t = \|s_{\pi^{(t+1)}(i)}^{(t)} - s_{\pi^{(t)}(i)}^{(t)}\|_2$ is the distance between latent vectors involved in a **switch**. It should be noted that the negative gradient term appears in a typical proof of convergence for gradient descent while the terms with d_t are introduced by the assignment step (1). In the absence of a switch, task loss always decreases. However, if d_t is large, the first two terms can dominate.

4.3 Convergence of LSP

By viewing the right-hand side of the inequality above as a quadratic function of d_t , we can see that if the magnitude of task gradients are larger than some factor of d_t , then the right-hand side must be negative. This implies that the task loss decreases. On the other hand, if the task gradients are small, the **Polyak-Lojasiewicz** condition

$$\frac{1}{2} \|\nabla_s l_{\text{task}}(x, y)\|_2^2 \geq \mu \cdot (l_{\text{task}}(x, y) - l_{\text{task}}(x^*, y)), \text{ where } x^* \text{ is a global minimum} \quad (3)$$

then implies that our optimization is already near a minimum. Thus, we put the two cases together to obtain the following key result.

Theorem 4.1. *If $l_{\text{task}}(\cdot, y)$ is **L-smooth** and satisfies the **Polyak-Lojasiewicz** condition, then*

$$l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_i^*, y_i) \leq \begin{cases} C \alpha^{2t}, & \text{if } \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\| \leq \frac{3d_t}{\eta(1-\frac{L}{2})} \\ \delta \left(l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) - l_{\text{task}}(s_i^*, y_i) \right), & \text{otherwise} \end{cases}$$

where appropriate choices of **step size** η will ensure that all constants are positive and $\alpha, \delta < 1$.

This implies that the difference between the current task loss and the global minimum is bounded above by the maximum of two sequences of positive real numbers, both converging to zero with linear rates. Therefore, the task loss of LSP converges to a local minimum with a linear rate.

5 Experiments

We first demonstrate that GC and GCR help reduce switches in a synthetic dataset. Without our methods, the models might not converge. Then, we apply LSP on common set prediction tasks such as object detection and point cloud prediction (see Appendix B). One unique ability that LSP enables is allowing teacher forcing in set of texts prediction scenarios which we demonstrate on a CLEVR object description task and on a challenging MIMIC chest x-ray report generation task. Without LSP, these kind of tasks were not possible to perform set prediction in due to the computational cost.

5.1 Synthetic dataset

This experiment aims to demonstrate the convergence properties of LSP variants (without GC, with GC, and with GCR ($d = 0$)). Three sets of N random points from standard normal distribution were generated. The three sets represent s 's, g 's, and y 's, all in \mathbb{R}^{dim} space. $\mathcal{L}_{\text{task}}(s_\pi, y) = \|s_\pi - y\|_2^2$, i.e. no prediction head. The loss is defined as $\mathcal{L}_{\text{task}}(s_\pi, y) + \alpha \mathcal{L}_{\text{latent}}(s_\pi, g)$ where α serves as a relative

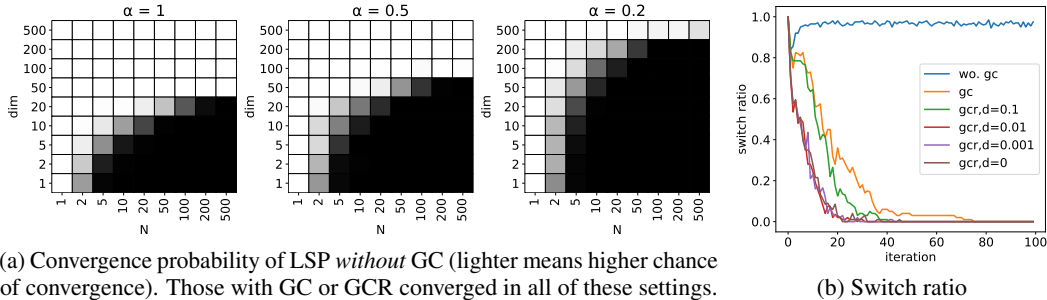


Figure 4: Comparing LSP with/without GC and GCR on the synthetic dataset.

strength between $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{latent}}$. We altered α to better demonstrate their behaviors in practice since both losses may be of different magnitudes. No neural networks were used in this experiment.

Convergence probability is how likely a trial will converge (from 100 trials). A trial is considered converged if after 300 iterations¹ $\mathcal{L}_{\text{task}} < 0.01$. A robust algorithm should always converge. Figure 4a demonstrates LSP *without* GC on different N 's and \dim 's and α 's. The results confirmed that without GC the training may not converge while those with GC converged robustly in all of these settings. Positive factors for convergence are: smaller N , larger \dim , larger α . In other words, keeping s closer to g than the other s 's. Larger N and smaller \dim reduce average spaces between points, and smaller α leaves a larger gap between a leading s and a trailing g weakening the bond.

Switch ratio is the fraction of s 's that were matched to different y 's after an update. Decreasing switch ratio as the training progresses is a good sign for convergence. We experimented with $N = 200, \dim = 2, \alpha = 0.5$ and penalized the gradient towards g 's to be 0.5 times smaller than those of s 's. This setting demonstrates a suboptimal encoder that cannot easily follow s 's. Figure 4b that, with a suboptimal encoder, GCR ($d = 0$) reduces the switch ratio the fastest due to its ability to slow down the faster s 's the most.

5.2 Object detection

We will demonstrate that LSP is applicable to common set prediction tasks such as object detection and point cloud prediction (see Appendix B). The goal is to show that LSP achieves a competitive performance compared to a manually designed assignment cost. We compared LSP against DETR [2], which is a reasonably strong baseline, that can be adapted to work with LSP with minimal changes. We used our modified MNIST dataset [22] in this experiment. The dataset contains 5,000 training and 1,000 test images. Each data point contains multiple randomly placed digits from the MNIST dataset. To increase the difficulty of the dataset, each digit in the image was augmented by using a random photometric distortion, morphological transformation, and random resizing. We reported test AP of the last training epoch. AP_L was not reported because there is no large object in our dataset. See Appendix Figure 6 for example images in our dataset.

Table 1 shows that **LSP** achieved a competitive result compared to a DETR baseline. Particularly, LSP with GCR outperformed the baseline when a small value of d was used. The gain primarily came from an increase in predicted bounding box quality shown by +3.6 AP_{75} over the baseline. The result also suggested that **LSP** led to a more robust matching for different object sizes as the matching cost is learnt. This is contrary to a manually designed fixed matching cost that usually puts smaller importance on smaller objects (smaller bounding boxes). As a result, LSP improved the detection performance on small objects (AP_S) by +2.4 points over the baseline. A complete description of this task is provided in Appendix A.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M
DETR [2]	43.5	71.0	49.1	39.8	64.5
LSP (GC)	31.1	62.3	26.8	28.3	48.9
LSP (GCR, $d = 10^{-3}$)	45.0	71.2	51.4	41.6	64.7
LSP (GCR, $d = 10^{-4}$)	45.6	71.2	52.7	42.2	64.8
LSP (GCR, $d = 0$)	43.8	69.8	50.2	40.0	64.8

Table 1: Performance of object detection task on the test set of our modified MNIST dataset.

¹We observed that non-convergent trials demonstrated plateau loss curves within 300 iterations.

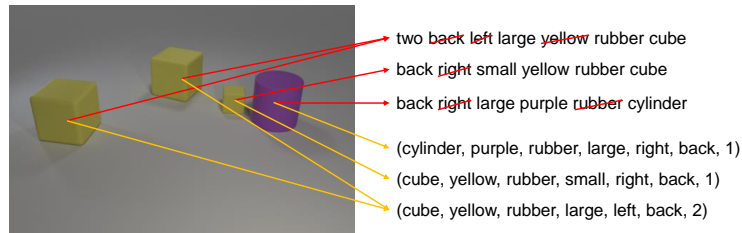


Figure 5: We repurposed CLEVR dataset [23] for object description task. The descriptions (red arrows) were from objects’ attributes (yellow arrows) which came from the metadata. We randomly dropped attributes from descriptions to make the description generation task more challenging.

Method	Precision	Recall	F1
Concat	0.931	0.910	0.920
Ordered set	0.957	0.526	0.679
LSP (w/o GC)	0.976	0.900	0.936
LSP (GC)	0.986	0.972	0.979
LSP (GCR, $d = 10^{-3}$)	0.983	0.975	0.979
LSP (GCR, $d = 0$)	0.983	0.972	0.978

Table 2: CLEVR object description generation task. Reported averages of three trials.

Method	micro avg. BLEU		
	$\hat{y} \rightarrow y$	$y \rightarrow \hat{y}$	hmean
RM+MCLN [26]*	16.7	15.6	16.2
Concat	16.7	15.2	15.9
Ordered set	20.2	18.5	19.3
LSP (GC)	17.9	24.2	20.6
LSP (GCR, $d = 10^{-3}$)	19.1	24.6	21.5
LSP (GCR, $d = 0$)	18.9	24.7	21.4

Table 3: MIMIC-CXR report generation task. Reported averages of three trials except * which was run once.

5.3 CLEVR object description generation

Image captioning or paragraph captioning [24] can be considered a kind of set of texts prediction, yet are usually tackled as one long text. Set of texts is hard because teacher forcing does not work with Hungarian assignment. In this experiment and the next, we demonstrate that LSP enables set of texts prediction that leads to superior performances.

We re-purposed the CLEVR dataset [23], which was originally designed for visual reasoning, for image captioning. We selected CLEVR to represent a clean dataset which we know the ground truth exactly. Each image contains objects of different kinds (attributes) described by a text description derived from its attributes. Keywords were randomly dropped from the description to make the task more challenging. The final description was guaranteed to be unambiguous for each object to keep the task tractable. See Figure 5 for examples. We evaluated the models by the ratio of ground truth objects that were described by the model (recall) and the ratio of predicted descriptions that were supported by the ground truths (precision). We reported micro average precision, recall, and F1.

Image captioning is usually tackled by concatenating descriptions into a single caption. We imbued a deterministic alphabetical ordering² of descriptions to help the model learn. The model called **Concat** which resembles show-attend-tell [25] albeit with Transformer. A more reasonable approach is to describe each object description individually as a set. However, PIT does not facilitate teacher forcing during training. A practical approach is to turn a set into an **Ordered set** to circumvent the matching problem. Each prediction head in the model is responsible for generating the description of each object in an alphabetical order. The most appropriate approach should assume no fixed ordering as true set prediction which is possible with our **LSP** model. The results shown in Table 2 demonstrated strong performances from **LSP** while **Ordered set** lacked behind in recall. **Concat** was strong due to its fully autoregressive nature, yet not as strong as **LSP**. LSP results also improve with the addition of our proposed GC or GCR. Furthermore, modeling as sets allows the prediction heads to specialize. We observed this effect with the LSP model which was further described in Appendix Figure 10. A complete description of this task is provided in Appendix C.

