Modular Diffusion Models for Structured Visual Recognition

Anonymous authors
Paper under double-blind review

Abstract

Traditional supervised methods for structured visual recognition tasks – such as object detection, segmentation, and scene graph generation – often produce deterministic, fixed outputs, limiting their ability to capture the inherent uncertainty in complex visual scenes. As a consequence, such point estimates are unable to capture the prediction uncertainty (or multi modality) intrinsic to these problems, often arising from natural ambiguities (e.g., ambiguity in size of partially occluded objects, local ambiguity of exact segmentation boundary, etc.) as well as noise and sparsity of training data. To address this limitation, we present Modular Diffusion Models (MDMs), a simple and novel framework that learns a distribution over structured outputs for a given input image. MDMs decompose the diffusion process into distinct, task-specific modules, each focused on capturing a different aspect of the structured information space, such as object categories, spatial locations, and inter-object relationships. This modular design allows each component to be learned independently, with seamless integration at inference without additional training. Furthermore, the modularity of MDMs enables the diffusion process to easily operate over the heterogeneous output space common in many structured learning tasks (*e.g.*, a continuous bounding boxes and discrete class labels). Experimental results over three distinct structured tasks – object detection, instance segmentation, and scene graph generation – highlight the benefits of our proposed framework.

1 Introduction

Structured visual recognition tasks – such as object detection (Ren et al., 2015; Girshick, 2015; Uijlings et al., 2013; Redmon et al., 2016; Carion et al., 2020), segmentation (He et al., 2017; Carion et al., 2020; Cheng et al., 2022; Li et al., 2023), and scene graph generation (Desai et al., 2021; Newell & Deng, 2017; Qi et al., 2019; Tang et al., 2020; 2019; Xu et al., 2017; Yang et al., 2018; Zareian et al., 2020; Zellers et al., 2018) – require models to capture different levels of detail and context within a visual scene (*e.g.*, an image or video) using structured representations. These representations range from simple bounding boxes that localize objects in object detection, to graphical structures in scene graph generation, where objects and their corresponding interactions are represented as nodes and edges respectively. These structures are fundamental to methods in areas like medical imaging, autonomous driving, and security systems. They are also frequently used as intermediate representation in a variety of complex downstream problems, and have proven effective in improving performance on tasks such as image captioning (Gu et al., 2019; Yang et al., 2019; Nguyen et al., 2021), visual question answering (Hudson & Manning, 2019; Tang et al., 2019; Lee et al., 2019), and image generation (Johnson et al., 2018; Li et al., 2019).

Learning these representations can be interpreted as modeling the conditional distribution over the desired type of structured information (*e.g.* segmentation masks, or scene graphs) given an input image (or video). Classical approaches to structured visual understanding were often explicitly probabilistic, utilizing graphical models such as Markov random fields and conditional random fields (Quattoni et al., 2004; Held et al., 1997; Lafferty et al., 2001; Boykov & Funka-Lea, 2006). These methods explicitly represented uncertainty and dependencies between output variables, allowing multiple plausible interpretations of a scene to be expressed through a structured probability distribution. However, estimating such distributions is inherently challenging due to the combinatorial growth of the output space. For instance, in scene graph generation, the number of possible graphs grows combinatorially with the number of objects and interactions involved.

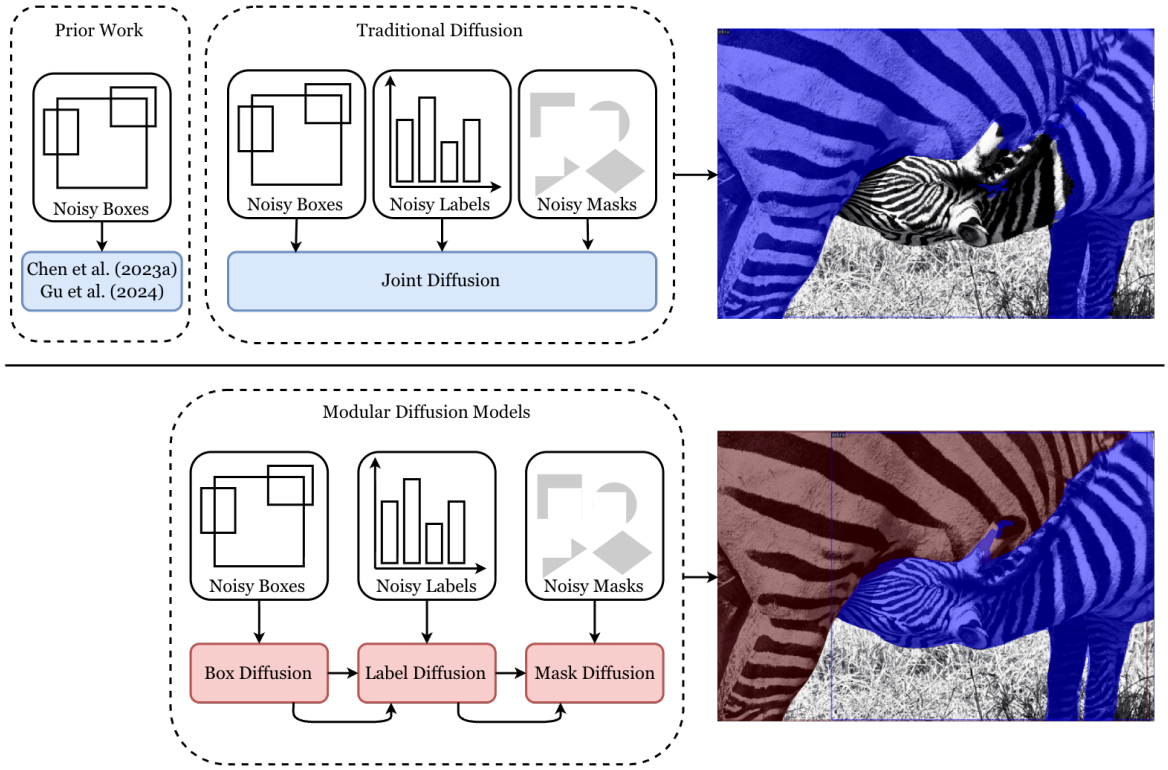


Figure 1: **Modular Diffusion Models** factorize the complex joint structured distribution into several easy-to-estimate conditionals, each modeled using a diffusion process. In contrast to existing work (top), MDM (bottom) models a diffusion process over all components of the structured distribution. Additionally, compared to a naive joint approach (top), MDM preserves cross-component alignment during denoising, avoiding incoherent intermediate states and enabling more reliable reconstruction of structured outputs. In the example shown above, MDM is able to correctly identify and accurately segment the two zebras in the scene.

To avoid the aforementioned issue, recent deep learning-based approaches often approximate the conditional distribution by learning to predict the distribution mean (a point estimate) (Dong et al., 2021; Liu et al., 2021; Newell & Deng, 2017; Tang et al., 2020; Xu et al., 2017; Yang et al., 2018; Zareian et al., 2020; Suhail et al., 2021; Khandelwal & Sigal, 2022), usually by employing supervised learning techniques on an annotated training set. While this shift has led to substantial gains in accuracy and scalability, it has also resulted in models that produce a single, fixed structured output for each visual input. As a consequence, modern approaches often struggle to represent inherent ambiguities present in complex visual scenes.

Learning generative models to approximate complex distributions is a well-established problem in the domain of image generation (Kingma & Dhariwal, 2018; Razavi et al., 2019; Brock et al., 2018; Karras et al., 2017; Goodfellow et al., 2014). Diffusion-based processes (Ho et al., 2020; Rombach et al., 2022; Ruiz et al., 2023; Nair et al., 2023; Saharia et al., 2022), in particular, have shown impressive ability to generate realistic, high-resolution images, both unconditionally and under the guidance of textual prompts. Diffusion processes fundamentally consists of two stages – (i) a forward phase, where noise is incrementally added to the input data, and (ii) a denoising phase, during which a neural network learns to gradually denoise a noisy input. Extending this framework to structured recognition tasks, however, is uniquely challenging due to the heterogeneous and interdependent nature of the output space. For instance, in the case of instance segmentation, the desired structured result includes continuous outputs (bounding boxes for individual objects ; *e.g.* "person" and "bicycle") and discrete outputs (corresponding object labels and spatial segmentation masks). These components are inherently coupled – the object category often determines the expected size

and shape of its bounding box and mask, while, in turn, the geometry of the mask and the spatial placement of the box provide important cues for identifying the object label.

A naive extension of diffusion models to structured prediction would involve defining a joint diffusion process over all output components, independently corrupting each element and training a model to jointly denoise them. However, modeling such complex structured distributions is challenging, particularly when they combine discrete (*e.g.* class labels) and continuous elements (*e.g.* bounding boxes and segmentation masks). Using the aforementioned example of instance segmentation, independently noising object bounding boxes, class labels, and segmentation masks can destroy the alignment between these components, such that they no longer correspond to the same underlying entities. For example, a segmentation mask originally associated with a "person" may become paired with a label such as "bicycle", or with a mismatched bounding box that no longer tightly encloses it. These inconsistencies produce incoherent intermediate states, making denoising substantially more difficult. This is shown qualitatively in Figure 1 and quantitatively in Table 4. Modeling the aforementioned joint diffusion process leads to a loss in performance across multiple denoising steps, suggesting the inability of the model to accurately capture the underlying structured data distribution.

As a result, existing diffusion-based approaches typically avoid full joint modeling and instead operate on simplified or homogeneous representations. For instance, in the diffusion-based object detection model proposed in Chen et al. (2023a), the forward phase only applies noise to the bounding box coordinates, while generating class labels is treated as an auxiliary task, which is learned through an additional head in the denoising model. Consequently, such an approach limits the model's capacity to learn a joint distribution over all components, resulting in fixed estimations for parts of the structured representation.

In this work, we introduce Modular Diffusion Models (MDMs), a novel framework designed to effectively learn distributions over structured outputs. The central concept of our approach is to decompose the joint conditional distribution, parameterized by the diffusion process, into a series of conditional distributions – each over a homologous portion of the output space. Each of these conditional distributions, we parameterize using a separate diffusion process, resulting in the complete model being a modular chain of diffusion processes. For instance, in the task of instance segmentation, rather than attempting to learn a single diffusion process that concurrently denoises bounding boxes, class labels, and segmentation masks, our proposed MDM framework independently learns separate diffusion processes for each of these components. The benefits of such a modular factorization are manyfold - (i) it simplifies the learning process by allowing individual modules to concentrate solely on denoising a single homogeneous input; (ii) it effectively accommodates heterogeneous inputs, as each diffusion module can be specifically designed to work with either continuous or categorical data; and (iii) the independently trained diffusion processes can be seamlessly integrated during inference, enabling the generation of complete structured representations without requiring any additional training.

We instantiate the individual diffusion processes in the MDM framework using a simple transformer-based architecture that performs iterative denoising of structured components conditioned on the input image and ancestral context. Concretely, we construct input queries directly from noisy variables and modulate them with time-aware conditioning, enabling dynamic adaptation across diffusion steps. These queries are processed by a multi-layer transformer decoder, which progressively refines them through attention-based interactions to produce clean structured predictions. We highlight the effectiveness of our proposed model and framework by achieving strong performance across diverse structured visual recognition tasks – object detection, instance segmentation, and scene graph generation – without requiring task-specific specialized design components. Specifically, on object detection and instance segmentation, MDM outperforms existing diffusion-based approaches by up to $8.4\text{AP}^{\text{mask}}$ and $3.9\text{AP}^{\text{box}}$, while achieving parity with other strong deterministic approaches. Furthermore, on the task of scene graph generation, MDM outperforms the most competitive baselines by 2.6/1.6 on hR@50/100. Beyond absolute performance metrics, contrary to determinstic models that are constrained to point estimates, MDM is able to capture the uncertainity in the underlying data distribution and sample multiple plausible structured outputs for a single image.

Contributions. Our contributions can be summarized as follows – (i) We introduce Modular Diffusion Models (MDMs), a general framework for modeling complex structured distributions by factorizing them into a sequence of simpler conditional diffusion processes; (ii) We propose a simple and general transformer-based instantiation of MDMs that performs iterative, context-aware denoising; (iii) We demonstrate the

effectiveness of the framework across multiple structured visual recognition tasks with heterogeneous outputs, without requiring task-specific architectural components; (iv) We highlight the flexibility of MDMs, enabling fine-grained control over individual diffusion processes during inference without additional training.

2 Related Work

2.1 Structured Learning

Object Detection. Object detection is a fundamental task in computer vision that involves identifying and localizing objects in an image. The field has seen substantial progress due to advancements in convolutional neural networks (CNNs), with most modern methods divided into *one-stage* and *two-stage* categories. Two-stage detectors, such as Faster R-CNN (Ren et al., 2015) and Fast R-CNN (Girshick, 2015), first generate candidate box proposals – either using external algorithms (Uijlings et al., 2013) or through a learned proposal network (RPN) (Ren et al., 2015). The second stage shares convolutional features across all region proposals, and generates refinements over the proposal boxes and corresponding labels. One-stage detectors were introduced to address the slower runtimes of two-stage models. These methods aim to simplify the process by detecting objects in a single pass (Redmon et al., 2016; Lu et al., 2016). More recently, transformers-based approaches like DETR (Carion et al., 2020) have gained significant popularity. These approaches formulate object detection as a set prediction problem, wherein boxes are generated via repeated self- and cross-attention (Vaswani et al., 2017) over fixed set of learnable queries, trained via a matching algorithm. Subsequent improvements in transformer-based methods have focused on enhancing convergence speeds and detection quality. For example, Zhu et al. (2020); Liu et al. (2022) use anchor-boxes to guide image conditioning, and Zhang et al. (2022); Li et al. (2022a) leverage denoising queries during training to improve learning stability and accelerate convergence.

Instance Segmentation. Instance segmentation extends object detection by not only localizing objects but also generating precise pixel-level masks for each object instance. Early methods, such as Mask-RCNN (He et al., 2017), introduce an additional mask head that operates over proposed box aligned features, allowing simultaneous prediction of object classes and segmentation masks. Transformer-based approaches have recently seen increased use. DETR (Carion et al., 2020) combines image-based cross-attention (Vaswani et al., 2017) with a convolutional up-scaling network to generate instance masks. Subsequent methods (Cheng et al., 2022; Li et al., 2023) leverage multi-scale features from CNN backbones to directly generate masks from transformer queries via a channel-wise inner product between the query vectors and feature map at each spatial location, eliminating the need for separate upsampling networks.

