第1章　提案手法

第 3 章では，LLM を用いたシーングラフ生成の手法について述べる．

1.1　使用モデル

1.1.1　Grounding DINO

Grounding DINO[?] は，Liu らによって提案されたオープンセット物体検出モデルである．従来の物体検出モデルは COCO などのデータセットに含まれる固定カテゴリしか検出できないという制約があった．これに対して，Transformer ベースの物体検出器である DINO を基盤とし言語情報を導入することで，自然言語の入力を条件として任意の物体を検出できるように設計されている．そのため，未知のカテゴリを含むオープンワールド環境でも柔軟に対応可能である．

Grounding DINO は，画像とテキストの情報を密接に統合するために，Dual-encoder-single-decoder アーキテクチャを採用している．従来モデルが一部の段階でしか特徴量を融合していなかったのに対し，Grounding DINO では以下の 3 つのフェーズで視覚と言語を融合させている．図 1.1 に Grounding DINO のフレームワークを示す．

Dual-encoder-single-decoder アーキテクチャ

1. Feature Enhancer
   自己注意機構に加え，テキスト画像，画像からテキストへの双方向の交差注意機構（Cross-Attention）を積み重ねることで，特徴量を強化・融合する．

2. Language-guided Query Selection
   入力テキストに最も関連性の高い画像特徴をデコーダのクエリとして選択する．

3. Cross-modality Decoder
   画像とテキストの両方の特徴量をクエリに注入し，モーダル間のアライメントをさらに高め，最終的な物体ボックスとラベル語句を出力する．

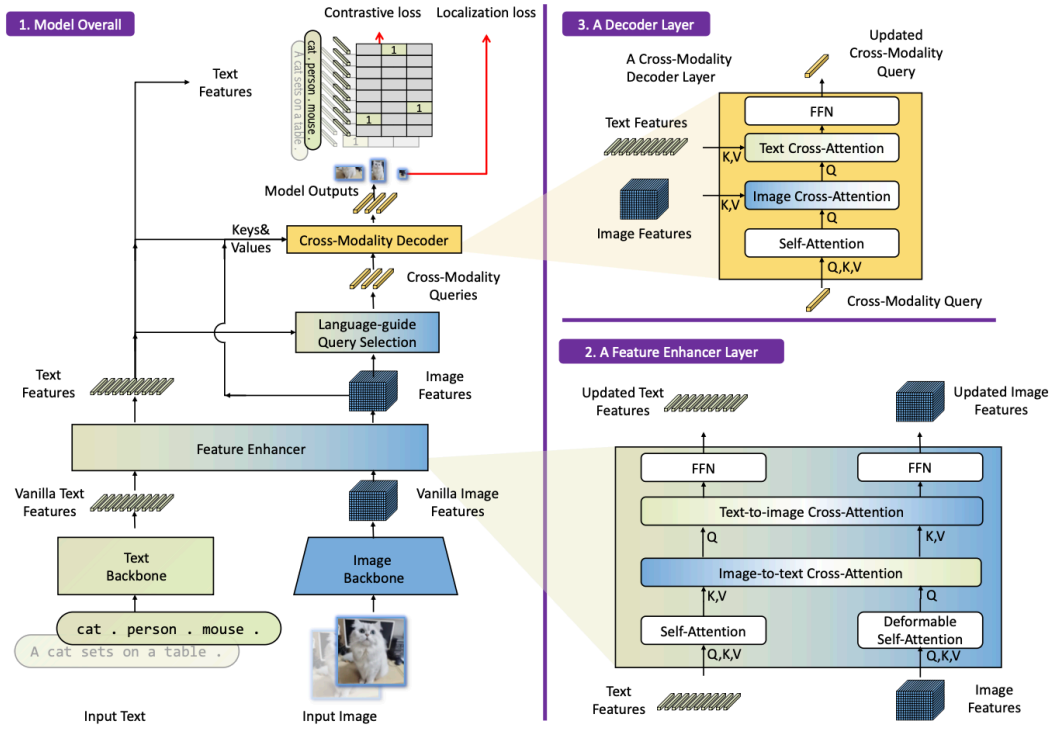

図 1.1: Grounding DINO のフレームワーク [1]

さらに，複数のカテゴリ名を入力する際，無関係な単語同士が干渉し合うのを防ぐため，Sub-sentence level Representation という技術を導入している．Attention mask を用いて，特定の単語間の関連性を強制的に遮断し，単語ごとの細かい情報