5.4 Chest radiograph report generation

With the ability to do set of texts prediction, LSP has potentials for a challenging image captioning task such as chest radiograph report generation [26, 27] which has real-world applications [28]. We

²Alphabetical ordering was better than ordering by length in our preliminary experiments.


X-ray	Ground truth	LSP	Ordered set	Concat
	frontal and lateral views of the chest were obtained. there are streaky linear opacities at the lung bases which are likely due to atelectasis with chronic changes. no definite focal consolidation is seen. there is no pleural effusion or pneumothorax. no pneumothorax is seen. the aorta is calcified and tortuous. the cardiac silhouette is top normal to mildly enlarged. dual-lead left-sided pacemaker is seen with leads in the expected positions of the right atrium and right ventricle. chronic-appearing rib deformities on the right is again seen.	frontal and lateral views of the chest were obtained. there is a small left pleural effusion with overlying atelectasis. there is no focal consolidation, pleural effusion or pneumothorax. there is no pleural effusion or pneumothorax. the aorta is calcified and tortuous. the heart is mildly enlarged. a left-sided pacemaker is seen with leads in the expected position of the right atrium and right ventricle. the patient is status post median sternotomy and cabg. the lungs are otherwise clear.	pa and lateral views of the chest provided. there is no pneumothorax. no pneumothorax is seen. the heart is mildly enlarged. the patient is status post median sternotomy and cabg. the lungs are hyperinflated with flattening of the diaphragms, suggesting chronic obstructive pulmonary disease.	pa and lateral views of the chest provided. no large effusion or pneumothorax. the heart is mildly enlarged. midline sternotomy wires and mediastinal clips are noted. there is a left chest wall pacer device with lead tips extending to the region of the right atrium and right ventricle. there is mild pulmonary edema. no convincing signs of pneumonia. bony structures are intact. mediastinal contour is normal. no free air below the right hemidiaphragm.

Table 4: A chest radiograph report generation example. Similar sentences were reordered and color coded for the ease of comparison. Highlighted segments are major discrepancies from the ground truth pointed out by a radiologist.

used MIMIC-CXR dataset [16] which contains 377,110 chest x-rays with 227,835 reports from 65,379 patients with the average length of 50 words per report. It is important to note that set of texts prediction may not be much beneficial on shorter caption datasets such as MS-COCO [14]. MIMIC-CXR is considered noisy because multiple radiologists contributed to the dataset. The report may be incomplete and/or inconsistent depending on the writer. Also, the report is not entirely predictable because it usually refers to previous studies or to preconditions of the patient. We used a specific kind of BLEU score [29, 30] for evaluation which focuses on correctness and completeness. We calculated sentence-level BLEU scores from every source sentence to the highest BLEU target sentence. It was calculated both ways from predictions to ground truths $\hat{y} \rightarrow y$ and vice versa $y \rightarrow \hat{y}$. To get a single summary metric we compute the harmonic mean between $\hat{y} \rightarrow y$ and $y \rightarrow \hat{y}$. Note that we discarded all *blank* predictions before scoring. This score does not penalize duplicated predictions.

We included **Concat**, **Ordered set**, and Transformer with relational memory (**RM+MCLN**) [26], which is also a kind of **Concat**, as baselines against our **LSP** model. The three baselines followed the original ordering in the reports which was found to work better than alphabetical ordering. **Ordered set** and **LSP** modeled the task as a set of sentences. To better capture report diversity, we trained both models to always predict 10 sentences (from average 5.4 sentences per report). The same cannot be done with concatenation baselines. A representative example was shown in Table 4 (duplicate sentences were removed). Qualitatively, **Concat** generated a sound report. Like most radiologists, it mentioned just a few frequent negative findings. This shows that **Concat** was heavily biased by the imperfection of this dataset. **Ordered set** predicted the most duplicated sentences. Due to report diversity, the order of a particular finding sentence can differ between reports. This prevents **Ordered set** to specialize its prediction heads resulting in duplicated sentences being predicted. **LSP** predicted the most diverse sentences and was the best at capturing negative findings thanks to its head specialization. Quantitatively, Table 3 shows the superiority of both **Ordered set** and **LSP** models mainly due to their ability to over-predict. Although **Ordered set** has a slight edge on the $\hat{y} \rightarrow y$ metric, **LSP** has a substantial improvement in $y \rightarrow \hat{y}$ resulting in the best harmonic mean score. A complete description of this task and more prediction examples are provided in Appendix D.

6 Broader Impact and Limitation

LSP is applicable to all set prediction tasks as long as the set members are representable as latent vectors. Mature set prediction tasks like object detection are likely to receive only incremental improvements from LSP. However, LSP has larger implications on tasks that were previously hard to implement as set prediction including acoustic source separation. A questionable application like mass surveillance might be made possible by a practical acoustic source separation using LSP.

The proof of convergence (Section 4) relies on an assumption that both s 's and g 's respond to gradient updates exactly. This assumption is only satisfied without a neural network. Hence, we cannot mathematically guarantee the convergence in the general case.

Although LSP does away with the need for specifying distance metrics, it requires a *reasonable* encoder to be designed instead. One may argue that designing an encoder is not an easy task in some cases. One such example is the point cloud autoencoder task. Another example is the task of acoustic source separation where one wants to decompose a mixture of sounds. It might not be obvious what kind of encoder should be used in order to learn the latent information required in order to reconstruct each source.

In this paper, we investigated and designed encoders for a few tasks. One can use these as guidelines. However, the design of encoder in a completely different domain may require a non-trivial investment.

7 Conclusion

Set prediction requires a suitable distance metric that is also efficient to calculate. We proposed LSP as a potential answer to both criteria. We gave a theoretical model of LSP and showed its convergence properties under assumptions. This encourages usages in practical settings as we have shown with object detection and image captioning. LSP did away with the need for hand-crafted distance measures in object detection and made teacher forcing a viable option for set of text predictions. We envision that LSP will broaden the applicability of set prediction to other domains where a distance metric is hard to obtain or define.

References

- [1] Adam R Kosiorek, Hyunjik Kim, and Danilo J Rezende, "Conditional set generation with transformers," June 2020.
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, et al., "End-to-End object detection with transformers," May 2020.
- [3] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al., "Attention is all you need," , no. Nips, June 2017.
- [4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z Ghahramani, M Welling, C Cortes, et al., Eds., pp. 2672–2680. Curran Associates, Inc., 2014.
- [5] Xuankai Chang, Yanmin Qian, Kai Yu, et al., "End-to-End monaural multi-speaker ASR system without pretraining," Nov. 2018.
- [6] Xuankai Chang, Wangyou Zhang, Yanmin Qian, et al., "End-to-End multi-speaker speech recognition with transformer," Feb. 2020.
- [7] Xiaodong Liu, Kevin Duh, Liyuan Liu, et al., "Very deep transformers for neural machine translation," Aug. 2020.
- [8] S Hamid Rezaatofghi, Kumar B G Vijay, Anton Milan, et al., "DeepSetNet: Predicting sets with deep neural networks," Nov. 2016.
- [9] Hamid Rezaatofghi, Roman Kaskman, Farbod T Motlagh, et al., "Learn to predict sets using Feed-Forward neural networks," Jan. 2020.
- [10] S Hamid Rezaatofghi, Roman Kaskman, Farbod T Motlagh, et al., "Deep Perm-Set net: Learn to predict sets with unknown permutation and cardinality using deep neural networks," May 2018.
- [11] David W Zhang, Gertjan J Burghouts, and Cees G M Snoek, "Set prediction without imposing structure as conditional density estimation," Sept. 2020.
- [12] Yan Zhang, Jonathon Hare, and Adam Prugel-Bennett, "Deep set prediction networks," in *Advances in Neural Information Processing Systems 32*, H Wallach, H Larochelle, A Beygelzimer, et al., Eds., pp. 3212–3222. Curran Associates, Inc., 2019.
- [13] Malte Probst, "The set autoencoder: Unsupervised representation learning for sets," Feb. 2018.

- [14] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al., “Microsoft COCO: Common objects in context,” May 2014.
- [15] Ranjay Krishna, Yuke Zhu, Oliver Groth, et al., “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *Int. J. Comput. Vis.*, , no. 1, pp. 32–73, Feb. 2016.
- [16] Alistair E W Johnson, Tom J Pollard, Nathaniel R Greenbaum, et al., “MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs,” Jan. 2019.
- [17] Shaoqing Ren, Kaiming He, Ross Girshick, et al., “Faster R-CNN: Towards Real-Time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, C Cortes, N Lawrence, D Lee, et al., Eds. 2015, vol. 28, Curran Associates, Inc.
- [18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, et al., “SSD: Single shot MultiBox detector,” in *Computer Vision – ECCV 2016*. 2016, pp. 21–37, Springer International Publishing.
- [19] Kaiwen Duan, Song Bai, Lingxi Xie, et al., “CenterNet: Keypoint triplets for object detection,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Apr. 2019, pp. 6568–6577.
- [20] Zhi Tian, Chunhua Shen, Hao Chen, et al., “FCOS: Fully convolutional one-stage object detection,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019, IEEE.
- [21] Hamed Karimi, Julie Nutini, and Mark Schmidt, “Linear convergence of gradient and Proximal-Gradient methods under the Polyak-Łojasiewicz condition,” Aug. 2016.
- [22] Yann LeCun, Corinna Cortes, and C J Burges, “MNIST handwritten digit database,” *AT&T Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, vol. 2, pp. 18, 2010.
- [23] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, et al., “CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning,” Dec. 2016.
- [24] Jonathan Krause, Justin Johnson, Ranjay Krishna, et al., “A hierarchical approach for generating descriptive image paragraphs,” Nov. 2016.
- [25] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, et al., “Show, attend and tell: Neural image caption generation with visual attention,” in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*. July 2015, ICML’15, pp. 2048–2057, JMLR.org.
- [26] Zhihong Chen, Yan Song, Tsung-Hui Chang, et al., “Generating radiology reports via memory-driven transformer,” Oct. 2020.
- [27] Guanxiong Liu, Tzu-Ming Harry Hsu, Matthew McDermott, et al., “Clinically accurate chest X-Ray report generation,” in *Proceedings of the 4th Machine Learning for Healthcare Conference, PMLR, Finale Doshi-Velez, Jim Fackler, Ken Jung, et al., Eds., Ann Arbor, Michigan, 2019*, vol. 106 of *Proceedings of Machine Learning Research*, pp. 249–269, PMLR.
- [28] Yongsik Sim, Myung Jin Chung, Elmar Kotter, et al., “Deep convolutional neural network-based software improves radiologist detection of malignant lung nodules on chest radiographs,” *Radiology*, vol. 294, no. 1, pp. 199–209, Jan. 2020.
- [29] Kishore Papineni, Salim Roukos, Todd Ward, et al., “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, July 2002, ACL ’02, pp. 311–318, Association for Computational Linguistics.
- [30] Matt Post, “A call for clarity in reporting BLEU scores,” in *Proceedings of the Third Conference on Machine Translation: Research Papers*, Brussels, Belgium, Oct. 2018, pp. 186–191, Association for Computational Linguistics.
- [31] K He, X Zhang, S Ren, et al., “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.
- [32] Yan Zhang, Jonathon Hare, and Adam Prügél-Bennett, “FSPool: Learning set representations with featurewise sort pooling,” June 2019.
- [33] Yinhan Liu, Myle Ott, Naman Goyal, et al., “RoBERTa: A robustly optimized BERT pretraining approach,” July 2019.
- [34] Thomas Wolf, Lysandre Debut, Victor Sanh, et al., “HuggingFace’s transformers: State-of-the-art natural language processing,” Oct. 2019.