Scene Graph Generation. Scene graph generation focuses on modeling the relationships between object entities within an image, requiring methods that can both accurately detect objects and reason about their inter-object semantics. This task has been extensively studied in computer vision (Desai et al., 2021; Khandelwal et al., 2021; Li et al., 2022b; Liu et al., 2021; Newell & Deng, 2017; Qi et al., 2019; Tang et al., 2020; 2019; Xu et al., 2017; Yang et al., 2018; Zareian et al., 2020; Zellers et al., 2018; Khandelwal & Sigal, 2022; Chen et al., 2023c; Desai et al., 2022; Kim et al., 2024; Fu et al., 2025). As object detection is a critical component, existing approaches in this domain can also be broadly classified into two categories: *one-stage* or *two-stage* approaches. The initial phase of two-stage methods involves pre-training a strong object detector (like Faster-RCNN (Ren et al., 2015)) to obtain candidate entity representations (boxes and corresponding features) from an image. The second stage then involves training a predicate generation network, conditioned on these fixed entity representations. Existing works have explored various architectures for the predicate generation network, like Bi-directional LSTMs (Zellers et al., 2018), Tree-LSTMs (Tang et al., 2019), Graph Neural Networks (Yang et al., 2018), and message-passing algorithms (Qi et al., 2019; Xu et al., 2017). One-stage approaches instead generate both the entities and predicates jointly via end-to-end trainable modules, like fully convolutional networks (Liu et al., 2021) and transformer-based architectures (Cong et al., 2022; Dong et al., 2021; Li et al., 2022b; Shit et al., 2022; Desai et al., 2022; Kim et al., 2024; Fu et al., 2025).

All the aforementioned methods generate deterministic structured estimates for a given input image. In contrast, our proposed MDM framework is able to learn a distribution, enabling sampling of multiple plausible structured representations.

2.2 Generative Modeling

Diffusion Models. Diffusion-based generative models have gained substantial popularity in recent years, largely due to their impressive performance on image generation tasks (Ho et al., 2020; Rombach et al., 2022; Ruiz et al., 2023; Nair et al., 2023; Saharia et al., 2022; Ho et al., 2022; Epstein et al., 2024; Gu et al., 2022). Introduced in Ho et al. (2020), at its core, the diffusion models are built upon a Markov process and consists of two primary steps: (i) a forward process, in which noise is progressively added to the input data, and (ii) a denoising process, where a neural network is trained to iteratively remove noise. Popular architectures for this denoising process include U-Net (Ronneberger et al., 2015) and transformer-based models (Vaswani et al., 2017). Nichol & Dhariwal (2021); Ho & Salimans (2022) demonstrate that these diffusion processes can be extended to model conditional distributions, wherein the generative process can be guided by external inputs, such as image labels. This can be achieved either through an explicit classifier or by training a single model that learns both unconditional and conditional distributions simultaneously. Rombach et al. (2022) further improve the computational efficiency of diffusion processes by proposing a method that operates in a lower-dimensional latent space instead of directly over image pixels. Specifically, images are first encoded into a latent space using a pre-trained variational autoencoder (VAE), diffusion is then applied in the latent space, and the generated output is mapped back to the image space through the VAE. For diffusion models with long Markov chains, inference can be time-consuming as the entire chain must be traversed. However, Song et al. (2020) show that the denoising process can be formulated as a non-Markovian chain, significantly reducing the number of steps required while avoiding the need for retraining the model. While Gaussian noise is most commonly used due to its smooth interaction with the continuous nature of images, Hoogeboom et al. (2021); Austin et al. (2021) have also explored the extension of diffusion processes to model discrete distributions.

Diffusion for Structured Learning. Although diffusion models have primarily been used for image generation, they have recently also been extended towards structured learning problems. Chen et al. (2023a) explore the application of diffusion models for object detection, formulating a diffusion process over bounding boxes, learning to denoise them by conditioning on an image, and generating corresponding labels as an auxiliary task. Gu et al. (2024) extend this further to instance segmentation by incorporating an additional segmentation head. Diffusion models have also been applied in semantic segmentation, which focuses on generating accurate pixel-level labels for images. Methods in Tan et al. (2022); Zbinden et al. (2023); Wang et al. (2023) define a diffusion process over pixel labels, learning to progressively denoise them to produce more precise segmentations. In all the aforementioned methods, diffusion is applied only to a single modality—whether bounding boxes or pixel labels—while other outputs, such as object categories or segmentations, are typically generated via auxiliary heads. In contrast, our proposed diffusion framework operates across multiple input modalities simultaneously (*e.g.* boxes, labels, segmentation masks). Finally, while research in graph generation closely aligns with the task of scene graph generation, diffusion-based graph generation methods in Jo et al. (2022); Vignac et al. (2022); Yan et al. (2023) are unconditional in nature. This is in contrast to our proposed method that grounds graphs in a given image.

3 Background

3.1 Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are probabilistic generative models that capture complex data distributions by learning to reverse a gradual noising process, generating samples via a sequence of iterative denoising steps. The diffusion framework is composed of two processes – a *forward diffusion process*, which progressively adds noise to data, and a *reverse diffusion process*, wherein a model is trained to recover samples by removing the added noise step-by-step.

Forward Process. Let $\mathbf{x}_0 \sim q(\mathbf{x}0)$ denote a data point drawn from the empirical data distribution. The forward process defines a Markov chain $\mathbf{x}_{1:T} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ over T discrete timesteps, where each random variable $\mathbf{x}_t \in \mathbb{R}^d$ represents a progressively corrupted version of the original data sample. The joint distribution of this forward process is defined as

$$q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) := \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right), \tag{1}$$

where each transition distribution $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ injects a small amount of noise into the previous state. The forward process is fixed and does not involve any learnable parameters.

Reverse Process. The objective of the reverse process is to learn a set of parameters $\boldsymbol{\theta}$ that can effectively remove the corruption introduced by the forward process. For each step in the reverse process, the model estimates a conditional distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t)$ to approximate the ground-truth denoising transition $q(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t, \mathbf{x}_0)$. The training objective then is to minimize the distance between these two distributions, typically by using the Kullback-Leibler (KL) divergence.

$$\arg\min_{\theta} D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t, \mathbf{x}_0\right) \,||\, p_{\boldsymbol{\theta}}\left(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t\right)\right). \tag{2}$$

The type of noise added in each transition of the Markov chain described in Eq. (1) can vary depending on the specific application and data characteristics. We will now outline the Gaussian diffusion framework, which is a widely used instantiation of diffusion models, and is particularly suitable for continuous data distributions such as images.

3.2 Gaussian Diffusion

Gaussian diffusion models define a forward process that incrementally adds Gaussian noise to the data at each timestep. Formally, the forward transition at timestep t is defined as,

$$q\left(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}\right) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}). \tag{3}$$

$\beta_t \in (0, 1)$ is a predefined variance schedule that controls the amount of noise added at each timestep t, where $\beta_t \to 1$ as $t \to T$.

Over multiple steps, this process gradually corrupts the original data. When the magnitude of β_t remains sufficiently small and the length of the Markov chain T is large, the distribution of $\mathbf{x}_T$ converges to a standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$.

A key property of the Gaussian diffusion forward process is that the marginal distribution of a noised $\mathbf{x}_t$ given data $\mathbf{x}_0$ admits a closed-form expression (Ho et al., 2020),

$$\begin{aligned} q\left(\mathbf{x}_t \,|\, \mathbf{x}_0\right) &:= \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \\ \bar{\alpha}_t &:= \prod_{i=0}^{t} \alpha_i; \quad \alpha_i := (1-\beta_i) \end{aligned} \tag{4}$$

This allows an arbitrary noised sample $\mathbf{x}_t$ to be efficiently obtained directly from $\mathbf{x}_0$, without the need to traverse through the Markov chain described in Eq. (1). Such a formulation allows computation of the ground-truth denoising transition $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ to also be formulated in a closed-form as,

$$\begin{aligned} q\left(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t, \mathbf{x}_0\right) &:= \mathcal{N}\left(\mathbf{x}_{t-1};\, \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0), \boldsymbol{\Sigma}(t)\right) \\ &:= \mathcal{N}\left(\mathbf{x}_{t-1};\, \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\,\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\,\mathbf{x}_0}{1-\bar{\alpha}_t}}_{\boldsymbol{\mu}(\mathbf{x}_t, \mathbf{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}}_{\boldsymbol{\Sigma}(t)}\right) \end{aligned} \tag{5}$$

To approximate the aforementioned posterior transition as closely as possible, the learned denoising step $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t)$ is also assumed to be Gaussian,

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t) &:= \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \boldsymbol{\Sigma}(t)) \\ &:= \mathcal{N}\left(\mathbf{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\,\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\,\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)}{1-\bar{\alpha}_t}}_{\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}}_{\boldsymbol{\Sigma}(t)}\right) \end{aligned} \tag{6}$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ is the mean function parameterized by a neural network conditioned on both the noisy input $\mathbf{x}_t$ and the timestep t. In practice, this mean function is often implemented to predict the denoised sample $\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)$, which serves as an estimate of the original sample $\mathbf{x}_0$. Additionally, note that the covariance is assumed to be identical to the true posterior transition, which simplifies the learning problem to matching the mean of $p_{\boldsymbol{\theta}}$ to that of q.

As shown in Ho et al. (2020), by choosing an appropriate parameterization of the mean function $\boldsymbol{\mu}_{\boldsymbol{\theta}}$, the KL divergence objective simplifies in Eq. (2) to a mean squared error loss,

$$\arg\min_{\theta} ||\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0||_2^2 \tag{7}$$

Once trained, diffusion models generate new samples by iteratively applying the reverse diffusion process starting from pure noise. Specifically, sampling begins by drawing an initial latent $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, and then repeatedly applying the learned reverse transitions $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$, as defined in Eq. (6), for $t = T, T-1, \ldots, 1$.

Conditioning. Diffusion models can be extended to learn conditional distributions $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$, where $\mathbf{c}$ denotes auxiliary information. This conditioning may represent class labels, text embeddings, or other semantic signal that guides generation. This information is typically passed as an additional input to the denoising network, for example through feature modulation or cross-attention. The overall training objective is analogous to unconditional case,

$$\arg\min_{\theta} ||\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{x}_0||_2^2 \tag{8}$$

4 Approach

In this section, we first outline the general MDM framework. We then present specific instantiations of this framework on three structured learning tasks – object detection, instance segmentation, and scene graph generation.

4.1 Modular Diffusion Models

Let $\mathbf{S} = \{\mathbf{S}_i\}_{i \leqslant n}$ represent the set of structured representations for a specific image $\mathbf{I}$. Each structured representation can be further decomposed into distinct components $\mathbf{S}_i = (\mathbf{s}_i^1, \ldots, \mathbf{s}_i^K)$. As an example, for the task of instance segmentation, each $\mathbf{S}_i$ corresponds to an individual object instance and is represented as $\mathbf{S}_i = \langle \mathbf{b}_i, \mathbf{l}_i, \mathbf{m}_i \rangle$, with $\mathbf{b}_i$, $\mathbf{l}_i$, and $\mathbf{m}_i$ denoting the bounding box, class label, and segmentation mask respectively. Therefore, the underlying learning problem can be formulated as estimating the joint distribution $\Pr(\mathbf{S} \,|\, \mathbf{I})$. Directly modeling this joint distribution through a diffusion process, however, can present significant challenges, particularly as the output space grows combinatorially with the increasing cardinality of $\mathbf{S}$. Furthermore, the components $\mathbf{s}_i$ may exhibit complex interdependencies and have distinct data types (*e.g.* boxes, labels), adding an additional layer of difficulty in directly capturing the joint distribution. Modeling these complex, heterogeneous dependencies efficiently therefore requires a more structured formulation – one that factorizes the joint distribution into simpler, easier-to-learn conditional distributions.

In particular, for a given structured representation $\mathbf{S}$, one can define a plausible factorization via a Directed Acyclic Graph (DAG) $\mathbf{G_S} = (\mathbf{S}, \mathbf{E_S})$, where $\mathbf{E_S}$ represents the directed edges denoting the conditional

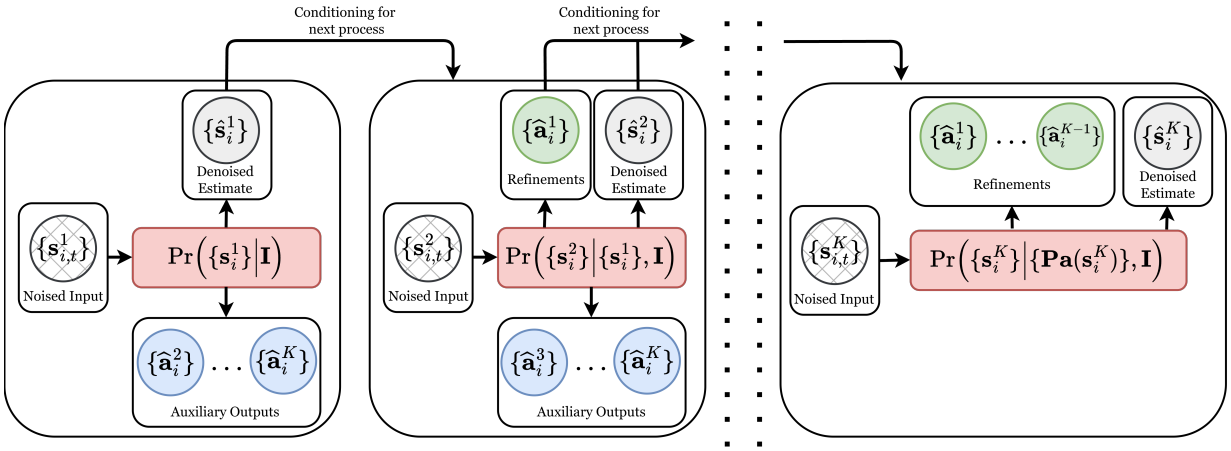


Figure 2: **Modular Diffusion Models.** We factorize the joint distribution over structured outputs into a sequence of conditional distributions, each modeled by its own diffusion process (red). At each stage, a module takes a noisy input along with relevant conditioning, and produces a denoised estimate, updated conditioning (green), and auxiliary outputs (blue).

dependencies between the components $\mathbf{s}_i$. Under this DAG structure, the joint distribution $\Pr(\mathbf{S} \,|\, \mathbf{I})$ can be factorized as,

$$\Pr\Big(\mathbf{S} \,\Big|\, \mathbf{I}\Big) = \prod_{i=1}^{K} \Pr\Big(\{\mathbf{s}_i^k\} \,\Big|\, \{\mathbf{Pa}(\mathbf{s}_i^k)\}, \mathbf{I}\Big) \tag{9}$$

where $\mathbf{Pa}(\mathbf{s}_i^k)$ denotes the set of parents of $\mathbf{s}_i^k$ in the DAG. Note that most existing methods for structured learning tasks often implicitly incorporate such a factorization via model architecture design.