[1] [?] から引用

を保持している．学習には，物体検出データ（Objects365，OpenImages，COCO），グラウンディングデータ（Flickr30k，Visual Genome，RefCOCO など），キャンプションデータ（Cap4M）を含む大規模データセットで事前学習を行うことで，高度な概念の一般化を実現している．

1.1.2 SAM

1.2 シーングラフ生成手法

本節では LLM を用いたシーングラフ生成手法について述べる．本手法の一連の流れは図の通りである．

1.2.1 物体列挙

このステップでは，画像を LLM に入力し，画像内に含まれる可能性のある主要な物体を列挙する．これは未知の物体名を含むオープンボキャブラリな物体候補集合を得ることを目的とする．出力は修飾語を含まない単純名詞のリストであり，以下のプロンプトを LLM（GPT-4o）に与えることで行う．なお，実際のプロンプトは付録に示す．

プロンプトの概要

以下の指示，出力形式に基づいて，画像内の物体を抽出することを指示
#指示

1. 画像内に見える**主要な物体**を全て列挙
2. 各要素は**単純な単数系の名詞**
3. **形容詞・色・素材・状態**などは含めない
4. **認識された物体名のみ**を出力する

#出力形式
カンマで区切る

大規模言語モデルは，事前学習によって膨大な知識や文脈理解能力を獲得しているが，その能力を効果的に引き出すためにはプロンプトの設計が極めて重要である．プロンプトはモデルに対する入力文であり，その内容や構造によって出力結果の精度・一貫性・創造性が大きく変化する．そこで上記のプロンプトでは，出力形式を明確に定義することでノイズを排除し，語彙の統一性を保つことで後段の処理を容易にすることを目的として設計した．このような設計により，モデルが曖昧な表現や冗長な記述を避け，画像内の実体に対応する一般的な語彙を抽出できるようになる．

なお，ここで得られた物体名の集合を

$$\hat{O} = \{\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_n\} \tag{1.1}$$

とする．

1.2.2 座標取得

ここでは得られた物体名をテキストクエリとして Grounding DINO（第 1.1.1 項）に入力し，対応するバウンディングボックス $box = (x_1,\, y_1,\, x_2,\, y_2)$ と信頼度を得る．ここで，物体ごとの中心座標は次式で求める．

$$\mathbf{c}_i = \left(\frac{x_1 + x_2}{2},\ \frac{y_1 + y_2}{2} \right) \tag{1.2}$$

さらに，各物体領域をより明確に抽出するため，SAM（第 1.1.2 項）によるマスク生成を併用する．SAM は，Grounding DINO で得られた矩形領域を入力とし，物体境界をピクセルレベルで抽出する．これにより，矩形境界のみでは不明確だった物体の輪郭が精密に得られ，重なり合う物体や接触判定における曖昧性を低減することができる．SAM の出力は，COCO 形式のポリゴン表現 $segmentation = (x_0,\, y_0,\, x_1,\, y_1,\, \ldots,\, x_n,\, y_n)$ で得る．これにより，物体 o_i の位置情報として，矩形＋ポリゴンのハイブリッド表現が得られる．

検出された物体を $\mathcal{O} = \{o_1, o_2, \ldots, o_N\}$ とし，画像に存在しない物体（Grounding DINO で検出されなかった候補）は自動的に除外される．本処理により，LLM が列挙した候補のうち実際に画像に存在するものだけが残るため，フィルタリングの役割を果たす．

1.2.3 関係語生成

本ステップでは，第 1.2 項で得られた物体集合 $\hat{O} = \{\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_n\}$ に対して，全ての物体ペア (o_i, o_j) を組み合わせ，それぞれの関係を大規模言語モデルによって生成する．生成結果は，関係語（relation）に加え，位置関係（position）および接触関係（attach）を含む JSON 形式で出力される．

関係語の生成には，LLM に明確な出力仕様を与えるため，指示文と出力形式を明示した以下のプロンプトを使用する．このプロンプトは，抽象的な意味関係ではなく，日常的・物理的に成立する空間関係を想定して設計している．また，プロンプト内に具体例な出力例を含めることで，LLM の出力を安定化させ，より一貫した関係文の生成を促している．なお，実際に使用したプロンプト全文は付録に示す．

プロンプトの概要

現実世界の日常的な空間関係や物体の配置に焦点を当て，抽象的で孤立した記述ではなく，現実で物体的にもっともらしい関係の生成を指示

#指示

1. n 個の文章を生成すること
2. 各項目は以下のキーを持つ JSON 形式で出力すること
    - ”sentence”：現実的な空間関係を明確に表す文脈に基づいた文章
    - ”object”：比較対象のオブジェクトのリスト
    - ”relation”：表現される空間的関係
    - ”position”：3 次元空間でどの位置にあるか
    - ”attach”：物体が物理的に接触しているか
3. 出力全体を JSON 配列で包む
4. コメントや説明を含めない