- [35] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, et al., “spacy: Industrial-strength natural language processing in python,” 2020.
- [36] Robert M Gower, “Convergence theorems for gradient descent,” Tech. Rep., Sept. 2019.

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
 - (b) Did you describe the limitations of your work? [Yes]
 - (c) Did you discuss any potential negative societal impacts of your work? [Yes]
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? [Yes]
 - (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We will publish the code as a supplementary material.
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We observed relatively small standard deviations (where applicable on multiple trials) compared to the performance gaps. To reduce clutter, we reported the standard deviations in the Appendix.
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We included a typical training time for a run on all experiments.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? [Yes]
 - (b) Did you mention the license of the assets? [No]
 - (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
 - (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

Table of Contents

A Object detection	13
A.1 Dataset	13
A.2 Models	14
A.3 Training details	14
A.4 Ablation studies	15
A.5 Convergence speed	15
A.6 Qualitative results	15
B MNIST Point Cloud Autoencoding	15
B.1 Model	16
B.2 Experimental setup	16
B.3 Results	16
C CLEVR object description generation	17
C.1 Dataset	17
C.2 Models	18
C.3 Training details	19
C.4 Results with standard deviations	19
C.5 Effect of hyperparameter in asymmetric latent loss	19
C.6 Effect of GCR’s d	19
C.7 Head specialization	19
C.8 Convergence speed comparison	19
D Chest radiograph report generation	20
D.1 Dataset	20
D.2 Models	20
D.3 Training details	22
D.4 Results with standard deviations	22
D.5 Effect of GCR’s d	22
D.6 Prediction samples	22
E Proof of convergence	24
E.1 Recall of the setup	24
E.2 Exponential decay of the switch distance	25
E.3 Convergence of the main task’s loss function	26
E.4 Impact of $\beta > 0$	28
E.5 Impact of gradient cloning with rejection	30
F Computational resources	30

A Object detection

A.1 Dataset

We used a modified MNIST dataset consisting of 5,000 training and 1,000 testing images. Each datapoint is a 160×160 image canvas containing multiple randomly placed digits from the MNIST dataset. To increase the task difficulty, each digit had to go through random image transformations

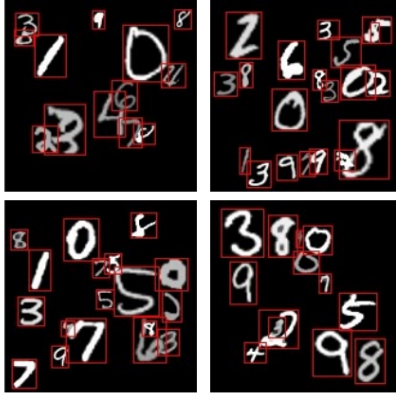


Figure 6: Example of datapoints in our object detection dataset and its ground truth (red boxes).

before being placed on the canvas. First, an image was randomly selected from the MNIST dataset and randomly resized to a square image with a size ranging from 16×16 to 64×64 . After that, the resized image (digit) was further augmented by random brightness, contrast, Gaussian blur, opening, and closing morphological operations. Then, the augmented image (digit) was randomly placed on the canvas with a constraint preventing it from having an excessive amount of highly overlapped objects. We limited the maximum number of objects in a single image to 50. The digit placement was not allowed when the summation of gray-scale value (intensity) in the bounding box area is higher than 20,000. The bounding box of each digit in the MNIST dataset was obtained by finding the border of the largest connected component in the digit image. An example of the generated dataset is shown in Figure 6. The dataset will be released with the code base.

A.2 Models

Every experiment was conducted using **DETR** with ResNet-50 [31] backbone, and 6 layers of both Transformer encoder and decoder with 256 hidden units and 8 attention heads. The backbone was ImageNet-pretrained. The size of class prediction heads was adjusted to match the number of classes in our dataset.

LSP model was based on **DETR** model with an extra encoder \mathcal{E} component. We used an encoder $\mathcal{E}(\text{box}, \text{class}, x)$ where x is the average pooled features from the ResNet-50 backbone. The encoder had the following architecture:

$$\begin{aligned} a &= \text{MLP}(\text{box}) \\ b &= \text{MLP}(\text{class}) \\ c &= \text{MLP}(x) \\ g &= \text{Linear}(\text{concat}(a, b, c)) \end{aligned}$$

Each MLP is a three-layer MLP with 256 hidden units with layer norm and ReLU activation after each layer. g has 256 units.

Remarks on the LSP model. The DETR model supervises on *all* layers of the Transformer decoder sharing the same prediction head. This aims to help training the deep architecture more effectively. Under the LSP terms, each intermediate state of the Transformer decoder layers is considered a set of s 's. There are *six* such layers hence six sets of s 's. We treated them individually in the experiments. That is we have six sets of g 's predicted by six different instances of encoders \mathcal{E} . This simplification disregards the fact that one set of s may affect the other five, yet was found to work well in practice.

A.3 Training details

We used the same set of training hyperparameters as DETR except for the number of training iterations, batch size, and augmentation strategy. Every model was trained for 100,000 iterations with the initial Transformer's learning rate of 10^{-4} , and the backbone's learning rate of 10^{-5} . The learning rate was divided by 10 after 75,000 iterations. Batch size of 32 and 8 were used for an

Encoder \mathcal{E}	AP	AP ₅₀	AP ₇₅	AP _S	AP _M
shared	43.6	69.5	49.8	40.2	63.8
separated	45.6	71.2	52.7	42.2	64.8

Table 5: Comparing shared vs. separated encoders \mathcal{E} in the object detection task.

Method	Batch size	AP	AP ₅₀	AP ₇₅	AP _S	AP _M
DETR [2]	8	33.5	66.3	30.0	30.2	51.1
	16	41.7	70.7	45.4	38.4	61.6
	32	43.5	71.0	49.1	39.8	64.5
	64	41.4	68.5	46.2	37.5	63.0
Ours	8	45.6	71.2	52.7	42.2	64.8
	16	40.1	66.0	44.8	36.2	62.7
	32	38.1	63.7	42.0	34.5	56.9

Table 6: The effect of different batch size on model performance.

original DETR and the DETR with LSP respectively. Training images were augmented using random horizontal flip, random brightness, random contrast, and random Gaussian blur. The training image resolution was set to 160×160 pixels.

A.4 Ablation studies

All proposed model used in this section is DETR with LSP (GCR), with $d = 10^{-4}$. We used the same training schedule as the main experiments.

A.4.1 Choice of encoder \mathcal{E}

We evaluate the necessity of having a different encoder \mathcal{E} for each Transformer decoder by comparing it with the one which has a single shared encoder \mathcal{E} for every Transformer decoder. Table 5 shows that although having multiple sets of s may affect the others, having different \mathcal{E} outperformed a single shared one. Nevertheless, having shared \mathcal{E} still achieved a competitive performance to the DETR baseline.

A.4.2 Batch size

We evaluate the effect of different training batch sizes of the proposed method. Table 6 shows that our method achieved the best performance at the batch size of 8. Surprisingly, the performance dropped drastically as the batch size increases which is opposite from the DETR baseline. The cause for this problem is unclear and is open for future research.

A.5 Convergence speed

Figure 7 shows a training progression plot of LSP (GCR) with different d 's against the DETR baseline. Both methods achieved convergence without much instability. However, we observed slightly worse small object localization performance from the DETR baseline compared to that of learned distance function from LSP.

A.6 Qualitative results

Figure 8 shows a qualitative result of DETR and our method on our generated object detection dataset. The images in the figure are randomly selected from the test set.

B MNIST Point Cloud Autoencoding

In this section, we aim to show that LSP is competitive on the often used set prediction task, namely point cloud autoencoding on the MNIST dataset [12]. We provided a comparison of our method against DSPN [12], and TSPN [1]. In contrast to prior works which use the chamfer loss, we performed comparison based on the Hungarian assignment.

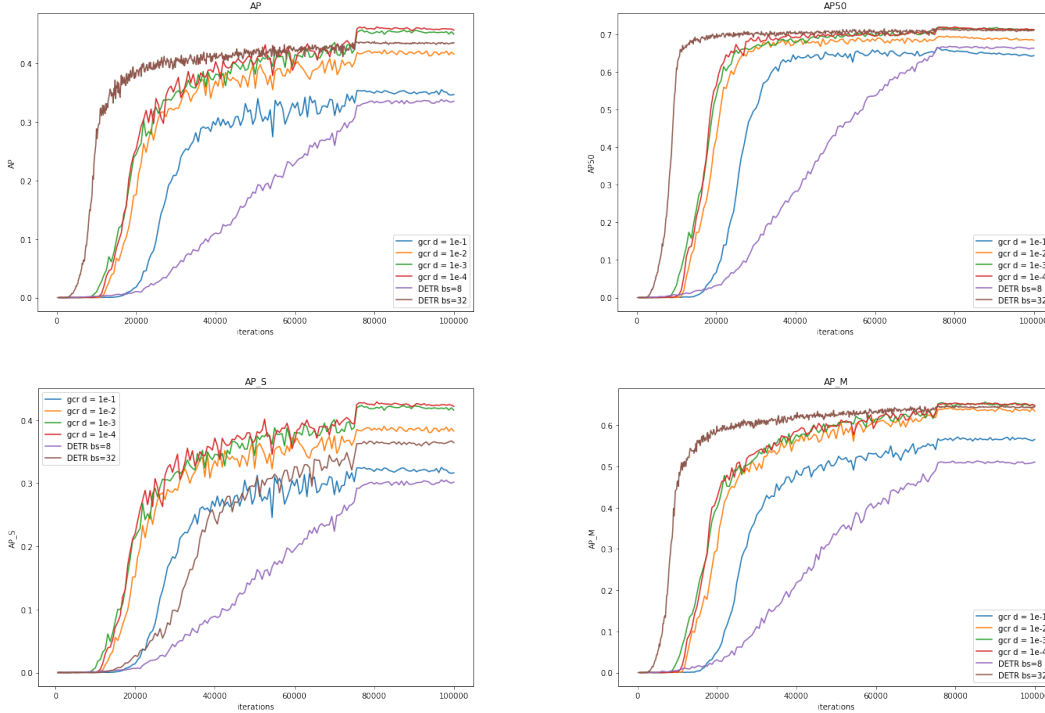


Figure 7: Convergence plot of GCR against the DETR baseline on the object detection task.