Our proposed MDM framework leverages this factorization under a DAG structure to make jointly learning a diffusion process easier. Specifically, MDM *independently* models each conditional $\Pr(\{\mathbf{s}_i\} \,|\, \{\mathbf{Pa}(\mathbf{s}_i)\}, \mathbf{I})$ as a separate diffusion model. Since each model is now responsible for learning the distribution over a single component $\mathbf{s}_i^k$, this approach reduces the complexity of the learning task. This modularity enables each diffusion process to be tailored to its corresponding component, allowing for flexibility in the choice of diffusion process and architectural design. Additionally, as each model now operates over a smaller output space, learning is more tractable.

During inference, a valid structured representation can easily be obtained without any additional training, via ancestral sampling over the specified DAG. For each component $\mathbf{s}_i^k$, this involves taking multiple denoising steps starting from random noise. Since each individual diffusion model is trained independently, it can be run for a varying number of denoising steps, allowing for fine-grained control over the generation process. As such, this flexibility in the number of denoising steps for each model leads to MDMs being able to generate exponentially large number of potential structured representations while only requiring linear number of diffusion processes.

4.2 MDM for Detection and Instance Segmentation

In this section, we present a specific instantiation of our proposed MDM framework on the task of instance segmentation, which requires models to accurately localize, label, and outline the boundaries of each distinct object instance within an image. Note that, within our framework, the task of object detection can be viewed as a special case of instance segmentation in which the mask component is omitted.

For an input image $\mathbf{I}$, the instance segmentation task outputs a set of representations $\mathbf{S}^{\text{seg}} = \{(\mathbf{b}_i, \mathbf{l}_i, \mathbf{m}_i)\}$, which typically consists of three components – a bounding box $\mathbf{b}_i$ that provides a spatial location for each instance, a distribution over class labels $\mathbf{l}_i$ that specifies the likelihood of the instance belonging to each object category, and a binary mask $\mathbf{m}_i$ that defines a pixel-wise segmentation for the object.

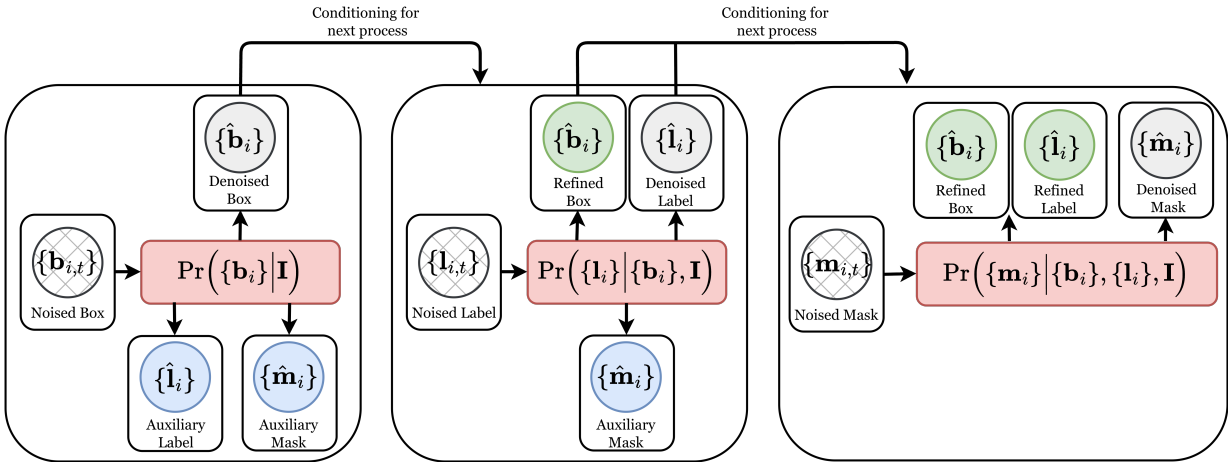


Figure 3: **Modular Diffusion Model for Instance Segmentation.** This consists of three separate diffusion processes - (i) box diffusion, which denoises bounding boxes conditioned on image features while producing auxiliary label and mask predictions; (ii) label diffusion, which denoises class labels conditioned on the current box estimates and image features, while refining the box and generating auxiliary outputs; and (iii) mask diffusion, which denoises segmentation masks conditioned on the estimated boxes, labels, and image features, while refining both box and label predictions.

For the task of instance segmentation, our proposed MDM framework factorizes the joint distribution $\Pr(\mathbf{S}^{\text{seg}} \,|\, \mathbf{I})$ using the dependency structure specified by the DAG,

$$\mathbf{b} \longrightarrow \mathbf{l} \longrightarrow \mathbf{m}$$

$$\Pr\Big(\mathbf{S}^{\text{seg}} \,\Big|\, \mathbf{I}\Big) = \Pr\Big(\{\mathbf{b}_i\} \,\Big|\, \mathbf{I}\Big) \;\cdot\; \Pr\Big(\{\mathbf{l}_i\} \,\Big|\, \{\mathbf{b}_i\}, \mathbf{I}\Big) \;\cdot\; \Pr\Big(\{\mathbf{m}_i\} \,\Big|\, \{(\mathbf{b}_i, \mathbf{l}_i)\}, \mathbf{I}\Big) \tag{10}$$

This factorization follows the natural structure of the instance segmentation task. The model first predicts the locations of candidate objects in the image using bounding boxes $\{\mathbf{b}_i\}$. Given these locations, it then predicts the semantic category distribution $\{\mathbf{l}_i\}$ for each detected instance. Finally, conditioned on the image, the predicted box, and the corresponding class label distribution, the model generates the instance mask $\{\mathbf{m}_i\}$, which captures the detailed pixel-level shape of the object.

We define three separate diffusion processes, each corresponding to one of the conditional distributions in Eq. (10). The forward process for each of these conditionals is described below.

Box Diffusion. Each bounding box is parameterized by the continuous tuple (x, y, w, h), representing the box center coordinates, and its width and height. Gaussian diffusion can therefore be applied directly by perturbing these parameters at timestep t using the standard forward process (Eq. 4).

Label Diffusion. The label associated with each box is a categorical variable in $[1, C]$, where C denotes the number of classes. Although diffusion can be defined directly over discrete variables (Austin et al., 2021), we instead adopt a simpler Gaussian formulation using a continuous relaxation of discrete labels (Dieleman et al., 2022; Chen et al., 2022; 2023b). Specifically, each label is represented as a one-hot vector $\mathbf{l}_i = [0, \ldots, 1, \ldots]$, where the j-th component $\mathbf{l}_i[j] = 1$.

Mask Diffusion. Traditionally, diffusion over masks is treated as a semantic segmentation problem, where each pixel is associated with a categorical label (Zbinden et al., 2023). In our MDM framework, as labels are modeled separately, we treat masks as binary spatial masks $\in [0, 1]^{W \times H}$ corresponding to individual object instances. Similar to label diffusion, we assume a continuous relaxation by downsampling masks to a smaller resolution $[0, 1]^{W/\tau \times H/\tau}$, where τ is the reduction factor. This has two benefits – First, the relaxed mask representation allows Gaussian diffusion to be applied directly. Second, performing diffusion at a reduced

spatial resolution significantly lowers computational cost, as the diffusion process operates on a smaller spatial grid. This is reasonable in practice, as instance segmentation methods often predict masks at relatively coarse resolutions before projecting them back to the original image size (He et al., 2017).

4.3 MDM for Scene Graph Generation

Scene graphs are a graphical representation wherein the nodes correspond to the entities present in the scene, with edges denoting the relationships between pairs of nodes. Each node is grounded within the image through its spatial location, detailed by a bounding box, and the corresponding class label. This graph is often represented using a set of `<subject, predicate, object>` triplets, where `subject` and `object` are nodes in the graph, and `predicate` is the relationship between them.

Therefore, the task of scene graph generation can be viewed as learning a set of the aforementioned triplets, and the corresponding representation $\mathbf{S}^{\text{sg}} = \{(\mathbf{b}_i^s, \mathbf{b}_i^o, \mathbf{l}_i^s, \mathbf{l}_i^o, \mathbf{l}_i^p)\}$ that is expected as output consists of five components - bounding boxes $\mathbf{b}_i^s$ and $\mathbf{b}_i^o$ that localize the subject and object entities, category labels $\mathbf{l}_i^s$ and $\mathbf{l}_i^o$ for the corresponding boxes, and a relationship label $\mathbf{l}_i^p$ defining the interaction between them.

We define the following dependency structure to instantiate our proposed MDM framework for the task of scene graph generation.

$$
\begin{aligned}
&[\mathbf{b}^s, \mathbf{b}^o] \longrightarrow [\mathbf{l}^s, \mathbf{l}^o, \mathbf{l}^p] \\
&\Pr\Big(\mathbf{S}^{\text{sg}} \,\Big|\, \mathbf{I}\Big) = \Pr\Big(\{(\mathbf{b}_i^s, \mathbf{b}_i^o)\} \,\Big|\, \mathbf{I}\Big) \;\cdot\; \Pr\Big(\{(\mathbf{l}_i^s, \mathbf{l}_i^o, \mathbf{l}_i^p)\} \,\Big|\, \{(\mathbf{b}_i^s, \mathbf{b}_i^o)\}, \mathbf{I}\Big)
\end{aligned}
\tag{11}
$$

Intuitively, the aformentioned factorization first generates candidate of bounding boxes that are likely to exhibit an interaction. Given these box pairs, the model then generates the corresponding object categories and the relationship label.

We therefore define two separate diffusion processes, each corresponding to one of the conditional distributions in Eq. (11). The forward process for each of these conditionals is described below.

Proposal Diffusion. Scene graph grounds each relation to the image by a pair of bounding boxes. Therefore, the input representation for the proposal diffusion then is the concatenated continuous vector $[\mathbf{b}_i^s, \mathbf{b}_i^o]$, where each box $\mathbf{b}_i^e$ is parameterized by the continuous tuple (x, y, w, h), representing the box center coordinates and its width and height. As these parameters lie in a continuous space, Gaussian diffusion can be applied directly.

Triplet Diffusion. The subject, object label, and predicate labels are categorical variables defined over the entity and relationship vocabularies. Similar to the label diffusion process described for instance segmentation (Section 4.2), we adopt a continuous relaxation for these discrete variables. The input to triplet diffusion is therefore the concatenated continuous vector $[\mathbf{l}_i^s, \mathbf{l}_i^o, \mathbf{l}_i^p]$, where each component is a one-hot vector corresponding to its respective category. Gaussian noise is then applied to this continuous representation during the forward diffusion process.

4.4 Transformer Based Denoising Process

In this section we present a concrete instantiation of the reverse diffusion process for the MDM framework. Specifically, we modify the transformer based architecture in Li et al. (2023) to operate within MDM framework, as highlighted in Figure 4. The choice of a transformer-based architecture is motivated by their ability to effectively model set-based structured outputs and capture complex interactions between elements through attention mechanisms. Importantly, it should be noted that the MDM framework itself is architecture agnostic and can be implemented using a variety of design choices.

Recall that our proposed MDM framework defines a diffusion process for each conditional $\Pr(\{\mathbf{s}_i^k\} \,|\, \{\mathbf{Pa}(\mathbf{s}_i^k)\}, \mathbf{I})$ in Eq. 9. As such, the reverse process is implemented via a multi-layer transformer-based decoder, which denoises a noisy input at timestep t, $\mathbf{s}_{i,t}^k$, conditioned on the image features $\mathbf{I}$ and any ancestral input $\mathbf{Pa}(\mathbf{s}_i^k)$. For example, in the case of the mask diffusion described in Section 4.2, these ancestral inputs would include

the denoised box $\{\hat{\mathbf{b}}_i\}$ and label estimates $\{\hat{\mathbf{l}}_i\}$ from the previous diffusion processes. Additionally, it should be noted that the image features $\mathbf{I}$ are obtained by passing the raw image through a combination of convolutional backbone and a multi-layer transformer encoder, akin to methods in Cheng et al. (2022); Li et al. (2023).

We now describe the key components of this architecture and explain how they are adapted to serve as the denoising network for the reverse diffusion process.

4.4.1 Image Encoder

Following Li et al. (2023), for each image $\mathbf{I}$, the encoder first employs a deep convolutional network (*e.g.* ResNet (He et al., 2016)) to extract multi-scale spatial feature maps $[\mathbb{R}^{c\times W_1\times H_1}, \ldots, \mathbb{R}^{c\times W_r\times H_r}]$ where c is the number of channels, and (W_r, H_r) denotes the spatial resolution at each scale. These feature maps are concatenated and passed through a multi-layer transformer encoder, which transforms them into position-aware flattened image representations $\mathbf{I} \in \mathbb{R}^{c\times \sum_r W_r H_r}$. These are subsequently used in the decoder to condition the denoising process.

4.4.2 Denoising Decoder

The denoising decoder is tasked with recovering a set of clean estimates $\{\hat{\mathbf{s}}_i^k\}$ from the noisy input set $\{\mathbf{s}_{i,t}^k\}$ at a given timestep t, and is implemented as a multi-layer transformer decoder. Unlike traditional transformer-based methods (*e.g.* Carion et al. (2020)) that use learnable queries to initialize the decoding process, we formulate the input queries $\{\mathbf{q}_i^k\}$ as a function of the noisy input, modulated by time-aware ancestral conditioning. Specifically,

$$\begin{aligned} \boldsymbol{\beta}_i^k, \boldsymbol{\gamma}_i^k &= f_a\left(\mathbf{Pa}(\mathbf{s}_i^k),\, t\right), \\ \mathbf{q}_i^k &= f_q\left(\mathbf{s}_{i,t}^k\right) \odot \left(1 + \boldsymbol{\beta}_i^k\right) + \boldsymbol{\gamma}_i^k \end{aligned} \tag{12}$$

where f_a, f_q are neural networks, and $\odot$ refers to elementwise multiplication. This formulation enables the queries to dynamically adapt to the noisy inputs as well as any additional contextual information, conditioned on the current diffusion timestep.

The transformer decoder, consisting of M layers, takes as input the set of queries $\{\mathbf{q}_i^k\}$ and iteratively denoises them. Each layer applies self-attention (sa) on the queries, cross-attention (ca) to the image features $\mathbf{I}$, and a feed-forward network (ffn). To incorporate timestep information, we adopt the zero-initialized adaptive layer normalization (adaLN-Zero) (Peebles & Xie, 2023), which modulates each update with time embeddings.