このプロンプト設計により，モデル出力をそのまま機械処理に利用できるようになっている．以下に，生成方法の詳細を解説する．

2 物体を含む文の生成

まず，列挙された物体集合 $\mathcal{O}$ の全ての組み合わせ (o_i, o_j) に対して，LLM の持つ一般的な知識を活用し,「現実世界で起こりうる」関係を文形式で大量に生成する．

ここで，文の主語は必ずしも固定されず，どちらの物体を主語としてよいものとする．これにより,「机の上にりんごがある」「りんごの下に机がある」など，同一の関係を異なる視点から柔軟に表現できる．

また，単純に 2 物体を含む文を生成させるだけでは,「There is an apple on the table.」のような幾何的な記述が中心となり，日常的なシーンを反映した文が得られにくい．そこで，LLM に対し**文脈的に自然な状況を反映するヒント（contextual hints）**を含めるよう指示することで,「The book is on the desk in an office.」や

「The mug is placed next to the sink in the kitchen.」のように，現実的なシーンとして成立する文の生成を誘導する．この設計により，得られる関係データは単なる座標的関係にとどまらず，意味的に妥当な日常シーンの中で成立する関係表現も獲得される．

生成された文は”sentence”フィールドに格納し，対応する物体ペアは”object ”にリスト形式で記録する．さらに，この情報をもとに”relation”に関係語を格納する．

位置関係の生成

関係語の生成をする際，単に 2 物体を含む文章を生成させるだけでは，後段の探索過程で得られる関係推論が曖昧になりやすい．例えば,「apple」と「table」の 2 つの物体が水平方向に位置し，正しい関係は「next to」である場合でも,「on」という関係が一般的な文脈で頻出するため，物体名のみを入力として探索を行うと「on」が誤って選択されてしまう可能性がある．つまり，それらがどのような位置情報にあるかを明示されなければ，グラフ探索時に両者を結ぶ経路を適当に特定することができず，結果として推論が偶発的で不安定なものとなってしまう．

そこで本研究では，画像から得られる位置情報（第 1.2.2 項）を探索の手がかりとするため，LLM に対して文章生成と同時に 2 物体間の空間的な位置関係を明記させるように設計する．これにより,「apple」と「table」のような語彙的関係にとどまらず，空間的に整合する構造的情報を持つようになる．

具体的には，各文において Object1 が Object2 に対して，垂直方向（vertical），水平方向（horizontal），奥行方向（depth）の 3 軸における相対的な位置を，up，down，left，right，front，back，None のいずれかで符号化する．

$$\text{vertical} \in \{\text{up}, \text{down}, \text{None}\},\ \text{horizontal} \in \{\text{left}, \text{right}, \text{None}\},\ \text{depth} \in \{\text{front}, \text{back}, \text{None}\}$$

例えば「The apple is on the table.」の場合,「apple」は「table」の上方向に存在するため，

$\text{position} = \{\text{vertical} : \text{down},\ \text{horizontal} : \text{None},\ \text{depth} : \text{None}\}$

として出力される．

このようにして，関係語に加え位置情報を保持することで，意味的関係と空間的配置の両面を反映したネットワーク構造を構築できる．後続のグラフ探索において，単語の出現頻度に依存しない空間的に妥当な重み付けに基づく関係推論が可能になる．

接触関係の生成

さらに，物体間の関係をより正確に表現するため，物理的な接触の有無を明示的に生成させる．位置情報のみでは関係性を十分に区別できない場合が多く，例えば「apple」が「table」の上に位置している場合，その関係は「on」である可能性もあれば「above」である場合もある．しかし，位置情報だけでは両者の違いを識別できず，いずれも垂直方向（vertical）が”up”であるという同一の特徴として扱われてしまう．その結果，本来「above」である関係が「on」として出力されるなど，関係語が誤って推定される可能性がある．

この問題を解消するため，本研究ではLLMに対して，生成された文が想定する状況において2物体が接触しているか否かを同時に出力させる．接触情報は”attach”に記録され，その値は”True”または”False”のいずれかで表される．基準は以下の通りである．

接触判定

- ”True”：物体が実際に接している
  （例：placed on, touching, leaning, stacked, held など）
- ”False”：空間的に近くても直接の接触はない
  （例：in front of, next to, above など）

このように接触情報を明示的に付与することで,「垂直方向に上方にある」という単純な位置情報だけでなく，物理的な相互関係を含む関係性を捉えることが可能となる．得られた”attach”フィールドは，後段のTripletRankerにおいて探索区間の制

約条件として用いられ，物理的に矛盾しない三つ組〈主語, 関係, 目的語〉の推論を助ける役割を果たす.