B.1 Model

We followed DSPN and TSPN model architectures by using a 3-layer MLP with FSPool [32] for the set encoders. For LSP, we augmented the TSPN architecture with an encoder, which is simply a linear layer from a tuple of $(x, y, \text{presence})$ to a 256-sized vector. For both the TSPN and LSP variants, we used the learned embeddings instead of Gaussian random vectors which were found to work better for Hungarian matching. In addition, both predicted each element’s presence to derive the set cardinality instead of using an MLP to predict the number of set cardinality explicitly.

B.2 Experimental setup

We followed DSPN and TSPN point cloud MNIST dataset, but limited the maximum number of points to 150. All experiments were run for 50 epochs. Each performance number was the minimum of the run. We varied the “hidden dimension” of the encoder for DSPN to scale it up for a comparable parameter count. DSPN used a learning rate of 0.01 (following DSPN, grid searched from $[0.01, 0.001, 0.0001]$) while the others used a learning rate of 0.0001. We used batch normalization in the set encoder for faster convergence.

B.3 Results

Table 7 shows a result of LSP against the DSPN and TSPN baseline. DSPN lags behind both TSPN and LSP by a large margin both quantitatively and qualitatively (Figure 9). DSPN also did not scale up well with wider models. TSPN performed better than LSP in this experiment, yet qualitatively hard to perceive the differences. We want to point out that this is possibly a task where designing a good encoder is harder than designing a good distance metric since each set element is simply $(x, y, \text{presence})$.



(a) DETR



(b) DETR with LSP (GCR, $d = 10^{-4}$)

Figure 8: Qualitative result of DETR and DETR with LSP on our object detection dataset.

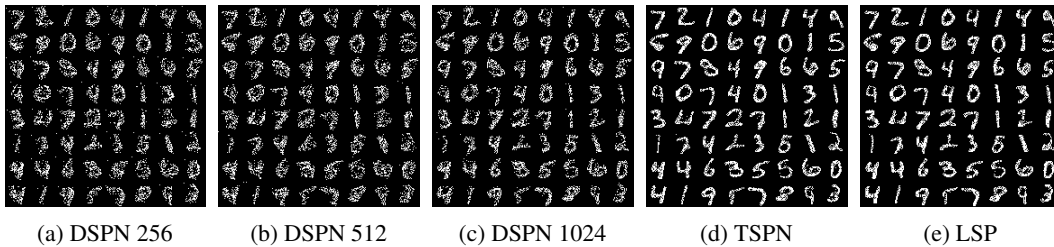


Figure 9: The qualitative comparison between reconstructed images from different models for the point cloud autoencoding task. DSPN’s performed poorly qualitatively compared to TSPN and LSP.

C CLEVR object description generation

C.1 Dataset

We constructed the attribute prediction dataset from images and metadata of CLEVR dataset [23]. The dataset released with 70,000/10,000 train/val scenes. Note that we used the *val* scenes as our test dataset and split the *train* into 60,000/10,000 for training and development purposes. There are seven attributes: shape, color, material, size, left/right, front/back, object count. Four of them

Model	Hidden dim.	#Params (M)	Chamfer L2 distance
DSPN	256	0.14	4.99 \pm 0.37
DSPN	512	0.40	4.52 \pm 0.23
DSPN*	1024	1.33	4.42 \pm 0.01
TSPN	256	1.50	3.78 \pm 0.01
LSP (GCR, $d = 0.001$)	256	1.50	3.90 \pm 0.02

Table 7: The performance comparison for test set of MNIST point cloud autoencoder task. The symbol \pm represents one standard deviation of 3 different random seeds, except DSPN* that runs with 2 seeds.

were obtained from the dataset’s metadata. The additional left/right and front/back attributes were calculated from the object’s pixel-wise quadrant. The object count attribute (≤ 3) is the number of objects that share the same attributes. The object description was generated from these attributes with random corruption. All attributes except *shape* were dropped with 50% chance. Note that the corruption always retained the unambiguity of the description.

C.2 Models

All models used byte-level byte pair encoding same as RoBERTa [33] via Huggingface [34]. We used ResNet-34 [31] pretrained on Imagenet as the backbones for all models.

Concat model is a model where the input was the concatenation of all object descriptions separated by “<sep>” tokens. It has the following architecture.

$$\begin{aligned} \mathbf{feat} &= \text{Conv1}(\text{ResNet34}(x)) \\ \hat{\mathbf{Y}} &= \text{TransformerDecoder}(\mathbf{Y}, \mathbf{feat}) \end{aligned}$$

where TransformerDecoder is a 3-layer Transformer decoder with 256 hidden units and 4 attention head connected to a linear layer for predicting the softmax distribution of the vocabularies, and \mathbf{Y} is supplied as a teacher forcing signal. At the evaluation time, the network was greedily decoded.

Ordered set model predicts an alphabetically sorted list of object descriptions. It has the following architecture.

$$\begin{aligned} \mathbf{feat} &= \text{Conv1}(\text{ResNet34}(x)) \\ \mathbf{R} &= \text{TransformerDecoder}(\mathbf{seed}, \mathbf{feat}) \\ \hat{\mathbf{Y}} &= \mathcal{D}(\mathbf{R} + \mathbf{Y}, \text{repeat}(\mathbf{feat})) \end{aligned}$$

where TransformerDecoder is a 3-layer Transformer decoder with 256 hidden units and 4 attention heads, \mathbf{seed} is fixed sinusoidal vectors as queries for set elements, \mathcal{D} is 3-layer Transformer decoder for generating object descriptions with 256 hidden units and 4 heads connected to a linear layer for predicting the softmax distribution of the vocabularies, and \mathbf{Y} is supplied as a teacher forcing signal. \mathbf{R} is added to the input of \mathcal{D} to dictate the topic about which it generates. The network always predicted K sentences ($K = 10$). We padded the ground truths (N sentences) with blank lines until they have K sentences. At the evaluation time, the network was greedily decoded, and all the blank lines were removed.

LSP model predicts a set of object descriptions. It shared the above architecture with an addition of an encoder \mathcal{E} as follows

$$\mathbf{B} = \text{TransformerDecoder}(\mathbf{Y}, \text{repeat}(\mathbf{feat}))$$

where TransformerDecoder is a 3-layer Transformer decoder with 256 hidden units and 4 attention heads. It performs cross-attention between the ground truth \mathbf{Y} and the image feature \mathbf{feat} . In summary, object descriptions \mathbf{Y} were encoded as \mathbf{B} . Then, the Hungarian algorithm was performed to find the minimum assignment π between \mathbf{R} and \mathbf{B} under Euclidean distance. The decoding step of this network became $\hat{\mathbf{Y}}_{\pi} = \mathcal{D}(\mathbf{R}_{\pi} + \mathbf{Y}, \text{repeat}(\mathbf{feat}))$ following the seq of text model’s notation. Finally, the loss function was calculated directly as $\mathcal{L}_{\text{task}}(\hat{\mathbf{Y}}_{\pi}, \mathbf{Y})$. Like the seq of text model, the ground truths \mathbf{Y} were padded by blank lines to have the total of K sentences. All the blank lines were removed at inference time.

C.3 Training details

The model sizes and other hyperparameters were not heavily tuned. Our goal is to show the improvement from modeling the task as set prediction. **Optimization.** Adam with learning rate 10^{-4} . Batch size 64. The learning rate was reduced by 5 after the validation loss of $\mathcal{L}_{\text{task}}$ is not reducing for 2 epochs. When the learning rate was below 10^{-6} , the training stopped. **Augmentation.** Besides resizing the image to 256×256 , there is no other augmentation.

C.4 Results with standard deviations

We reported averages of three trials. \pm denotes single standard deviation.

Method	Precision	Recall	F1
Concat	0.931 ± 0.02	0.910 ± 0.02	0.920 ± 0.01
Ordered set	0.957 ± 0.01	0.526 ± 0.02	0.679 ± 0.02
LSP (GC)	0.986 ± 0.01	0.972 ± 0.01	0.979 ± 0.01
LSP (GCR, $d = 10^{-1}$)	0.983 ± 0.01	0.970 ± 0.01	0.977 ± 0.01
LSP (GCR, $d = 10^{-2}$)	0.979 ± 0.01	0.957 ± 0.02	0.968 ± 0.01
LSP (GCR, $d = 10^{-3}$)	0.983 ± 0.01	0.975 ± 0.01	0.979 ± 0.01
LSP (GCR, $d = 10^{-4}$)	0.984 ± 0.01	0.973 ± 0.02	0.978 ± 0.02
LSP (GCR, $d = 0$)	0.983 ± 0.01	0.972 ± 0.02	0.978 ± 0.01

C.5 Effect of hyperparameter in asymmetric latent loss

In this section, we studied the effect of hyperparameter β and γ of the asymmetric latent loss ($\mathcal{L}_{\text{latent}}$) by fixing the value of γ to 1 and adjusted the β . The table below shows us that $\beta = 0.1$ yielded the best performance. Nevertheless, given the variances, there was no significant performance change over different β .

Model	Precision	Recall	F1
GC ($\beta = 0$)	0.979 ± 0.01	0.970 ± 0.02	0.975 ± 0.01
GC ($\beta = 0.1$)	0.989 ± 0.01	0.979 ± 0.01	0.984 ± 0.01
GC ($\beta = 0.2$)	0.987 ± 0.01	0.976 ± 0.02	0.982 ± 0.01
GC ($\beta = 0.5$)	0.980 ± 0.01	0.972 ± 0.01	0.976 ± 0.01
GC ($\beta = 1$)	0.983 ± 0.01	0.966 ± 0.02	0.974 ± 0.02

C.6 Effect of GCR's d

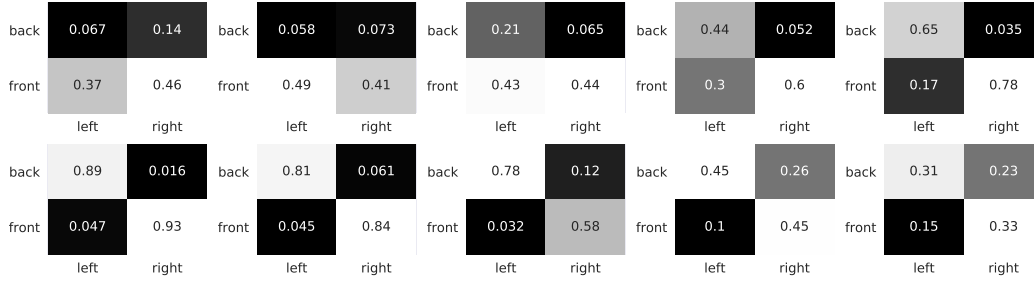
In Section C.4. We saw no real performance differences between GC and GCR and GCR with different d 's. This task represents a non-trivial set prediction yet has clean labels and few set members. Under this scenario, the choice of GC or GCR did not really matter.