For layer $m \leqslant M$, the update can be written as a sequence of operations,

$$\begin{aligned} \mathbf{q}_{i,\text{sa}}^{k,m} &= \mathcal{F}_{\text{sa}}^m\big(\mathbf{q}_i^{k,m-1}, t\big), \\ \mathbf{q}_{i,\text{ca}}^{k,m} &= \mathcal{F}_{\text{ca}}^m\big(\mathbf{q}_{i,\text{sa}}^{k,m}, \mathbf{I}, t\big), \\ \mathbf{q}_i^{k,m} &= \mathcal{F}_{\text{ffn}}^m\big(\mathbf{q}_{i,\text{ca}}^{k,m}, t\big), \end{aligned} \tag{13}$$

where each transformation $\mathcal{F}_x^m;\ x \in \{\text{sa}, \text{ca}, \text{ffn}\}$ can be expressed as,

$$\mathcal{F}_x^m(\mathbf{q}, \mathbf{c}, t) = \mathbf{q} + \boldsymbol{\alpha}_x^{k,m} \odot L_{x,m}\Big(\mathbf{q} \odot (1 + \boldsymbol{\beta}_x^{k,m}) + \boldsymbol{\gamma}_x^{k,m}, \mathbf{c}\Big) \tag{14}$$

where $L_{x,m}(\cdot)$ is the transformer module (self-attention, cross-attention, or feed-forward network). Additionally, conditioning input $\mathbf{c} = \varnothing$ for $x \in \text{sa}, \text{ffn}$, and $\mathbf{c} = \mathbf{I}$ for $x = \text{ca}$. The time-dependent modulation parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ are computed as,

$$(\boldsymbol{\alpha}_x^{k,m}, \boldsymbol{\beta}_x^{k,m}, \boldsymbol{\gamma}_x^{k,m}) = f_{x,m}(t) \tag{15}$$

where $f_{x,m}(t)$ is a learned neural network. After M iterations of denoising, the set of clean estimate $\{\hat{\mathbf{s}}_i^k\}$ are obtained by passing the output of the last layer $\{\mathbf{q}_i^{k,M}\}$ through a prediction network.

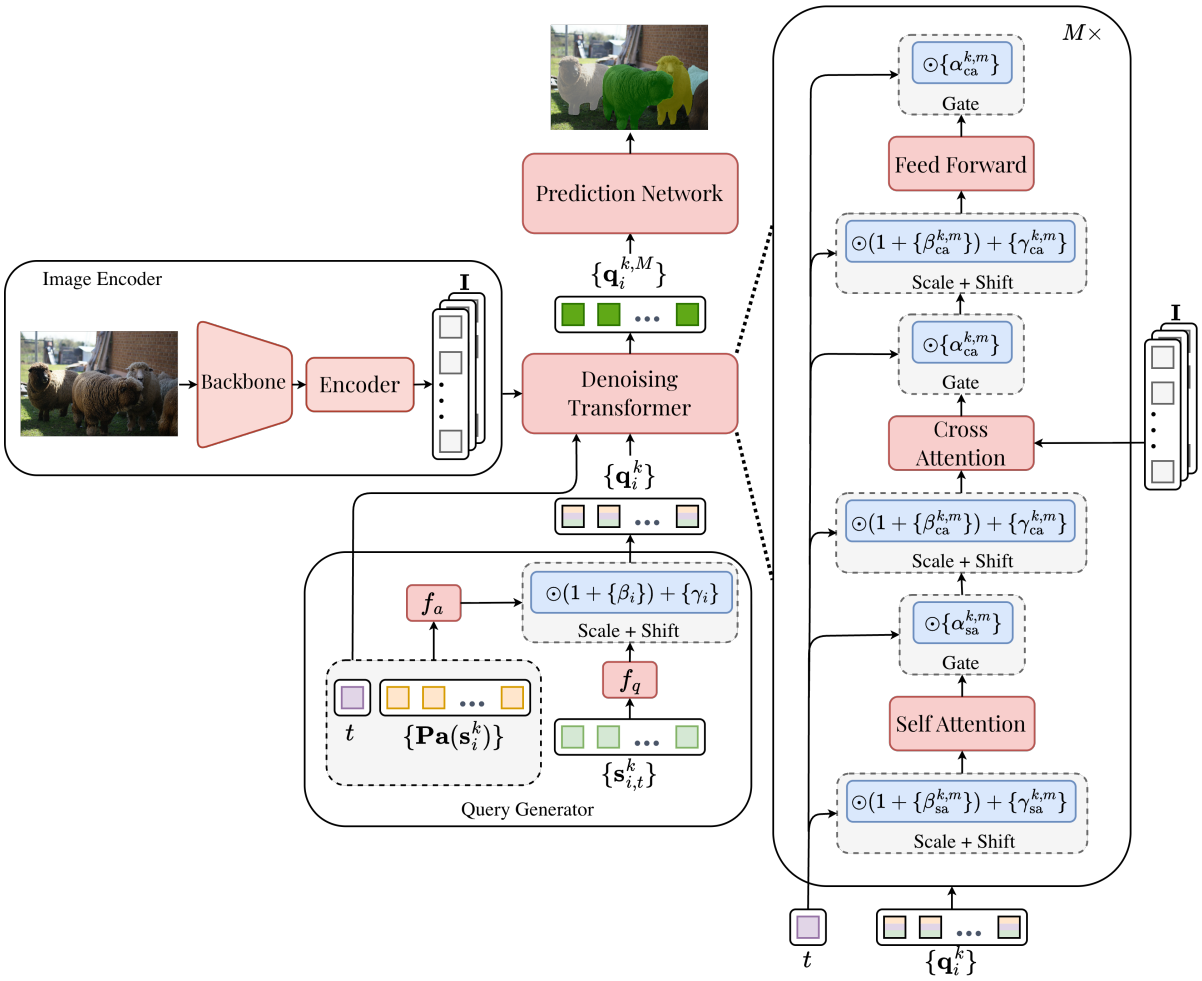


Figure 4: **Transformer-based Architecture.** A simple transformer-based instantiation of the proposed MDM framework that consists of – (i) an image encoder that extracts multi-scale features from the input image, (ii) a query generation module that maps noisy inputs into queries for the denoising transformer, (iii) a stack of M time-conditioned denoising decoder layers that iteratively refine representations through attention and feed-forward operations, and (iv) a prediction head that produces the final structured outputs.

4.5 Training

Each conditional diffusion process $\Pr(\{\mathbf{s}_i^k\} \,|\, \{\mathbf{Pa}(\mathbf{s}_i^k)\}, \mathbf{I})$ can be trained in an end-to-end fashion. Note that, following transformer-based set prediction frameworks (Carion et al., 2020; Zhu et al., 2020), we use a fixed number of queries Q_{train} during training. As the number of ground-truth elements is typically significantly smaller than Q_{train}, the ground-truth set is padded with background (no-object) entries. For instance, in instance segmentation, this padding consists of random boxes, a background label, and zero-valued segmentation masks. The noisy input $\{\mathbf{s}_{i,t}^k\}$ for each diffusion process is then computed over this padded set.

As the denoised estimates $\{\hat{\mathbf{s}}_i^k\}$ obtained from a diffusion process form an unordered set, we adopt a bipartite matching algorithm (Carion et al., 2020) to align them with the ground truth set $\{\mathbf{s}_i^k\}$. However, as most structured predicted tasks involves correlated components, computing this matching independently for each element can be suboptimal. For example. matching based on boxes alone can be ambiguous when multiple boxes are close or overlapping, potentially assigning the wrong ground-truth object and producing inconsistent conditioning for subsequent labels and mask diffusion processes. To mitigate this, we additionally compute estimates for the other components of the structured output, $\{\hat{\mathbf{a}}_i^v\}, v \neq k$, by passing the output queries

$\{\mathbf{q}_i^{k,M}\}$ through the corresponding prediction networks. These estimates can be divided into two sets – (i) refinements over previous predictions $v \in \mathbf{Pa}(\mathbf{s}_i^k)$, and (ii) auxiliary outputs $v \notin \mathbf{Pa}(\mathbf{s}_i^k)$. It should be noted that the auxiliary outputs are only computed to guide model training and are *not* passed as conditioning to subsequent diffusion processes.

We define the joint matching cost $\mathcal{L}_\sigma$ for a given matching σ as

$$\mathcal{L}_\sigma = \left[\sum_i \mathcal{L}^k \left(\hat{\mathbf{s}}_i^k, \mathbf{s}_{\sigma(i)}^k \right) + \sum_i \mathcal{L}^{\text{diff}} \left(\hat{\mathbf{s}}_i^k, \mathbf{s}_{\sigma(i)}^k \right) + \sum_{v \neq k} \sum_i \mathcal{L}^v \left(\hat{\mathbf{a}}_i^v, \mathbf{s}_{\sigma(i)}^v \right) \right]. \tag{16}$$

where $\mathcal{L}^k$ is the function that defines the loss for component k (*e.g.* combination of L-1 and generalized IoU loss for the boxes), and $\mathcal{L}^{\text{diff}}$ is the diffusion loss (Eq. 8 in case of Gaussian diffusion). A matching algorithm computes optimal permutation $\widehat{\sigma}$ as,

$$\widehat{\sigma} = \arg\min_{\sigma} \mathcal{L}_\sigma. \tag{17}$$

The choice of matching algorithm is task-dependent. For instance, in object detection and instance segmentation, the Hungarian algorithm is used (Carion et al., 2020), whereas for scene graph generation a multi-assignment heuristic can be employed (Kim et al., 2024).

Finally, to improve convergence, we apply the loss $\mathcal{L}_{\widehat{\sigma}}$ to the outputs from every intermediate layer, as is the case in Carion et al. (2020); Li et al. (2023); Cheng et al. (2022). Specifically, for each intermediate decoder layer $m < M$, we compute the corresponding denoised and auxiliary outputs from $\{\mathbf{q}_i^{k,m}\}$, determine an optimal matching, and apply the joint matching loss.

Our proposed MDM framework allows each diffusion network corresponding to a conditional distribution (Eq. 9) to be trained independently. In practice, however, this would require maintaining separate parameters for both the image encoders and denoising decoders for each task, which can be impractical under memory constraints. To address this, we share the encoder and denoising decoder weights across all tasks to reduce memory overhead. Task-specific behavior is induced through projection parameters f_q (Eq. 12) and $f_{x,m}$ (Eq. 13), which modulate the query representations. This design enables the shared parameters to operate across tasks while still allowing each conditional diffusion process to retain task-specific conditioning. An additional benefit of this parameter sharing is improved training and inference speeds as the encoder needs to compute image features $\mathbf{I}$ only once per batch, which can then be reused across tasks during both training and inference.

4.6 Inference

As described in Section 4.1, inference in the proposed MDM framework is performed via ancestral sampling over the underlying DAG. However, as noted in Song et al. (2020), naively traversing the full Markov chain during sampling can be computationally expensive when the number of diffusion timesteps T is large. As such, we adopt the DDIM sampling procedure (Song et al., 2020), which accelerates inference by transforming the original diffusion process into a non-Markovian process, allowing sampling with substantially fewer timesteps.

Furthermore, since each conditional distribution in the DAG is modeled as a separate diffusion process, the number of sampling steps can be chosen separately for each component. For example, in an instance segmentation model trained using the proposed MDM framework, the box diffusion process can be run for T^{box} steps, the label diffusion can be run for T^{label} steps, and the mask diffusion can be run for T^{mask} steps, enabling fine-grained control over the generation process.

5 Experiments

We evaluate the proposed MDM framework on three structured prediction tasks: object detection, instance segmentation, and scene graph generation. While MDM achieves competitive performance across all tasks, our primary objective is to emphasize two key points: (i) the framework's generality, demonstrating its

	#	Method					$\mathbf{AP}^{\mathbf{mask}}$	$\mathbf{AP}^{\mathbf{box}}$	$\mathbf{AP}^{\mathbf{mask}}_{50}$	$\mathbf{AP}^{\mathbf{mask}}_{75}$	$\mathbf{AP}^{\mathbf{mask}}_{s}$	$\mathbf{AP}^{\mathbf{mask}}_{m}$	$\mathbf{AP}^{\mathbf{mask}}_{l}$
Deterministic	①	Mask-RCNN (He et al., 2017)					42.5	48.2	-	-	23.8	45.0	60.0
	②	HTC (Chen et al., 2019a)					39.7	44.9	61.4	43.1	22.6	42.2	50.6
	③	QueryInst (Fang et al., 2021)					40.6	45.6	63.0	44.0	23.4	42.5	52.8
	④	Mask2Former (Cheng et al., 2022)					43.7	46.2	66.0	46.9	23.4	47.2	64.8
	⑤	Mask DINO (Li et al., 2023)					**46.0**	50.5	68.9	**50.3**	26.0	**49.3**	65.5
Diffusion	⑥	DiffusionInst; $T = 1$ (Gu et al., 2024)					37.3	-	60.3	39.3	18.9	40.1	54.7
	⑦	DiffusionInst; $T = 4$ (Gu et al., 2024)					37.5	-	60.9	39.3	19.2	40.4	54.8
			Q_{eval}	T^{box}	T^{label}	T^{mask}							
	⑧	MDM	300	1	-	-	45.0	49.1	67.7	48.9	25.0	48.4	65.8
	⑨			4	-	-	44.9	49.1	68.1	48.5	25.0	48.3	65.1
	⑩			4	1	-	45.7	50.3	68.9	49.7	26.1	49.0	65.8
	⑪			4	4	-	45.8	50.5	68.9	49.8	26.2	49.1	66.0
	⑫			4	4	1	45.8	50.6	68.9	49.8	26.4	49.1	66.0
	⑬			4	4	4	45.9	50.7	**68.9**	49.9	**26.5**	49.1	**66.1**
	⑭		100	4	4	4	45.1	49.5	68.0	48.8	25.5	48.4	65.8
	⑮		600				45.9	**50.9**	68.8	50.1	26.1	49.2	65.9
	⑯		900				45.8	50.8	68.6	49.9	26.0	49.1	65.7

Table 1: **Instance segmentation.** Results on MS-COCO *val2017* with ResNet-50 backbone. The best results are shown in **bold**, and the second-best results are underlined. MDM ⑬ achieves performance on par with the strong baseline ⑤. Note that although MDM is trained with 300 queries, it can operate over different query budgets (Q_{eval}) ⑭-⑯ without any additional training.

applicability across a diverse set of structured prediction problems, and (ii) the effectiveness of a simple transformer-based instantiation, which attains strong results without relying on task-specific architectural components, data augmentation, or optimization techniques.

5.1 Object Detection and Instance Segmentation

Dataset. We evaluate performance on the MS-COCO 2017 dataset (Lin et al., 2014). MS-COCO contains approximately 160K images across 80 object categories, split into 118K training images, 5K validation images, and 41K test images. Since ground-truth annotations for the test set are not publicly available, following prior work (Li et al., 2022a), we report results on the validation set. Performance is measured using average precision (AP) across multiple IoU thresholds and object scales.

Implementation Details. We train the transformer-based framework described in Section 4.4 on the task of object detection using the MDM formulation in Section 4.2. For fair comparison with prior methods, we adopt a ResNet-50 backbone (He et al., 2016) as the feature extractor in the image encoder. The encoder comprises 6 transformer layers, while the denoising decoders use 6 layers for object detection and 9 layers for instance segmentation. As discussed in Section 4.5, parameters are shared across all denoising decoders, resulting in a model with a comparable number of parameters to the baseline methods. Finally, we set $Q_{\text{train}} = 300$ and, following the baseline in (Chen et al., 2023a), employ a box-renewal strategy (Chen et al., 2023a) during box diffusion inference to better align the inference procedure with training.