エラー処理

LLM による出力は確率分布に基づく生成過程を通じて行われるため，同一のプロンプトを与えても常に完全に同一の結果が得られるとは限らない．出力形式や構造を詳細に指定した場合においても，トークン生成時の確率的揺らぎや文脈解釈の不確実性により，意図しない形式の文章や構造上の誤り，あるいは値の欠落を生じることがある，これは，LLM が厳密な規則に従って動作するルールベースモデルではなく，膨大な事例から統計的に言語パターンを学習した確率的生成モデルであることに起因する．そこで本研究では，このような生成結果の不確実性に対処するため，出力が事前に定義したフォーマットに適合しているかを自動的に検証する機構を導入する。具体的には、生成結果において必要な値が欠落している場合や、構造がフォーマット仕様と一致しない場合を検出し、それらが確認された際には再生成を自動的に実行する．これにより，生成結果の構造的信頼性を確保しつつ、プロンプトの自由度と創発性を維持したまま、安定した関係データの取得を実現している。

1.2.4 関係グラフ構築

本ステップでは，前段で得られた関係データをもとに，物体，関係語，位置関係，および接触情報を統合して関係グラフを構築する．構築される関係グラフは，有向グラフ $G = (V, E)$ で表され，ノード集合 V は物体や関係に関する要素を，エッジ集合 E はそれらの共起確率を表す．各ノードは，画像中の物体と LLM 出力の意味情報を対応づけ，関係文から抽出された主語・関係語・目的語を多段的に結ぶ構造を持つ．これにより，単なる文表現のつながりではなく，空間的配置と物理的生合成を伴う関係ネットワークを表現できる．構築する関係グラフの一例を図に示す.

（図）

ノード設計

本研究で用いるノード種別を表 1.1 に示す．これらのノードは，物体 → 位置付き物体 → 関係表現 → 三つ組という階層的接続を通じて結ばれており，言語的関係と空間的配置を統合的に扱うことが可能となっている．

表 1.1: ノード種類ごとの例と役割

ノード種別	例	役割
object	apple	画像内物体（主語・目的語）を表す基本ノード
area_object	up_apple	主語側の位置情報を付与したノード（up/down/left/right/front/back）
relation_object	apple on	主語と関係語を結合したノード
triplet	apple on table	完全な三つ組（主語・関係語・目的語）
relation_object_to	on table	関係語と目的語を結合したノード
area_object_to	down_table	目的語側に反転位置を付与したノード（逆方向探索用）

area_object ノードは，言語的な関係表現が必ずしも一意な物理配置を示さないという問題を解決するために導入する．例えば,「The apple is on the table.」という文を考えると，人間は直感的に「リンゴが机の上にあり，上方向の接触が生じている」状況を理解できる．しかし同じ”on”には,「壁にかかった絵」や「ドアに貼られたメモ」のように，接触方向が上方向ではなく側面方向である場合にも用いられる．このとき，”on”を単独の関係語として集約し，単純に object ノード →relation_object ノード（apple→apple on）のような表現だけでグラフを作ると，異なる物理配置に基づく”on”が同一ノードに混在してしまう．その結果，探索時に参照される重みは「”up”だから”on”になった」「”in front of”だから”on”になった」といった位置条件を反映したものではなく,「単に”on”という単語がよく生成されたから」という頻度

バイアスに引っ張られた結果になり得る．これでは，探索経路が表す意味が曖昧になり，なぜその関係を選んだのかを位置情報に基づいて説明できない．

そこで本研究では，位置情報をノードとして明示化し，up_apple のような area_object ノードを relation_object ノードの前段に挿入する．この設計により，「上方向にある”apple”からは”on”/”above”が候補になりやすい」といった位置条件付きの関係傾向がグラフ上に分離され，探索が単語頻度に支配されることを抑制できる．

さらに，目的語側にも area_object_to ノードを生成する．これは記述された位置情報の逆方向（up↔down，right↔left，front↔back,）に基づいて，主語側の位置属性と整合する形で目的語側の位置を表現するものである．これにより，探索は主語側だけに依存せず，目的語側からも一貫した制約で関係を評価できる．

エッジ設計と重み付け

接触関係の付与

1.2.5　グラフ探索

1.2.6　シーングラフ生成