C.7 Head specialization

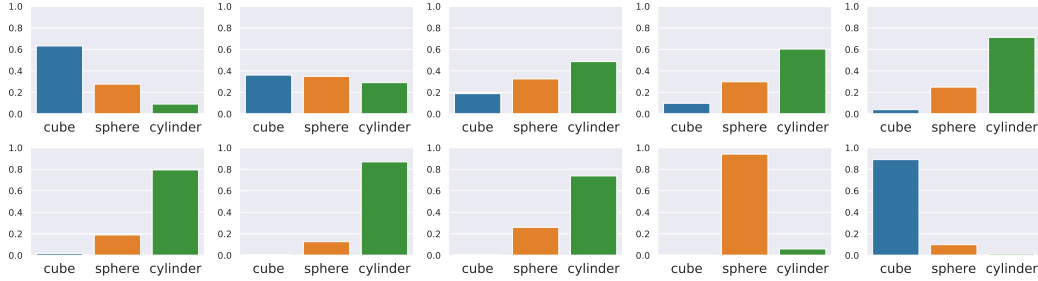
We observed prediction head specialization from the **LSP** model both in location specialization (Figure 10a) and shape specialization (Figure 10b).

C.8 Convergence speed comparison

We depicted the speed of validation metrics over training epochs between Ordered Set, Concat, and LSP (GC) in Figure 11. Two baselines converged faster than LSP but to worse solutions. Since these methods did not converge to solutions of the same quality, it was unfair to compare the convergence time directly. However, at any point in time, LSP was either on par or better with the other methods, showing training stability. Noted that it is not a perfect comparison because the baselines are not set prediction methods.



(a) Specialization in locations



(b) Specialization in shapes

Figure 10: All 10-head specialization of the LSP model on CLEVR object description task.

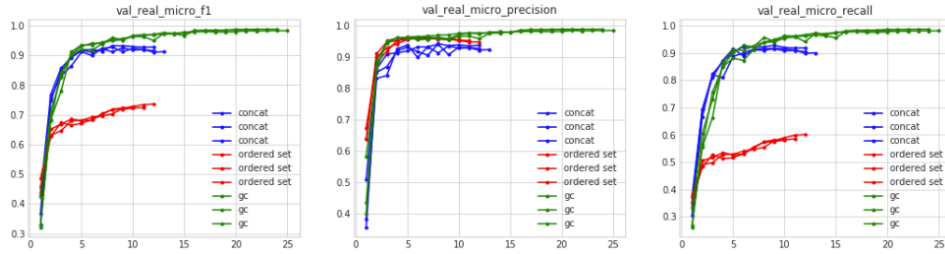


Figure 11: The comparison between validation performance progresses of different methods for CLEVR dataset. Different lines are different random seeds.

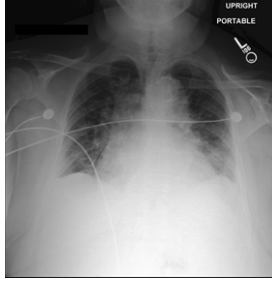
D Chest radiograph report generation

D.1 Dataset

We used MIMIC-CXR dataset [16] containing 377,110 chest x-rays with 227,835 reports from 65,379 patients. We selected only reports with apparent “FINDING” keyword and extracted only the finding section with regular expression to focus on a specific part on the report that is most predictable from a chest x-ray. This resulted in 149,766 reports. From these, we selected only “frontal” images (PA or AP). Finally, we have 160,291 images and 143,778 reports which were split into train/val/test as 112,025/15,994/32,272 images and 100,531/14,391/28,856 reports respectively. We broke a document into sentences with Spacy [35].

D.2 Models

All models used byte-level byte pair encoding same as RoBERTa [33] via Huggingface [34]. We used ResNet-34 [31] pretrained on Imagenet as the backbones for all models.



FINDINGS: In comparison to study performed on of ___ there is new mild pulmonary edema with small bilateral pleural effusions. Lung volumes have decreased with crowding of vasculature. No pneumothorax. Severe cardiomegaly is likely accentuated due to low lung volumes and patient positioning.

IMPRESSION: 1. New mild pulmonary edema with persistent small bilateral pleural effusions. 2. Severe cardiomegaly is likely accentuated due to low lung volumes and patient positioning.

Figure 12: A sample report from MIMIC-CXR dataset [16]. A report usually contains header section, finding section, and impression section. We extracted only the finding section with regular expression (highlighted in yellow).

Concat model is a model where the input was the concatenation of finding sentences in the report separated by “<sep>” tokens. It has the following architecture.

$$\begin{aligned} \text{feat} &= \text{Conv1}(\text{ResNet34}(x)) \\ \text{feat} &= \text{TransformerEncoder}(\text{feat} + \text{2D position encoding}) \\ \hat{\mathbf{Y}} &= \text{TransformerDecoder}(\mathbf{Y}, \text{feat}) \end{aligned}$$

where `TransformerEncoder` is a 3-layer Transformer encoder with 256 hidden units and 4 attention heads, the position encoding is 2D sinusoidal features following [2], `TransformerDecoder` is a 3-layer Transformer decoder with 256 hidden units and 4 attention heads connected to a linear layer for predicting the softmax distribution of the vocabularies, and \mathbf{Y} is supplied as a teacher forcing signal. At the evaluation time, the network was greedily decoded.

Ordered set model predicts an alphabetically sorted list of finding sentences. It has the following architecture.

$$\begin{aligned} \text{feat} &= \text{Conv1}(\text{ResNet34}(x)) \\ \text{feat} &= \text{TransformerEncoder}(\text{feat} + \text{2D position encoding}) \\ \mathbf{R} &= \text{TransformerDecoder}(\text{seed}, \text{feat}) \\ \hat{\mathbf{Y}} &= \mathcal{D}(\mathbf{R} + \mathbf{Y}, \text{repeat}(\text{feat})) \end{aligned}$$

where `TransformerEncoder` is a 3-layer Transformer encoder with 256 hidden units and 4 attention heads, the position encoding is 2D sinusoidal features following [2], `TransformerDecoder` is a 3-layer Transformer decoder with 256 hidden units and 4 attention heads, `seed` is fixed sinusoidal vectors as queries for set elements, \mathcal{D} is 3-layer Transformer decoder for generating object descriptions with 256 hidden units and 4 heads connected to a linear layer for predicting the softmax distribution of the vocabularies, and \mathbf{Y} is supplied as a teacher forcing signal. \mathbf{R} is *added* to the input of \mathcal{D} to dictate the topic about which it generates. The network always predicted K sentences ($K = 10$). We padded the ground truths (N sentences) with blank lines until they have K sentences. The losses on these padded blank lines were *zero* out to allow the model to always predict K non-blank sentences. At the evaluation time, the network was greedily decoded.

LSP model predicts a set of finding sentences. It shared the above architecture with an addition of an encoder \mathcal{E} as follows

$$\mathbf{B} = \text{TransformerDecoder}(\mathbf{Y}, \text{repeat}(\text{feat}))$$

where `TransformerDecoder` is a 3-layer Transformer decoder with 256 hidden units and 4 attention heads. It performs cross-attention between the ground truth \mathbf{Y} and the image feature `feat`. In summary, object descriptions \mathbf{Y} were encoded as \mathbf{B} . Then, the Hungarian algorithm was performed to find the minimum assignment π between \mathbf{R} and \mathbf{B} under Euclidean distance. The decoding step of this network became $\hat{\mathbf{Y}}_{\pi} = \mathcal{D}(\mathbf{R}_{\pi} + \mathbf{Y}, \text{repeat}(\text{feat}))$ following the sequence of text model’s notation. Finally, the loss function was calculated directly as $\mathcal{L}_{\text{task}}(\hat{\mathbf{Y}}_{\pi}, \mathbf{Y})$. This model always predicted $K = 10$ sentences. However, we *did not* pad \mathbf{Y} with blank sentences. That is the Hungarian match of N out of K sentences. We found this to work better than that with blank sentence padding.

RM+MCLM [26] also takes the concatenation of finding sentences in the report. We used the official implementation from <https://github.com/cuhksz-nlp/R2Gen>. We ran it with our dataset using batch size 64 and our augmentation scheme. We kept the rest of the settings original apart from that of the other models.

D.3 Training details

The model sizes and other hyperparameters were not heavily tuned. Our goal is to show the improvement from modeling the task as set prediction. **Optimization.** Adam with learning rate 10^{-4} . Batch size 64. The learning rate was reduced by 5 after the validation loss of $\mathcal{L}_{\text{task}}$ is not reducing for 2 epochs. When the learning rate was below 10^{-6} , the training stopped. **Augmentation.** Random rotation up to 90 degrees, random horizontal flip, random contrast and brightness in range (0.5, 1.5), random crop with random size in range (0.7, 1.0), and random aspect ratio from 4:3 to 3:4. Images were resized to 256×256 pixels.

D.4 Results with standard deviations

We reported averages of three trials except * which was run once. \pm denotes single standard deviation.

Method	micro avg. BLEU		
	$\hat{y} \rightarrow y$	$y \rightarrow \hat{y}$	hmean
RM+MCLN [26]*	16.7	15.6	16.2
Concat	16.7 ± 0.1	15.2 ± 0.1	15.9 ± 0.1
Ordered set	20.2 ± 0.2	18.5 ± 0.2	19.3 ± 0.2
LSP (GC)	17.9 ± 0.5	24.2 ± 0.5	20.6 ± 0.3
LSP (GCR, $d = 10^{-1}$)	19.1 ± 0.5	24.5 ± 0.5	21.5 ± 0.5
LSP (GCR, $d = 10^{-2}$)	18.7 ± 0.8	24.6 ± 0.5	21.3 ± 0.7
LSP (GCR, $d = 10^{-3}$)	19.1 ± 0.2	24.6 ± 0.1	21.5 ± 0.1
LSP (GCR, $d = 10^{-4}$)	18.6 ± 0.7	24.3 ± 0.2	21.1 ± 0.5
LSP (GCR, $d = 0$)	18.9 ± 0.8	24.7 ± 0.4	21.4 ± 0.7

D.5 Effect of GCR's d


In Section D.4, we observed high $y \rightarrow \hat{y}$ across GC and GCR with all d 's. The difference was in $\hat{y} \rightarrow y$ where GCR excelled. We observed no real differences between different d 's of GCR. We began to saw superior performances from GCR over GC in this difficult image captioning task with noisy labels.

D.6 Prediction samples


We selected a few reports to qualitatively evaluate the models. Duplicate sentences were dropped in the post process. We reordered and color-coded sentences related to abnormalities by hand to the best of our non-expert ability. Finally, we asked a radiologist for opinions on these reports and **highlighted areas** that significantly deviated from the ground truth. The radiologist was also asked to select the best model from each example. For example number 1-3 the LSP is preferred, while the best choice for number 4 is inconclusive.

In general, there are abnormalities such as opacity, consolidation, and nodule that were not frequently recognized by the models. The models were quite able to recognize abnormalities like pleural effusions, cardiomegaly, and devices including lines and pacemaker. However, we observed better prediction coverage from the LSP model than its peers especially on negative abnormality predictions. In our opinion, this task is not yet solved. It is questionable how much more we can expect models to learn from this noisy and partially predictable dataset.