Results – Instance Segmentation. We evaluate MDM for instance segmentation on MS-COCO in Table 1. Our approach demonstrates a significant performance advantage over existing diffusion-based methods ⑥-⑦ and achieves parity with state-of-the-art deterministic segmentors ⑧. Specifically, MDM ⑬ outperforms DiffusionInst (Gu et al., 2024) ⑦ – which only applies diffusion to bounding boxes – by 7.4 AP^{mask} using the same number of box refinement steps ($T^{\text{box}} = 4$). By further incorporating label and mask diffusions, MDM extends this margin to 8.4 AP^{mask} ⑬, highlighting the effectiveness of our factorized approach.

When compared to determinstic methods, MDM achieves performance on-par with the strong baseline in Mask DINO (Li et al., 2023) ⑤. Beyond raw metrics, the generative nature of MDM offers a distinct advantage over deterministic models – Mask DINO is limited to a single deterministic point-estimate prediction. In contrast, MDM attempts to model the underlying data distribution, enabling it to sample multiple plausible outputs for the same input image. This allows our model to handle uncertainty and resolve ambiguous

	#	Method	Q_{eval}	T^{box}	T^{label}	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Deterministic	1	RetinaNet (Lin et al., 2017)				38.7	58.0	41.5	23.3	42.3	50.3
	2	Faster R-CNN (Ren et al., 2015)				40.2	61.0	43.8	24.2	43.5	52.0
	3	DETR (Carion et al., 2020)				42.0	62.4	44.2	20.5	45.8	61.1
	4	Conditional-DETR (Meng et al., 2021)				43.0	64.0	45.7	22.7	46.7	61.5
	5	Cascade R-CNN (Wang et al., 2022)				44.2	64.7	47.5	24.7	48.2	60.6
	6	Cascade R-CNN (Cai & Vasconcelos, 2018)				44.3	62.2	48.0	26.6	47.7	57.7
	7	Sparse R-CNN (Sun et al., 2021)				45.0	63.4	48.2	26.9	47.2	59.5
	8	Efficient-DETR (Yao et al., 2021)				45.1	63.1	49.1	28.3	48.4	59.0
	9	SMCA-DETR (Gao et al., 2021)				45.6	65.5	49.1	25.9	49.3	62.6
	10	Deformable-DETR (Zhu et al., 2020)				46.2	65.2	50.0	28.8	49.2	61.7
	11	DAB-DETR† (Liu et al., 2022)				46.9	66.0	50.8	30.1	50.4	62.5
	12	DN-DETR† (Li et al., 2022a)				48.6	67.4	52.7	31.0	52.0	63.7
	13	DINO-DETR† (Zhang et al., 2022)				**50.9**	69.0	55.3	<u>34.6</u>	**54.1**	64.6
Diffusion	14	DiffusionDet; $T = 1$ (Chen et al., 2023a)				45.8	64.1	50.4	27.6	48.7	62.2
	15	DiffusionDet; $T = 4$ (Chen et al., 2023a)				46.6	65.1	51.3	28.9	49.2	62.1
	16	MDM	300	1	-	49.3	67.7	53.9	32.9	52.8	64.0
	17			4	-	49.2	68.9	52.9	33.2	52.1	63.5
	18			4	1	50.3	**69.4**	54.7	34.5	53.5	64.4
	19			4	4	50.5	<u>69.2</u>	55.0	34.3	53.5	64.8
	20		100	4	4	49.4	68.1	53.6	33.5	52.2	63.9
	21		600			50.7	69.2	<u>55.4</u>	34.5	53.8	**65.1**
	22		900			<u>50.7</u>	69.0	**55.5**	**34.7**	<u>53.9</u>	<u>64.9</u>
Not Directly Comparable											
	23	RT-DETR (Zhao et al., 2024)				53.1	71.3	57.7	34.8	58.0	70.0
	24	RT-DETRv2 (Lv et al., 2024)				53.4	71.6	57.4	36.1	57.9	70.8
	25	DEIM-RT-DETR (Huang et al., 2025)				54.3	72.3	58.8	37.5	58.7	70.8

Table 2: **Object detection.** Results on MS-COCO *val2017* with a ResNet-50 backbone. † denotes models trained with an effective query count of 900. Unlike deterministic methods that require retraining to increase the number of queries, MDM trained with 300 queries can be evaluated with 900 queries (22) without retraining. The best results are shown in **bold**, and the second-best results are <u>underlined</u>. Methods in (23)–(25) are not directly comparable, as they rely on specialized architectural components that are orthogonal to our formulation. Notably, these components can be incorporated into MDM to provide further improvements.

cases in complex scenes that a single-pass deterministic model might overlook. A qualitative analysis of this behavior is provided in Section 5.3.

Results – Object Detection. We contrast MDM against existing detection methods in Table 2. MDM (16) - (20) consistently outperforms most deterministic baselines (1) – (12) and all diffusion-based counterparts (14) – (15) across all metrics. Additionally, we achieve performance on-par with the competitive baseline in DINO-DETR (Zhang et al., 2022) ((13)), which represents the most comparable baseline to our specific transformer-based implementation (Section 4.4). Additionally, MDM benefits from a generative formulation (as discussed in the instance segmentation setting), enabling it to model the underlying data distribution and produce multiple plausible predictions.

Another mode of comparison is the number of object queries used at inference. Deterministic baselines operate under a fixed query budget (typically 300 queries). Recently, several high-performing variants (Liu et al., 2022; Li et al., 2022a; Zhang et al., 2022) (11)–(13) report results using an effective query count of 900. Although increasing the number of queries generally improves detection performance, this typically requires retraining,

which is computationally expensive and time-consuming. In contrast, our proposed MDM framework can scale to an arbitrary number of queries at inference without any retraining. As shown in ◇22, a model trained with 300 queries can be directly evaluated in the 900-query regime, yielding consistent performance without additional optimization. This provides a more flexible mechanism for test-time performance improvement, enabling a controllable trade-off between inference-speed and accuracy within a single trained model.

Compared to the diffusion-based approach DiffusionDet (Chen et al., 2023a) ◇15, which models diffusion only over bounding boxes, our approach ◇17 provides a 2.6 AP improvement under identical number of number of box refinement steps ($T^{\text{box}} = 4$). This margin increases to 3.9 AP when additionally running the label diffusion ◇19, thus highlighting the effectiveness of our proposed factorization that models labels and boxes separately.

It should be noted that although recent state-of-the-art methods like RT-DETR (Zhao et al., 2024) ◇21 - ◇22 and DEIM-RT-DETR (Huang et al., 2025) ◇23 report higher absolute numbers, these methods utilize specialized architectural components – such as hybrid encoders and optimized matching strategies. As MDM provides a general diffusion-based framework for modeling structured tasks, these design choices are orthogonal to our formulation and can be incorporated into a specific instantiation of MDM to improve performance.

5.2 Scene Graph Generation

Dataset. We evaluate performance on the Visual Genome benchmark dataset (Krishna et al., 2017), using the widely adopted preprocessed subset introduced by Xu et al. (2017). This subset comprises 108k images, spanning 150 object categories and 50 relation labels.

Evaluation Protocol. To measure performance, we report results on standard scene graph evaluation metrics, namely Recall (R@K) (Xu et al., 2017), Mean Recall (mR@K) (Chen et al., 2019b; Tang et al., 2019), and Harmonic Recall (hR@K) (Khandelwal & Sigal, 2022). Additionally, we also report the mR@100 performance on each long-tail category subset, namely *head*, *body*, and *tail* (Li et al., 2021). R@K is computed in a class agnostic manner, whereas mR@K measures the average recall across predicate classes independently. However, as noted in Khandelwal & Sigal (2022), there exists an inherent tradeoff between R@K and mR@K – mR@K can be improved at the expense of R@K, and vice versa, depending on the strength of the biasing towards long-tail predicate classes. Existing approaches typically control this trade-off through data sampling (Desai et al., 2021; Li et al., 2021) or loss reweighting (Yan et al., 2020; Khandelwal & Sigal, 2022) techniques. However, these biasing mechanisms are fixed during training, making the trade-off difficult to adjust without additional optimization.

To avoid expensive retraining, we instead adopt a simple test-time biasing strategy (Menon et al., 2020) that enables flexible control over the trade-off between R@K and mR@K at inference time. Specifically, let $\widetilde{\mathbf{l}}^p \in \mathbb{R}^C$ be the unnormalized logits over predicate classes. Following Khandelwal & Sigal (2022), we define a class weight w_c as $\max\left\{(\eta/f_c)^\delta, 1.0\right\}$ where f_c is the frequency of the predicate class $c \in C$ in the training set, and η, δ are scaling parameters. Following the best performing variants in Khandelwal & Sigal (2022); Kim et al. (2024); Fu et al. (2025), we set $\eta = 0.07, \delta = 0.75$. Intuitively, less frequent predicate classes are assigned larger weights, thereby compensating for the long-tail imbalance in the data distribution. We then define the biased distribution $\mathbf{l}^p$ as,

$$\mathbf{l}^p[c] = \sigma\left(\widetilde{\mathbf{l}}^p[c] + \lambda \log w_c\right) \tag{18}$$

where λ controls the biasing strength. Increasing λ amplifies the contribution of infrequent predicate classes, thus improving mR@K at the cost of R@K, while smaller values favor frequent predicates, leading to higher overall recall. Importantly, since this adjustment is applied directly to the predicted logits at inference time, the trade-off can be tuned without any additional training or modification to the learned model parameters.

Implementation Details. Scene graph generation can be formulated as predicting a set of `<subject, object, predicate>` triplets. Accordingly, recent methods (Khandelwal & Sigal, 2022; Kim et al., 2021; Fu et al., 2025) adopt transformer-based architectures in which each component of the triplet is predicted by a separate decoder. To ensure a fair comparison with prior works we set $Q_{\text{train}} = 300$, use a ResNet-101

	B	#	Method	Q_{eval}	λ	T^{box}	T^{label}	mR@50/100	R@50/100	hR@50/100	Head	Body	Tail
Deterministic	X101-FPN	1	RelDN (Zhang et al., 2019)					6.0 / 7.3	31.4 / 35.9	10.1 / 12.1	-	-	-
		2	MOTIF (Zellers et al., 2018)					5.5 / 6.8	32.1 / 36.9	9.4 / 11.5	-	-	-
		3	VCTree (Tang et al., 2019)					6.6 / 7.7	31.8 / 36.1	10.9 / 12.7	-	-	-
		4	BGNN (Li et al., 2021)					10.7 / 12.6	31.0 / 35.8	15.9 / 18.6	**34.0**	12.9	6.0
		5	VCTree-TDE (Tang et al., 2020)					9.3 / 11.1	19.4 / 23.2	12.6 / 15.0	-	-	-
		6	VCTree-DLFE (Chiou et al., 2021)					11.8 / 13.8	22.7 / 26.3	15.5 / 18.1	-	-	-
		7	VCTree-EBM (Suhail et al., 2021)					9.7 / 11.6	20.5 / 24.7	13.2 / 15.8	-	-	-
		8	VCTree-BPLSA (Guo et al., 2021)					13.5 / 15.7	21.7 / 25.5	16.6 / 19.4	-	-	-
		9	DT2-ACBS (Desai et al., 2021)					**22.0** / **24.4**	15.0 / 16.3	17.8 / 19.5	-	-	-
		10	PE-Net (Zheng et al., 2023)					12.4 / 14.5	30.7 / 35.2	17.7 / 20.5	-	-	-
		11	VETO (Sudhakaran et al., 2023)					10.6 / 13.8	28.6 / 34.0	15.5 / 19.6	-	-	-
		12	DRM (Li et al., 2024)					20.4 / <u>24.1</u>	19.0 / 22.9	19.7 / 23.5	-	-	-
	ResNet 50	13	RelTR (Cong et al., 2023)					10.8 / 12.3	27.5 / 30.7	15.5 / 17.6	-	-	-
		14	EGTR (Im et al., 2024)					17.8 / 21.7	18.7 / 20.5	18.2 / 21.1	-	-	-
		15	Hydra-SGG (Chen et al., 2024)					15.9 / 19.4	28.6 / 33.4	20.5 / 24.7	-	-	-
	ResNet-101	16	BGNN (Li et al., 2021; 2022b)					8.6 / 10.3	28.2 / 33.8	13.2 / 15.8	29.1	12.6	2.2
		17	RelDN (Zhang et al., 2019; Li et al., 2022b)					4.4 / 5.4	30.3 / 34.8	7.7 / 9.3	31.3	2.3	0.0
		18	AS-Net (Chen et al., 2021)					6.1 / 7.2	18.7 / 21.1	9.2 / 10.7	19.6	7.7	2.7
		19	HOTR (Kim et al., 2021)					9.4 / 12.0	23.5 / 27.7	13.4 / 16.7	26.1	16.2	3.4
		20	SGTR (Li et al., 2022b)					15.8 / 20.1	20.6 / 25.0	17.9 / 22.3	21.7	21.6	17.1
		21	ISG* (Khandelwal & Sigal, 2022)					15.7 / 17.8	27.2 / 29.8	19.9 / 22.3	28.5	18.8	13.3
		22	SpeaQ*$_{\text{ISG}}$ (Kim et al., 2024)					15.1 / 17.6	32.1 / 35.5	20.5 / 23.5	-	-	-
		23	HQSG*‡ (Fu et al., 2025)					16.0 / 20.5	**34.1** / **38.3**	21.8 / 26.7	-	-	-
Diffusion	ResNet-101	24	MDM	300	0.4	1	-	18.3 / 21.0	29.2 / 32.8	22.5 / 25.6	28.4	26.1	13.8
		25				4	-	18.6 / 21.8	30.2 / 34.3	23.0 / 26.7	29.3	27.2	14.3
		26				4	1	19.3 / 22.7	30.2 / 34.5	23.5 / 27.4	30.1	28.2	15.0
		27				4	4	19.7 / 22.9	30.2 / 34.6	23.8 / 27.6	30.2	28.1	15.6
		28		100		4	4	18.8 / 21.3	28.2 / 32.0	22.6 / 25.6	27.9	26.5	14.3
		29		600				20.1 / 23.5	31.2 / 35.7	<u>24.4</u> / <u>28.3</u>	31.0	<u>29.0</u>	15.8
		30		900				20.2 / 23.5	31.5 / 36.1	**24.6** / **28.5**	31.2	**29.2**	15.6
		31		300	0.0	4	4	13.5 / 15.9	<u>32.9</u> / <u>37.3</u>	19.1 / 22.3	32.2	23.6	9.8
		32			0.1			15.3 / 17.9	32.8 / 37.3	20.9 / 24.2	<u>33.8</u>	23.8	7.2
		33			0.2			16.7 / 19.7	32.3 / 36.7	22.0 / 25.6	32.9	25.8	9.7
		34			0.3			18.3 / 21.5	31.6 / 36.1	23.2 / 26.9	31.8	27.2	12.6
		35			0.5			<u>20.7</u> / 24.0	27.7 / 31.7	23.7 / 27.3	27.2	28.7	**18.5**

Table 3: **Scene Graph Generation on Visual Genome.** Mean Recall (mR@K), Recall (R@K), and Harmonic Recall (hR@K) shown for baselines and our approach on the Visual Genome (Krishna et al., 2017) test set. The best results are shown in **bold**, and the second-best results are <u>underlined</u>. * denotes model trained using the loss-reweighting strategy proposed in (Khandelwal & Sigal, 2022). ‡ denotes model trained with $Q_{\text{train}} = 600$. MDM trained with 300 queries can be evaluated with 600 queries ◇29 without retraining.

backbone (He et al., 2016) as the feature extractor in the image encoder, and instantiate the MDM framework with a denoising decoder consisting of three separate decoders for subject, object, and predicate, each with 6 layers. It should be noted that architecturally each decoder is similar to the formulation described in Section 4.4, and we do not introduce any task-specific components. We provide additional details in the Appendix (Section A.1).

Results. As highlighted in Table 3, MDM 27 outperforms deterministic baselines on the Visual Genome test set, demonstrating strong performance across head, body, and tail categories. Compared to baselines 1–12 using the stronger ResNeXt-101-FPN backbone (Xie et al., 2017), MDM 27 outperforms the best baseline in Li et al. (2024) 12 by 4.1/4.1 on hR@50/100. When compared to baselines 16–22 using the same ResNet-101 backbone (He et al., 2016), MDM 27 consistently achieves better overall performance, improving over the strong baseline in Kim et al. (2024) 22 by 3.3/4.1 on hR@50/100.

It should be noted that the most competitive baseline in HQSG Fu et al. (2025) 23 is trained with 600 input queries ($Q_{\text{train}} = 600$). As highlighted in Section 5.1, MDM can seamlessly function across a wide range of

Method	T			Box						Mask					
				AP	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_s$	**AP**$_m$	**AP**$_l$
Joint Diffusion	1			46.9	66.6	50.8	30.0	49.9	62.3	43.1	65.4	46.7	23.1	46.3	63.2
	4			45.7	65.9	49.1	29.4	48.4	61.6	42.2	64.6	45.4	22.7	45.7	62.5
MDM	T^{box}	T^{label}	T^{mask}												
	1	-	-	49.1	68.6	53.3	32.7	52.2	65.0	45.0	67.7	48.9	25.0	48.4	65.8
	4	-	-	49.1	69.5	52.7	32.5	51.9	64.3	44.9	68.1	48.5	25.0	48.3	65.1
	4	1	-	50.3	70.3	54.4	34.2	53.6	65.5	45.7	68.9	49.7	26.1	49.0	65.8
	4	4	-	50.5	70.2	54.8	34.1	53.6	66.3	45.8	68.9	49.8	26.2	49.1	66.0
	4	4	1	<u>50.6</u>	**70.3**	<u>55.0</u>	<u>35.3</u>	<u>53.8</u>	<u>66.3</u>	<u>45.8</u>	<u>68.9</u>	<u>49.8</u>	<u>26.4</u>	<u>49.1</u>	<u>66.0</u>
	4	4	4	**50.7**	<u>70.2</u>	**55.0**	**35.4**	**53.8**	**66.5**	**45.9**	**68.9**	**49.9**	**26.5**	**49.1**	**66.1**

Table 4: **Drawbacks of Joint Modeling.** Results on MS-COCO *val2017* with ResNet-50 backbone. Both formulations use a 6-layer encoder, a 9-layer denoising decoder, and $Q_{\text{train}} = Q_{\text{eval}} = 300$. In "Joint Diffusion", boxes, labels, and masks are corrupted simultaneously, and passed to the decoder to be denoised jointly. The best results are shown in **bold**, and the second-best results are <u>underlined</u>. Note that MDM has a comparable number of parameters to the "Joint Diffusion" model.

input queries without any additional training. As such, when evaluated under the 600 query setting, MDM [29] outperforms the baseline in HQSG, achieving a 2.6/1.6 improvement on hR@50/100.

Additionally, existing approaches such as ISG [21], SpeaQ$_{\text{ISG}}$ [22], and HQSG [23] utilize training-time strategies to facilitate the trade off between recall and mean recall. As such, making any changes to this trade off requires expensive retraining. MDM, on the other hand, can employ the simple test-time biasing strategy described in 18 to allow effectively function across a wide range of recall and mean recall values without additional optimization ([31] - [35]). Specifically, by tuning the parameter λ during inference, a single trained MDM model can biased to maximize performance on the rare body and tail categories [35] ($\lambda = 0.5$) or shifted to favor frequent head classes [32] ($\lambda = 0.1$).

5.3 Qualitative Analysis

MDM models the joint distribution over structured outputs, allowing it to produce multiple plausible predictions given a single input image.We illustrate this property for instance segmentation in Figure 6 by comparing outputs from our proposed MDM with those from the deterministic Mask R-CNN model (He et al., 2017). For MDM, given an input image, we draw 1000 samples and aggregate the resulting masks into a heatmap that reflects prediction variability. Regions of higher intensity indicate areas that are consistently predicted across samples, while darker regions highlight uncertainty. We also visualize variation in the predicted bounding boxes by plotting the mean box for each object and shading the region spanned by the boxes with the minimum and maximum areas across samples. The extent of this shaded region reflects the variability in the predictions, with larger regions indicating greater uncertainty in the object's localization.

Contrary to deterministic approaches that generate a fixed prediction, MDM is able to identify challenging instances in at least a subset of samples. Subsequently, it is able to provide an estimate of uncertainty by aggregating information across samples. For example, in Figure 6a, the "bird" on the right is partially occluded by tree leaves. MDM reflects this ambiguity through uncertainty in both localization (the green shaded region), and shape (darker regions in the segmentation heatmap). Similar behavior is observed in Figures 6b (the "giraffe") and 6c (the "skis").

This diversity is more pronounced for the task of scene graph generation, where the combinatorial output space allows for many valid interpretations. As a result, MDM is able to generate distinct but plausible graphs given the same input image. We highlight some examples in Figure 7.

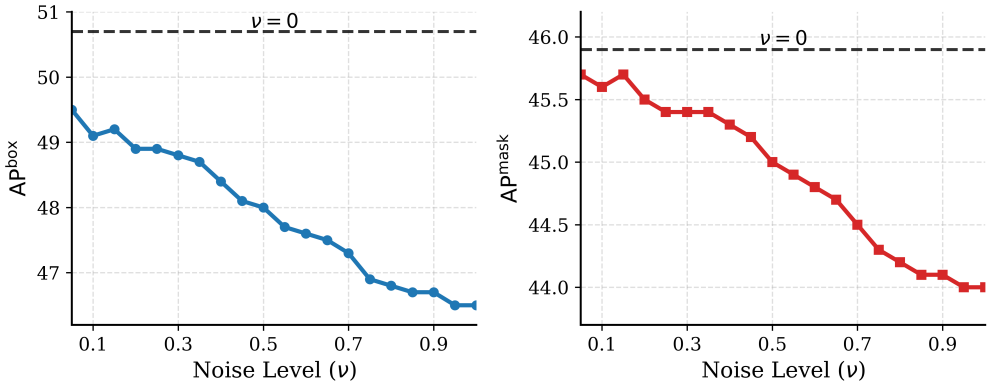


Figure 5: **Robustness to Noise in Conditioning**. $\mathbf{AP^{box}}$ (left) and $\mathbf{AP^{mask}}$ (right) results shown for the MDM model trained on the instance segmentation task. We progressively corrupt conditioning signals using the forward diffusion process with noise level $\nu \in [0, 1]$. Here, $\nu = 0$ corresponds to clean conditioning, while $\nu = 1$ corresponds to fully noised inputs drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Performance degrades smoothly under increasing corruption, indicating that downstream diffusion modules remain robust to errors propagated from earlier stages of ancestral inference.

5.4 Ablation Study

Drawbacks of Joint Diffusion Modeling. For structured prediction tasks, MDM factorizes the joint conditional distribution into a sequence of simpler conditionals, each modeled by a separate diffusion process. As discussed in Section 1, an alternative is to model the full joint distribution using a single diffusion process, where all components are corrupted simultaneously, fed into a denoising network, and jointly denoised.

We empirically compare this joint formulation with MDM in Table 4 on the instance segmentation task. Both approaches use the transformer-based architecture described in Section 4.4, consisting of a 6-layer encoder and a 9-layer denoising decoder. Importantly, as noted in Section 4.5, MDM has a comparable number of parameters to the joint diffusion model due to parameter sharing.

As shown in Table 4, the joint diffusion formulation consistently underperforms MDM across all metrics, and its performance further degrades as the number of diffusion steps increases (*e.g.* box AP drops from 46.9 to 45.7, and mask AP from 43.1 to 42.2 when increasing T from 1 to 4). In contrast, MDM (i) achieves a substantially higher performance (50.7 box AP and 45.9 mask AP), and (ii) shows continual improvement with more diffusion steps. This corroborates our hypothesis that jointly corrupting all components breaks the alignment between boxes, labels, and masks, leading to incoherent intermediate states that are difficult to denoise. As a result, the joint formulation is unable to accurately capture the underlying structured data distribution, as is evidenced by its poor performance with additional denoising steps. MDM, on the other hand, preserves cross-component consistency through its conditional factorization, allowing for improved modeling of the structured output space.

Controllable Inference with MDMs. As MDMs learn a separate diffusion process for each component of a structured representation, during inference, the number of times each diffusion process is run can be independently controlled. As such, this allows flexibility and fine-grained controlled over the generation process without any additional training. We highlight this in Tables 1, 2, and 3.

Although MDM is capable of producing valid structured outputs from each individual diffusion process, chaining these processes together leads to improved performance. For example, in instance segmentation (Table 1), while the box diffusion process by itself is competitive, we observe steady gains when progressively incorporating additional components. Adding label diffusion improves performance by 0.8 AP^{mask} and 1.4 AP^{box} (⑨ → ⑪), and further including mask diffusion yields additional gains of 0.1 on AP^{mask} and 0.2 AP^{box} (⑪ → ⑬). This further highlights the complementary nature and benefits of modeling each component of the structured space as an individual diffusion process.

Impact of Number of Queries. As discussed in Section 4.5, our model is trained with a fixed number of queries (Q_{train}). Unlike deterministic approaches, which require the number of inference queries to match the training setup ($Q_{\text{eval}} = Q_{\text{train}}$), MDMs enable flexible adjustment of the query count without requiring any additional training. As such, we vary the number of queries (Q_{eval}) for each task and report the results in Tables 1, 2, and 3.

The impact of increasing Q_{eval} depends strongly on the structure and complexity of the output space. For problems with relatively constrained output spaces – such as object detection and instance segmentation – where the number of valid predictions scales linearly with the number of objects, a limited number of queries is sufficient to cover most plausible outputs. For example, increasing Q_{eval} from 100 to 300 yields moderate improvements, with gains of 1.1 AP for object detection (◇20 → ◇19) and 0.8 AP^{mask} for instance segmentation (14 → 13). Beyond this performance quickly saturates, and further increasing the number of queries provides diminishing returns, with negligible improvements observed beyond $Q_{\text{eval}} = 600$.

In contrast, scene graph generation is a fundamentally more complex problem, where the output space is combinatorial in the number of objects. In this setting, increasing Q_{eval} substantially improves coverage of plausible relations triplets. For example, going from 100 to 300 queries results in gains of 0.9/1.6 in mR@50/100 and 2.0/2.6 in R@50/100 (28 → 27). This trend continues as Q_{eval} is increased further, where going from 300 to 900 queries provides additional improvements of 0.5/0.6 in mR@50/100 and 1.3/1.5 in R@50/100 (27 → 30).

Importantly, while increasing Q_{eval} improves coverage in complex output spaces, it also increases inference runtime. However, this overhead is sublinear, as a substantial portion of computation is independent of the number of queries (image encoder). In particular, increasing Q_{eval} from 100 to 900 makes inference approximately 1.3× slower.

Robustness to Noise in Ancestral Conditioning. As described in Section 4.6, inference in MDM follows an ancestral sampling procedure, where each diffusion module conditions on the outputs of its parent processes. As a result, errors introduced in earlier stages of the sampling process may propagate to subsequent modules. To explicitly study this effect, we perform a controlled stress test in which the conditioning inputs to each diffusion process are deliberately corrupted. Concretely, we apply the forward diffusion process defined in Eq. (4) to the conditioning variables and perturb them to an intermediate noise level corresponding to timestep νT, where T is the total number of diffusion steps and $\nu \in [0, 1]$ controls the severity of corruption. Note that $\nu = 0$ corresponds to no corruption of the conditioning, while $\nu = 1$ corresponds to fully noised conditioning, *i.e.* samples drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

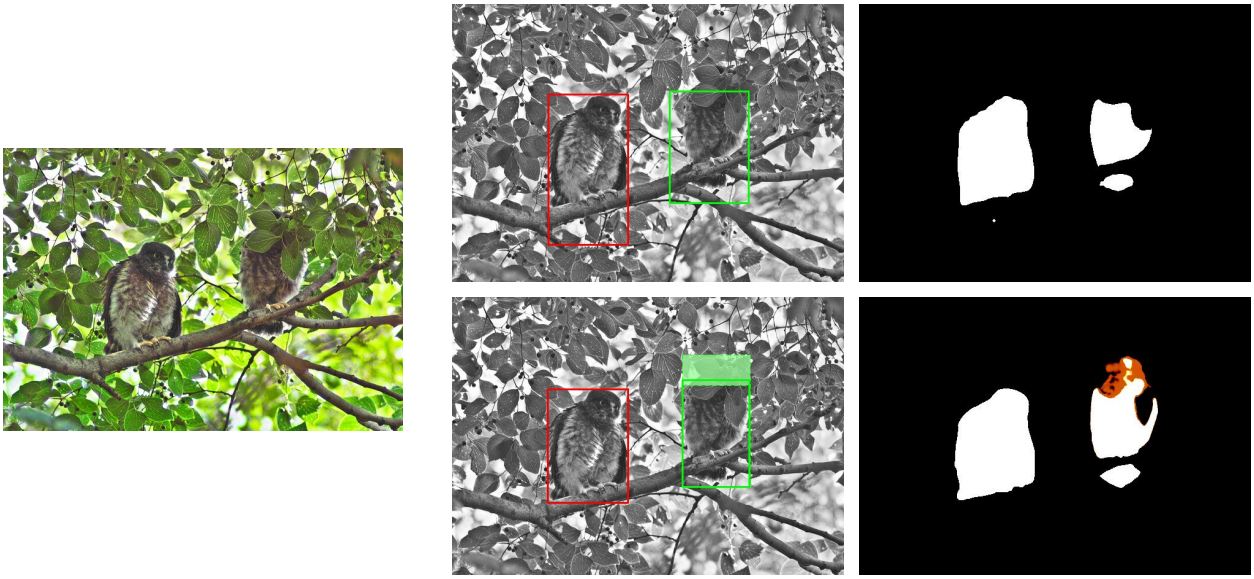
We highlight the results for instance segmentation in Figure 5. Across both box and mask AP, MDM exhibits a smooth and gradual degradation in performance as ν increases, rather than an abrupt collapse. Each diffusion stage is therefore able to recover meaningful structure even when conditioned on heavily corrupted inputs. Notably, even at $\nu = 1$, where the conditioning is fully noised and drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, MDM retains a substantial fraction of its performance, indicating strong robustness to severe upstream corruption in the ancestral inference process.