Report generation example 1 (curated)

X-ray	Ground truth	LSP	Ordered set	Concat
	frontal and lateral views of the chest were obtained. there are streaky linear opacities at the lung bases which are likely due to atelectasis with chronic changes. no definite focal consolidation is seen. there is no pleural effusion or pneumothorax. the aorta is calcified and tortuous. the cardiac silhouette is top normal to mildly enlarged. dual-lead left-sided pacemaker is seen with leads in the expected positions of the right atrium and right ventricle. chronic-appearing rib deformities on the right is again seen.	frontal and lateral views of the chest were obtained. there is a small left pleural effusion with overlying atelectasis. there is no focal consolidation, pleural effusion or pneumothorax. there is no pleural effusion or pneumothorax. the aorta is calcified and tortuous. the heart is mildly enlarged. a left-sided pacemaker is seen with leads in the expected position of the right atrium and right ventricle. the patient is status post median sternotomy and cabg. the lungs are otherwise clear.	pa and lateral views of the chest provided. there is no pneumothorax. no pneumothorax is seen. the heart is mildly enlarged. the patient is status post median sternotomy and cabg. the lungs are hyperinflated with flattening of the diaphragms, suggesting chronic obstructive pulmonary disease.	pa and lateral views of the chest provided. no large effusion or pneumothorax. the heart is mildly enlarged. midline sternotomy wires and mediastinal clips are noted. there is a left chest wall pacer device with lead tips extending to the region of the right atrium and right ventricle. there is mild pulmonary edema. no convincing signs of pneumonia. bony structures are intact. mediastinal contour is normal. no free air below the right hemidiaphragm.


Report generation example 2 (curated)

X-ray	Ground truth	LSP	Ordered set	Concat
	the patient is status post coronary artery bypass graft surgery. the heart is mildly enlarged. there is a large hiatal hernia with an air-fluid level. otherwise, the mediastinal and hilar contours are unremarkable. the lungs appear clear. the chest is hyperinflated. there is no pleural effusion or pneumothorax. bony structures are unremarkable.	the patient is status post median sternotomy and cabg. the heart is mildly enlarged. there is a large hiatal hernia with an air-fluid level. the mediastinal and hilar contours are unremarkable. the lungs are clear. there is no pleural effusion or pneumothorax. there is no focal consolidation, pleural effusion or pneumothorax. no acute osseous abnormalities. frontal and lateral views of the chest were obtained.	the patient is status post median sternotomy and cabg. the cardiac silhouette is top normal to mildly enlarged. the lungs are clear. no acute osseous abnormalities are detected. pa and lateral views of the chest were provided. no free air below the right hemidiaphragm.	midline sternotomy wires and mediastinal clips are again noted. the cardiomeastinal silhouette is normal. there is a large retrocardiac opacity containing an air-fluid level concerning for a large hiatal hernia. there is no focal consolidation, effusion, or pneumothorax. imaged osseous structures are intact. pa and lateral views of the chest provided. no free air below the right hemidiaphragm is seen.

Report generation example 3 (random)

X-ray	Ground truth	LSP	Ordered set	Concat
	pa and lateral views of the chest provided. left chest wall dual lead pacer is again noted with leads extending the region the right atrium and right ventricle. the heart is mildly enlarged. no convincing signs of edema or pneumonia. mediastinal contour is normal. bony structures are intact. mild hilar congestion is suspected.	ap and lateral views of the chest. left-sided dual-chamber pacemaker device is noted with leads terminating in the right atrium and right ventricle. the heart is mildly enlarged. the mediastinal and hilar contours are unchanged. no acute osseous abnormalities. there is mild pulmonary vascular congestion. no pleural effusion or pneumothorax is seen. there is no pleural effusion or pneumothorax. there is no focal consolidation, pleural effusion or pneumothorax. the lungs are clear.	left-sided dual-chamber pacemaker device is noted with leads terminating in the right atrium and right ventricle. the heart is mildly enlarged. the cardiac silhouette is mildly enlarged. no acute osseous abnormality is detected. the lungs are clear without focal consolidation or effusion. the lungs are clear without focal consolidation, pleural effusion or pneumothorax. no free air below the right hemidiaphragm.	left-sided dual-chamber pacemaker device is noted with leads terminating in the right atrium and right ventricle. mild cardiomegaly is unchanged. there is no pulmonary edema. the mediastinal and hilar contours are similar. there are no acute osseous abnormalities. no focal consolidation, pleural effusion or pneumothorax is present.

Report generation example 4 (random)

X-ray	Ground truth	LSP	Ordered set	Concat
	a portable view of the chest demonstrates continous improved aeration of the left mid and upper lung. a moderate left pleural effusion remains. small right pleural effusion is stable. the right lung is grossly clear. a left pigtail catheter and right picc are unchanged in position. there is no pneumothorax.	as compared to the previous radiograph, the patient has been extubated. there is a large left pleural effusion with associated atelectasis. the right lung is essentially clear. the tip of the right picc line projects over the mid svc. the monitoring and support devices are constant. there is no pneumothorax. no pneumothorax. the size of the cardiac silhouette is unchanged. the heart is enlarged. there is mild pulmonary edema.	as compared to the previous radiograph, the patient has received a new right-sided picc line. there is a right-sided picc line with distal lead tip in the distal svc. there is no pneumothorax.	the pre-existing left pleural effusion has slightly increased in extent. as compared to the previous radiograph, the patient has received a right-sided picc line. the course of the line is unremarkable, the tip of the line projects over the mid svc. there is no evidence of complications, notably no pneumothorax. unchanged appearance of the right lung. unchanged appearance of the cardiac silhouette.

E Proof of convergence

We begin by showing the proof for the simplest situation where **gradient cloning** is used without the **rejection** mechanism and β is set to 0. This will illustrate the intuitions behind the convergence of latent set prediction, which can be modified to accommodate the **rejection** mechanism and other values of β later on.

E.1 Recall of the setup

In the deep set prediction setting, we wish to train a set predictor which generates latent vectors $s_i \in \mathbb{R}^C$ to match a set of targets y_i in the output space, which are designated by the generated guiding vectors $g_i \in \mathbb{R}^C$. With a permutation π , we introduce the main task's loss

$$\mathcal{L}_{\text{task}} = \sum_i l_{\text{task}}(s_{\pi(i)}, y_i).$$

Note that the permutation π can change over the course of model training and a **switch** (Figure 2) is said to occur when $\pi^{(t+1)} \neq \pi^{(t)}$. To discourage **switch**, we define the following latent loss with squared Euclidean distance to drive g_i toward $s_{\pi(i)}$

$$\mathcal{L}_{\text{latent}} = \sum_i l_{\text{latent}}(s_{\pi(i)}, g_i) = \sum_i \frac{1}{2} \|s_{\pi(i)} - g_i\|_2^2.$$

The training at each time point $t + 1$ consists of two steps. First, the permutation π is updated according to the values of $s_{\pi(i)}^{(t)}$'s and $g_i^{(t)}$'s from the previous time point

$$\pi^{(t+1)} = \operatorname{argmin}_{\pi' \in \mathcal{P}} \sum_i \|s_{\pi'(i)}^{(t)} - g_i^{(t)}\|_2, \text{ where } \mathcal{P} \text{ is the set of all permutations.} \quad (4)$$

Then, the values of $s_{\pi(i)}$'s and g_i 's are updated based on the gradients from $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{latent}}$ with step size η and latent loss strength γ

$$\begin{aligned} s_{\pi^{(t+1)}(i)}^{(t+1)} &= s_{\pi^{(t+1)}(i)}^{(t)} - \eta \nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) \\ g_i^{(t+1)} &= g_i^{(t)} - \eta \left(\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) + \gamma \nabla_g l_{\text{latent}}(s_{\pi^{(t+1)}(i)}^{(t)}, g_i^{(t)}) \right). \end{aligned} \quad (5)$$

This is the **gradient cloning** technique with a special case of **asymmetric latent loss** (as in (2) where $\beta = 0$).

Even though the set matching step may increase task loss through the switch from $s_{\pi(i)}^{(t)}$ to $s_{\pi(i)}^{(t+1)}$, it turns out that sufficient conditions for the convergence of LSP are the same as those needed in a typical proof of convergence of gradient descent [21]. Namely, we assume that $l_{\text{task}}(\cdot, y)$ is **L-smooth** and satisfies the **Polyak-Lojasiewicz** condition.

The **L-smooth** assumption states that there is a constant $L \in \mathbb{R}^+$ such that for any x and y

$$L \cdot \|x - y\|_2 \geq \|\nabla_s l_{\text{task}}(x, y) - \nabla_s l_{\text{task}}(y, y)\|_2, \quad (6)$$

which also implies (Lemma 1.2 in [36])

$$\langle \nabla_s l_{\text{task}}(x, y), y - x \rangle + \frac{L}{2} \|y - x\|_2^2 \geq l_{\text{task}}(y, y) - l_{\text{task}}(x, y). \quad (7)$$

The **Polyak-Lojasiewicz** condition guarantees that there is a constant $\mu \in \mathbb{R}^+$ such that for any x

$$\frac{1}{2} \|\nabla_s l_{\text{task}}(x, y)\|_2^2 \geq \mu \cdot (l_{\text{task}}(x, y) - l_{\text{task}}(x^*, y)), \quad \text{where } x^* \text{ is where the function reaches its minimum} \quad (8)$$

E.2 Exponential decay of the switch distance

In this section, we show that our training dynamics effectively drive g_i 's toward $s_{\pi(i)}$'s so strongly that the distance between a **switch** at time t decays exponentially.

First, we show that the total distance between respective g 's and s 's decay exponentially.

Lemma E.1. For $0 < \eta\gamma \leq 1$,

$$\sum_i \|s_{\pi^{(t)}(i)}^{(t)} - g_i^{(t)}\|_2 \leq (1 - \eta\gamma)^t \sum_i \|s_{\pi^{(0)}(i)}^{(0)} - g_i^{(0)}\|_2.$$

Proof. From the gradient descent update formula for $s_{\pi^{(t+1)}(i)}^{(t+1)}$ and $g_i^{(t+1)}$ in (5) we have

$$\begin{aligned} s_{\pi^{(t+1)}(i)}^{(t+1)} - g_i^{(t+1)} &= s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} + \eta\gamma \nabla_g l_{\text{latent}}(s_{\pi^{(t+1)}(i)}^{(t)}, g_i^{(t)}) \\ &= s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} - \eta\gamma \left(s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right), \quad \text{because } l_{\text{latent}} \text{ is the squared Euclidean distance} \\ &= (1 - \eta\gamma) \left(s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right) \end{aligned}$$

and so

$$\begin{aligned} \sum_i \|s_{\pi^{(t+1)}(i)}^{(t+1)} - g_i^{(t+1)}\|_2 &= (1 - \eta\gamma) \sum_i \|s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)}\|_2 \\ &\leq (1 - \eta\gamma) \sum_i \|s_{\pi^{(t)}(i)}^{(t)} - g_i^{(t)}\|_2, \quad \text{by the definition of } \pi^{(t+1)} \text{ in (4)} \end{aligned}$$

The desired result then follows through induction. \square

An important implication of the above behavior is that the distance $\|s_{\pi^{(t+1)}(i)}^{(t)} - s_{\pi^{(t)}(i)}^{(t)}\|_2$ due to a **switch** also decays exponentially.