6 Conclusion

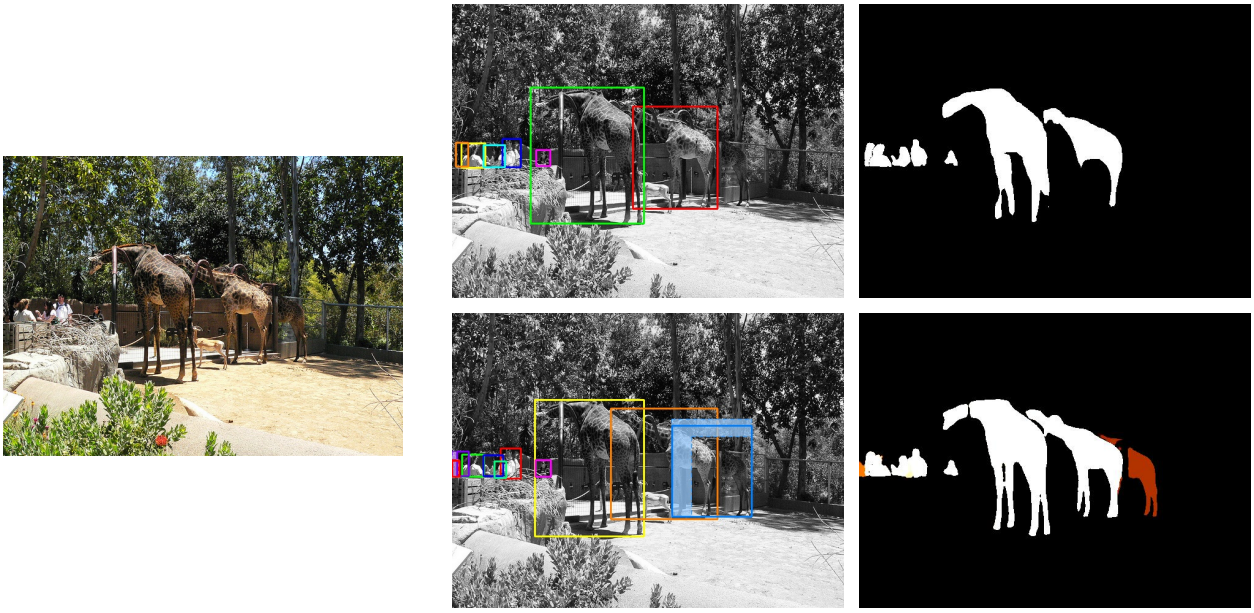
In this work, we propose Modular Diffusion Models (MDMs), a general framework for learning complex distributions in structured learning tasks. Instead of modeling the joint distribution over heterogeneous components directly, our approach factorizes it into a sequence of simpler conditional distributions. We provide a simple transformer-based instantiation of the proposed framework, and highlight the generality and effectiveness of MDMs through extensive experiments across three distinct structured learning problems.

Ethical Considerations. The proposed MDM framework is a general-purpose approach for modeling structured distributions and is not tied to any particular application domain. We argue that improved uncertainty modeling can be beneficial in many vision tasks by providing a richer representation of model predictions and ambiguity. However, as is the case with any AI approach, the framework could be applied in settings such as surveillance, monitoring, or automated decision-making, where prediction errors or biased model behavior may have adverse consequences. Additionally, machine learning models trained on real-world

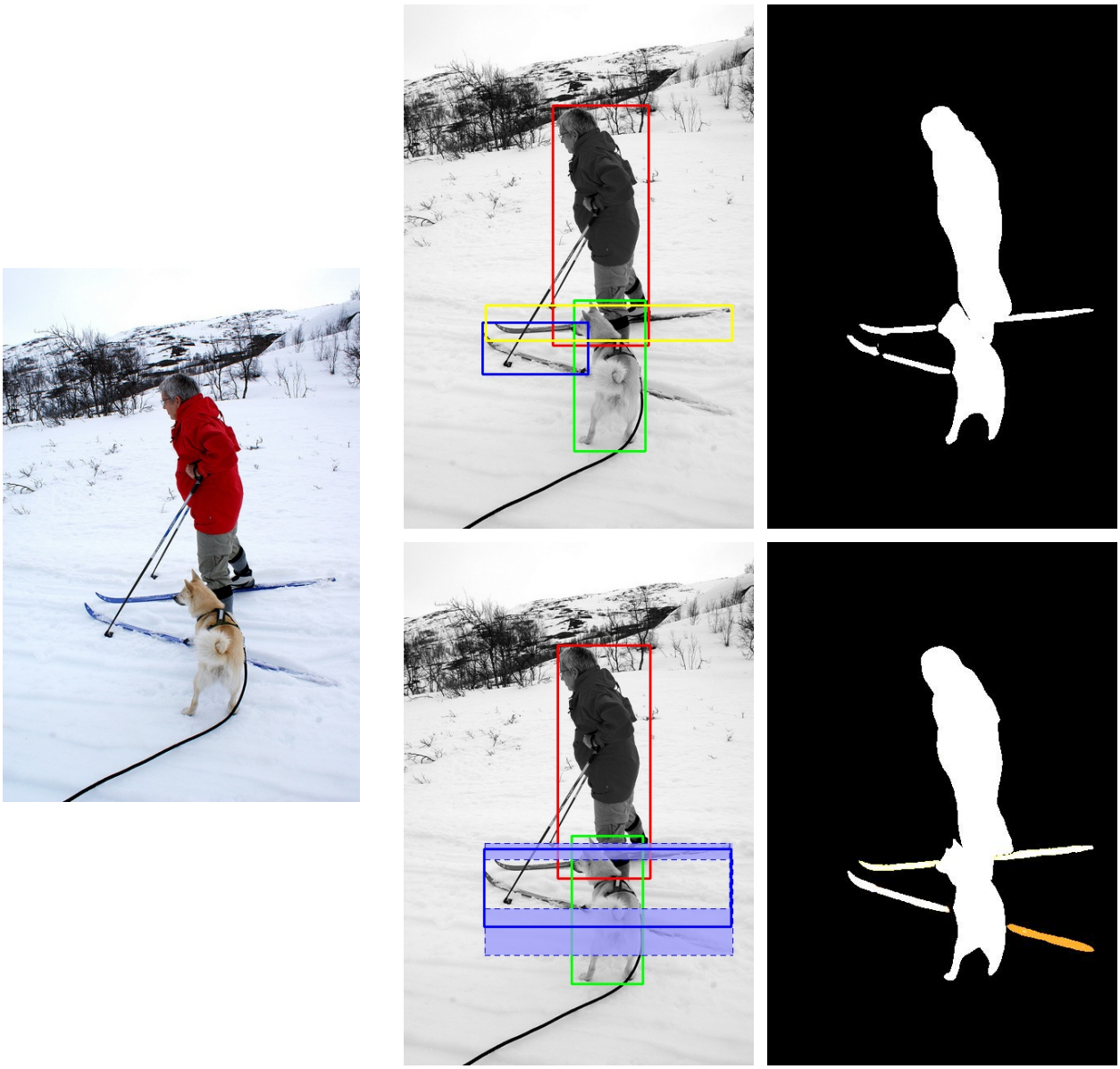
data can inherit and amplify biases present in the training datasets. For example, under representation of certain object categories, environments, or demographics may lead to uneven performance across different groups. This behavior is not unique to MDM and is a challenge faced by many AI systems. Finally, our work does not involve the collection of new human-subject data, and all experiments are conducted using publicly available benchmark datasets commonly used in computer vision research (Lin et al., 2014; Krishna et al., 2017).



(a) The "bird" object on the right is partially occluded by tree leaves. Unlike the deterministic Mask-RCNN (He et al., 2017) (top), MDM (bottom) captures ambiguity in both the "bird's" head location (green) and its shape (darker regions in the heatmap).

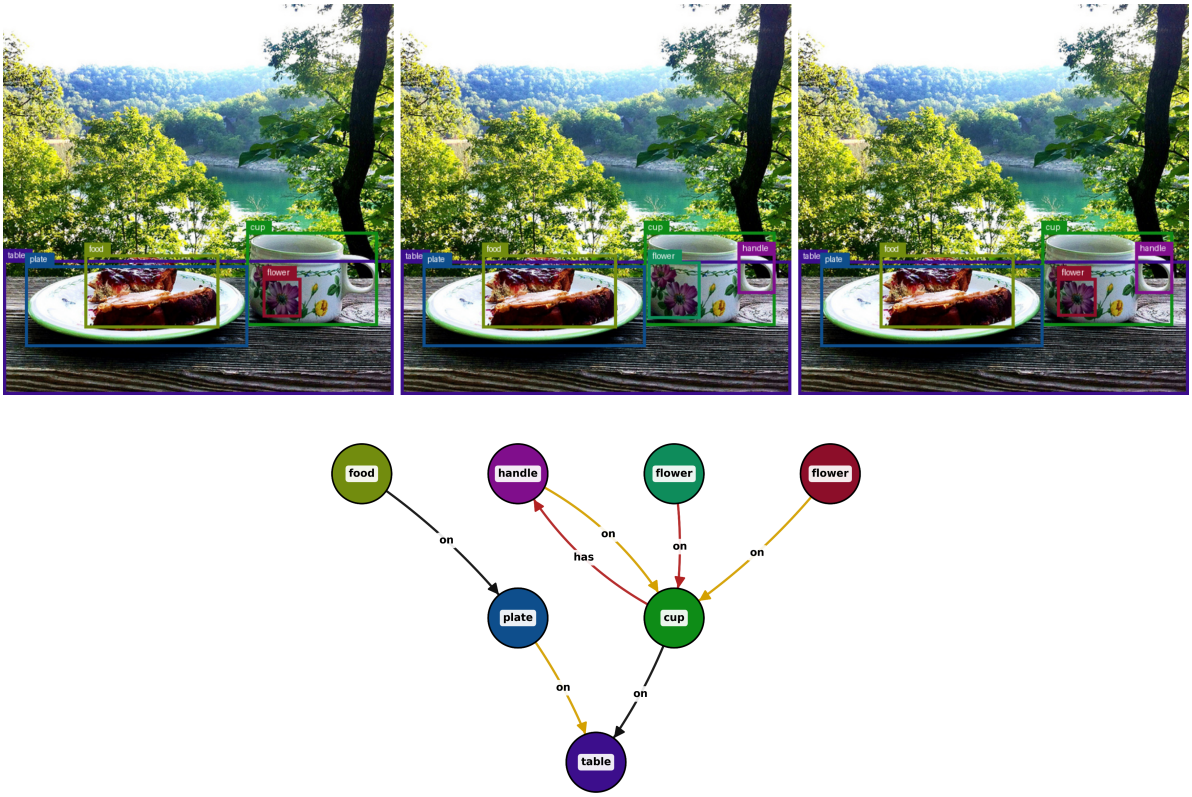


(b) The rightmost "giraffe" is partially occluded by others. Unlike the deterministic Mask R-CNN (He et al., 2017) (top), MDM (bottom) captures uncertainty in localization due to the "giraffe's" upper body (blue).

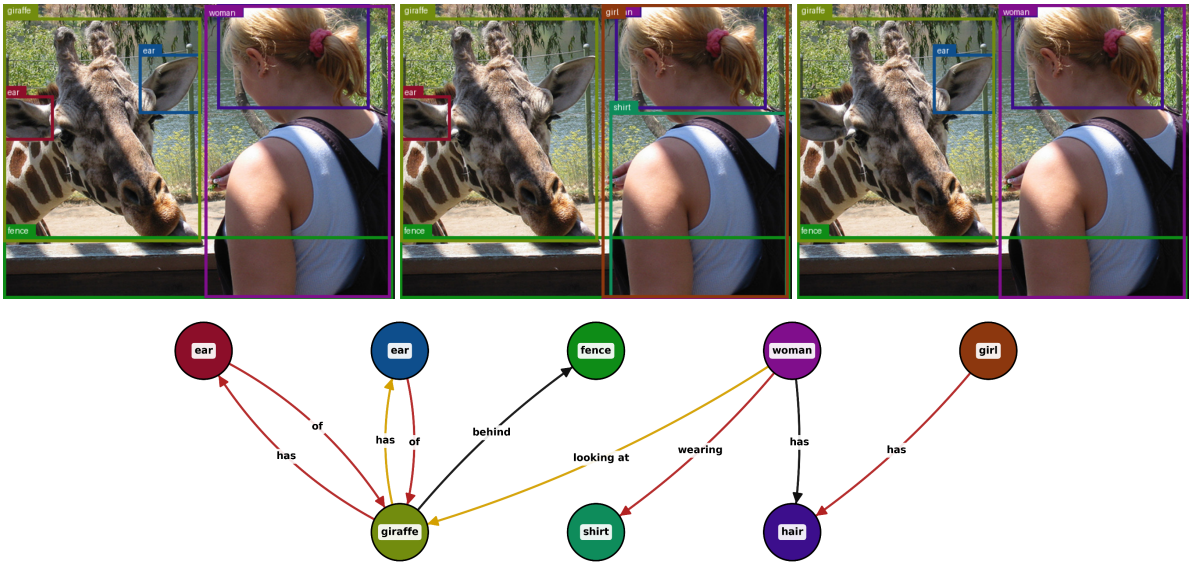


(c) The tail of the left "ski" is covered by the snow. Unlike the deterministic Mask R-CNN (He et al., 2017) (top), MDM (bottom) captures uncertainty in both localization (blue) and shape (darker regions in the heatmap).