Theorem E.2.

$$\|s_{\pi^{(t+1)}(i)}^{(t)} - s_{\pi^{(t)}(i)}^{(t)}\|_2 \leq 2(1 - \eta\gamma)^t \sum_i \|s_{\pi^{(0)}(i)}^{(0)} - g_i^{(0)}\|_2$$

Proof. By triangle inequality, we have

$$\begin{aligned} \|s_{\pi^{(t+1)}(i)}^{(t)} - s_{\pi^{(t)}(i)}^{(t)}\|_2 &\leq \|s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)}\|_2 + \|s_{\pi^{(t)}(i)}^{(t)} - g_i^{(t)}\|_2 \\ &\leq \sum_i \|s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)}\|_2 + \sum_i \|s_{\pi^{(t)}(i)}^{(t)} - g_i^{(t)}\|_2 \\ &\leq 2 \sum_i \|s_{\pi^{(t)}(i)}^{(t)} - g_i^{(t)}\|_2, \quad \text{by the definition of } \pi^{(t+1)} \text{ in (4)} \end{aligned}$$

and the desired result follows immediately from Lemma E.1. \square

E.3 Convergence of the main task's loss function

For convenience, we introduce the following notations:

$$\begin{aligned}\alpha &= 1 - \eta\gamma, \\ \mathcal{B} &= L\eta \left(1 - \frac{L\eta}{2}\right), \\ \mathcal{C} &= 4L \left(\frac{1}{2} + \frac{3}{\mathcal{B}} + \frac{9L}{2\mu\mathcal{B}^2}\right) \left(\sum_i \|\mathbf{s}_{\pi^{(0)}(i)}^{(0)} - g^{(0)}\|_2\right)^2, \\ \delta &= 1 - \frac{\mu}{9L}(2\mathcal{B}^3 - 13\mathcal{B}^2 + 12\mathcal{B}), \\ d_t &= \|\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} - \mathbf{s}_{\pi^{(t)}(i)}^{(t)}\|_2, \\ x_t &= l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i) - l_{\text{task}}(\mathbf{s}_i^*, \mathbf{y}_i),\end{aligned}$$

where α , \mathcal{B} , \mathcal{C} , and δ are constants and d_t and x_t are sequences of positive real numbers. If η is chosen to be smaller than $\min\{1/\gamma, 2/L\}$, then $\alpha \in (0, 1)$ and $\mathcal{B}, \mathcal{C} > 0$. With AM-GM inequality, it is clear that $\mathcal{B} \leq 1/2$. In fact, \mathcal{B} can be made arbitrarily small from the choice of η . Furthermore, since the cubic polynomial $p(x) = 2x^3 - 13x^2 + 12x$ is increasing on $[0, 1/2]$, we can set η to make \mathcal{B} small enough that $p(\mathcal{B}) < 9L/\mu$ and ensure that $\delta \in (0, 1)$.

In the proofs below, we rely on the properties of these constants, especially that $\alpha, \delta \in (0, 1)$, to show the convergence of LSP.

Lemma E.3. *If $l_{\text{task}}(\cdot, \mathbf{y})$ is L -smooth, we have the following inequalities*

$$\|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i)\|_2 \geq \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i)\|_2 - Ld_t, \quad (9)$$

$$l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)}, \mathbf{y}_i) - l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i) \leq \frac{L}{2}d_t^2 + \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i)\|_2 \cdot d_t - \frac{\mathcal{B}}{L} \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i)\|_2^2. \quad (10)$$

Proof. From L-smoothness (6), we have

$$\begin{aligned}L \cdot \|\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} - \mathbf{s}_{\pi^{(t)}(i)}^{(t)}\|_2 &\geq \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i) - \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i)\|_2 \\ &\geq \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i)\|_2 - \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i)\|_2\end{aligned}$$

by the triangle inequality. This proves the first inequality.

For the second inequality, we apply L-smoothness (7) with $x = \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}$ and $y = \mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)}$ and obtain

$$\begin{aligned}l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)}, \mathbf{y}_i) - l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i) &\leq \langle \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i), \mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)} - \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} \rangle + \frac{L}{2} \cdot \|\mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)} - \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}\|_2^2 \\ &= \langle \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i), -\eta \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i) \rangle + \frac{L}{2} \|\eta \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i)\|_2^2 \\ &= -\eta \left(1 - \frac{L\eta}{2}\right) \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i)\|_2^2 \\ &= -\frac{\mathcal{B}}{L} \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i)\|_2^2.\end{aligned}$$

We also consider (7) when $x = \mathbf{s}_{\pi^{(t)}(i)}^{(t)}$ and $y = \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}$

$$\begin{aligned}l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i) - l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i) &\leq \langle \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i), \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} - \mathbf{s}_{\pi^{(t)}(i)}^{(t)} \rangle + \frac{L}{2} \cdot \|\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} - \mathbf{s}_{\pi^{(t)}(i)}^{(t)}\|_2^2 \\ &\leq \|\nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i)\|_2 \cdot d_t + \frac{L}{2}d_t^2,\end{aligned}$$

where we use Cauchy-Schwarz inequality on the first term. Adding the two inequalities together gives us the desired result. \square

We are now ready to prove the main Theorem 4.1. With the above notation, we rewrite the inequality as

$$x_{t+1} = l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_i^*, y_i) \leq \begin{cases} C \alpha^{2t}, & \text{if } \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\| \leq \frac{3Ld_t}{\mathcal{B}} \\ \delta x_t, & \text{otherwise} \end{cases}$$

Proof of Theorem 4.1. In the first case, we assume $\|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 \leq \frac{3Ld_t}{\mathcal{B}}$. From the Polyak-Lojasiewicz condition (8), we have

$$l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) - l_{\text{task}}(s_i^*, y_i) \leq \frac{1}{2\mu} \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2^2 \leq \frac{9L^2}{2\mu\mathcal{B}^2} d_t^2.$$

From (10) in Lemma E.3, we also have

$$\begin{aligned} l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) &\leq \frac{L}{2} d_t^2 + \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 \cdot d_t - \frac{\mathcal{B}}{L} \|\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i)\|_2^2 \\ &\leq \frac{L}{2} d_t^2 + \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 \cdot d_t \\ &\leq \frac{L}{2} d_t^2 + \frac{3L}{\mathcal{B}} d_t^2. \end{aligned}$$

We combine above inequalities and Theorem E.2 to conclude that

$$\begin{aligned} x_{t+1} &\leq L \left(\frac{1}{2} + \frac{3}{\mathcal{B}} + \frac{9L}{2\mu\mathcal{B}^2} \right) d_t^2 \\ &\leq L \left(\frac{1}{2} + \frac{3}{\mathcal{B}} + \frac{9L}{2\mu\mathcal{B}^2} \right) \left(2(1 - \eta\gamma)^t \sum_i \|s_{\pi^{(0)}(i)}^{(0)} - g^{(0)}\|_2 \right)^2 = C \alpha^{2t} \end{aligned}$$

In the second case, we assume $\|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 > \frac{3Ld_t}{\mathcal{B}}$. Notice that \mathcal{B} is chosen to be small and in particular $1/\mathcal{B} > 2$. From (9) in Lemma E.3, this implies

$$\|\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i)\|_2 \geq \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 - Ld_t > 5Ld_t \geq 0.$$

We now consider (10) from Lemma E.3

$$\begin{aligned} l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) &\leq \frac{L}{2} d_t^2 + \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 \cdot d_t - \frac{\mathcal{B}}{L} \|\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i)\|_2^2 \\ &\leq \frac{L}{2} d_t^2 + \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 \cdot d_t - \frac{\mathcal{B}}{L} \left(\|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 - Ld_t \right)^2 \\ &= \left(\frac{L}{2} - \mathcal{B}L \right) d_t^2 + (1 + 2\mathcal{B}) \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 \cdot d_t - \frac{\mathcal{B}}{L} \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2^2. \end{aligned}$$

Next, we use our assumption that $\mathcal{B} < 1/2$ and $d_t < \frac{\mathcal{B}}{3L} \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2$ to obtain

$$\begin{aligned} l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) &< \left(\left(\frac{L}{2} - \mathcal{B}L \right) \frac{\mathcal{B}^2}{9L^2} + (1 + 2\mathcal{B}) \frac{\mathcal{B}}{3L} - \frac{\mathcal{B}}{L} \right) \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2^2 \\ &= -\frac{1}{18L} (2\mathcal{B}^3 - 13\mathcal{B}^2 + 12\mathcal{B}) \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2^2 \\ &\leq -\frac{\mu}{9L} (2\mathcal{B}^3 - 13\mathcal{B}^2 + 12\mathcal{B}) \left(l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) - l_{\text{task}}(s_i^*, y_i) \right) \\ &= (\delta - 1) \left(l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) - l_{\text{task}}(s_i^*, y_i) \right) \end{aligned}$$

where we use the Polyak-Lojasiewicz condition (8) and the fact that $p(x) = 2x^3 - 13x^2 + 12x$ is positive on $(0, 1/2]$. Hence $x_{t+1} - x_t < (\delta - 1)x_t$ or $x_{t+1} < \delta x_t$ as desired. \square

Theorem E.4. (Convergence of latent set prediction) If the main task's loss function, $l_{\text{task}}(\cdot, y)$ is **L-smooth** and satisfies the **Polyak-Lojasiewicz** condition, and η is sufficiently small, then the training dynamics described in (5) guarantees its convergence at a linear rate

$$x_t = l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) - l_{\text{task}}(s_i^*, y_i) \leq C_0 \epsilon^{t-1},$$

where $C_0 := \max\{C, \delta x_0\}$ and $\epsilon := \max\{\alpha^2, \delta\}$.

Proof. This is a consequence of Theorem 4.1. It is easy to check when $t = 1$ and we will proceed by induction.

Suppose that $x_t \leq C_0 \epsilon^{t-1}$ and consider two cases. If $\|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\| \leq \frac{3Ld_t}{B}$, we have

$$x_{t+1} \leq C \alpha^{2t} \leq C_0 \epsilon^t.$$

If $\|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\| > \frac{3Ld_t}{B}$, we have

$$x_{t+1} \leq \delta x_t \leq C_0 \delta \epsilon^{t-1} \leq C_0 \epsilon^t.$$

Therefore x_t is bounded above by an exponential decay. This implies that the main task's loss function converges at least at a linear rate $\epsilon < 1$. This finishes the proof. \square

E.4 Impact of $\beta > 0$

Setting $\beta > 0$ adds a new term to the update of $s_{\pi^{(t+1)}(i)}^{(t+1)}$ in (5)

$$\begin{aligned} s_{\pi^{(t+1)}(i)}^{(t+1)} &= s_{\pi^{(t+1)}(i)}^{(t)} - \eta \left(\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) + \beta \nabla_s l_{\text{latent}}(s_{\pi^{(t+1)}(i)}^{(t)}, g_i^{(t)}) \right) \\ g_i^{(t+1)} &= g_i^{(t)} - \eta \left(\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) + \gamma \nabla_g l_{\text{latent}}(s_{\pi^{(t+1)}(i)}^{(t)}, g_i^{(t)}) \right). \end{aligned} \quad (11)$$

which only slightly change the algebra inside the proof of Lemma E.1 as follows

$$\begin{aligned} s_{\pi^{(t+1)}(i)}^{(t+1)} - g_i^{(t+1)} &= s_{\pi^{(t+1)}(i)}^{(t)} - \eta \beta \nabla_s l_{\text{latent}}(s_{\pi^{(t+1)}(i)}^{(t)}, g_i^{(t)}) - g_i^{(t)} + \eta \gamma \nabla_g l_{\text{latent}}(s_{\pi^{(t+1)}(i)}^{(t)}, g_i^{(t)}) \\ &= s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} - \eta(\beta + \gamma) \left(s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right) \\ &= (1 - \eta(\beta + \gamma)) \left(s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right) \end{aligned}$$

Hence, we can derive stronger exponential distance decays in Lemma E.1 and Theorem E.2

Lemma E.5. For $0 < \eta\gamma \leq 1$,

$$\sum_i \|s_{\pi^{(t)}(i)}^{(t)} - g_i^{(t)}\|_2 \leq (1 - \eta(\beta + \gamma))^t \sum_i \|s_{\pi^{(0)}(i)}^{(0)} - g^{(0)}\|_2.$$

Theorem E.6.

$$d_t = \|s_{\pi^{(t+1)}(i)}^{(t)} - s_{\pi^{(t)}(i)}^{(t)}\|_2 \leq 2(1 - \eta(\beta + \gamma))^t \sum_i \|s_{\pi^{(0)}(i)}^{(0)} - g^{(0)}\|_2$$

Next, we note that this also introduces an additional term to the Equation (10) in Lemma E.3 because the derivation of Equation (10) contains the dot product $\langle \nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i), s_{\pi^{(t+1)}(i)}^{(t+1)} - s_{\pi^{(t+1)}(i)}^{(t)} \rangle$. This dot product now contains an additional term $-\eta\beta \langle \nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i), \nabla_s l_{\text{latent}}(s_{\pi^{(t+1)}(i)}^{(t)}, g_i^{(t)}) \rangle$ which can be bounded from above by $\eta\beta \left\| \nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) \right\|_2 \cdot \left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2$.

Since the rest of the algebra in the proof of Lemma E.3 remains the same, we can readily revise the result of Equation (10) as follows

Lemma E.7. *If $l_{\text{task}}(\cdot, y)$ is L -smooth, we have the following inequalities*

$$l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \leq \frac{L}{2} d_t^2 + \|\nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i)\|_2 \cdot d_t - \frac{\mathcal{B}}{L} \|\nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i)\|_2^2 + \eta\beta \left\| \nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) \right\|_2 \cdot \left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2.$$

As an auxiliary result, it is clear that

$$\begin{aligned} \left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2 &\leq \sum_i \left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2 \\ &\leq \sum_i \left\| s_{\pi^{(t)}(i)}^{(t)} - g_i^{(t)} \right\|_2, \text{ by the definition of } \pi^{(t+1)} \text{ in (4)} \\ &\leq (1 - \eta(\beta + \gamma))^t \sum_i \left\| s_{\pi^{(0)}(i)}^{(0)} - g^{(0)} \right\|_2, \text{ by Lemma E.5} \end{aligned} \quad (12)$$

For convenience, we also introduce some new notations

$$\begin{aligned} \omega &= 1 - \eta(\beta + \gamma), \\ \mathcal{D} &= \sum_i \left\| s_{\pi^{(0)}(i)}^{(0)} - g^{(0)} \right\|_2, \end{aligned}$$

where $\omega \in (0, 1)$ given appropriate choices of β , η , and γ . This also implies that $\eta\beta \in (0, 1)$.

Now, we are ready to revise the proof of the main Theorem 4.1 to incorporate the effect of $\beta > 0$. First, we note that the second case of the proof where $\left\| \nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \right\|_2 > \frac{3Ld_t}{\mathcal{B}}$ is when the new $\left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2$ term interferes with our ability to manipulate the algebra to apply the **Polyak-Lojasiewicz** condition. This is mainly because the condition $\left\| \nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \right\|_2 > \frac{3Ld_t}{\mathcal{B}}$ does not tell us much about the value of $\left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2$. However, as both d_t and $\left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2$ are bounded above by constant factors of ω^t (Theorem E.6), we can address this issue by changing the threshold on $\left\| \nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \right\|_2$ for dividing the proof into two cases from $\frac{3Ld_t}{\mathcal{B}}$, which uses d_t as reference, to $\frac{6L\mathcal{D}}{\mathcal{B}}\omega^t$, which uses ω^t as reference.

For the first case where $\left\| \nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \right\|_2 \leq \frac{6L\mathcal{D}}{\mathcal{B}}\omega^t$, we have

$$l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) - l_{\text{task}}(s_i^*, y_i) \leq \frac{1}{2\mu} \left\| \nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \right\|_2^2 \leq \frac{36L^2\mathcal{D}^2}{\mathcal{B}^2} \omega^{2t}.$$

from the **Polyak-Lojasiewicz** condition, and

$$\begin{aligned} l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) &\leq \frac{L}{2} d_t^2 + \frac{6LD}{\mathcal{B}} \omega^t d_t + \eta\beta \left\| \nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) \right\|_2 \cdot \left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2 \\ &\leq \frac{L}{2} d_t^2 + \frac{6LD}{\mathcal{B}} \omega^t d_t + \frac{6LD\eta\beta}{\mathcal{B}} \omega^t \left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2. \end{aligned}$$

from Lemma E.7. Since both d_t and $\left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2$ are bounded above by constant factors ω^t (Theorem E.6), all terms in the above inequalities are bounded above by constant factors of ω^{2t} and the desired result follows.

For the second case where $\left\| \nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \right\|_2 > \frac{6L\mathcal{D}}{\mathcal{B}}\omega^t$, by applying Theorem E.6, we can show that $\left\| \nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \right\|_2 > \frac{3Ld_t}{\mathcal{B}}$ as in the original proof. Hence, we can follow the same algebraic manipulations to derive

$$\begin{aligned} l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t+1)}, y_i) - l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) &< -\frac{1}{18L} (2\mathcal{B}^3 - 13\mathcal{B}^2 + 12\mathcal{B}) \left\| \nabla_s l_{\text{task}}(s_{\pi^{(t)}(i)}^{(t)}, y_i) \right\|_2^2 \\ &\quad + \eta\beta \left\| \nabla_s l_{\text{task}}(s_{\pi^{(t+1)}(i)}^{(t)}, y_i) \right\|_2 \cdot \left\| s_{\pi^{(t+1)}(i)}^{(t)} - g_i^{(t)} \right\|_2 \end{aligned}$$

From (12) and the condition of this second case, we can bound the last term from above by the gradient of task loss

$$\left\| \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} - \mathbf{g}_i^{(t)} \right\|_2 \leq \mathcal{D}\omega^t < \frac{\mathcal{B}}{6L} \left\| \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i) \right\|_2$$

This yields

$$l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)}, \mathbf{y}_i) - l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i) < -\frac{1}{18L} (2\mathcal{B}^3 - 13\mathcal{B}^2 + 12\mathcal{B} - 3\eta\beta\mathcal{B}) \left\| \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t)}(i)}^{(t)}, \mathbf{y}_i) \right\|_2^2$$

Because $\eta\beta \in (0, 1)$, we can show that for $x \in (0, 1/2]$, the polynomial $p(x) = 2x^3 - 13x^2 + 12x - 3\eta\beta x$ is strictly greater than $q(x) = 2x^3 - 13x^2 + 9x$ which is always positive in this interval. Hence, we can follow the original proof to get the desired result.

E.5 Impact of gradient cloning with rejection

The rejection mechanism adds complexity to the variable update equation (5) by modifying the task loss gradient with

$$\nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task, rejected}} = \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task}} - \frac{\nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \cdot \langle \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task}}, \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \rangle}{\left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \right\|_2^2}$$

on either the update of $\mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)}$ or $\mathbf{g}_i^{(t+1)}$ depending on which variable is leading (Figure 3).

Since gradient rejection is activated when $\left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} - \nabla_{\mathbf{g}} \mathcal{L}_{\text{latent}} \right\|_2 > d \left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task}} \right\|_2$, we can derive an upper bound for this modification term as follows

$$\begin{aligned} \left\| \frac{\nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \cdot \langle \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task}}, \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \rangle}{\left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \right\|_2^2} \right\|_2 &\leq \left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task}} \right\|_2 \\ &< \frac{1}{d} \left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} - \nabla_{\mathbf{g}} \mathcal{L}_{\text{latent}} \right\|_2 \\ &= \frac{1}{d} (\beta + \gamma) \left\| \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} - \mathbf{g}_i^{(t)} \right\|_2 \end{aligned} \quad (13)$$

The proof of Lemma E.5 can be modified to incorporate this term by

$$\begin{aligned} \left\| \mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)} - \mathbf{g}_i^{(t+1)} \right\|_2 &\leq \omega \left\| \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} - \mathbf{g}_i^{(t)} \right\|_2 + \left\| \frac{\nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \cdot \langle \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task}}, \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \rangle}{\left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \right\|_2^2} \right\|_2 \\ &\leq \left(\omega + \frac{1}{d} (\beta + \gamma) \right) \left\| \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} - \mathbf{g}_i^{(t)} \right\|_2 \end{aligned}$$

where d must satisfy the conditions $d > 1/\eta$ to ensure that the constant factor lies in $(0, 1)$.

Similarly to the impact of setting $\beta > 0$, the modification of the task loss gradient can appear as an extra term in Lemma E.7 through the dot product $\langle \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i), \mathbf{s}_{\pi^{(t+1)}(i)}^{(t+1)} - \mathbf{s}_{\pi^{(t+1)}(i)}^{(t)} \rangle$ if the rejection mechanism is activated and $\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}$ is leading. Interestingly, this extra term is easy to handle as

$$\langle \nabla_{\mathbf{s}} l_{\text{task}}(\mathbf{s}_{\pi^{(t+1)}(i)}^{(t)}, \mathbf{y}_i), -\frac{\nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \cdot \langle \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task}}, \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \rangle}{\left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \right\|_2^2} \rangle = -\frac{\langle \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{task}}, \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \rangle^2}{\left\| \nabla_{\mathbf{s}_{\pi}} \mathcal{L}_{\text{latent}} \right\|_2^2}$$

which is strictly negative. Hence, even if the rejection mechanism is activated, the result of the Lemma E.7 still holds in its current form. This implies that the proof of the main theorem also holds.

F Computational resources

- A typical training time of models on the MNIST point cloud autoencoding task was around 2.5 GPU hours on an RTX 2080Ti.

- A typical training time of models on the CLEVR object description generation task was around 2 GPU hours on a V100.
- A typical training time of models on the chest radiograph report generation task was around 5 GPU hours on an A100.
- A typical training time of models on the object detection task was around 8 GPU hours on an RTX 3090 (batch size 8).