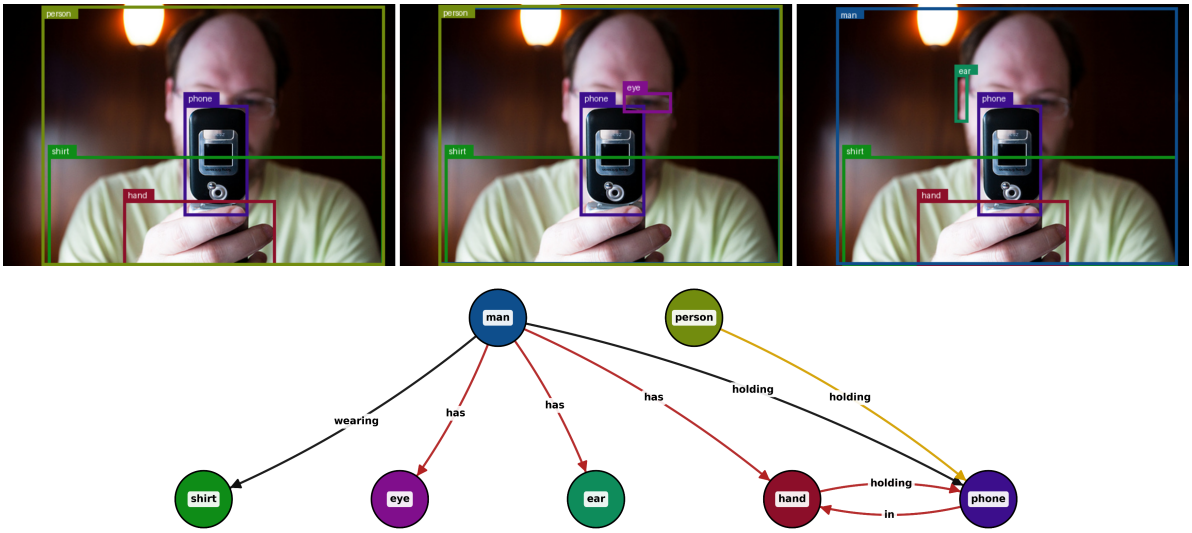
Figure 6: **Uncertainty analysis.** Results shown on the task of instance segmentation. We contrast predictions from a deterministic Mask R-CNN (He et al., 2017) model with those from our proposed MDM framework. For each image, MDM generates 1000 samples and aggregates masks into a heatmap; brighter regions indicate consistent predictions across samples, while darker regions reflect uncertainty. Variability in object localization is also highlighted by plotting the mean bounding box and shading the region spanning the minimum and maximum predicted box extents; larger shaded regions indicate greater uncertainty.


food
handle
flower
flower
on
on
has
on
on
plate
cup
on
on
table

(a)


ear
ear
fence
woman
girl
of
has
has
of
behind
looking at
wearing
has
has
giraffe
shirt
hair

(b)



(c)

Figure 7: **Qualitative Analysis.** Results for the scene graph generation task. We sample three outputs from our MDM model and visualize the aggregated scene graph together with the corresponding bounding boxes. **Black** edges denote triplets that appear in all three samples, **yellow** edges denote triplets that appear in exactly two samples, and **red** edges denote triplets that are unique to a single sample. MDM is able to generate distinct but plausible graphs given the same input image.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34: 17981–17993, 2021.

Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4974–4983, 2019a.

Mingfei Chen, Yue Liao, Si Liu, Zhiyuan Chen, Fei Wang, and Chen Qian. Reformulating hoi detection as adaptive set prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9004–9013, 2021.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Hydra-sgg: Hybrid relation assignment for one-stage scene graph generation. *arXiv preprint arXiv:2409.10262*, 2024.

Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19830–19843, 2023a.

Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6163–6171, 2019b.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 909–919, 2023b.

Zhanwen Chen, Saed Rezayi, and Sheng Li. More knowledge, less bias: Unbiasing scene graph generation with explicit ontological adjustment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4023–4032, 2023c.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

Meng-Jiun Chiou, Henghui Ding, Hanshu Yan, Changhu Wang, Roger Zimmermann, and Jiashi Feng. Recovering the unbiased scene graphs from the biased ones. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1581–1590, 2021.

Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *arXiv preprint arXiv:2201.11460*, 2022.

Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11169–11183, 2023.

Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15404–15413, 2021.

Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Single-stage visual relationship learning using conditional queries. *Advances in Neural Information Processing Systems*, 35:13064–13077, 2022.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Qi Dong, Zhuowen Tu, Haofu Liao, Yuting Zhang, Vijay Mahadevan, and Stefano Soatto. Visual relationship detection using part-and-sum transformers with composite queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3550–3559, 2021.

Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6910–6919, 2021.

Jiawei Fu, Tiantian Zhang, Kai Chen, and Qi Dou. Hybrid reciprocal transformer with triplet feature alignment for scene graph generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8953–8963, 2025.

Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3621–3630, 2021.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10323–10332, 2019.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10696–10706, 2022.

Zhangxuan Gu, Haoxing Chen, and Zhuoer Xu. Diffusioninst: Diffusion model for instance segmentation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2730–2734. IEEE, 2024.

Yuyu Guo, Lianli Gao, Xuanhan Wang, Yuxuan Hu, Xing Xu, Xu Lu, Heng Tao Shen, and Jingkuan Song. From general to specific: Informative scene graph generation via balance adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16383–16392, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Karsten Held, E Rota Kops, Bernd J Krause, William M Wells, Ron Kikinis, and H-W Muller-Gartner. Markov random field segmentation of brain mr images. *IEEE transactions on medical imaging*, 16(6): 878–886, 1997.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

Shihua Huang, Zhichao Lu, Xiaodong Cun, Yongjun Yu, Xiao Zhou, and Xi Shen. Deim: Detr with improved matching for fast convergence. In *Proceedings of the computer vision and pattern recognition conference*, pp. 15162–15171, 2025.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. Egtr: Extracting graph from transformer for scene graph generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24229–24238, 2024.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, pp. 10362–10383. PMLR, 2022.

Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1219–1228, 2018.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Siddhesh Khandelwal and Leonid Sigal. Iterative scene graph generation. *Advances in Neural Information Processing Systems*, 35:24295–24308, 2022.

Siddhesh Khandelwal, Mohammed Suhail, and Leonid Sigal. Segmentation-grounded scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15879–15889, 2021.

Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 74–83, 2021.

Jongha Kim, Jihwan Park, Jinyoung Park, Jinyoung Kim, Sehyung Kim, and Hyunwoo J Kim. Groupwise query specialization and quality-aware multi-assignment for transformer-based visual relationship detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 28160–28169, 2024.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Soohyeong Lee, Ju-Whan Kim, Youngmin Oh, and Joo Hyuk Jeon. Visual question answering over scene graph. In *2019 First International Conference on Graph Computing (GC)*, pp. 45–50. IEEE, 2019.

Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13619–13627, 2022a.

Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3041–3050, 2023.

Jiankai Li, Yunhong Wang, Xiefan Guo, Ruijie Yang, and Weixin Li. Leveraging predicate and triplet learning for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28369–28379, 2024.

Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11109–11119, 2021.

Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19486–19496, 2022b.

Yikang Li, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. Pastegan: A semi-parametric method to generate image from scene graph. *Advances in Neural Information Processing Systems*, 32, 2019.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11546–11556, 2021.

Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022.

Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European conference on computer vision*, pp. 852–869. Springer, 2016.

Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer. *arXiv preprint arXiv:2407.17140*, 2024.

Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3651–3660, 2021.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Nithin Gopalakrishnan Nair, Anoop Cherian, Suhas Lohit, Ye Wang, Toshiaki Koike-Akino, Vishal M Patel, and Tim K Marks. Steered diffusion: A generalized framework for plug-and-play conditional image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20850–20860, 2023.

Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. *Advances in neural information processing systems*, 30, 2017.

Kien Nguyen, Subarna Tripathi, Bang Du, Tanaya Guha, and Truong Q Nguyen. In defense of scene graphs for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1407–1416, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3957–3966, 2019.

Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. *Advances in neural information processing systems*, 17, 2004.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, et al. Relationformer: A unified framework for image-to-graph generation. *arXiv preprint arXiv:2203.10202*, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Gopika Sudhakaran, Devendra Singh Dhami, Kristian Kersting, and Stefan Roth. Vision relation transformer for unbiased scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21882–21893, 2023.

Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13936–13945, 2021.

Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14454–14463, 2021.

Haoru Tan, Sitong Wu, and Jimin Pi. Semantic diffusion network for semantic segmentation. *Advances in Neural Information Processing Systems*, 35:8702–8716, 2022.

Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6619–6628, 2019.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3716–3725, 2020.

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Mengyu Wang, Henghui Ding, Jun Hao Liew, Jiajun Liu, Yao Zhao, and Yunchao Wei. Segrefiner: Towards model-agnostic segmentation refinement with discrete diffusion process. *arXiv preprint arXiv:2312.12425*, 2023.

Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 2567–2575, 2022.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5410–5419, 2017.

Qi Yan, Zhengyang Liang, Yang Song, Renjie Liao, and Lele Wang. Swingnn: Rethinking permutation invariance in diffusion models for graph generation. *arXiv preprint arXiv:2307.01646*, 2023.

Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 265–273, 2020.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 670–685, 2018.

Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10685–10694, 2019.

Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021.

Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *European Conference on Computer Vision*, pp. 606–623. Springer, 2020.

Lukas Zbinden, Lars Doorenbos, Theodoros Pissas, Adrian Thomas Huber, Raphael Sznitman, and Pablo Márquez-Neila. Stochastic segmentation with conditional categorical diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1119–1129, 2023.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5831–5840, 2018.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11535–11543, 2019.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16965–16974, 2024.

Chaofan Zheng, Xinyu Lyu, Lianli Gao, Bo Dai, and Jingkuan Song. Prototype-based embedding network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22783–22792, 2023.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A Appendix

A.1 Transformer Architecture for Scene Graph Generation

In Section 5.2 of the main paper we briefly describe that for the task of scene graph generation, we instantiate the MDM framework with a denoising decoder consisting of three components – a subject decoder, an object decoder, and a predicate decoder. In this section we detail how the transformer-based formulation described in Section 4.4 is extended to this three-decoder setup without introducing any additional task-specific architectural components.

As described in Section 4.3, our proposed MDM framework for scene graph generation factorizes the joint distribution into two conditional distributions – $\Pr(\{(\mathbf{b}_i^s, \mathbf{b}_i^o)\} \,|\, \mathbf{I})$ (proposal diffusion) and $\Pr(\{(\mathbf{l}_i^s, \mathbf{l}_i^o, \mathbf{l}_i^p)\} \,|\, \{(\mathbf{b}_i^s, \mathbf{b}_i^o)\}, \mathbf{I})$ (triplet diffusion) (Eq. (11)). The forward diffusion processes for these conditionals independently corrupt the corresponding components as described in Section 4.3.

Naively modeling the denoising process for the aforementioned conditionals makes learning challenging due to the combinatorial nature of the output space. For example, in triplet diffusion, the output space has a cardinality of $\mathcal{N}_o \times \mathcal{N}_o \times \mathcal{N}_p$, where $\mathcal{N}_o$ and $\mathcal{N}_p$ are the number of entity and predicate categories respectively.

Therefore, to make learning tractable while still allowing the subject, object, and predicate components to depend on each other, we follow an approach similar to Khandelwal & Sigal (2022), and further factorize each denoising process. Concretely,

$$\Pr\Big(\{(\mathbf{b}_i^s, \mathbf{b}_i^o)\} \,\Big|\, \mathbf{I}\Big) = \underbrace{\Pr\Big(\{\mathbf{b}_i^s\} \,\Big|\, \mathbf{I}\Big)}_{\text{subject decoder}} \cdot \underbrace{\Pr\Big(\{\mathbf{b}_i^o\} \,\Big|\, \{\mathbf{b}_i^s\}, \mathbf{I}\Big)}_{\text{object decoder}} \tag{19}$$

$$\begin{aligned}\Pr\Big(\{(\mathbf{l}_i^s, \mathbf{l}_i^o, \mathbf{l}_i^p)\} \,\Big|\, \{(\mathbf{b}_i^s, \mathbf{b}_i^o)\}, \mathbf{I}\Big) = \; & \underbrace{\Pr\Big(\{\mathbf{l}_i^s\} \,\Big|\, \{(\mathbf{b}_i^s, \mathbf{b}_i^o)\}, \mathbf{I}\Big)}_{\text{subject decoder}} \\ & \cdot \underbrace{\Pr\Big(\{\mathbf{l}_i^o\} \,\Big|\, \{\mathbf{l}_i^s\}, \{(\mathbf{b}_i^s, \mathbf{b}_i^o)\}, \mathbf{I}\Big)}_{\text{object decoder}} \\ & \cdot \underbrace{\Pr\Big(\{\mathbf{l}_i^p\} \,\Big|\, \{(\mathbf{l}_i^s, \mathbf{l}_i^o)\}, \{(\mathbf{b}_i^s, \mathbf{b}_i^o)\}, \mathbf{I}\Big)}_{\text{predicate decoder}}\end{aligned} \tag{20}$$

Each of these three decoders ($r \in \{s, o, p\}$) is architecturally identical to the formulation introduced in Section 4.4 and processes its respective noisy input (subject/object/predicate box or label). Note that in proposal diffusion, given noisy subject and object boxes $\{(\mathbf{b}_{i,t}^s, \mathbf{b}_{i,t}^o)\}$, the noisy predicate box $\mathbf{b}_{i,t}^p$ is defined as the bounding box formed by the centers of $\mathbf{b}_{i,t}^s$ and $\mathbf{b}_{i,t}^o$. The input queries $\{\mathbf{q}_i^r\}; r \in \{s, o, p\}$ are subsequently obtained by following the formulation in Eq. (12).

For a particular layer m, the query features $\{\mathbf{q}_i^{s,k,m}\}$ for the subject decoder s are computed by applying the formulation in Eq. (13). For the object decoder o, we modulate the object queries based on the subject queries as follows,

$$\begin{aligned}\boldsymbol{\rho}_i^{o,m}, \boldsymbol{\tau}_i^{o,m} &= f_{o,m}(\mathbf{q}_i^{s,k,m}) \\ \mathbf{q}_i^{o,k,m-1} &\leftarrow \mathbf{q}_i^{o,k,m-1} \odot (1 + \boldsymbol{\rho}_i^{o,m}) + \boldsymbol{\tau}_i^{o,m}\end{aligned} \tag{21}$$

The updated features $\{\mathbf{q}_i^{o,k,m}\}$ are then obtained by applying the formulation in Eq. (13). Finally, the predicate queries are modulated using both subject and object queries,

$$\begin{aligned}\boldsymbol{\rho}_i^{p,m}, \boldsymbol{\tau}_i^{p,m} &= f_{p,m}(\mathbf{q}_i^{s,k,m}, \mathbf{q}_i^{o,k,m}) \\ \mathbf{q}_i^{p,k,m-1} &\leftarrow \mathbf{q}_i^{p,k,m-1} \odot (1 + \boldsymbol{\rho}_i^{p,m}) + \boldsymbol{\tau}_i^{p,m}\end{aligned} \tag{22}$$

The updated query features $\{\mathbf{q}_i^{p,k,m}\}$ are then similarly obtained by following Eq. (13). Crucially, it should be noted that there are no structural or architectural differences across the three decoders. The inherent dependencies between the subject, object, and predicate components are established entirely via the query modulation steps described above.